

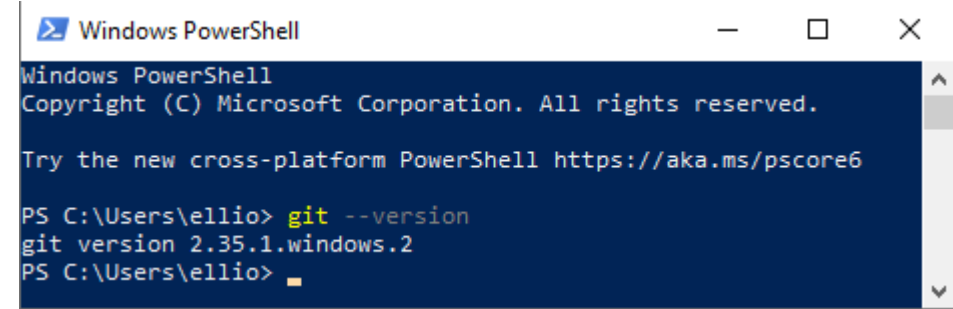
The Complete Git Guide: Understand and master Git and GitHub

1. Installing Git
2. Basic Shell Commands
3. How Git works under the hood
4. Basic Git Operations
5. Git Branches and HEAD
6. Cloning, Exploring and Modifying Public Repositories
7. Merging Branches
8. Git Hub and Remote Repositories
9. Git Push, Pull and Fetch
10. Git Pull requests
11. Forks and Contribution to the Public Repositories
12. Git Tags
13. Git Rebasing
14. Ignoring files in Git
15. Detached Head
16. Advanced Git
17. GitHub Pages
18. Git Hooks

INSTALLING GIT

Installing Git

From powershell use **git --version** command to see if git is installed. If not installed git can be downloaded from <https://git-scm.com>
For windows there are 32 bit and 64 bit versions.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ellio> git --version
git version 2.35.1.windows.2
PS C:\Users\ellio>
```

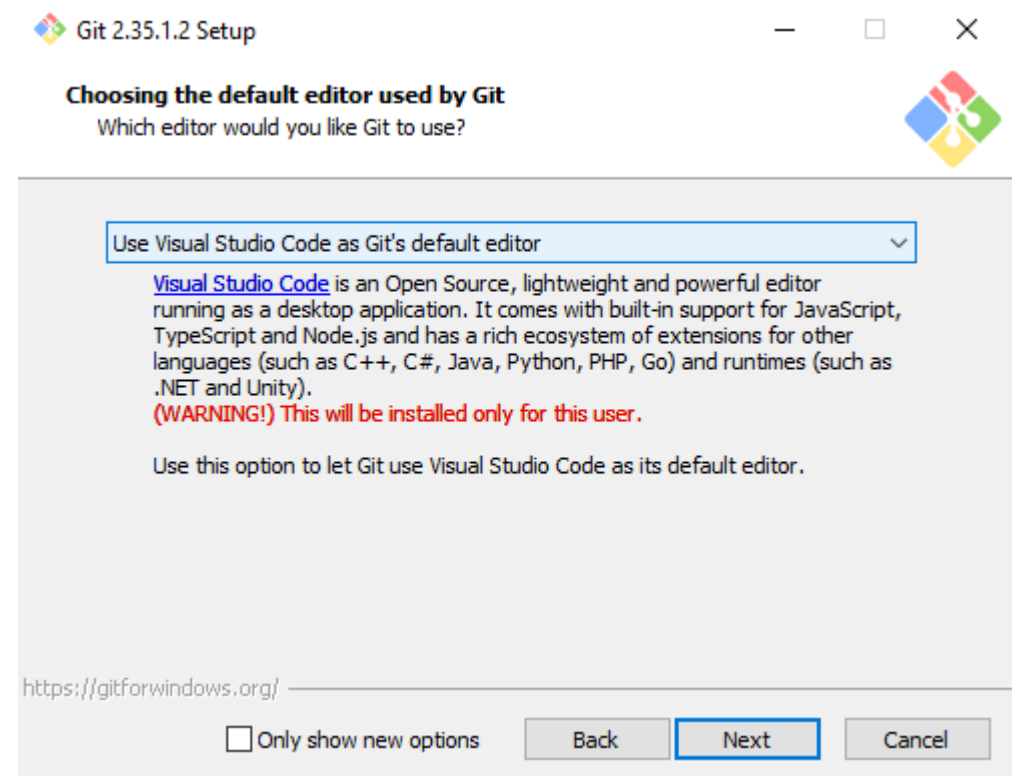
Important Notice: Git installation for windows comes with git bash terminal. You should use git bash terminal rather than built in terminal such as powershell.

First go to <https://code.visualstudio.com> and download and install.

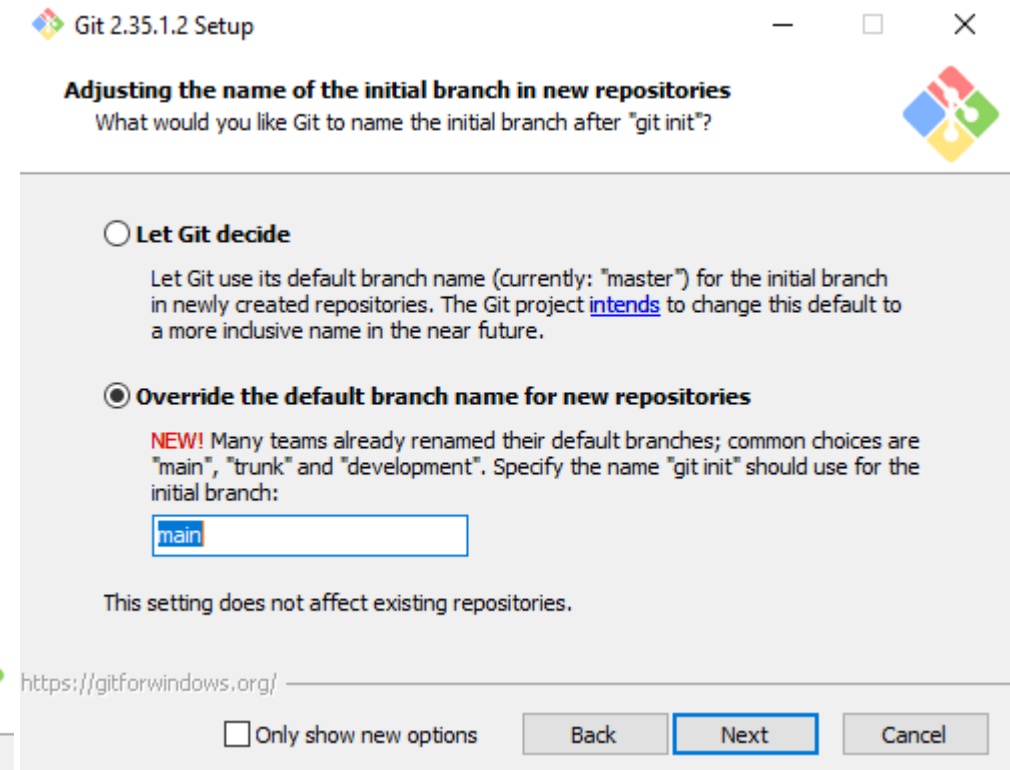
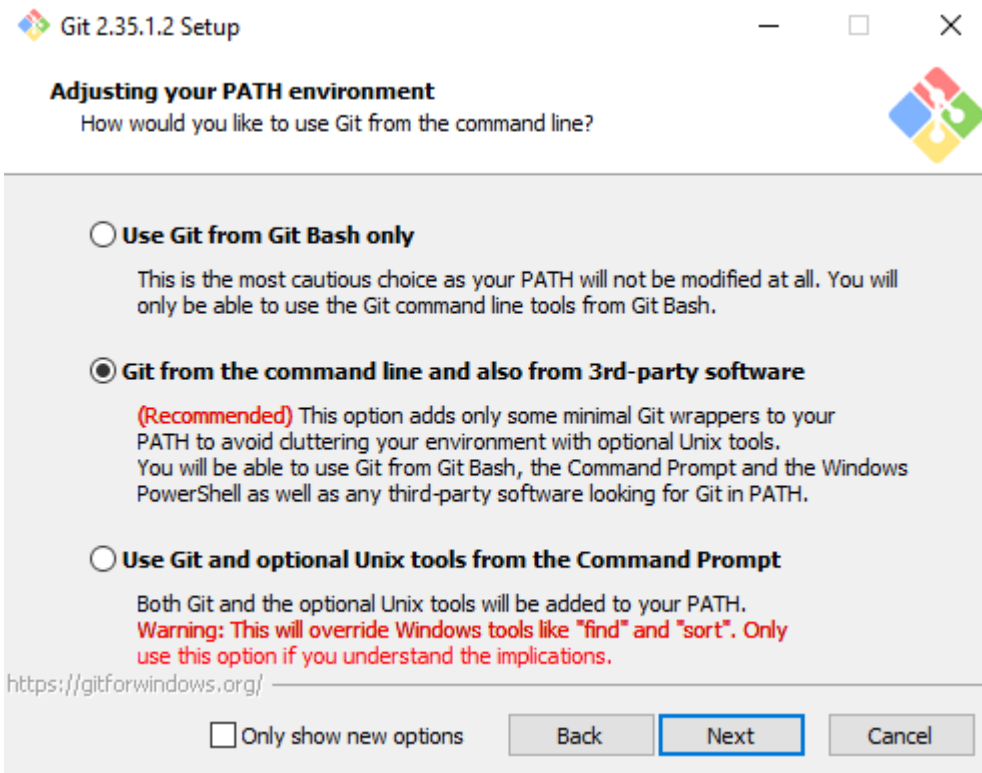
When installing check the box that says
“register code as default editor for supported file types”

Next click on the git downloaded exe file, i.e. Git-2.35.1.2-64-bit.exe.

When running through the git install options select visual studio code as the default git editor.

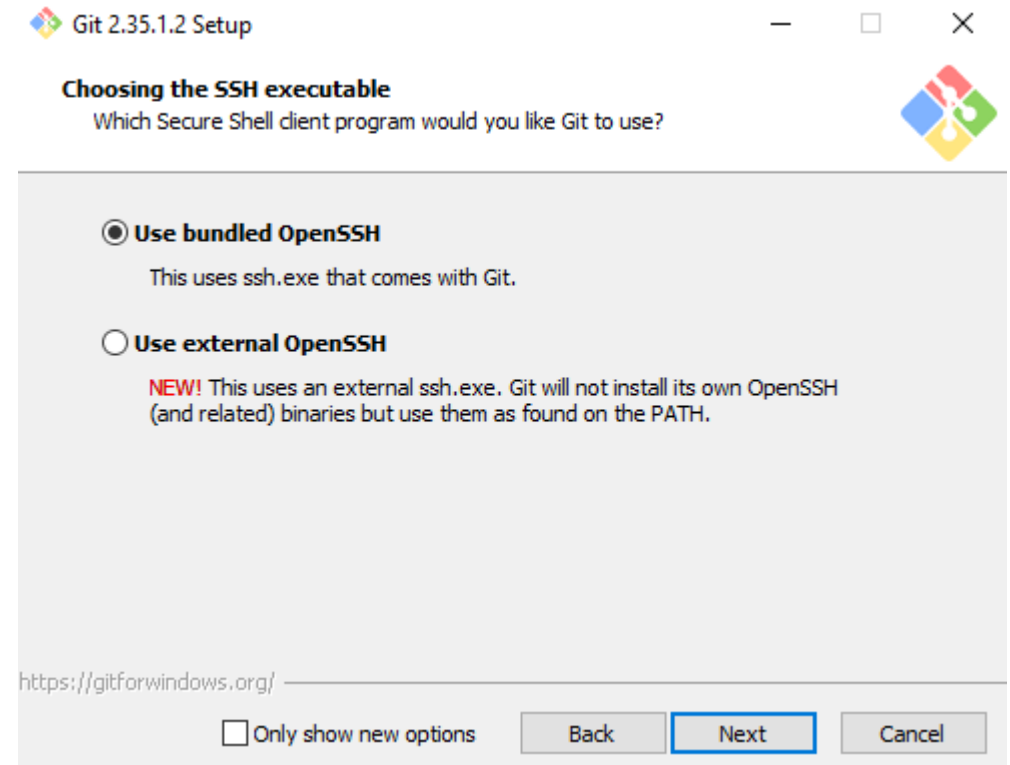
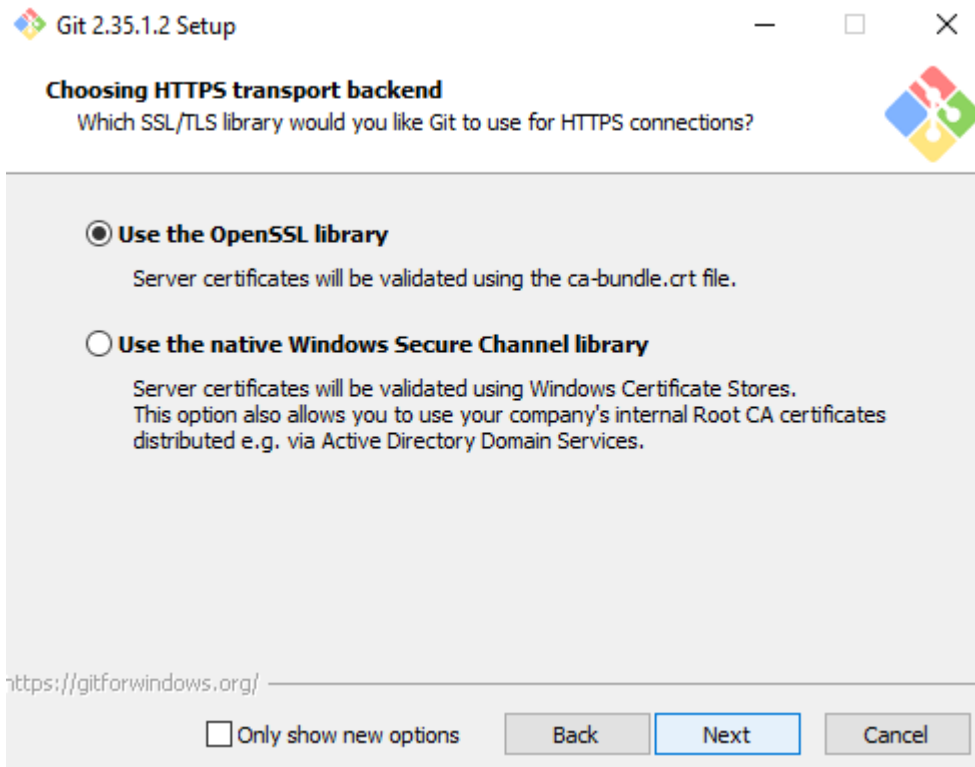


Override the default branch name for new repositories. Chose main. This will make sense further along in the course.



Leave as default setting of Git from command line and also from third party software.

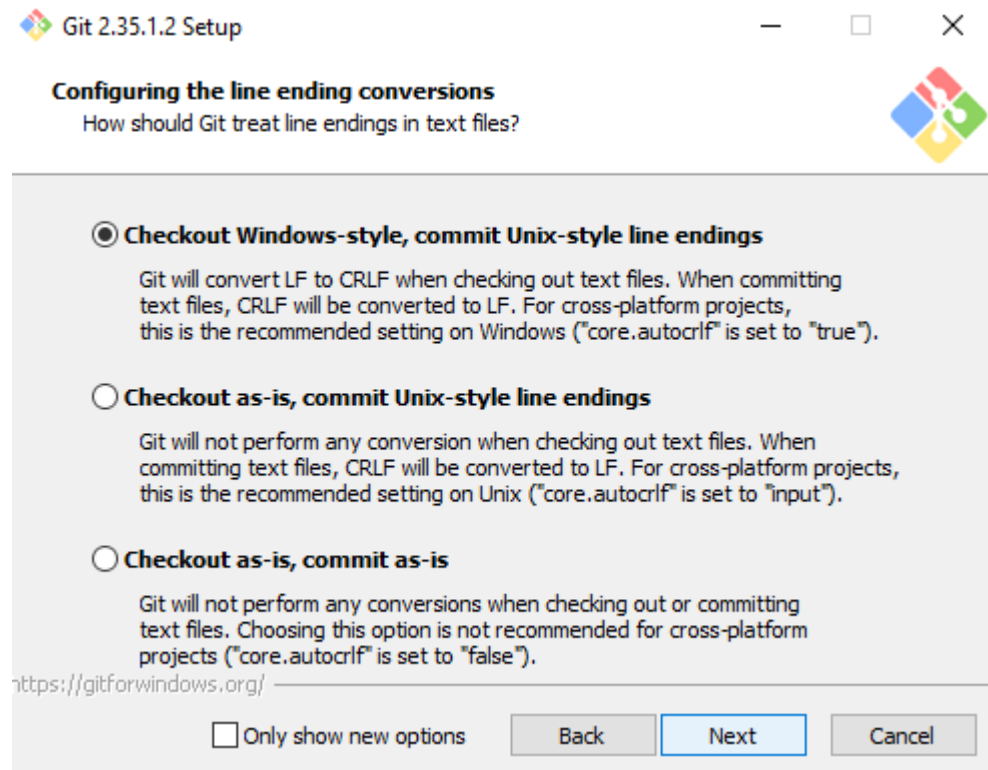
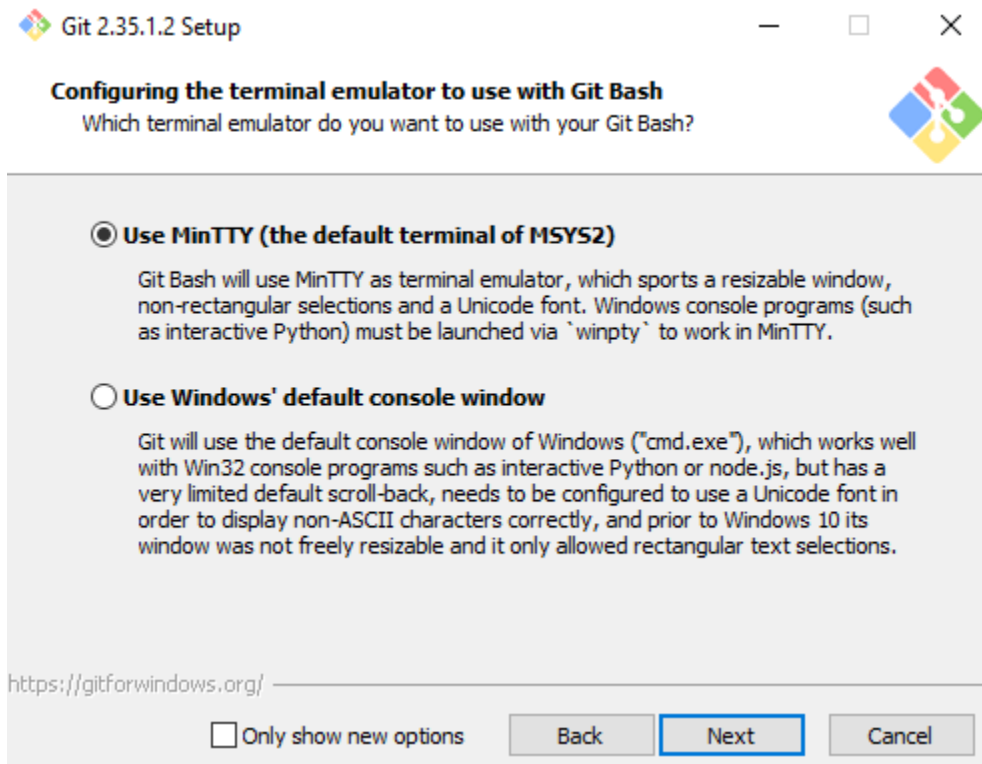
Leave as default setting of use bundled OpenSSH.



Leave as default setting of use OpenSSL Library.

Leave as default setting of checkout windows-style, commit unix style line endings.

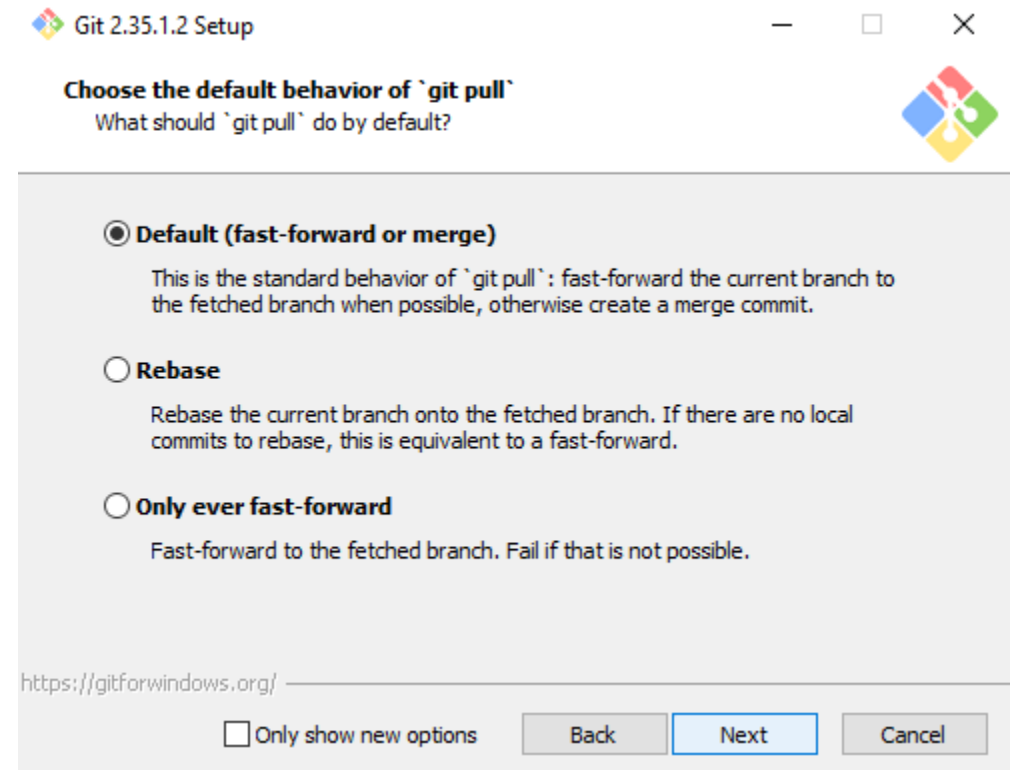
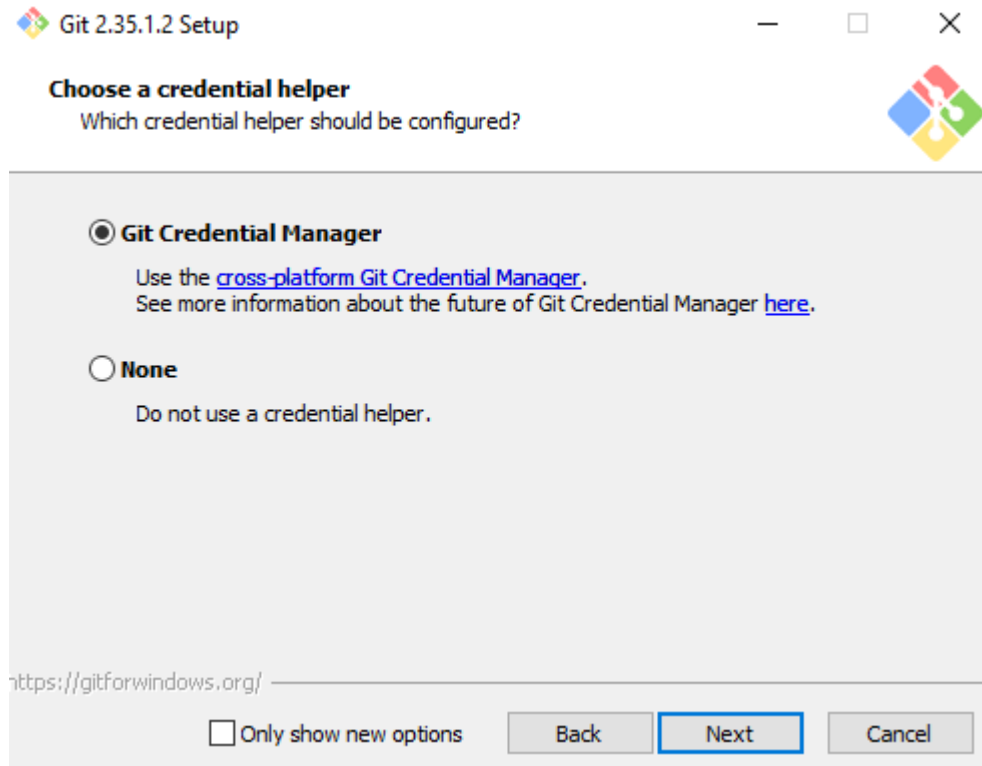
This option is best for cross platform projects where developers use different operating systems.



Leave as default setting of Use minTTY

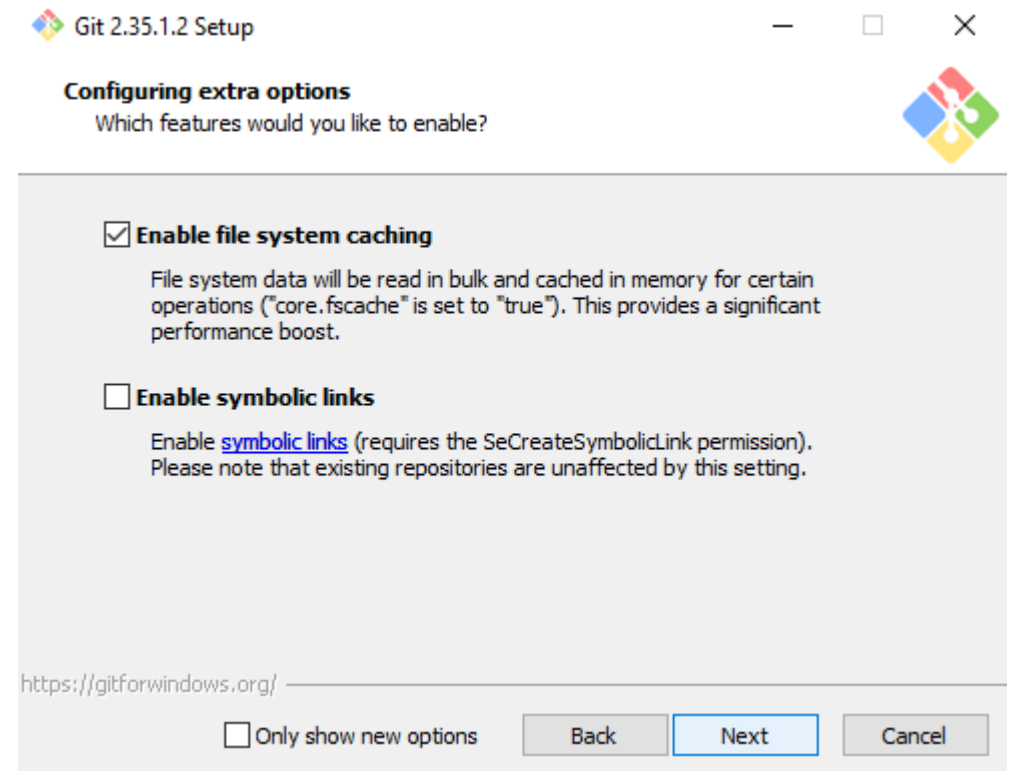
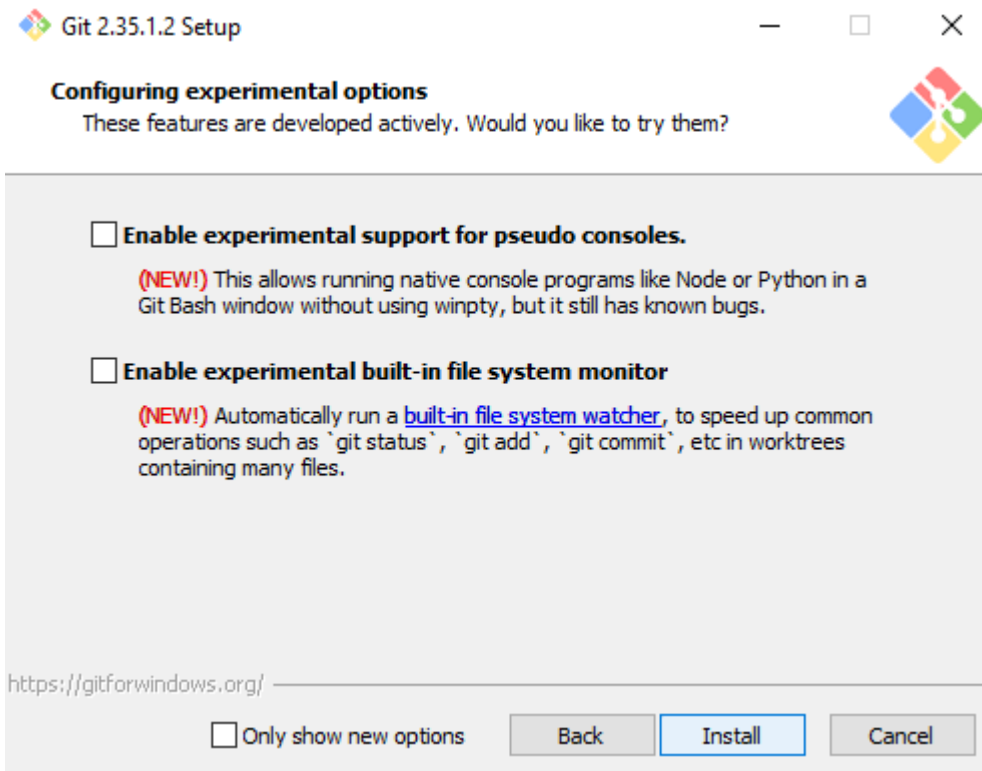
Leave as default setting of (fast forward or Merge).

This will make sense further along in the course.



Leave as default setting of Git Credential Manager

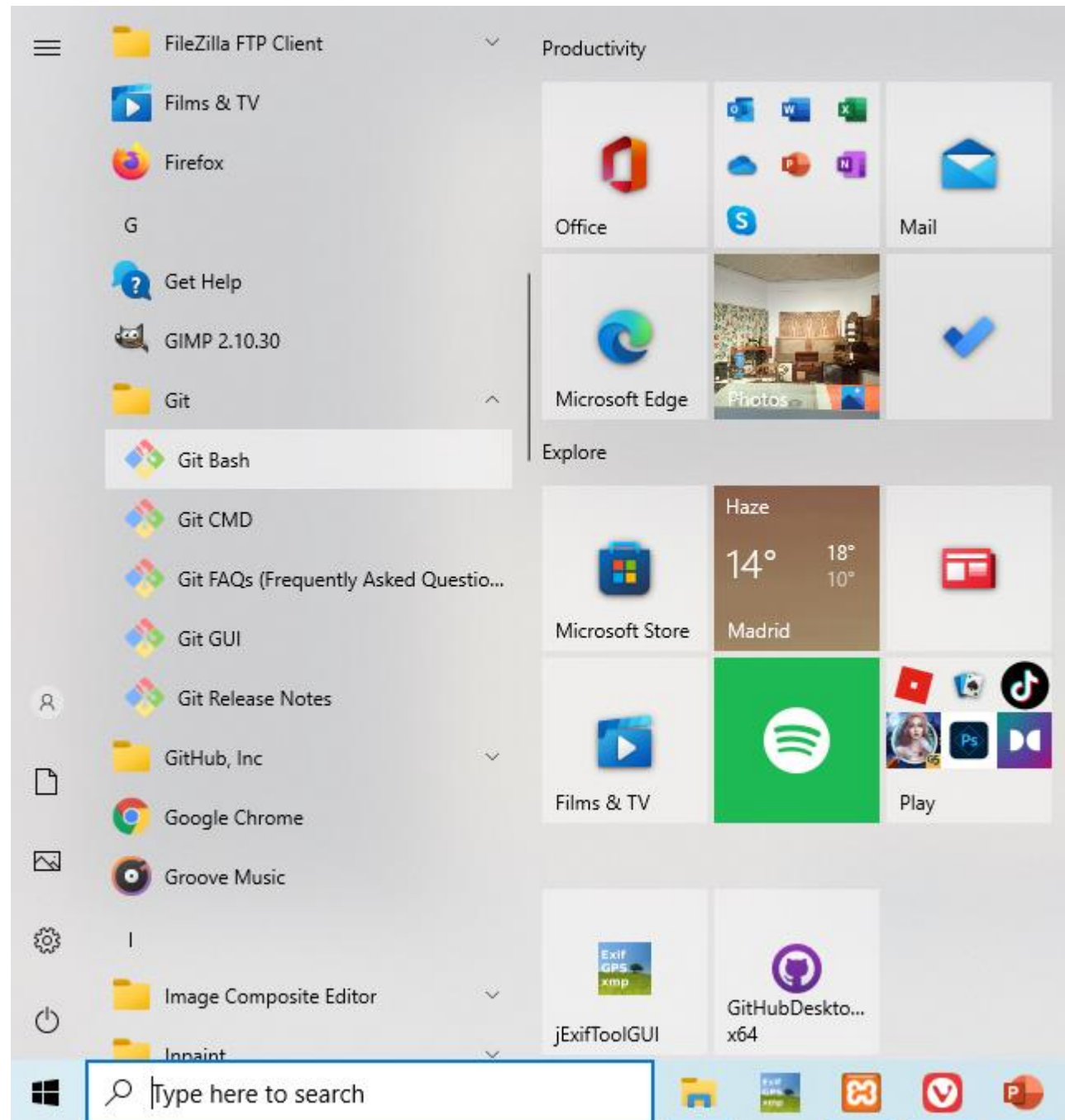
Leave as default setting of enable file system caching.



Leave as default setting of both unchecked.

Click **Install**

From the windows icon there is a git folder with **Git Bash** Terminal which is a command line terminal program with colours.
(better than Powershell!)



BASIC SHELL COMMANDS

Directory Management Shell Commands

pwd Shows the current directory.

cd .. Two dots is the alias for parent directory. So **cd..** Will go back one directory. Note that prior we were in `c/users/ellio` and now we are in `c/users`

ls Will list what is in the current directory, in this case folders.

clear This command will clear the bash terminal of all previous content.

ls -l Will list details about the folders or directories

Directory permissions

Owner of directory

Group of users

Date

Name of directory

```
MINGW64:/c/Users

ellio@DESKTOP-U93252R MINGW64 ~
$ pwd
/c/Users/ellio

ellio@DESKTOP-U93252R MINGW64 ~
$ cd ..

ellio@DESKTOP-U93252R MINGW64 /c/Users
$ ls
'All Users'@  Default/  'Default User'@  Public/  desktop.ini  ellio/

ellio@DESKTOP-U93252R MINGW64 /c/Users
$ clear
```

```
MINGW64:/c/Users

ellio@DESKTOP-U93252R MINGW64 /c/Users
$
```

```
MINGW64:/c/Users

ellio@DESKTOP-U93252R MINGW64 /c/Users
$ ls -l
total 25
drwxrwxrwx 1 ellio 197609 14 Dec 7 2019 'All Users' -> /c/ProgramData/
drwxr-xr-x 1 ellio 197609  0 Jul 1 2021 Default/
drwxrwxrwx 1 ellio 197609 16 Dec 7 2019 'Default User' -> /c/Users/Default/
drwxr-xr-x 1 ellio 197609  0 Jul 1 2021 Public/
-rw-r--r-- 1 ellio 197609 174 Dec 7 2019 desktop.ini
drwxr-xr-x 1 ellio 197609  0 Feb 23 00:59 ellio/

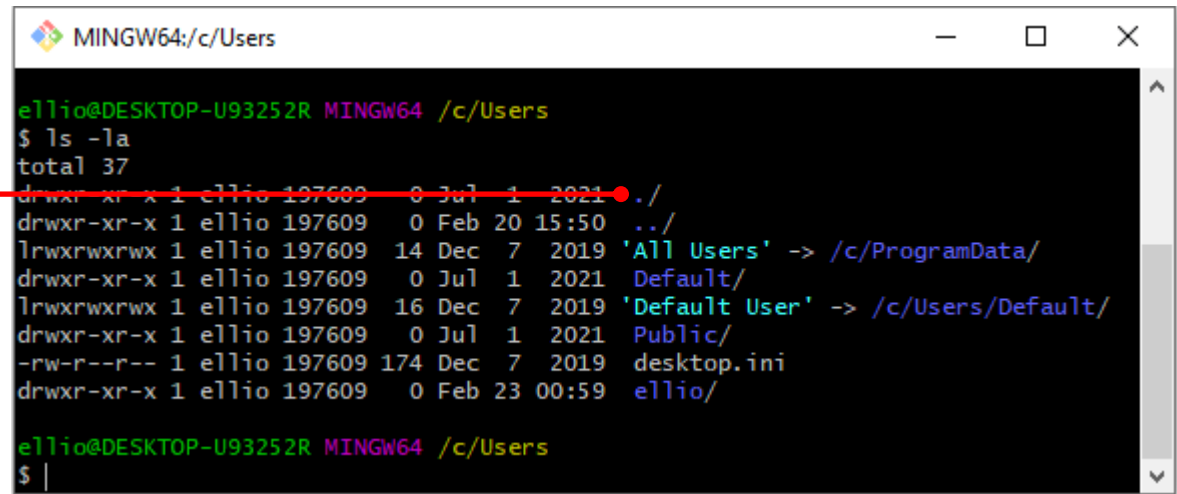
ellio@DESKTOP-U93252R MINGW64 /c/Users
$ |
```

ls -la – Will list more details about all the folders or directories.

Hidden directories start with a dot.
NOTE: Without option a you will not see hidden folders and files.

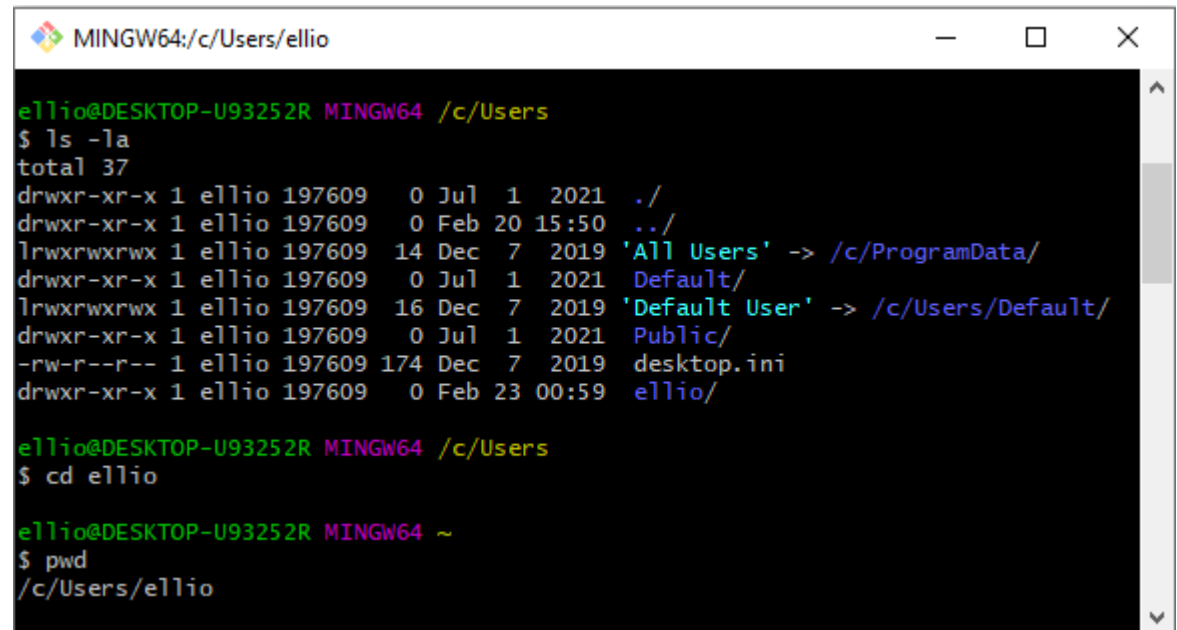
cd – can be used to change directory. In this case we are moving into the ellio directory.

pwd – confirms that we have moved into this directory.



```
MINGW64:/c/Users
ellio@DESKTOP-U93252R MINGW64 /c/Users
$ ls -la
total 37
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 ./
drwxr-xr-x 1 ellio 197609  0 Feb 20 15:50 ../
lrwxrwxrwx 1 ellio 197609 14 Dec  7  2019 'All Users' -> /c/ProgramData/
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 Default/
lrwxrwxrwx 1 ellio 197609 16 Dec  7  2019 'Default User' -> /c/Users/Default/
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 Public/
-rw-r--r-- 1 ellio 197609 174 Dec  7  2019 desktop.ini
drwxr-xr-x 1 ellio 197609  0 Feb 23 00:59 ellio/

ellio@DESKTOP-U93252R MINGW64 /c/Users
$
```

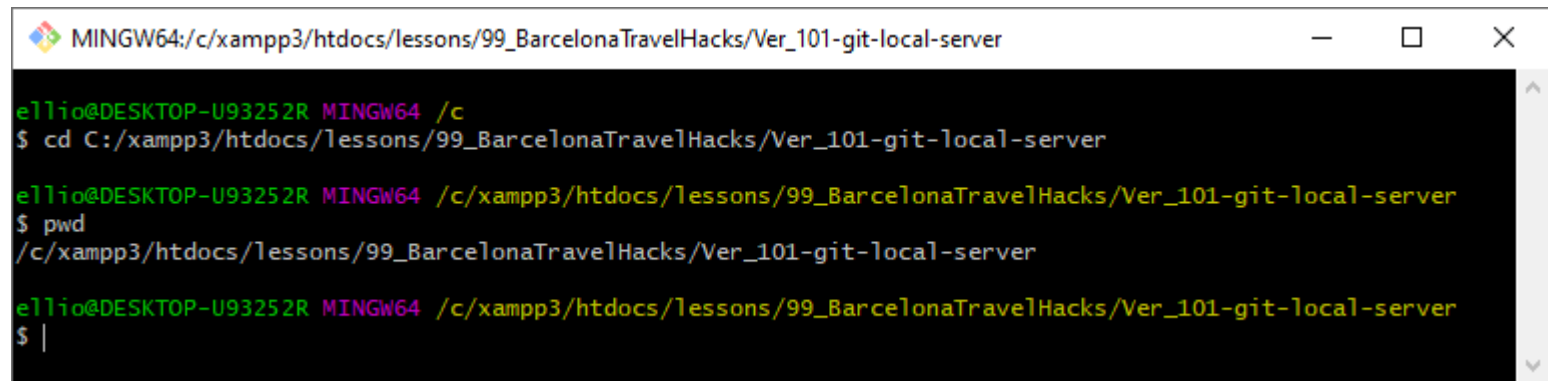


```
MINGW64:/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 /c/Users
$ ls -la
total 37
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 ./
drwxr-xr-x 1 ellio 197609  0 Feb 20 15:50 ../
lrwxrwxrwx 1 ellio 197609 14 Dec  7  2019 'All Users' -> /c/ProgramData/
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 Default/
lrwxrwxrwx 1 ellio 197609 16 Dec  7  2019 'Default User' -> /c/Users/Default/
drwxr-xr-x 1 ellio 197609  0 Jul  1  2021 Public/
-rw-r--r-- 1 ellio 197609 174 Dec  7  2019 desktop.ini
drwxr-xr-x 1 ellio 197609  0 Feb 23 00:59 ellio/

ellio@DESKTOP-U93252R MINGW64 /c/Users
$ cd ellio

ellio@DESKTOP-U93252R MINGW64 ~
$ pwd
/c/Users/ellio
```


cd I can use this command with any long path to jump into that directory.

A screenshot of a Windows command prompt window. The title bar at the top shows the path 'MINGW64:/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server' and standard window controls. The command prompt has a black background with yellow text. The user 'e11io@DESKTOP-U93252R' is in a 'MINGW64' environment. The sequence of commands and outputs is: 1. Command: 'cd C:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server'. 2. Prompt: '\$'. 3. Command: 'pwd'. 4. Output: '/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server'. 5. Prompt: '\$'. 6. Command: '|'. 7. Prompt: '\$ |'.

```
MINGW64:/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c
$ cd C:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ pwd
/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ |
```

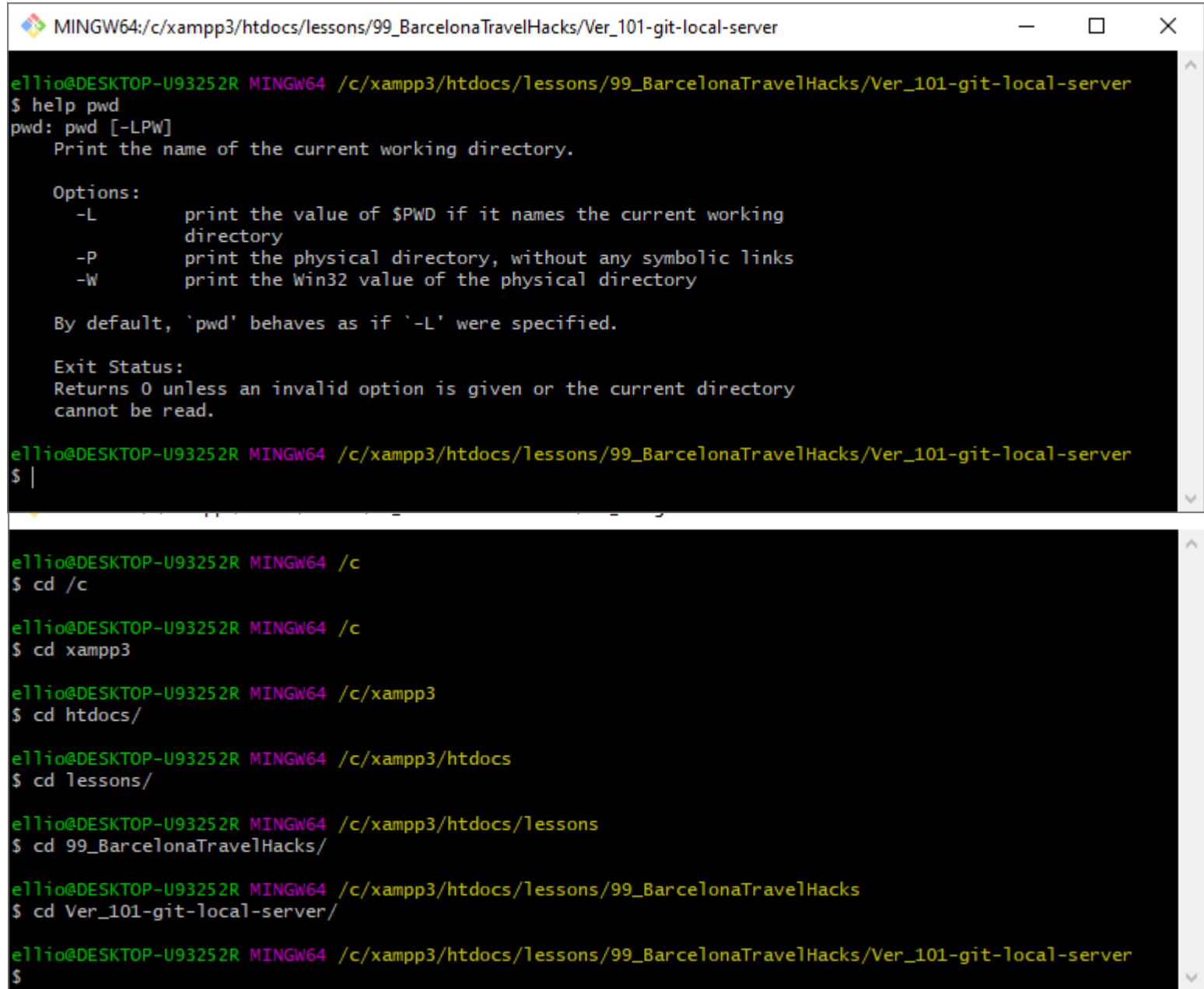
File management Shell Commands

man This command can be used to get help on any command except that it does not work on windows.

Alternatively use **help** then the command you want help with.

cd /c Will take me back to the root folder of C:

Tab key Performs an autocomplete function. For example if I type `cd les` then it would autocomplete to `cd lessons`.



```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ help pwd
pwd: pwd [-LPW]
    Print the name of the current working directory.

Options:
  -L      print the value of $PWD if it names the current working
          directory
  -P      print the physical directory, without any symbolic links
  -W      print the Win32 value of the physical directory

By default, 'pwd' behaves as if '-L' were specified.

Exit Status:
Returns 0 unless an invalid option is given or the current directory
cannot be read.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ |

ellio@DESKTOP-U93252R MINGW64 /c
$ cd /c

ellio@DESKTOP-U93252R MINGW64 /c
$ cd xampp3

ellio@DESKTOP-U93252R MINGW64 /c/xampp3
$ cd htdocs/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs
$ cd lessons/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons
$ cd 99_BarcelonaTravelHacks/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ cd Ver_101-git-local-server/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$
```

Double tab In this example I typed `cd ver` followed by double tab.

Because there were multiple options it asked me if I wanted to see all 50 possible options.

I was then able to continue typing `cd Ver_101` followed by tab which autocompleted to the correct command because there was only one option for `Ver_101`.

```
MINGW64:/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ cd ..

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ cd Ver_
Display all 50 possibilities? (y or n)
Ver_100_-_Photo-metadata-fixer-and-Sitemap-Maker/ Ver_75/
Ver_101-git-local-server/ Ver_76 - Upgrade 14 Server Version 02-09-2021/
Ver_54 - upgrade Server Version 15-05-21/ Ver_77/
Ver_55/ Ver_78 - Upgrade 15 Server Version 09-09-2021/
Ver_56 - Live server Backup 31-05-2021/ Ver_79 - Live Server backup 18-10-2021/
Ver_57 - Upgrade 6 Server Version 19-06-2021/ Ver_80/
Ver_58 - Live Server Backup 24-06-2021/ Ver_81 - Live Server Backup 05-11-2021/
Ver_59 - Live Server Backup 02-07-2021/ Ver_82/
Ver_60/ Ver_83 - Live Server Backup 07-11-2021/
Ver_61 - Live Server Backup 12-07-2021/ Ver_84 - Upgrade 16 Server Version 13-11-2021/
Ver_61 - Upgrade 7 Server Version 12-07-2021/ Ver_85/
Ver_62 - Upgrade 8 Server Version 23-07-2021/ Ver_86 - Upgrade 17 Server Version 23-11-2021/
Ver_63 - Experimental/ Ver_87 - Upgrade 18 Server Version 28-11-2021/
Ver_64/ Ver_88/
Ver_65/ Ver_89 - Upgrade 19 Server Version 05-12-2021/
Ver_66 - Upgrade 9 Server Version '6-08-2021/ Ver_90 - Photo metadata fixer/
Ver_67 - Upwork lite javascript gallery/ Ver_91 - Photo metadata fixer2/
Ver_68/ Ver_92_-_Photo-metadata-fixer-and-Sitemap-Maker/
Ver_69 - Upgrade 10 Server Version 11-08-2021/ Ver_93 - Live Server Backup 29-01-2022/
Ver_70 - Upgrade 11 Server Version 12-08-2021/ Ver_94 - Upgrade 20 Server Version 29-01-2021/
Ver_71/ Ver_95 - Live Server Backup 01-02-2022/
Ver_72 - Upgrade 12 Live Version 17-08-2021/ Ver_96 - Live Server Backup 02-02-2022/
Ver_72 - Upgrade 12 Server Version 22-08-2021/ Ver_97 - Live Server Backup 21-02-2021/
Ver_73 - Live Server Backup 29-08-2021/ Ver_98 - Upgrade 21 Server Version 21-02-2021/
Ver_74 - Upgrade 13 Server Version 30-08-2021/ Ver_99 - Live Server Backup 22-02-2021/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ cd Ver_101-git-local-server/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ pwd
/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$
```

touch This command is used to create a new file. In this case I have created new-fil.txt with txt being the file extension.

```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ touch new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ ls -la
total 24
drwxr-xr-x 1 ellio 197609 0 Feb 23 21:09 ./
drwxr-xr-x 1 ellio 197609 0 Feb 23 20:09 ../
-rw-r--r-- 1 ellio 197609 0 Feb 23 21:09 new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$
```

echo This command is used to output whatever to the bash terminal

```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ echo "hello world!"
hello world!

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$
```

> Using greater than symbol means write to. So now *another-file.txt* will contain *hello world!*.

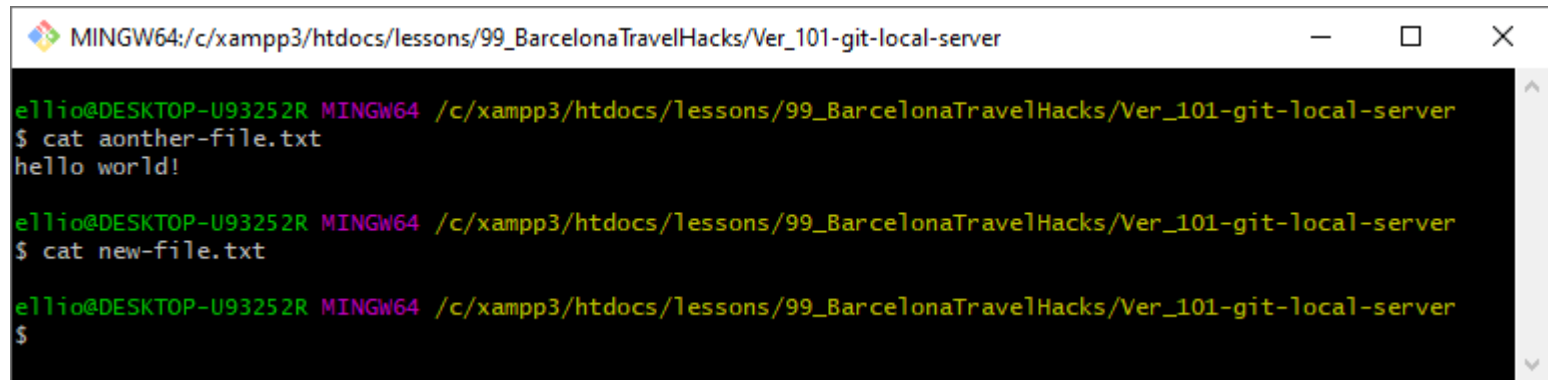
```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ echo "hello world!" > aonther-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ ls -la
total 25
drwxr-xr-x 1 ellio 197609 0 Feb 23 21:15 ./
drwxr-xr-x 1 ellio 197609 0 Feb 23 20:09 ../
-rw-r--r-- 1 ellio 197609 13 Feb 23 21:15 aonther-file.txt
-rw-r--r-- 1 ellio 197609 0 Feb 23 21:09 new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ |
```

cat This command is used to read contents of file.

We can see that *another-file.txt* contains *hello world!* And *new-file.txt* is empty.

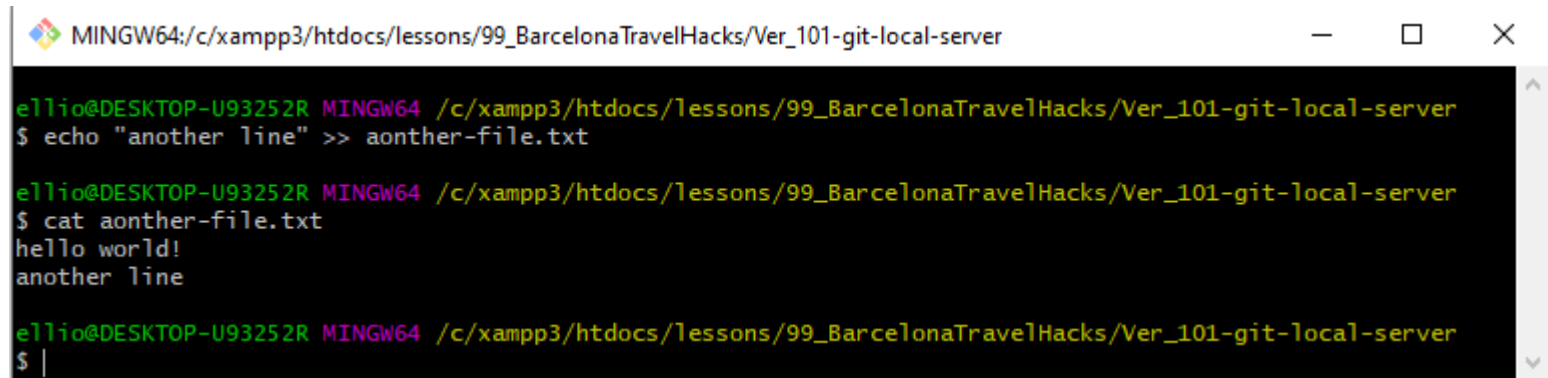
A terminal window titled 'MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server'. It shows three commands being executed: 'cat aonther-file.txt' (note the typo 'aonther'), 'cat new-file.txt', and a blank prompt. The output of the first command is 'hello world!'.

```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ cat aonther-file.txt
hello world!

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ cat new-file.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$
```

>> Using two greater than symbol means append to. So now *another-file.txt* will contain *hello world!* *Another line*.

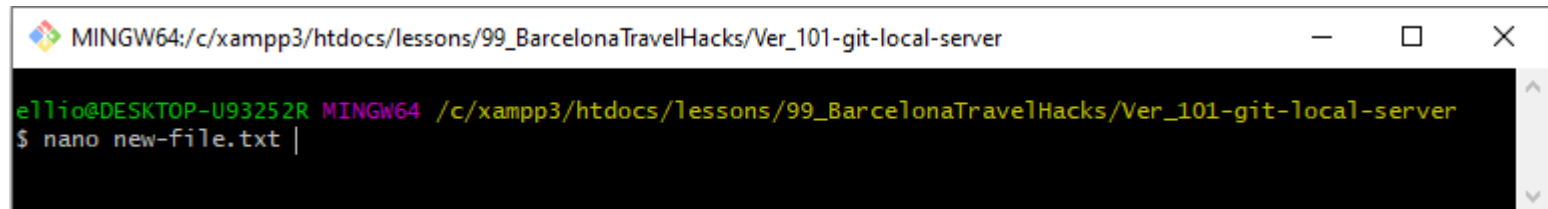
A terminal window titled 'MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server'. It shows three commands: 'echo "another line" >> aonther-file.txt' (note the typo 'aonther'), 'cat aonther-file.txt', and a blank prompt. The output of the second command is 'hello world!' followed by 'another line' on a new line.

```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ echo "another line" >> aonther-file.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ cat aonther-file.txt
hello world!
another line

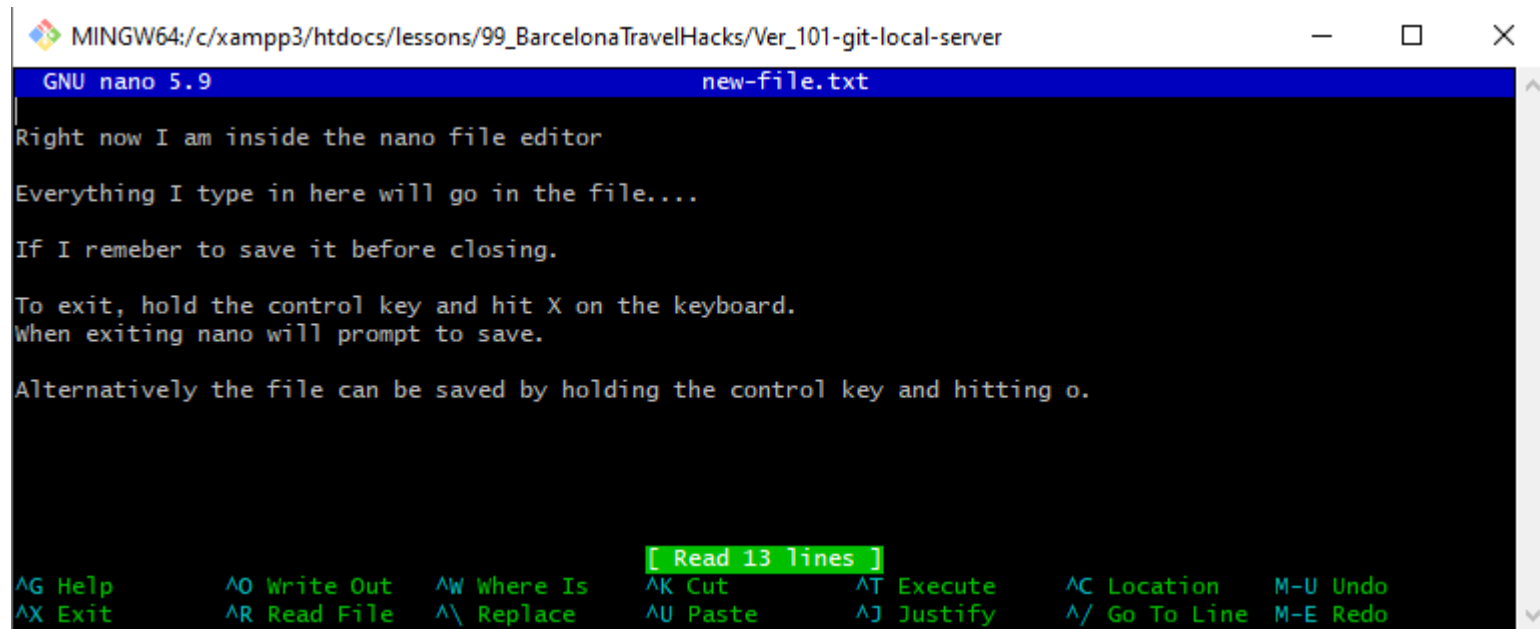
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ |
```

nano is an inbuilt file editor within the bash terminal

A terminal window titled 'MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server'. It shows the command 'nano new-file.txt' being executed, with a cursor at the end of the line.

```
MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ nano new-file.txt |
```

new-file.txt is now open within a nano terminal for editing.



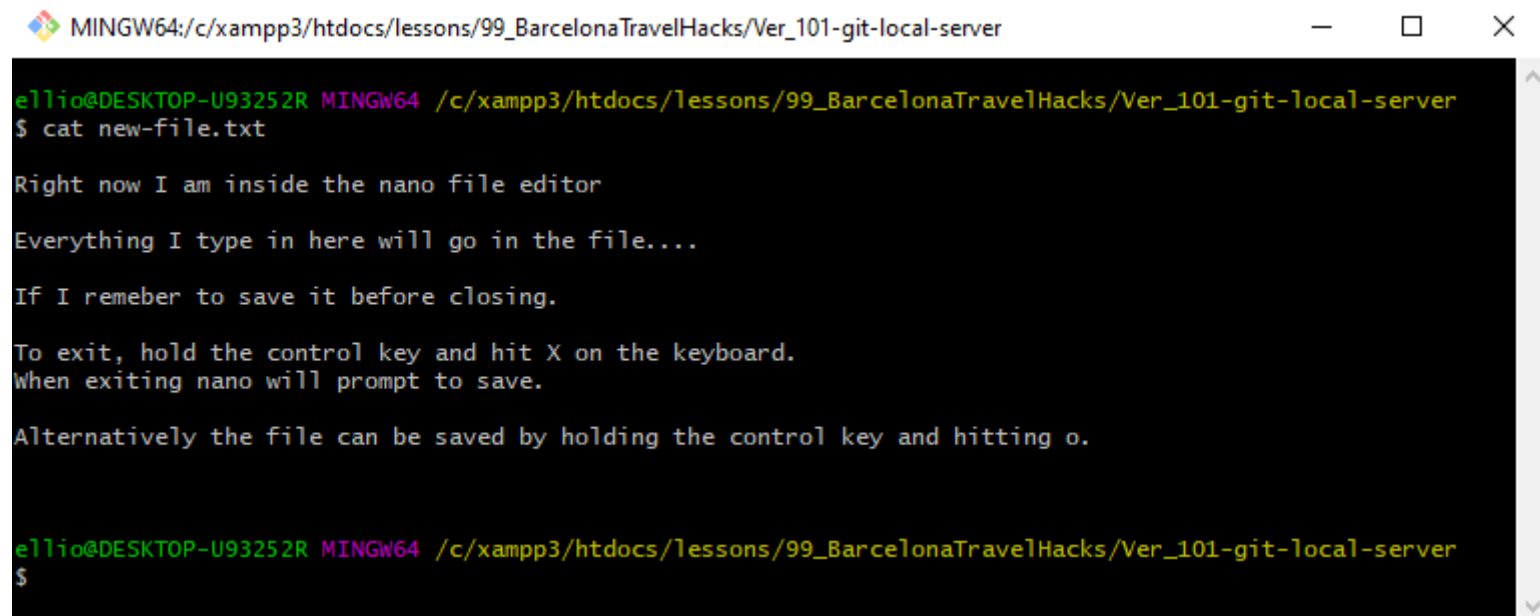
A screenshot of a Windows terminal window titled "MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server". The terminal is running the GNU nano 5.9 editor, editing a file named "new-file.txt". The file contains the following text:

```
Right now I am inside the nano file editor
Everything I type in here will go in the file...
If I remeber to save it before closing.
To exit, hold the control key and hit X on the keyboard.
When exiting nano will prompt to save.
Alternatively the file can be saved by holding the control key and hitting o.
```

The nano editor's status bar at the bottom shows "Read 13 lines" in a green box. Below this, a row of keyboard shortcuts is displayed in green text:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_/ Go To Line	M-E Redo

cat new-file.txt shows that the contents of the nano terminal has been saved to the file.

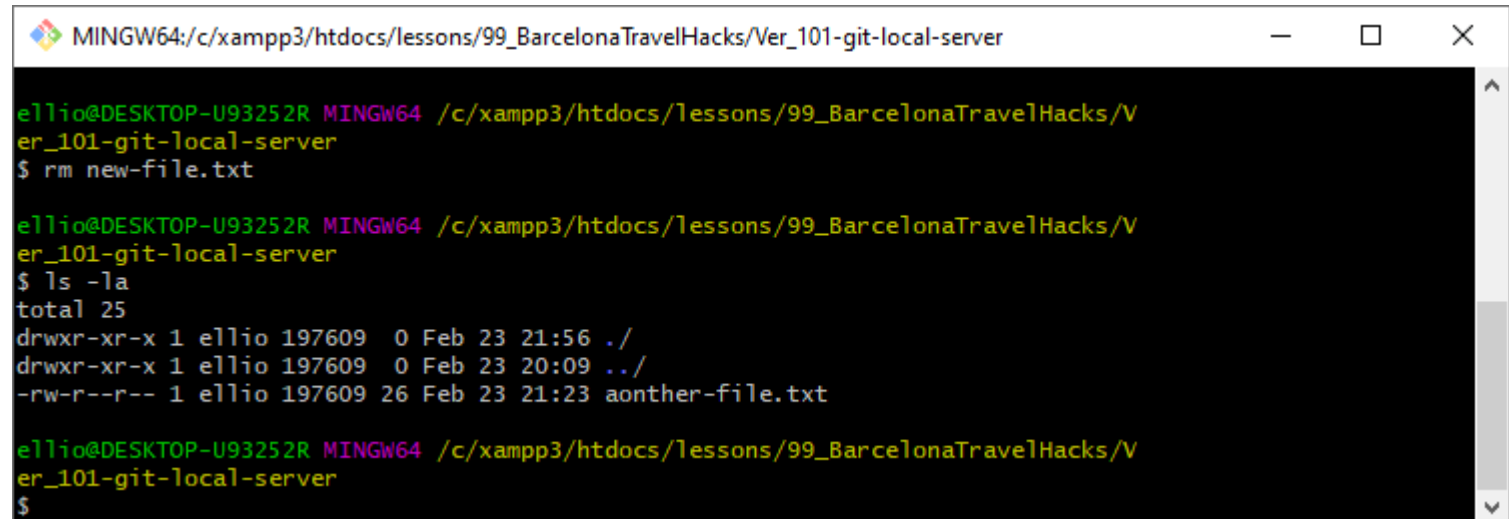


A screenshot of a Windows terminal window titled "MINGW64:/c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server". The terminal shows the command `cat new-file.txt` being executed. The output of the command is the same text that was in the nano editor window above:

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server
$ cat new-file.txt
Right now I am inside the nano file editor
Everything I type in here will go in the file...
If I remeber to save it before closing.
To exit, hold the control key and hit X on the keyboard.
When exiting nano will prompt to save.
Alternatively the file can be saved by holding the control key and hitting o.
```

The prompt `ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server $` is shown at the bottom of the terminal window.

rm This command removes (deletes) a file from the directory.



```
MINGW64:/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/Ver_101-git-local-server

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/V
er_101-git-local-server
$ rm new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/V
er_101-git-local-server
$ ls -la
total 25
drwxr-xr-x 1 ellio 197609  0 Feb 23 21:56 ./
drwxr-xr-x 1 ellio 197609  0 Feb 23 20:09 ../
-rw-r--r-- 1 ellio 197609 26 Feb 23 21:23 another-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks/V
er_101-git-local-server
$
```

Summary of Shell Commands

pwd	<i>Shows the current directory</i>
clear	<i>Clear the contents of the bash terminal</i>
.	<i>Is the alias for the current directory</i>
..	<i>Is the alias for the parent directory</i>
ls	<i>Gives a simple list of what is in the current directory</i>
ls -l	<i>Detailed list of current directory (file permissions, owner, users, date, and files or directories)</i>
ls -la	<i>Detailed list of current directory including hidden files or directories</i>
cd	<i>Change directory</i>
Cd ..	<i>Change directory to parent directory (go up one level)</i>
mkdir	<i>Make directory. This command will create a sub-directory in the current directory</i>
tab	<i>autocomplete</i>
touch	<i>Create file</i>
echo	<i>Output the contents of the echo command</i>
>	<i>Write to</i>
>>	<i>Append to</i>
cat	<i>Read contents of</i>
nano	<i>Nano text editor</i>
rm	<i>Delete file</i>
rm -rf	<i>Delete directory</i>

HOW GIT WORKS UNDER THE HOOD

Next I have used **cd ..** to go back one directory and **ls** to list all the directories.

rm is used to delete files but if I try to use same command to delete a folder it gives me a cannot remove error.

rm -rf command is used to delete directories.

Trying to **cd** to the folder gives a no such file or directory error because we have deleted the directory.

```
MINGW64:/c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ ls
Ver_100_-_Photo-metadata-fixer-and-Sitemap-Maker/  Ver_75/
Ver_101-git-local-server/                          'Ver_76 - Upgrade 14 Server Version 02-09-2021'/
'Ver_54 - upgrade Server Version 15-05-21'/        Ver_77/
Ver_55/                                             'Ver_78 - Upgrade 15 Server Version 09-09-2021'/
'Ver_56 - Live server Backup 31-05-2021'/          'Ver_79 - Live Server backup 18-10-2021'/
'Ver_57 - Upgrade 6 Server Version 19-06-2021'/    Ver_80/
'Ver_58 - Live Server Backup 24-06-2021'/          'Ver_81 - Live Server Backup 05-11-2021'/
'Ver_59 - Live Server Backup 02-07-2021'/          Ver_82/
Ver_60/                                             'Ver_83 - Live Server Backup 07-11-2021'/
'Ver_61 - Live Server Backup 12-07-2021'/          'Ver_84 - Upgrade 16 Server Version 13-11-2021'/
'Ver_61 - Upgrade 7 Server Version 12-07-2021'/    Ver_85/
'Ver_62 - Upgrade 8 Server Version 23-07-2021'/    'Ver_86 - Upgrade 17 Server Version 23-11-2021'/
'Ver_63 - Experimental'/                          'Ver_87 - Upgrade 18 Server Version 28-11-2021'/
Ver_64/                                             Ver_88/
Ver_65/                                             'Ver_89 - Upgrade 19 Server Version 05-12-2021'/
'Ver_66 - Upgrade 9 Server Version '6-08-2021"/    'Ver_90 - Photo metadata fixer'/
'Ver_67 - Upwork lite javascript gallery'/         'Ver_91 - Photo metadata fixer2'/
Ver_68/                                             Ver_92_-_Photo-metadata-fixer-and-Sitemap-Maker/
'Ver_69 - Upgrade 10 Server Version 11-08-2021'/   'Ver_93 - Live Server Backup 29-01-2022'/
'Ver_70 - Upgrade 11 Server Version 12-08-2021'/   'Ver_94 - Upgrade 20 Server Version 29-01-2021'/
Ver_71/                                             'Ver_95 - Live Server Backup 01-02-2022'/
'Ver_72 - Upgrade 12 Live Version 17-08-2021'/     'Ver_96 - Live Server Backup 02-02-2022'/
'Ver_72 - Upgrade 12 Server Version 22-08-2021'/   'Ver_97 - Live Server Backup 21-02-2021'/
'Ver_73 - Live Server Backup 29-08-2021'/          'Ver_98 - Upgrade 21 Server Version 21-02-2021'/
'Ver_74 - Upgrade 13 Server Version 30-08-2021'/   'Ver_99 - Live Server Backup 22-02-2021'/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ rm Ver_101-git-local-server/
rm: cannot remove 'Ver_101-git-local-server/': Is a directory

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ rm -rf Ver_101-git-local-server/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$ cd Ver_101-git-local-server
bash: cd: Ver_101-git-local-server: No such file or directory

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/99_BarcelonaTravelHacks
$
```

mkdir Is short for make directory. In this case I have created the directory **first-project** within the `c/xampp3/htdocs/lessons/07_GIT` directory.

cd to move into the first-project directory and a **ls -la** shows the directory as empty.

Git init Command will initialise the git repository.

ls -la Will show the git repository directory. Note that the **git repository is a HIDDEN FOLDER** and so will not show up with a **ls** command.

Note: The bash terminal prompt changes to (master) because we are now on the master branch

Initialise New GIT Repository

```
MINGW64/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons
$ ls
01_HTML_CSS/  04_JAVASCRIPT/  07_GIT/
02_PROC_PHP/  05_JQUERY/      98_PHP_DATABASES/
03_OOP_PHP/   06_AJAX/        99_BarcelonaTravelHacks/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons
$ cd 07_GIT/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls -la
total 16
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:13 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:03 ../

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ mkdir first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd first-project/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project
$ ls -la
total 0
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/first-project/.git/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 .git/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

Note: However, moving into the same directory from a normal terminal will not show the master branch.

Overview of GIT Repository

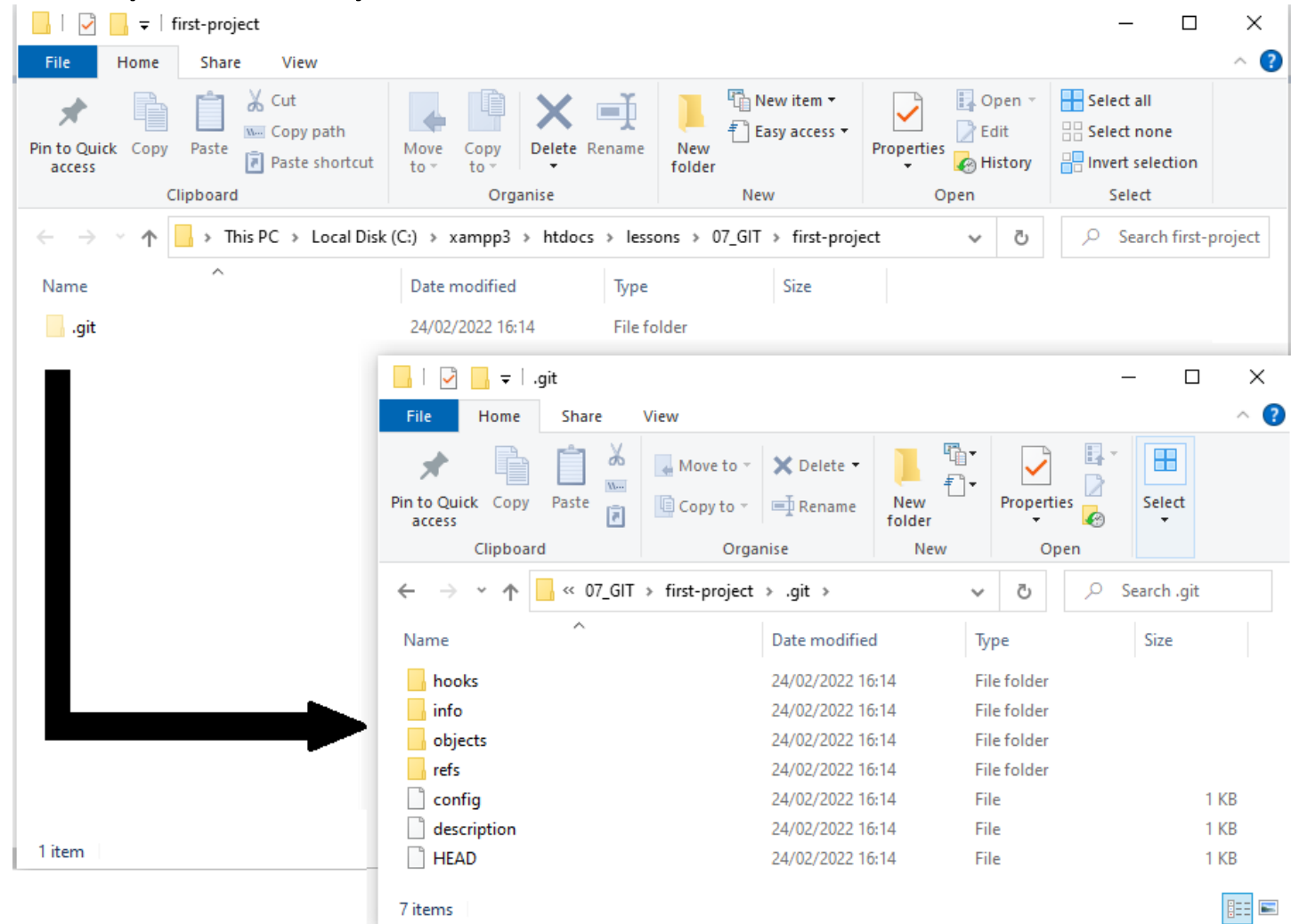
Explorer . From git bash terminal Will open the directory in windows explorer where we will see a hidden git folder with sub folders and files.

IMPORTANT: Never move or modify the git folder or its contents from explorer. It will be completely managed by git.

```
Command Prompt

c:\> cd xampp3\htdocs\lessons\07_GIT\first-project

c:\xampp3\htdocs\lessons\07_GIT\first-project>
```



To enter the git directory use **cd .git**. Note that the git directory is hidden so we have to denote this with a dot before the directory name.

ls -la Will list the contents of the .git directory and a **cat config** will show the contents of the config file.

cat description File is empty and says unnamed directory because the repository does not have a name.

cat HEAD File only contains the default refs/heads/master.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project/.git
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ ls -la
total 11
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 ../
-rw-r--r-- 1 ellio 197609 23 Feb 24 16:14 HEAD
-rw-r--r-- 1 ellio 197609 130 Feb 24 16:14 config
-rw-r--r-- 1 ellio 197609  73 Feb 24 16:14 description
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 hooks/
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 objects/
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 refs/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ cat config
[core]
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ cat description
Unnamed repository; edit this file 'description' to name the repository.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ cat HEAD
ref: refs/heads/master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$
```

GIT has its own file system which stores objects in the objects folder. There are four object types in GIT which allow it to perform file tracking, change tracking etc.

Blob

In blob object types Git stores files, any file types such as video, pictures, text, etc. any file types.

Tree

With help of tree object type git stores information about directories. Directory may be empty or contain files or a mix of files and directories. Directory may be a blob or tree.

Commit

Commit object type stores different versions of our project.

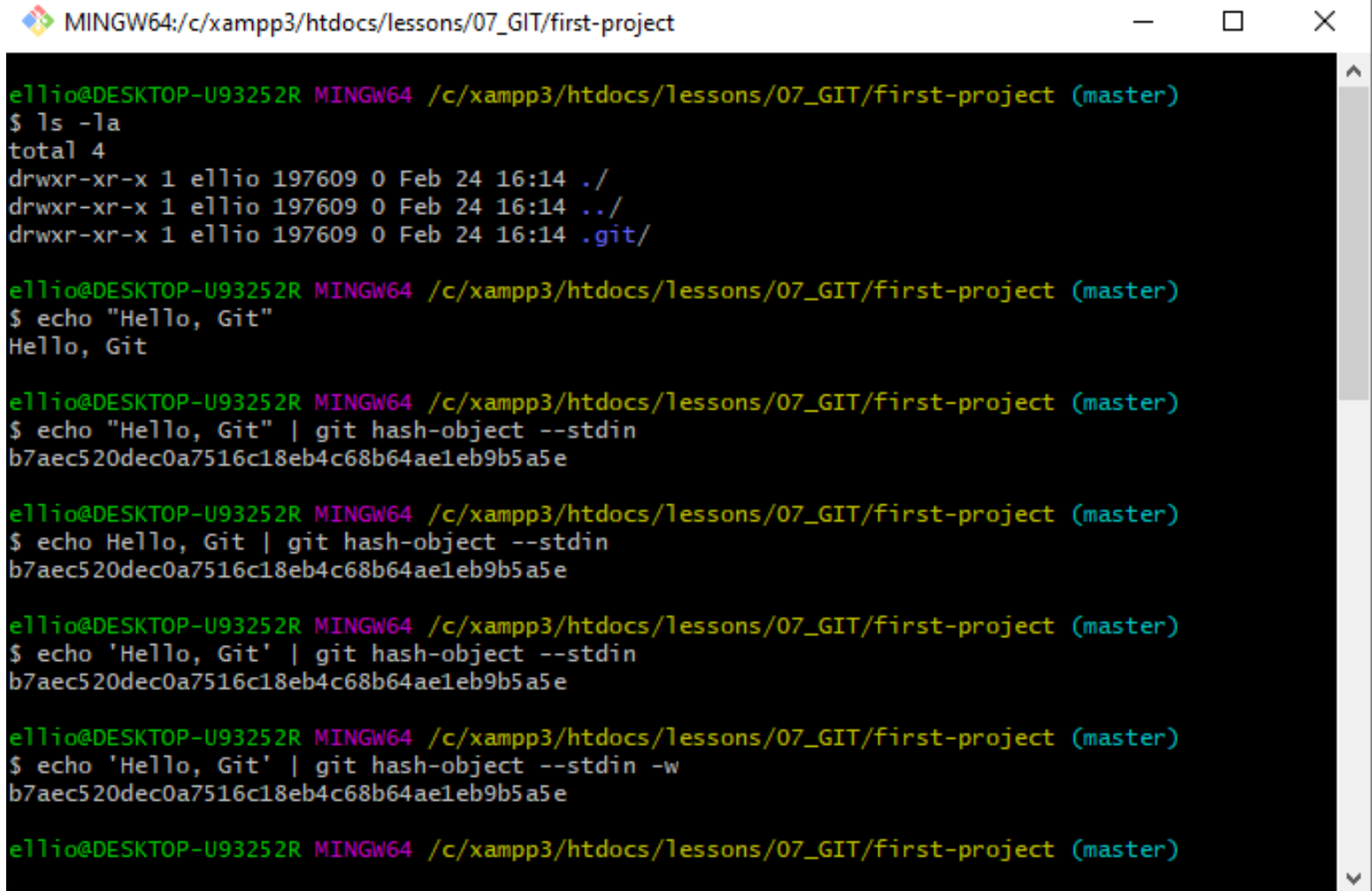
Annotated Tag

Annotated tag is persistent text pointer to specific commit.

For management of Blob and tree we will use low level git commands such as **git-hash-object** or **git cat-file**

Writing of new git object using git hash-object

The first-project folder contains a git repository as seen with a **ls -la** command. **git hash-object** Command will produce a hash of the *echo "Hello, Git"*. Note that no quotes or single quotes can be used. We use the *pipe* to say make a hash of what is before the pipe. **--stdin** signifies standard input and **-w** signifies write.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 .git/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ echo "Hello, Git"
Hello, Git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ echo "Hello, Git" | git hash-object --stdin
b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

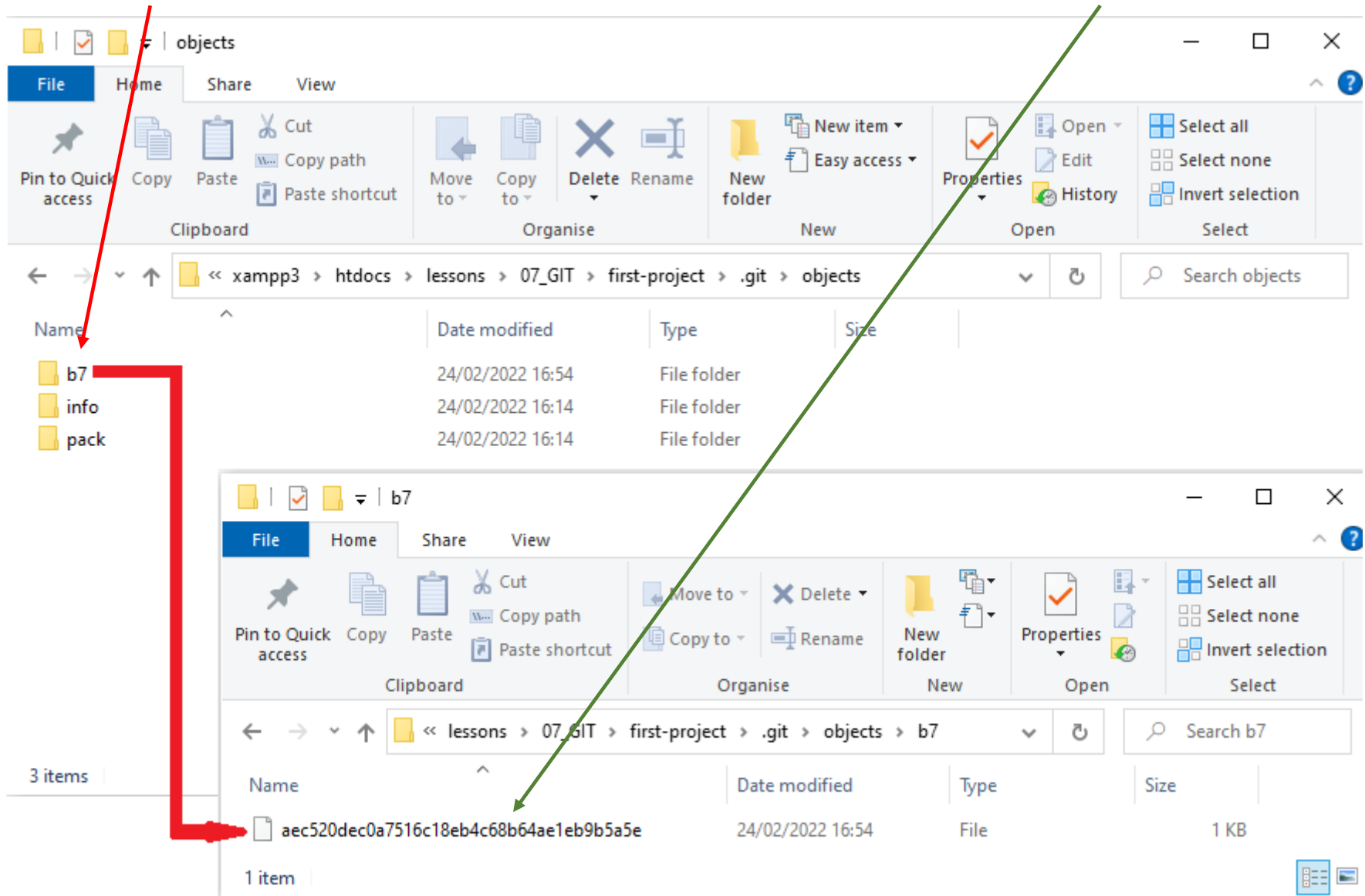
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ echo Hello, Git | git hash-object --stdin
b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ echo 'Hello, Git' | git hash-object --stdin
b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

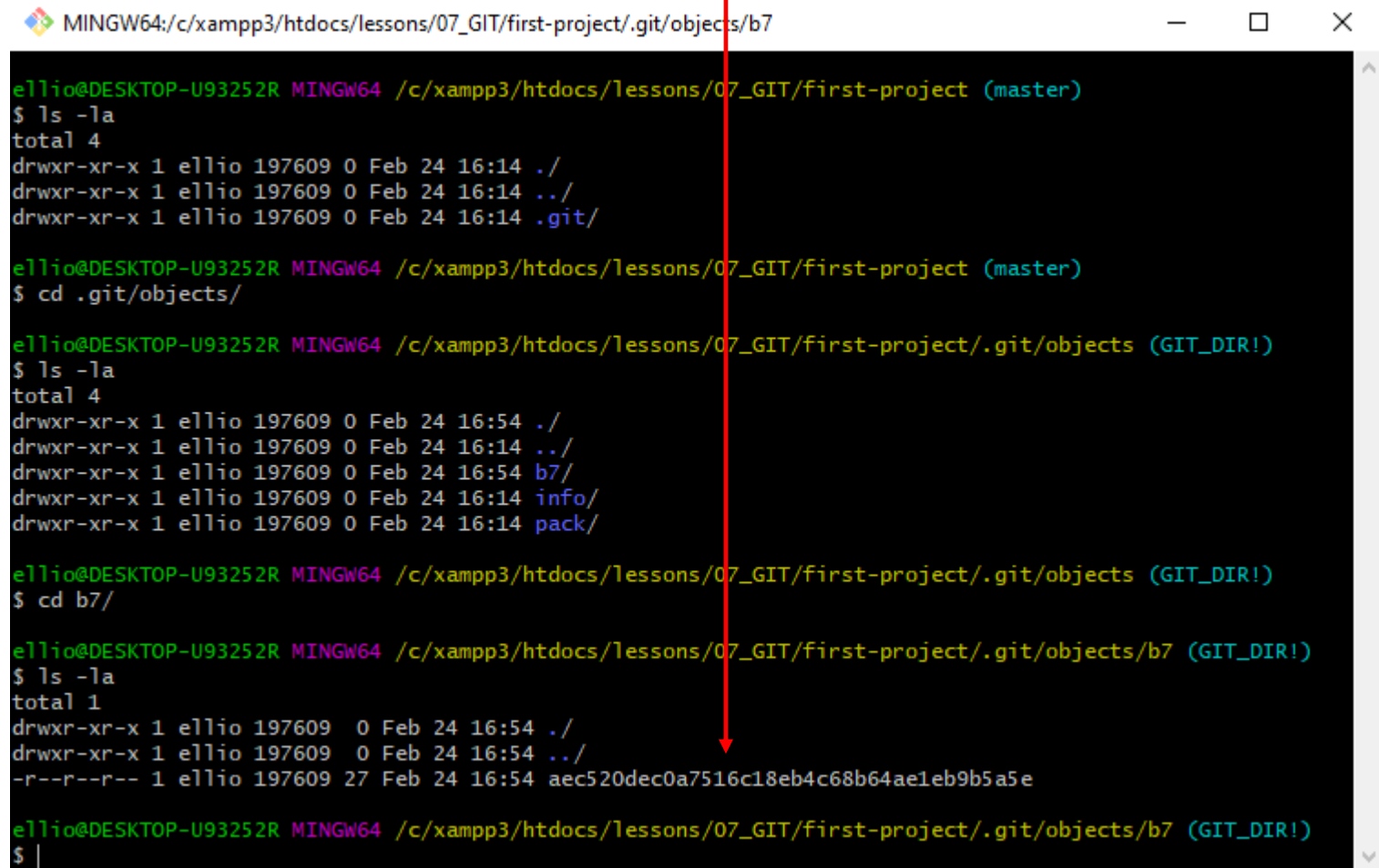
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ echo 'Hello, Git' | git hash-object --stdin -w
b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
```


The git hash-object command gave us the hash of **703a16bf2247dc35c60b5c2aa6a8783dc6222b5d**
Note: the **first two hex** are the folder name and the remaining hash is the **file name** in that folder.



The new hash file was not created in the main folder but in separate folder within the objects folder of the git directory.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/b7
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 .git/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git/objects/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 b7/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 pack/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$ cd b7/

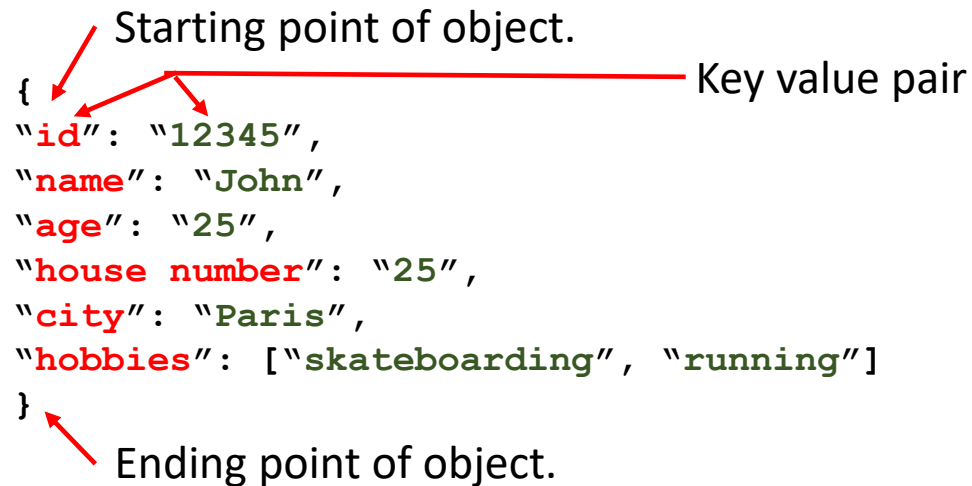
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/b7 (GIT_DIR!)
$ ls -la
total 1
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 ../
-r--r--r-- 1 ellio 197609 27 Feb 24 16:54 aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/b7 (GIT_DIR!)
$ |
```

I use `cd ..` Three times to return to the master level of the directory.

JSON vs Git Database

JSON is Javascript object notation using key value pairs to store data in javascript friendly form.



The diagram shows a JSON object with several annotations. A red arrow points from the text "Starting point of object." to the opening curly brace '{'. Another red arrow points from the text "Ending point of object." to the closing curly brace '}'. A red line connects the text "Key value pair" to the first key-value pair '"id": "12345"'. The JSON object is as follows:

```
{  
  "id": "12345",  
  "name": "John",  
  "age": "25",  
  "house number": "25",  
  "city": "Paris",  
  "hobbies": ["skateboarding", "running"]  
}
```

Each **key** is unique across entire JSON object.

Values can be the same across JSON keys.

Git also stores keys and values.

Keys in Git unique like in JSON. Git keys are equal to the hash of each object. Folder name + file name = hash.

Values in Git cannot be equal because keys are generated from values. – the hash.

What is hash?

Any variable length input (such as text, video file, image file etc.)

HASH FUNCTION is one way function. i.e. a hash can be derived from an input but an input cannot be derived from a hash.

HASH FUNCTION

Same input will always produce same HASH

A small change of input will produce a completely different hash.

Fixed length hash i.e. b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

Common hash functions:

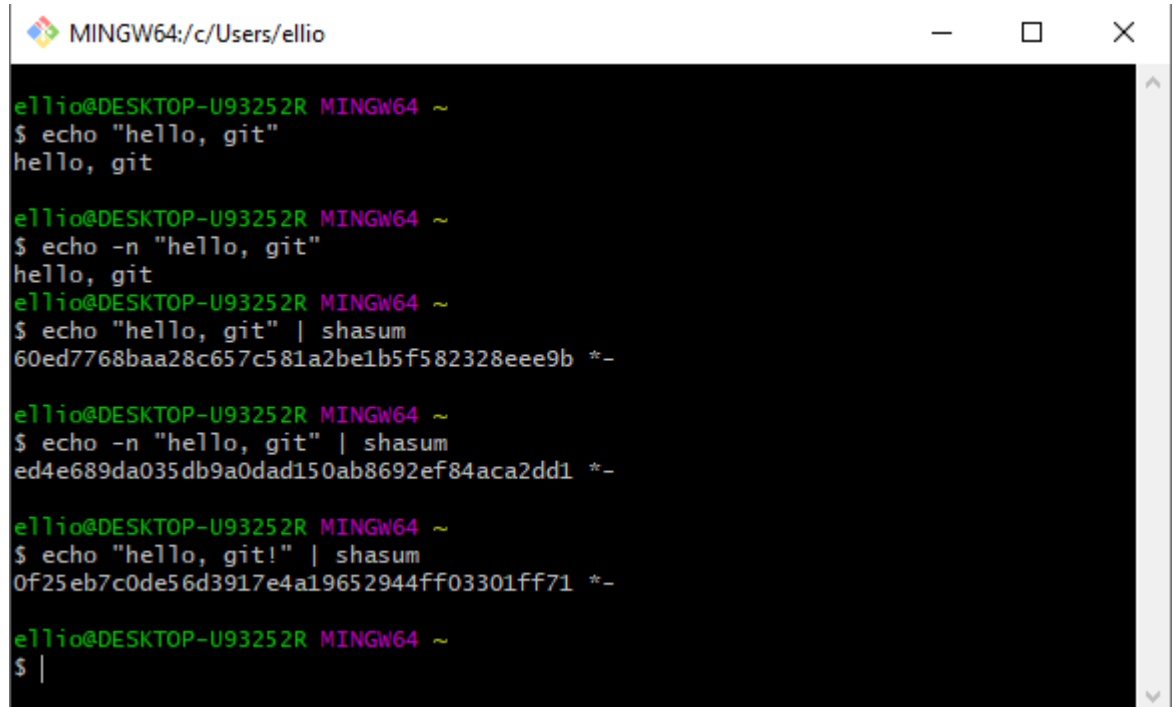
- MD5: 128 bit long hash
- SHA1: 160 bit long hash**
- SHA256: 256 bit long hash
- SHA384: 384 bit long hash
- SHA512: 512 bit long hash

**Git uses SHA1 hash
which is 40 hex long**

In a new terminal window we can **echo "Hello, Git"** and using the utility **shasum** we can generate a sha1 hash.

Note that **echo "hello, Git!"** produces a completely different sha1 hash.

By adding **-n echo "hello, git!"** I am specifying no line break at the end of **"hello, git!"** Echo automatically adds line break at the end of the text. This also produces a completely different sha1 hash.

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/ellio'. The terminal shows a series of commands and their outputs. The first command is '\$ echo "hello, git"', which outputs 'hello, git'. The second command is '\$ echo -n "hello, git"', which also outputs 'hello, git'. The third command is '\$ echo "hello, git" | shasum', which outputs a long hexadecimal string '60ed7768baa28c657c581a2be1b5f582328eee9b *'. The fourth command is '\$ echo -n "hello, git" | shasum', which outputs a different hexadecimal string 'ed4e689da035db9a0dad150ab8692ef84aca2dd1 *'. The fifth command is '\$ echo "hello, git!" | shasum', which outputs another hexadecimal string '0f25eb7c0de56d3917e4a19652944ff03301ff71 *'. The terminal prompt '\$ |' is visible at the bottom.

```
ellio@DESKTOP-U93252R MINGW64 ~
$ echo "hello, git"
hello, git

ellio@DESKTOP-U93252R MINGW64 ~
$ echo -n "hello, git"
hello, git

ellio@DESKTOP-U93252R MINGW64 ~
$ echo "hello, git" | shasum
60ed7768baa28c657c581a2be1b5f582328eee9b *-

ellio@DESKTOP-U93252R MINGW64 ~
$ echo -n "hello, git" | shasum
ed4e689da035db9a0dad150ab8692ef84aca2dd1 *-

ellio@DESKTOP-U93252R MINGW64 ~
$ echo "hello, git!" | shasum
0f25eb7c0de56d3917e4a19652944ff03301ff71 *-

ellio@DESKTOP-U93252R MINGW64 ~
$ |
```

How many files can Git store in the same repository and what is the chance of producing files with the same hash? i.e. hash collision.

SHA1 has 160 bits so the answer is a possible 2 to the power of 160 number of files. The number is so huge that there is no need to worry about git running out of possible hashes, even if the project has millions of files.

To understand probability of hash collision think about dice probability. The probability of throwing two dice and getting the same number is $(1/6) \times (1/6)$ which is about 16%. For SHA1 the formula is $(1/2 \text{ to power } 160) \times (1/2 \text{ to power } 160)$ which is $\frac{1}{2} \text{ to power } 320$ or 0. 96 zeros then some numbers. So small that it will almost never occur.

Reading git object using git cat-file

git cat-file -p <hash> Command will printout to screen the contents of the hash file.

git cat-file -s <hash> Command will show the size of the file in bytes

git cat-file -t <hash> Command will show the type of file in the repository, in this case a blob.

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
```

```
$ git cat-file -p b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e
```

```
Hello, Git
```

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
```

```
$ git cat-file -s b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e
```

```
11
```

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
```

```
$ git cat-file -t b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e
```

```
blob
```

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
```

```
$ git cat-file -t b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5f
```

```
fatal: git cat-file: could not get object info
```

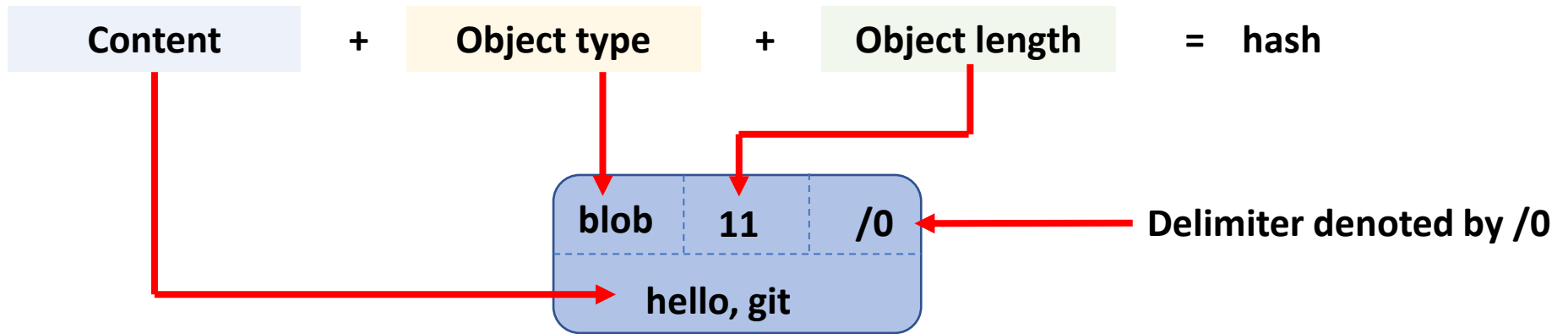
```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
```

```
$ |
```

Note: The file has to exist in the repository or we will get a fatal error.

Git blobs do not store filenames. Files are stored as a hash. Folder is first two hex digits of hash and the file is named after the remaining 38 hex digits of the hash.

Contents of git object



It does not matter how git object was created there are always 4 field inside of actual git object in git database.

b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

Note that our previous blob was created from “Hello, Git” with a file size of 11 bytes. Using shasum utility and manually adding blob, size and delimiter we get the same hash thus proving that the hash is made of blob, object type and delimiter with the contents. Note the `-e` is needed to escape slashed character.

```
MINGW64:/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 ~
$ echo -e "blob 11\0Hello, Git" | shasum
b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e *-
ellio@DESKTOP-U93252R MINGW64 ~
$ |
```

Create new git object using git hash-object

Previously we used **echo "Hello, Git" | git hash-object --stdin -w** to create a blob with contents of Hello, Git but we can also create a git object based on existing file.

In a sperate terminal a new file is created n the **c/user/ellio directory** with a name **new-file.txt** and contents **"Second file in Git repository"**.

Is shows the file in the **c/user/ellio directory** and not the git directory.

```
MINGW64:/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 ~
$ echo "Second file in Git repository" > new-file.txt
ellio@DESKTOP-U93252R MINGW64 ~
$ ls
3D Objects/
AppData/
Application Data/@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
IntelGraphicsProfiles/
Links/
Local Settings'@
Music/
My Documents'@
NTUSER.DAT
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer
00000000000000000001.regtrans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer
00000000000000000002.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
Recent@
Saved Games'/
Searches/
SendTo@
Start Menu'@
Templates@
Videos/
jexiftoolgui_data/
new-file.txt
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
temp-tree.txt
ellio@DESKTOP-U93252R MINGW64 ~
$ pwd
/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 ~
```

git hash-object <file path and file name> Command will generate a hash for that file.

To write it to the repository we append **-w** to the command.

If we **cd/.git/objects** into the objects if the git repository we can see two folders named after the first two hex digits of our hashed objects.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git hash-object /c:/users/ellio/new-file.txt
4400aae52a27341314f423095846b1f215a7cf08

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git hash-object /c:/users/ellio/new-file.txt -w
warning: LF will be replaced by CRLF in C:/users/ellio/new-file.txt.
The file will have its original line endings in your working directory
4400aae52a27341314f423095846b1f215a7cf08

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 4400aae52a27341314f423095846b1f215a7cf08
Second file in Git repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -t 4400aae52a27341314f423095846b1f215a7cf08
blob

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -s 4400aae52a27341314f423095846b1f215a7cf08
30

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git/objects/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 44/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 b7/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 pack/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$
```

The original file can be deleted but the git repository still retains the new file because it stores it in the objects folder independently of existing directories.

```
MINGW64:/c:/Users/el...

ellio@DESKTOP-U93252R MINGW64 ~
$ rm new-file.txt
```


If we try to read the contents of a git file using the cat command it returns some weird symbols because git uses compressed binary to store files.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/44

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git/objects/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 44/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 b7/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 pack/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects (GIT_DIR!)
$ cd 44

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/44 (GIT_DIR!)
$ ls -la
total 1
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 ../
-r--r--r-- 1 ellio 197609 46 Feb 24 18:08 00aae52a27341314f423095846b1f215a7cf08

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/44 (GIT_DIR!)
$ cat 00aae52a27341314f423095846b1f215a7cf08
xKOR06NMKQH IU Sp,Q(J-/,/,/?
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/objects/44 (GIT_DIR!)
$
```

However the hash can be read to find the contents, type (blob) and size of the file in the git repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

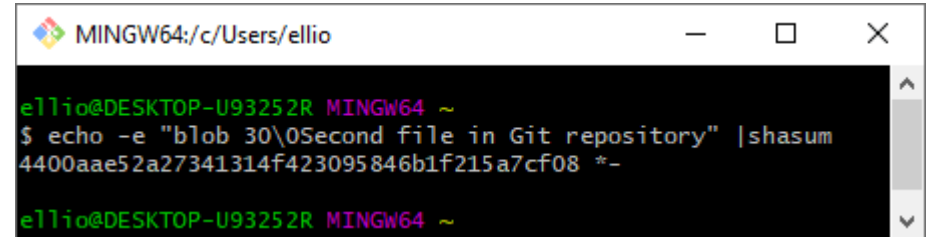
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 4400aae52a27341314f423095846b1f215a7cf08
Second file in Git repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -t 4400aae52a27341314f423095846b1f215a7cf08
blob

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -s 4400aae52a27341314f423095846b1f215a7cf08
30

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

The second file in git repository has content of “Second file in Git repository”. It has length of 30 bytes and is a blob.

A terminal window titled 'MINGW64:/c/Users/ellio' with standard window controls. The prompt is 'ellio@DESKTOP-U93252R MINGW64 ~'. The command entered is '\$ echo -e "blob 30\0Second file in Git repository" |shasum'. The output is '4400aae52a27341314f423095846b1f215a7cf08 *-'.

```
MINGW64:/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 ~
$ echo -e "blob 30\0Second file in Git repository" |shasum
4400aae52a27341314f423095846b1f215a7cf08 *-
ellio@DESKTOP-U93252R MINGW64 ~
```

Recap: we have two blobs in the objects directory. Note: blobs do not store filenames.

4400aae52a27341314f423095846b1f215a7cf08

Content: Second file in Git repository

Type: blob

Length: 30

b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

Content: Hello, git

Type: blob

Length: 11

Git Tree Objects

Git tree objects have four fields:

```
100644 blob 57537e1d8fba7d80c5bcca8b04e49666b1c1790f
100644 blob 1fed445333e85fb9996542978fa56866de90a2fb
040000 tree c167c406a72fd6f3ab232b905cc730588167340e
100644 blob cb82ecb6e7b8f3d4be8f188fc495063aa5ae0c48
040000 tree 5ddb0cfa6f45d4bce18247297b89353c9ae47af5
100644 blob 8ce3a3772a346a19afc3bc9ee3a7fa999b36c5d0
100644 blob 7fdb8f594ccda44eba69ea7ce95ea84c93bbfac0
```

↑
Permissions

↙
type

↑
SHA1 hash

.babelrc
.flowconfig
dist
package.json
src
yarn-error.log
yarn.lock
↑
filename

Git object permissions

040000	Directory, i.e a tree
100644	Regular Non-Executable file
100664	Regular Non-Executable group writable file
100755	Regular Executable file
120000	Symbolic link
160000	Gitlink

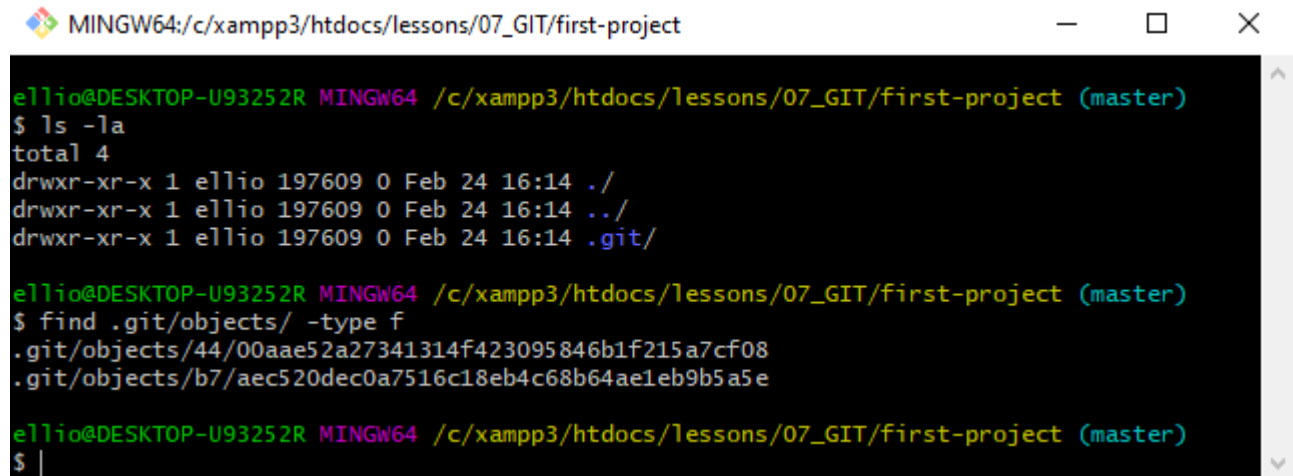
Permissions are needed because the git repository is independent of computer file system. Git is a separate file system so permissions are needed.

Creating Git Tree Object

We want to build the below tree for the two existing blobs in our git project:

```
100644 blob 4400aae52a27341314f423095846b1f215a7cf08    file2.txt
100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e    file1.txt
```

ls -la command shows that we have a git folder and **find .git/objects -type f** command finds all type files within the .git/objects directory.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 .git/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ find .git/objects/ -type f
.git/objects/44/00aae52a27341314f423095846b1f215a7cf08
.git/objects/b7/aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

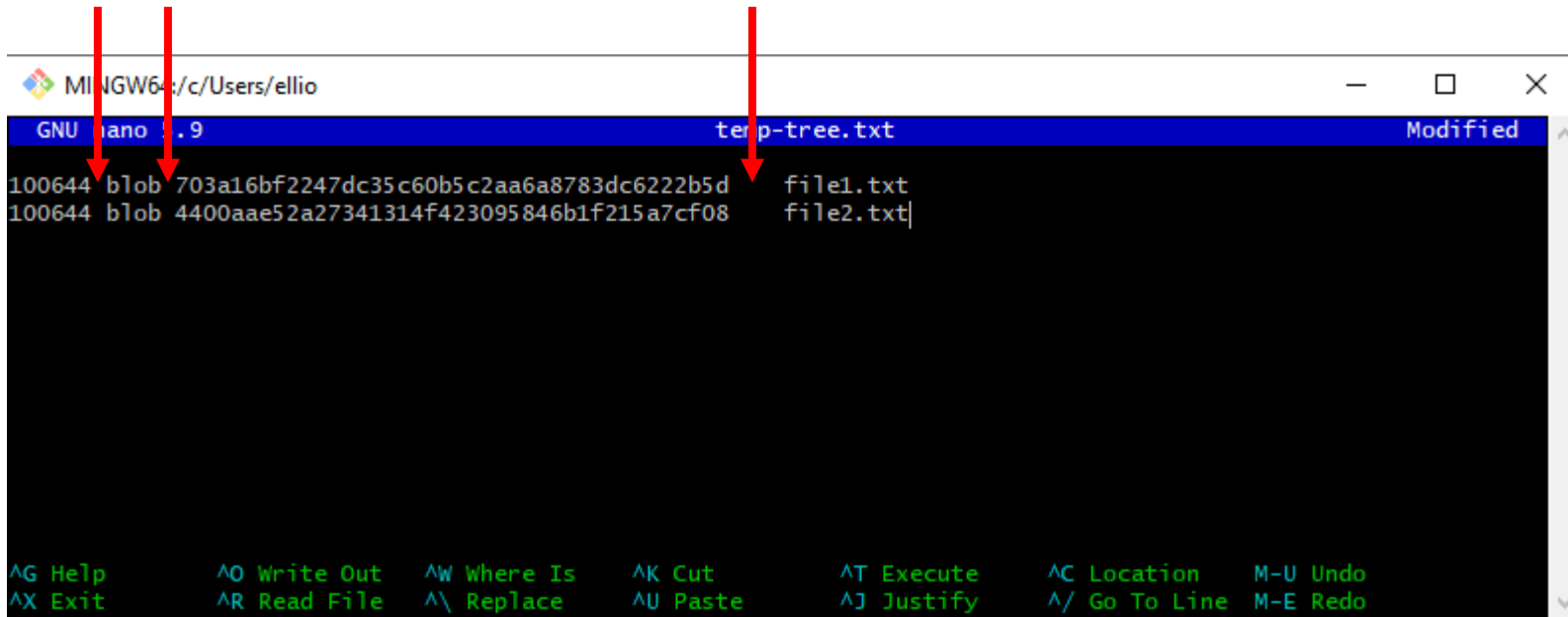
We can create a file called temp-tree.txt in nano editor.



```
MINGW64:/c/Users/ellio
ellio@DESKTOP-U93252R MINGW64 ~
$ nano temp-tree.txt
```

Space Space

Tab



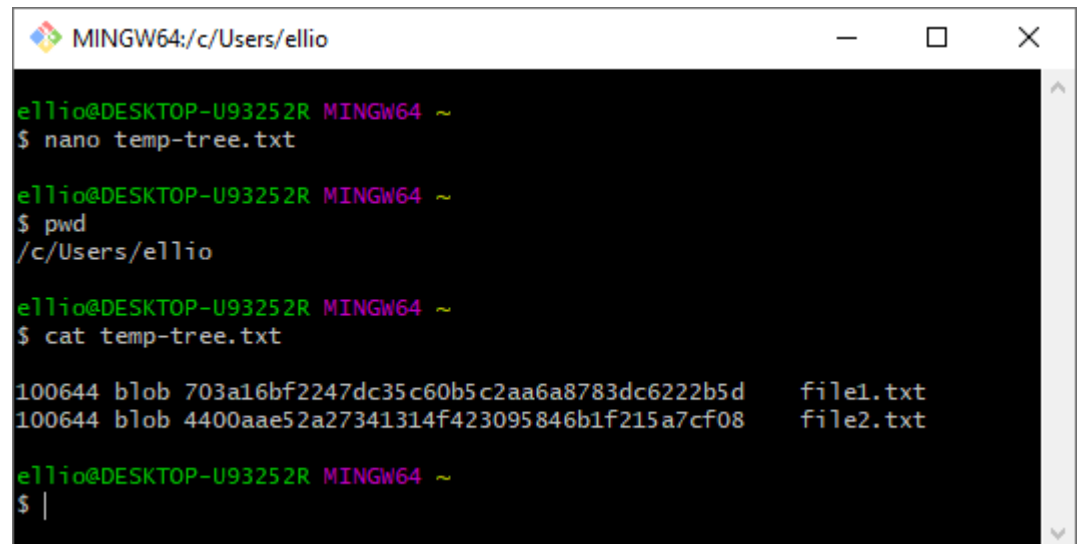
A screenshot of the nano text editor window. The title bar shows 'MINGW64:/c/Users/ellio'. The editor is editing a file named 'temp-tree.txt'. The content of the file is as follows:

```
100644 blob 703a16bf2247dc35c60b5c2aa6a8783dc6222b5d file1.txt
100644 blob 4400aae52a27341314f423095846b1f215a7cf08 file2.txt|
```

At the bottom of the window, there is a status bar with various shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, ^_ Go To Line, M-E Redo.

Control and O to save the file then control and X to exit nano editor.

Now there is a tree file in the c/users/ellio directory that can be viewed with a **cat temp-tree.txt** command.



A screenshot of a terminal window titled 'MINGW64:/c/Users/ellio'. It shows the following sequence of commands and output:

```
ellio@DESKTOP-U93252R MINGW64 ~
$ nano temp-tree.txt

ellio@DESKTOP-U93252R MINGW64 ~
$ pwd
/c/Users/ellio

ellio@DESKTOP-U93252R MINGW64 ~
$ cat temp-tree.txt

100644 blob 703a16bf2247dc35c60b5c2aa6a8783dc6222b5d file1.txt
100644 blob 4400aae52a27341314f423095846b1f215a7cf08 file2.txt

ellio@DESKTOP-U93252R MINGW64 ~
$ |
```

We can see the contents of the temp-tree.txt file in the c/users/ellio directory using the cat command.

We can then create this tree in the repository using a **git mktree** command and we see that it has been created in the objects folder with a hash.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat /c/users/ellio/temp-tree.txt
100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e    file1.txt
100644 blob 4400aae52a27341314f423095846b1f215a7cf08    file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat /c/users/ellio/temp-tree.txt | git mktree
3b95df0ac6365c72e9b0ff6c449645c87e6e1159

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ find .git/objects/ -type f
.git/objects/3b/95df0ac6365c72e9b0ff6c449645c87e6e1159 ← Tree
.git/objects/44/00aae52a27341314f423095846b1f215a7cf08
.git/objects/b7/aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

Now that the **tmp-tree.txt** is in the repository we can delete it with a **rm temp-tree.txt** from the c/users/ellio directory.

```
MINGW64:/c/Users/...

ellio@DESKTOP-U93252R MINGW64 ~
$ rm temp-tree.txt
```

cat command can be used to view the contents, type (tree) and size of the hash.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 3b95df0ac6365c72e9b0ff6c449645c87e6e1159
100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e    file1.txt
100644 blob 4400aae52a27341314f423095846b1f215a7cf08    file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -t 3b95df0ac6365c72e9b0ff6c449645c87e6e1159
tree

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -s 3b95df0ac6365c72e9b0ff6c449645c87e6e1159
74

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

3b95df0ac6365c72e9b0ff6c449645c87e6e1159

Content:

100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

100644 blob 4400aae52a27341314f423095846b1f215a7cf08

Type:

tree

Length:

102

file1.txt

file2.txt

b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

Content:

Hello, git

Type:

blob

Length:

11

4400aae52a27341314f423095846b1f215a7cf08

Content:

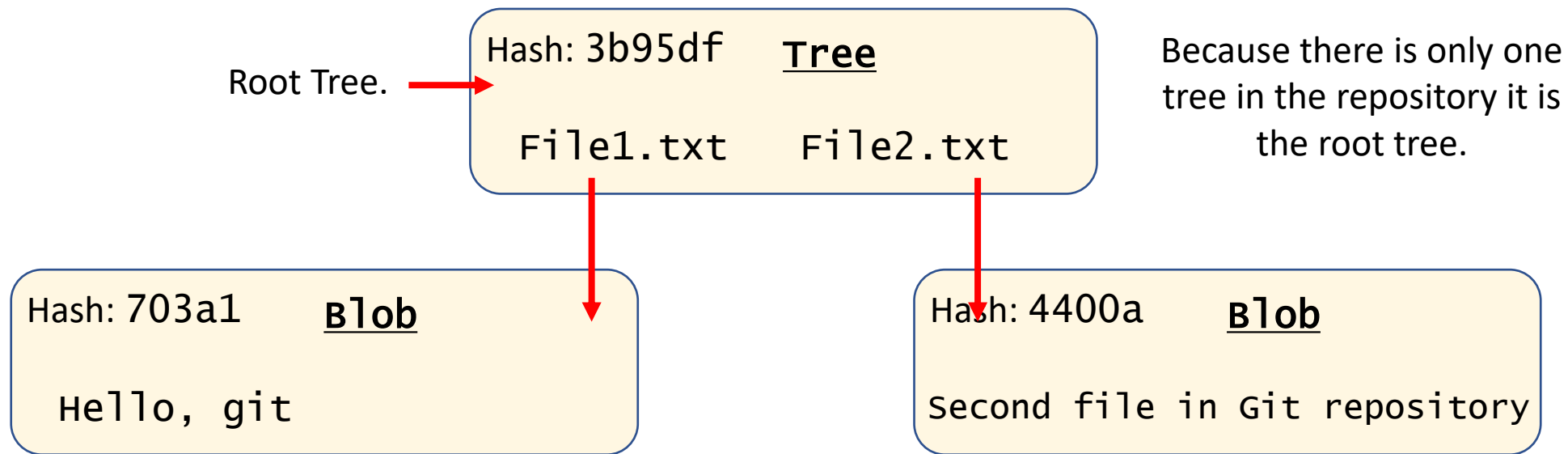
Second file in Git repository

Type:

blob

Length:

30

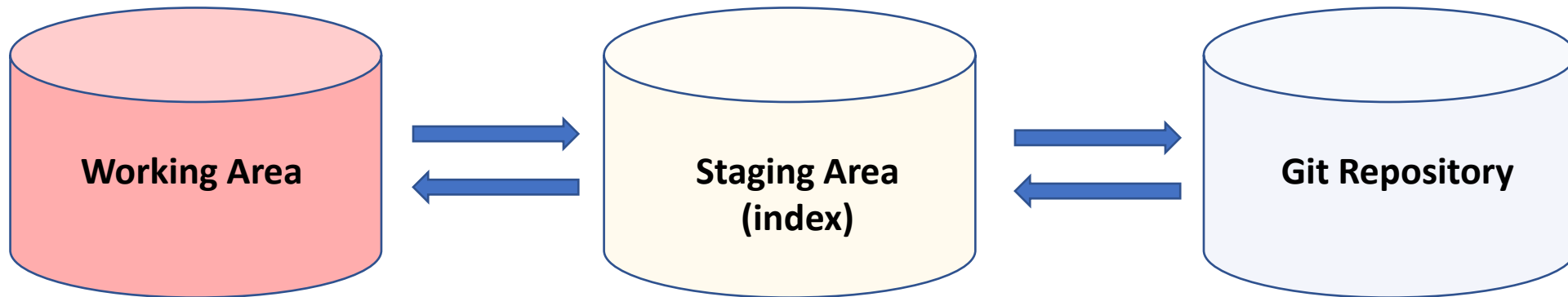


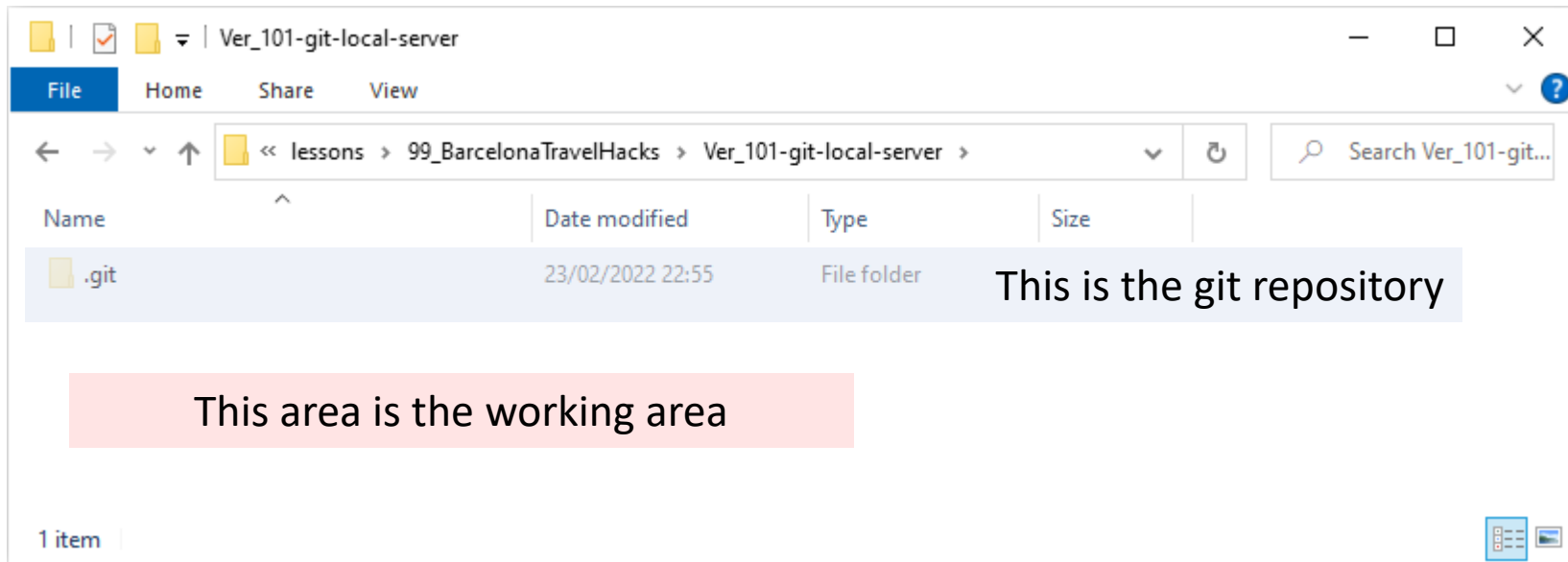
Working Area, Staging Area and Git Repository

This is where files are worked on

This is an intermediary between the depository and the Working Area

This is the warehouse for the project





git ls-files Command can be used to view the files in the Staging Area (index)

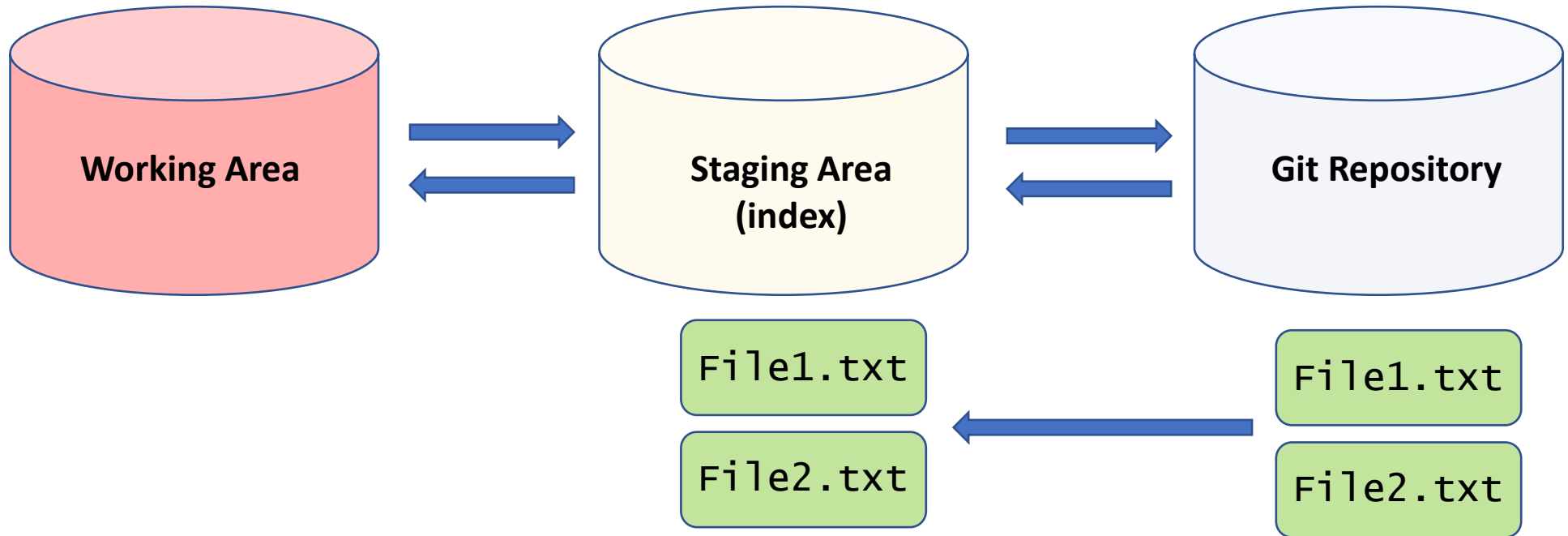
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

So at the moment there are no files in the Working Area or the Staging Area but the repository contains 2 files, 3 files including the tree.

Git read tree



git read-tree Command can be used to move files from git repository to staging area.

Note that we need to put in the hash of the tree.

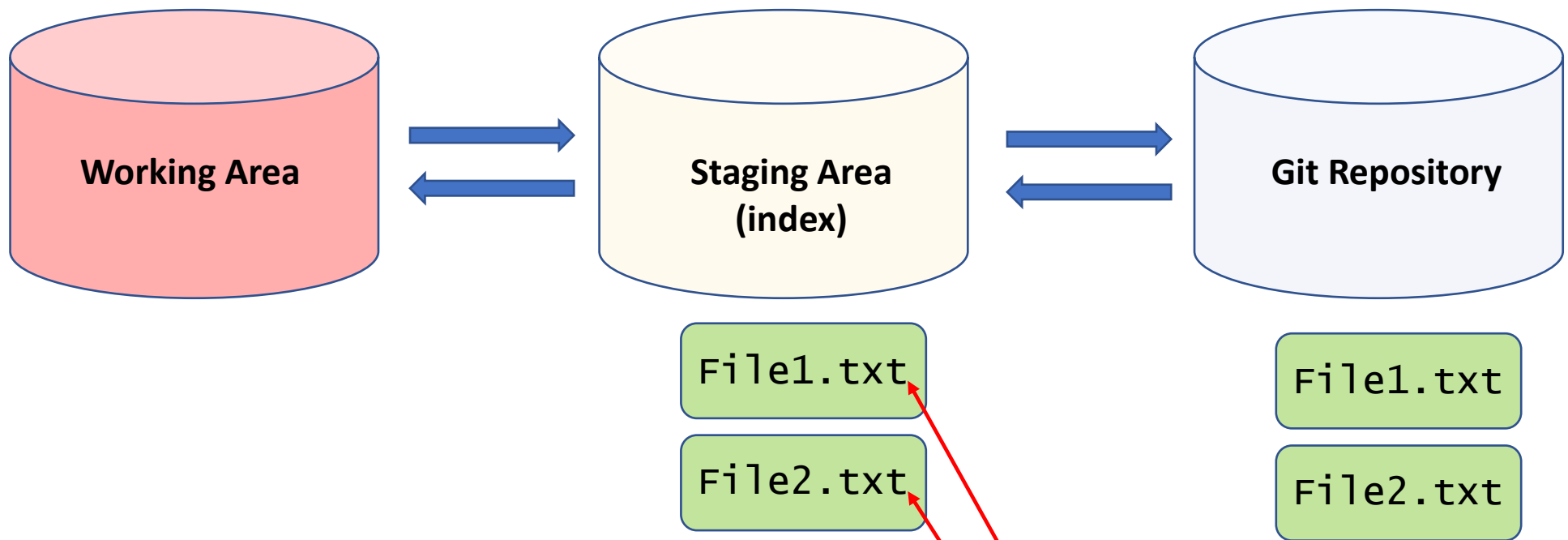
```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ find .git/objects/ -type f
.git/objects/3b/95df0ac6365c72e9b0ff6c449645c87e6e1159
.git/objects/44/00aae52a27341314f423095846b1f215a7cf08
.git/objects/b7/aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file 3b95 -t
tree

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git read-tree 3b95

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



With a **git ls-files** command we can read the Staging area. By adding a **-s** to this command we get a more detailed table output.

File permissions.

File Hash.

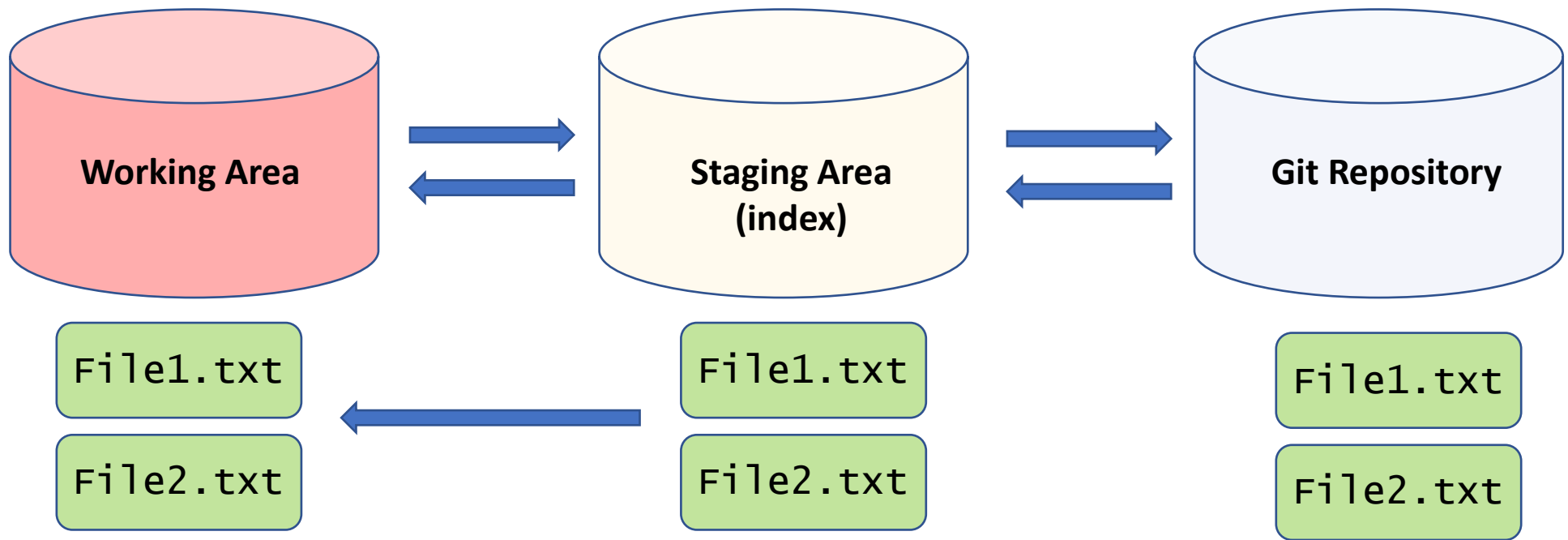
Zero means the file is the same as in the repository.

File name.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files
file1.txt
file2.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



In the Working Area we only have the git repository.

Using **git checkout-index -a** command we can copy *all* files from the staging area to the working area.

Now we see the two blob files have been copied from the Staging Area (index) to the working Area.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 4
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 19:14 .git/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout-index -a

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la
total 6
drwxr-xr-x 1 ellio 197609 0 Feb 24 19:28 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 19:14 .git/
-rw-r--r-- 1 ellio 197609 12 Feb 24 19:28 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 24 19:28 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

We can read the files in the working area using the cat command.

A ls -l command will show the files or directories with:

Permissions

Owner

Group of users

Date

Name

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat file1.txt
Hello, Git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat file2.txt
Second file in Git repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 2
-rw-r--r-- 1 ellio 197609 12 Feb 24 19:28 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 24 19:28 file2.txt

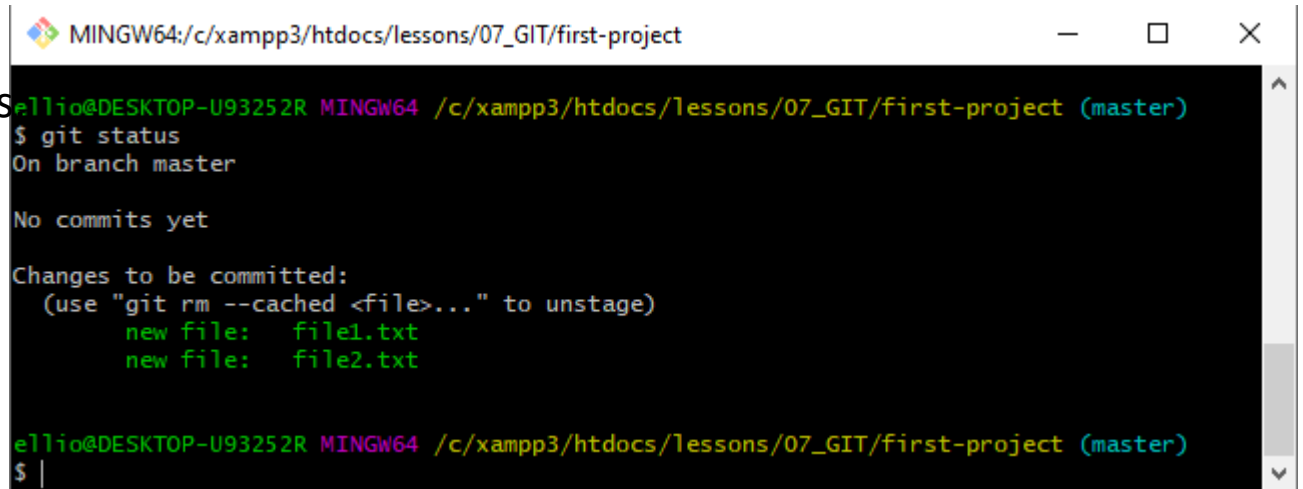
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

How many folders can git object create?

Git uses two hex bits to represent folder names so $16 \times 16 = 256$ maximum possible folders in a git project.

If hashing of a file produces a hash where the first two bits match those of an existing folder then that hash will be stored in the existing folder. Therefore the git object folders may contain multiple hashed files.

Git status shows us information about files, commits, and branches



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

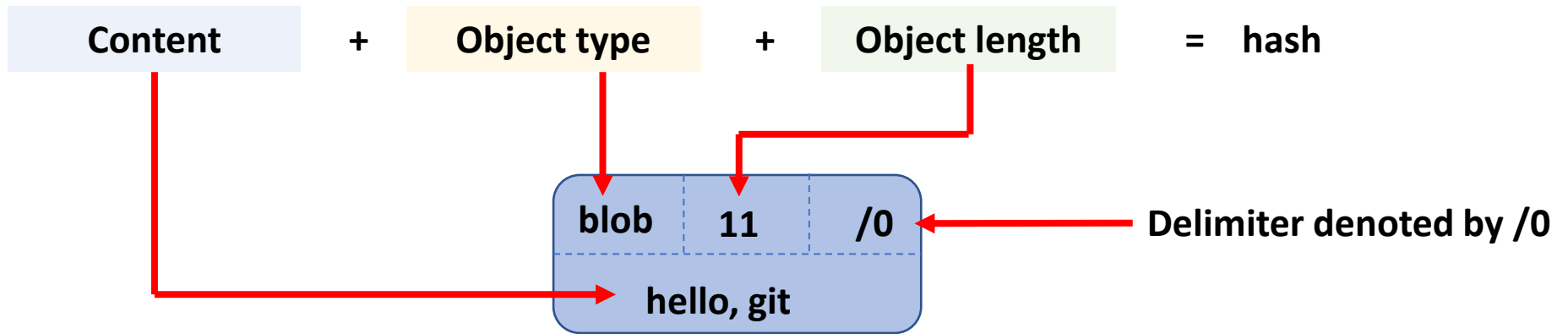
Summary of Shell Commands

git init	<i>Initialise git repository in current folder</i>
git cat-file -p <hash>	<i>Read a hash file contents</i>
git cat-file -s <hash>	<i>Read a hash file size</i>
git cat-file -t <hash>	<i>Read a hash file type</i>
echo "test" git hash-object --stdin -w	<i>Writes contents of echo to a hash in git objects folder</i>
git hash-object <file path and file name> -w	<i>Detailed list of current directory including</i>
find .git/objects -type f	<i>Find files (hashes) of git objects</i>
cat <file path and name> git mktree	<i>Create tree from file</i>
git ls-files	<i>List files in staging area (index)</i>
git read-tree <hash>	<i>Copy files from repository to staging area</i>
git checkout-index -a	<i>Copy files from staging area (index) to working area</i>
git cat <filename>	<i>Read contents of file in working area</i>

GIT BRANCHES AND HEAD

What is Git Commit Object?

A git commit object has the same structure as a tree and blob object type.



The contents contains Author name and Email, Commit description and optional Parent parameter.

Commits allows us to store different versions of our project and allows us to checkout rapidly a different version.

Hash: 8a31fb Commit

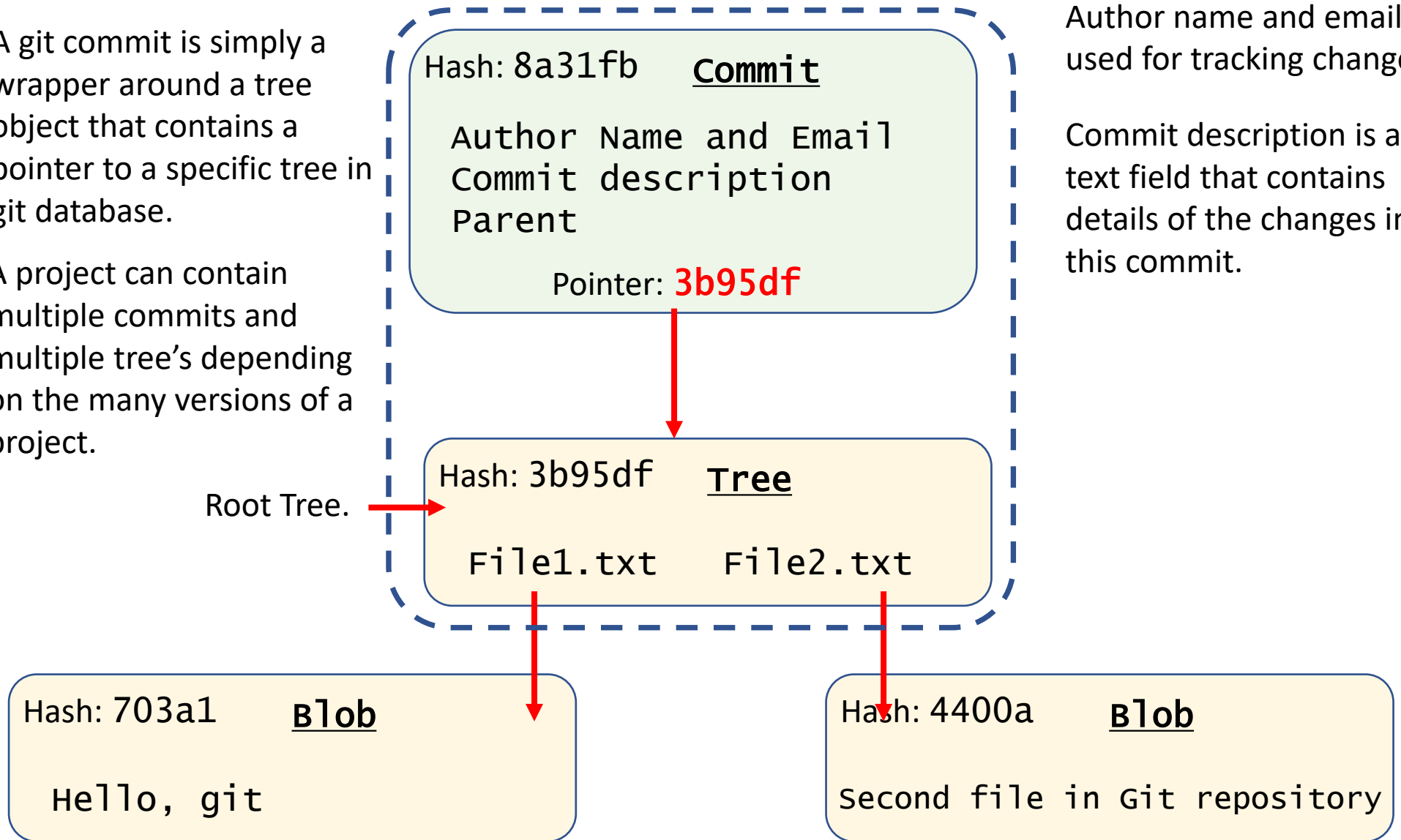
Author Name and Email
Commit description
Parent

A git commit is simply a wrapper around a tree object that contains a pointer to a specific tree in git database.

A project can contain multiple commits and multiple tree's depending on the many versions of a project.

Author name and email is used for tracking changes.

Commit description is a text field that contains details of the changes in this commit.



Git is distributed version control software so many people can work on the same project.

Set Author Name and email?

Set Author Name and email locally on your machine. These are global settings that will apply to all git projects.

git config --global user.name

“user name” Command is used to set username.

git config --global user.email

“user email” Command is used to set user email.

git config --list Command is used to verify username and email.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git config --global user.name "Elliott Farmer"
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git config --global user.email "myemail@provider.com"
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Elliott Farmer
user.email=myemail@provider.com
core.repositoryformatversion=0
core.filemode=false
...skipping...
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
```

Creating first Commit

git status Command shows us that we are on the master branch and that there are currently no commits.

It shows us that there are two files in the staging area that could be committed to the repository. This is another purpose of the staging area. There must be files in the staging area to perform a commit.

git commit -m "comment" Command will commit whatever is in the staging area (index) to the repository. In this case two files have been committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls -s
git: 'ls' is not a git command. See 'git --help'.

The most similar command is
    lfs

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0        file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0        file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git commit -m "Our very first commit in the project"
[master (root-commit) 25de26c] Our very first commit in the project
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

We can see that this commit was performed on the master branch.

Because it is our first commit it is a root commit and does not have any parents.

SHA1 hash of this new commit, the first 6 hex digits.

Here we can see the new commit object in folder 25 which is the first two hex digits of the commit hash.

We can read the contents of the commit with **git cat-file -p <hash>** command.

Note that the first line refers to a tree, specifically in this case 3b95df which is our manually created tree.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git commit -m "Our very first commit in the project"
[master (root-commit) 25de26c] Our very first commit in the project
2 files changed, 2 insertions(+)
create mode 100644 file1.txt
create mode 100644 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

Description of this commit

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ find .git/objects/ -type f
.git/objects/25/de26c3fff87e02b4bc3aabe34e835cef2eb21a
.git/objects/3b/95df0ac6365c72e9b0ff6c449645c87e6e1159
.git/objects/44/00aae52a27341314f423095846b1f215a7cf08
.git/objects/b7/aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 25de26c
tree 3b95df0ac6365c72e9b0ff6c449645c87e6e1159
author Elliott Farmer <myemail@provider.com> 1645733086 +0100
committer Elliott Farmer <myemail@provider.com> 1645733086 +0100

Our very first commit in the project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

Next are the author and committer. Usually these are the same. →

Commit comment. →

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 25de26c
tree 3b95df0ac6365c72e9b0ff6c449645c87e6e1159
author Elliott Farmer <myemail@provider.com> 1645733086 +0100
committer Elliott Farmer <myemail@provider.com> 1645733086 +0100

Our very first commit in the project

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

Here is a unix time stamp. The time stamp is number of seconds passed since 1st Jan 1970.

Timestamp. I.e UTC +1

The size of the commit object is 211 bytes →

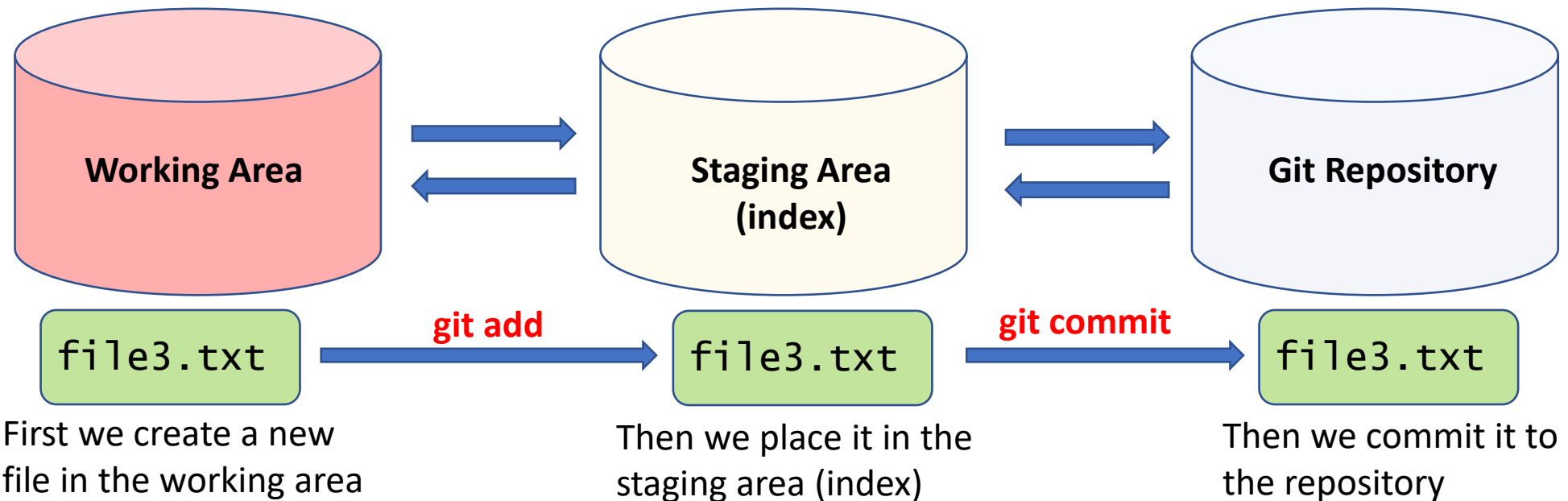
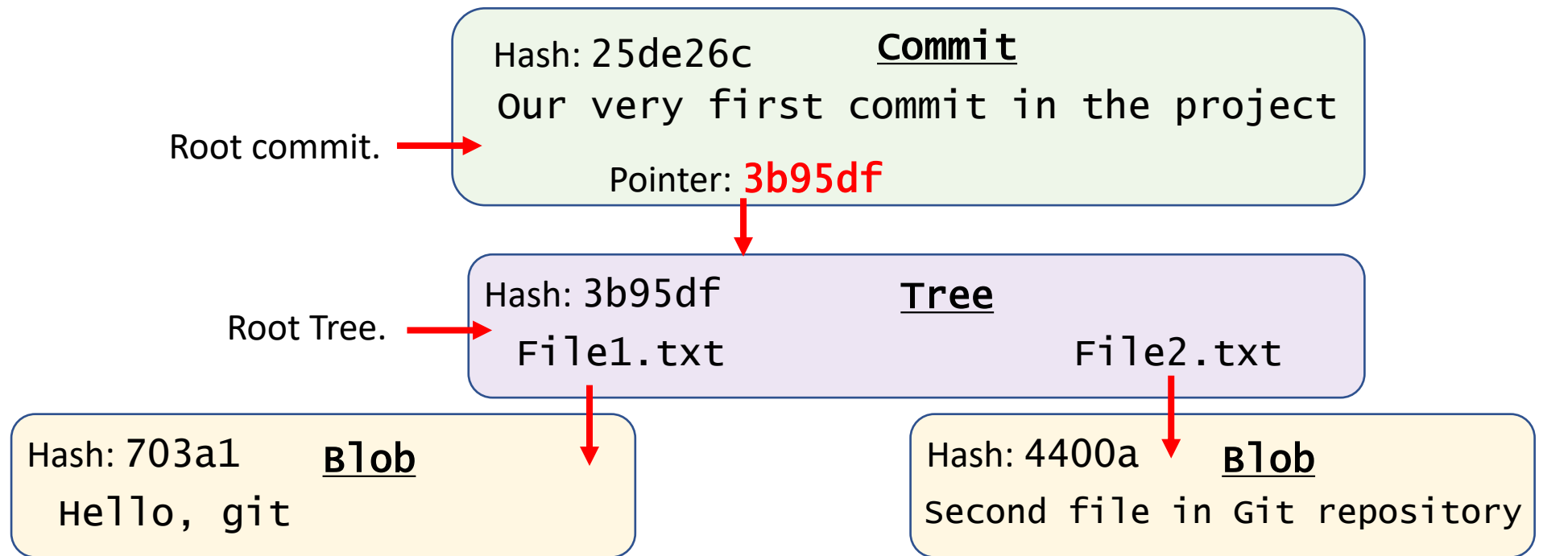
And the type of object is a commit. →

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -s 25de26c
211

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -t 25de26c
commit

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



Here are the two files located in git working area.

Here are the two files located in git staging area.

Here are the objects in the git repository.

Root Commit

Tree

Blob

Blob

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 2
-rw-r--r-- 1 ellio 197609 12 Feb 24 19:28 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 24 19:28 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files
file1.txt
file2.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ find .git/objects/ -type f
.git/objects/25/de26c3fff87e02b4bc3aabe34e835cef2eb21a
.git/objects/3b/95df0ac6365c72e9b0ff6c449645c87e6e1159
.git/objects/44/00aae52a27341314f423095846b1f215a7cf08
.git/objects/b7/aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

git status command shows that there is nothing to commit right now.

git Log command shows that there is only one commit action in the repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
nothing to commit, working tree clean

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a (HEAD -> master)
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```


Using nano editor a third file is created in the working area called file3.txt.

The file3.txt can be seen in the working area with a `ls -l` command.

But file3.txt is not in the staging area. Checked with `git ls-files -s`

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ nano file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 3
-rw-r--r-- 1 ellio 197609 12 Feb 24 19:28 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 24 19:28 file2.txt
-rw-r--r-- 1 ellio 197609 41 Feb 24 22:06 file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
GNU nano 5.9                                file3.txt
This is third file in our Git repository

[ Read 1 line ]

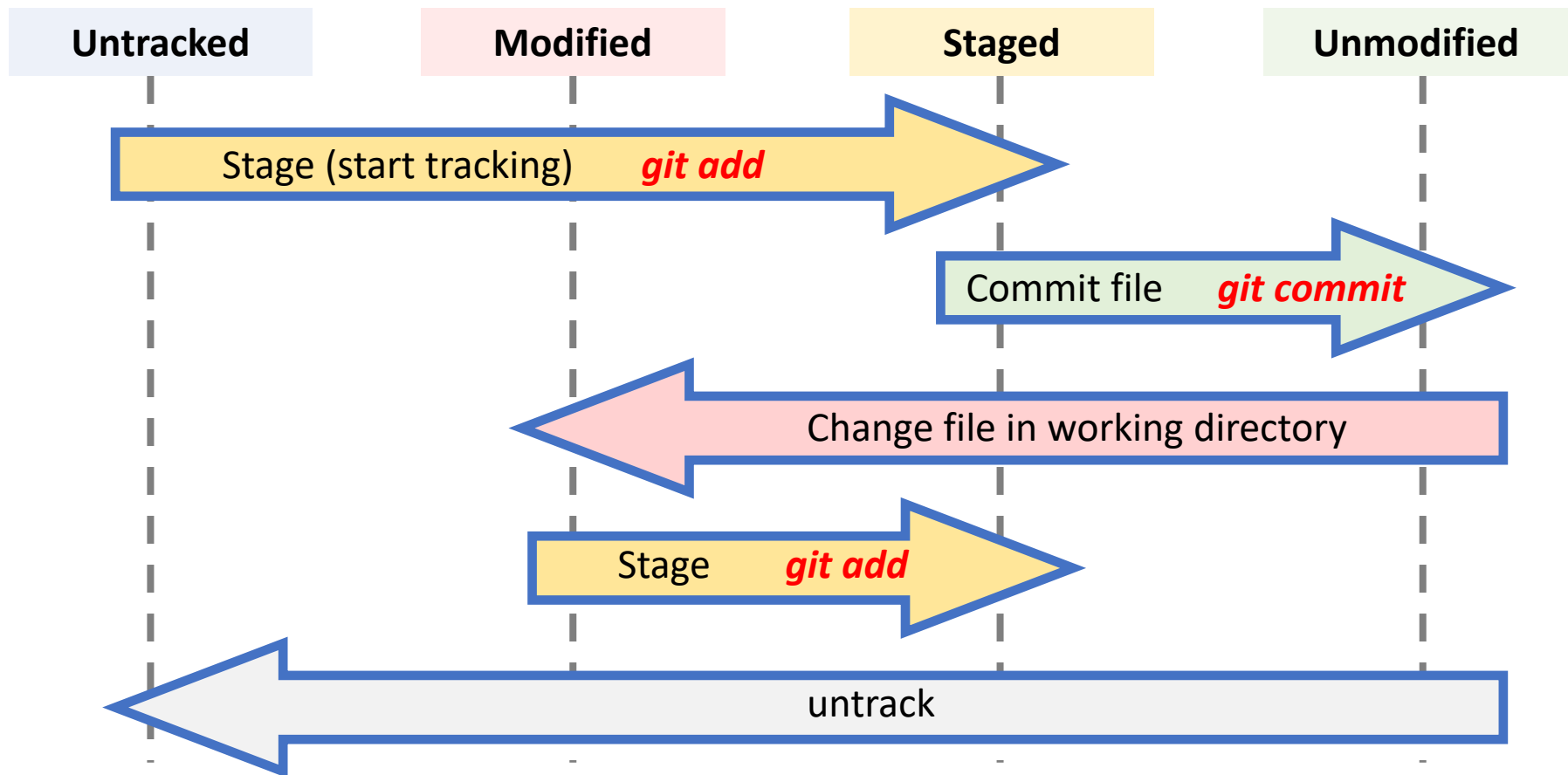
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

A git status command shows that file3.txt is untracked and it suggests we use git add to start tracking the file.

There are four git tracking status options and a file may change status many times.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
       file3.txt

nothing added to commit but untracked files present (use "git add" to track)
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



Git Add command

The git add command is used to move the new file file3.txt from untracked to staged. Note that the git add command requires parameters.

git add . Means add all files or
git add <filename> means add specified file.

git ls-files -s now shows file3.txt in the staging area.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

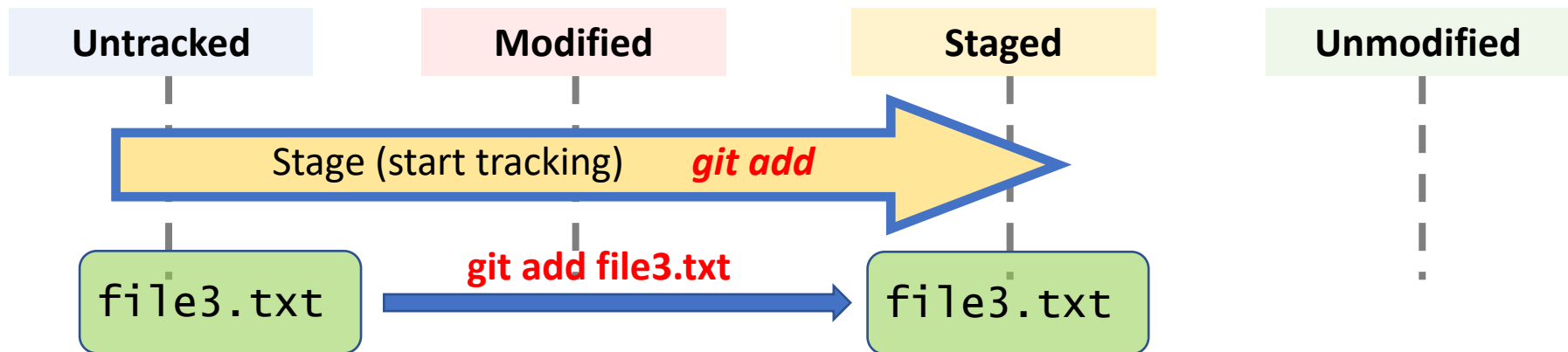
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git add file3.txt
warning: LF will be replaced by CRLF in file3.txt.
The file will have its original line endings in your working directory

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt
100644 289960fd4f172ef29a9bf30f9c49c1725ca253dc 0      file3.txt

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

File3.txt has now moved from untracked to tracked and is in the staging area.



The **git status** command shows file3.txt as staged and ready to be committed to the repository.

The change file3.txt from staged to untracked the command **git rm --cached file3.txt** can be used.

git ls-files -s shows that file3.txt is now removed from the staging area.

The **git status** command now shows file3.txt as untracked and ready to be tracked.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file3.txt

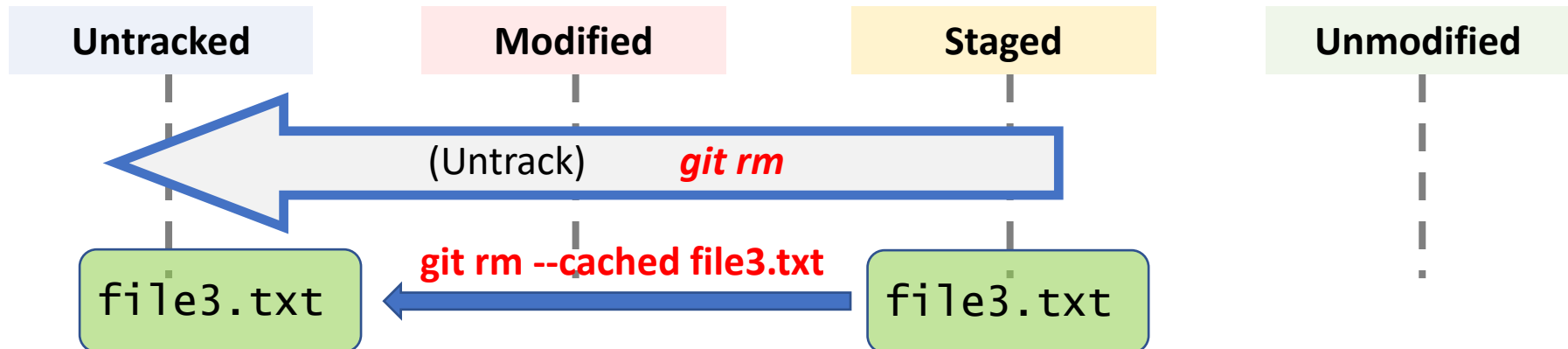
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git rm --cached file3.txt
rm 'file3.txt'

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3.txt

nothing added to commit but untracked files present (use "git add" to track)

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



Again we add file3.txt to tracked in the staged area.

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

```
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git add file3.txt
warning: LF will be replaced by CRLF in file3.txt.
The file will have its original line endings in your working directory

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt
100644 289960fd4f172ef29a9bf30f9c49c1725ca253dc 0      file3.txt

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

Now the file can be moved from the staged area to the repository with a **commit** command.

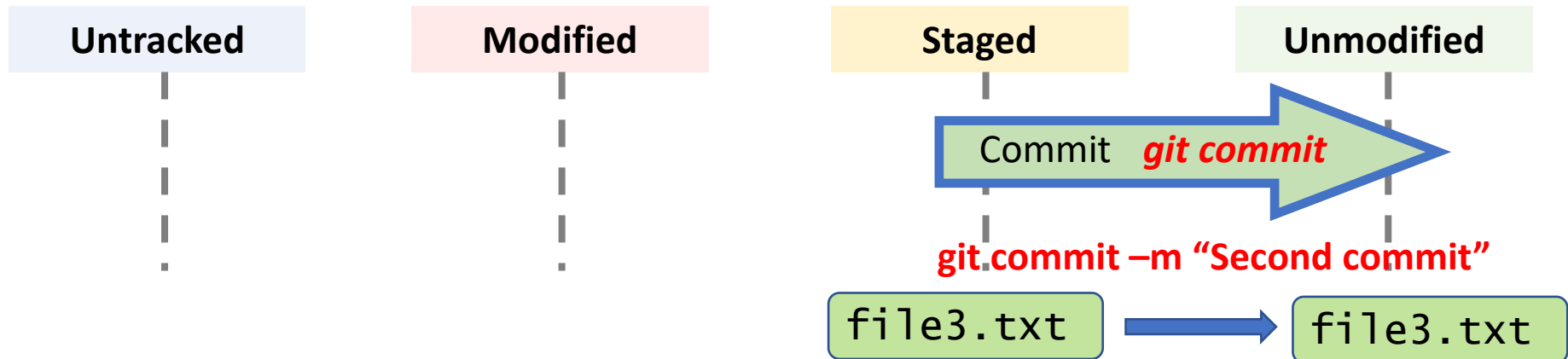
Note the output that the commit is in the master branch with a hash of fd127e1 and comment of "Second commit"

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

```
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git commit -m "Second commit"
[master fd127e1] Second commit
1 file changed, 1 insertion(+)
create mode 100644 file3.txt

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat file3.txt
This is third file in our Git repository

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



git status now shows nothing to commit meaning we are all synchronised in working area, tracking area and repository.

git log now shows two commits.

git cat-file -p fd127e1 shows more detail about the second commit. Note how the Second commit has a parent that is the hash of the root commit.

All subsequent commits will have a parent pointing to the root commit.

git cat-file -p 31b1e9 shows that the tree of the Second commit contains 3 files with hash links.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
nothing to commit, working tree clean

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31 (HEAD -> master)
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p fd127e1
tree 31b1e96ca47eb9a6c2a506d8a1d6530da7786ea7
parent 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
author Elliott Farmer <myemail@provider.com> 1645740056 +0100
committer Elliott Farmer <myemail@provider.com> 1645740056 +0100

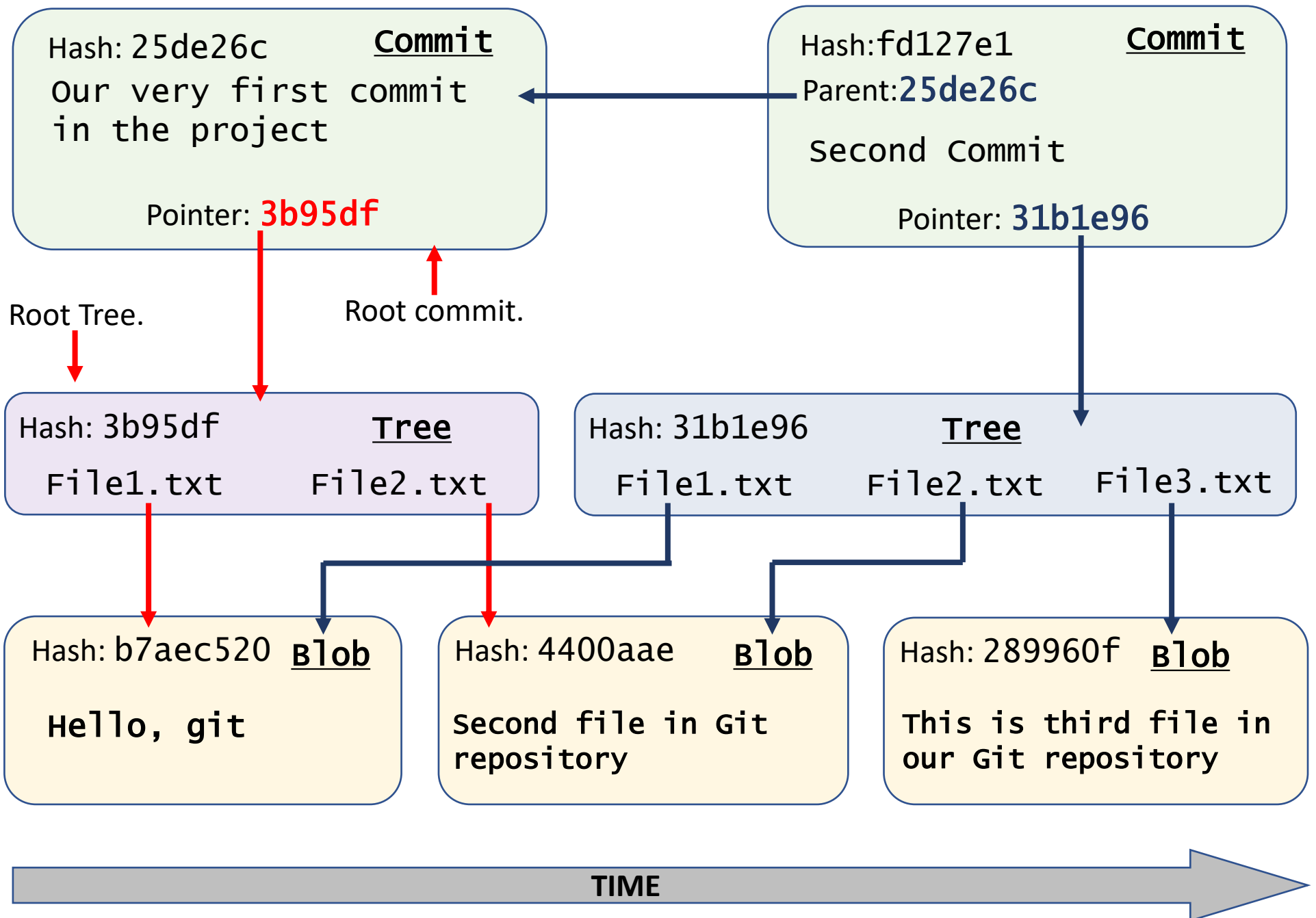
Second commit

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 31b1e9
100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e file1.txt
100644 blob 4400aae52a27341314f423095846b1f215a7cf08 file2.txt
100644 blob 289960fd4f172ef29a9bf30f9c49c1725ca253dc file3.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```



Now the project has two versions. The first version with tree 3b95df has two files in it and the second version with tree 31b1e96 has three files in it. This is how Git tracks changes. When files are modified the hash changes so a new tree is created.

Summary of Shell Commands

git config --global user.name "user name" *Set global username for local git*

git config --global user.email "user email" *Set global user email for local git*

git config --list *View git config list*

git status *Check status of git*

git add <filename>

Copy file from working area to staged area (tracked status)

git rm --cached <filename>

Remove file from staged area to untracked status

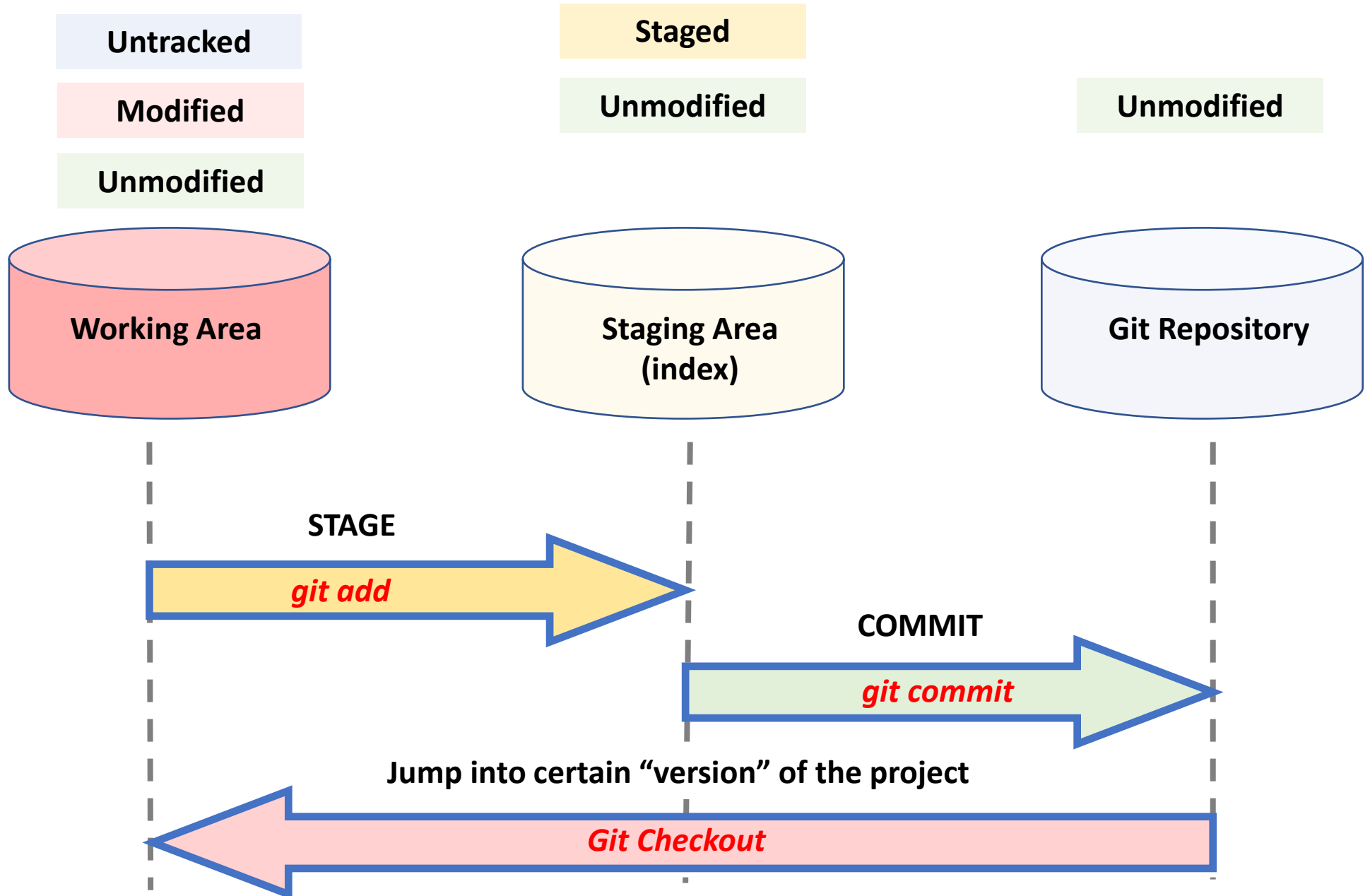
git commit -m "comment"

Commit changes in the staged area to the repository

git log

View log of commit actions

File States in Git Areas



Download the GitHub desktop exe from <https://desktop.github.com>



Let's get started!

Add a repository to GitHub Desktop to start collaborating



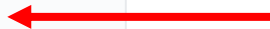
Clone a repository from the Internet...



Create a New Repository on your hard drive...



Add an Existing Repository from your hard drive...



Click on add an existing repository



ProTip! You can drag & drop an existing repository folder here to add it to Desktop





Let's get started!

Add a repository to GitHub Desktop to start collaborating



Clone a repository



Create a New Repository



Add an Existing Repository from your hard drive...



ProTip! You can drag & drop an existing repository folder here to add it to Desktop

Add local repository



Local path

C:\xampp3\htdocs\lessons\07_GIT\first-project

Choose...

Add repository

Cancel

Select the folder and choose
add repository

Current repository is first-project

Current branch is master

The screenshot shows the Visual Studio interface with the following elements:

- Toolbar:** Contains 'Current repository' (first-project), 'Current branch' (master), and a 'Publish repository' button.
- Changes/History View:** The 'Changes' tab is active, showing '0 changed files'. The 'History' tab is also visible.
- Commit Message Input:** A text box for the commit message, currently empty, with a 'Commit to master' button below it.
- Commit History:** Shows 'Committed 2 hours ago' and 'Second commit'.
- Main Area:** Displays the message 'No local changes' with a subtext: 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.'
- Suggestions:**
 - Publish your repository to GitHub:** This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others. Always available in the toolbar for local repositories or **Ctrl** **P**. Button: **Publish repository**.
 - Open the repository in your external editor:** Select your editor in [Options](#). Repository menu or **Ctrl** **Shift** **A**. Button: **Open in Visual Studio Code**.
 - View the files of your repository in Explorer:** Repository menu or **Ctrl** **Shift** **F**. Button: **Show in Explorer**.

**History will show
all of the commits**

**No local changes means that the files
are synchronised across working area,
staging area and the repository**

**Changes are
committed here**

File Edit View Repository Branch Help

Current repository first-project

Current branch master

Publish repository Publish this repository to GitHub

Changes History

No branches to compare

Second commit
Elliott Farmer • 2h

Our very first commit in the project
Elliott Farmer • 4h

Second commit
Elliott Farmer fd127e1 ± 1 changed file +1 -0

file3.txt	+	@@ -0,0 +1 @@
	1	+This is third file in our Git repository

In the second commit a third file was added to the project

File Edit View Repository Branch Help

Current repository first-project

Current branch master

Publish repository Publish this repository to GitHub

Changes History

No branches to compare

Second commit
Elliott Farmer • 2h

Our very first commit in the project
Elliott Farmer • 4h

Our very first commit in the project
Elliott Farmer 25de26c ± 2 changed files +2 -0

file1.txt	+	@@ -0,0 +1 @@
file2.txt	+	1 +Hello, Git

In the first commit two files were added to the project

What is branch?

Branch is just **text reference** to the commit

Default Branch is **Master**

Pointers for all branches are located in **.git/refs/heads** folder

Current Branch tracks new commits

Branch pointer moves automatically every each new commit

Change Branch git **checkout <branch>**

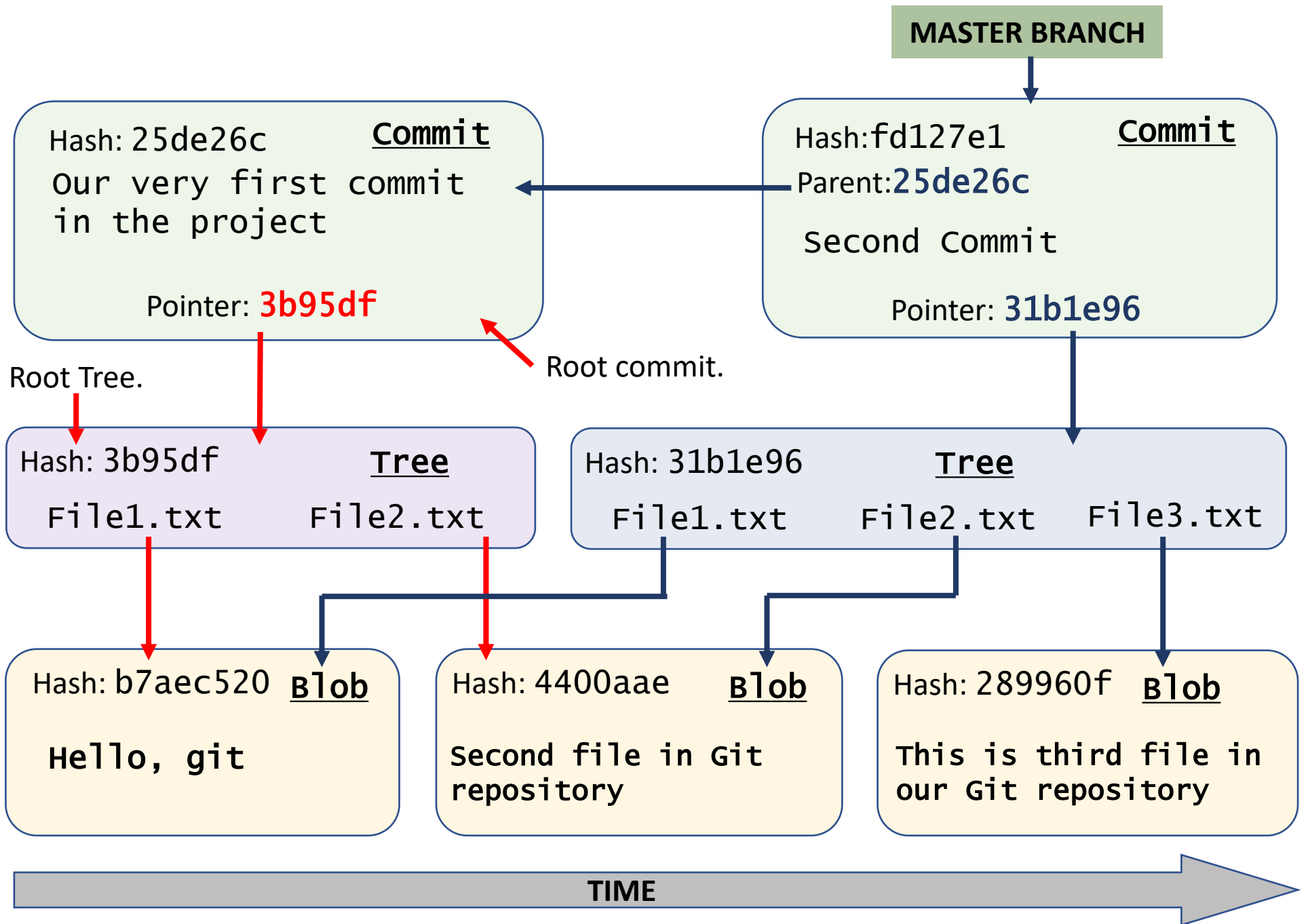
MASTER

New-Branch

DEV-1512

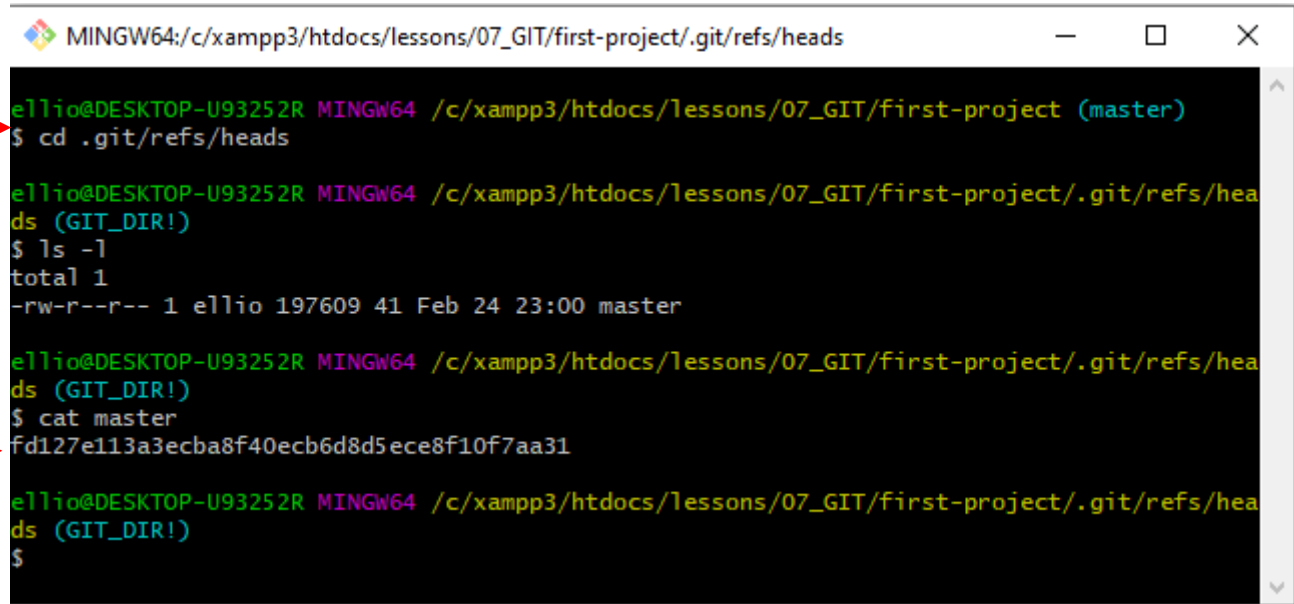
When switching between branches we will not see any new commits in the other branches.

Branch name cannot have spaces. i.e. New branch is not valid and would automatically be changed to New-Branch.



cd .git/refs/heads directory contains one file called **master**.

cat master shows that this file has one line only which is the hash pointer to the most recent commit in this branch. In this case **fd127e1**.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project/.git/refs/heads
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git/refs/heads
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/refs/heads (GIT_DIR!)
$ ls -l
total 1
-rw-r--r-- 1 ellio 197609 41 Feb 24 23:00 master
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/refs/heads (GIT_DIR!)
$ cat master
fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project/.git/refs/heads (GIT_DIR!)
$
```

What is HEAD?

HEAD is a pointer to current branch in the project. There can be multiple branches but only one HEAD – only one current branch.

HEAD is reference to the currently checked-out branch or commit.

HEAD is locally significant only so changing it on your git will not affect other users

Pointer is located in the **.git/HEAD** file.

Default pointer is **ref: refs/heads/master**

Change reference to specific branch **git checkout <branch>**

Change reference to specific commit **git checkout <hash>**

In the **.git** directory we have a file called **HEAD**.

cat head shows that this file has one line only which is the pointer file.

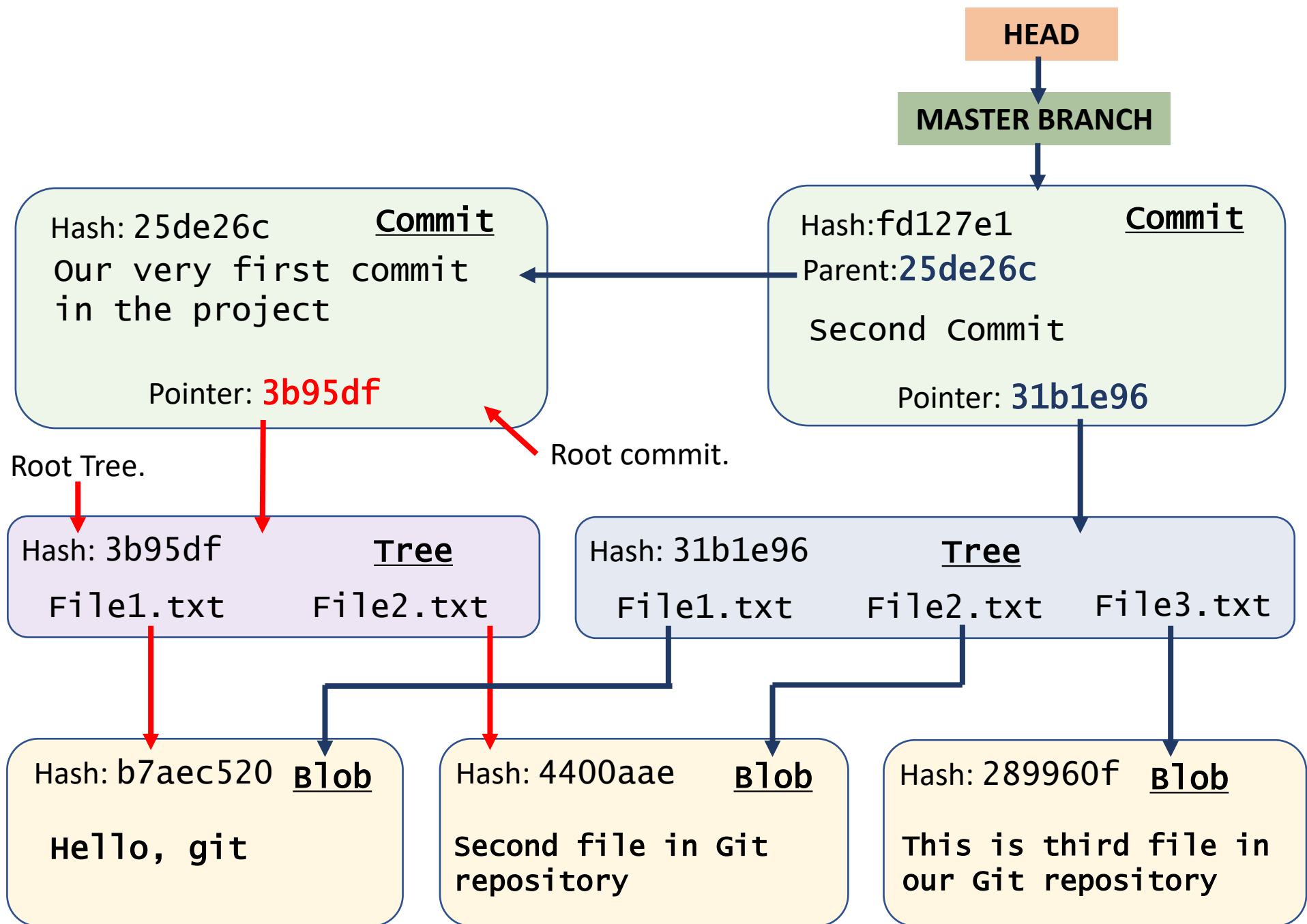
A **cat refs/heads/master** command shows that the hash in the HEAD pointer file is that of the second commit.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project/.git
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd .git
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ ls -la
total 17
drwxr-xr-x 1 ellio 197609  0 Feb 24 23:09 ./
drwxr-xr-x 1 ellio 197609  0 Feb 24 22:06 ../
-rw-r--r-- 1 ellio 197609 14 Feb 24 23:00 COMMIT_EDITMSG
-rw-r--r-- 1 ellio 197609 23 Feb 24 16:14 HEAD
-rw-r--r-- 1 ellio 197609 130 Feb 24 16:14 config
-rw-r--r-- 1 ellio 197609  73 Feb 24 16:14 description
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 hooks/
-rw-r--r-- 1 ellio 197609 281 Feb 24 23:00 index
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609  0 Feb 24 21:04 logs/
drwxr-xr-x 1 ellio 197609  0 Feb 24 23:00 objects/
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 refs/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ cat HEAD
ref: refs/heads/master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$ cat refs/heads/master
fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project/.git (GIT_DIR!)
$
```



ls -l command shows the three files in the working directory. →

rm <filename> Command can be used to remove all three files from the working directory. →

git status shows that the files are ready to be staged. →

All three files are staged using **git add .** →

git status show all three files as staged. →

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 3
-rw-r--r-- 1 ellio 197609 12 Feb 24 19:28 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 24 19:28 file2.txt
-rw-r--r-- 1 ellio 197609 41 Feb 24 22:06 file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ rm file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ rm file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ rm file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    file1.txt
        deleted:    file2.txt
        deleted:    file3.txt

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    file1.txt
        deleted:    file2.txt
        deleted:    file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

FileEditViewRepositoryBranchHelp

Current repository
first-project

Current branch
master

↑

Publish repository

Publish this repository to GitHub

Changes 3

History

file1.txt

3 changed files

file1.txt

file2.txt

file3.txt

@@ -1 +0,0 @@

-Hello, Git

Summary (required)

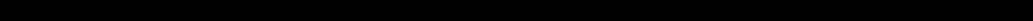
Description

Commit to master

Committed 3 hours ago
Second commit

Undo

The files show as staged in the git hub desktop and marked with a minus for remove.



The screenshot shows a Windows command prompt window with the title bar "MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project". The prompt is "elliott@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)". The command "git commit" has been entered and is highlighted by a red arrow.

[illegible]

```

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
Third commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   deleted:   file1.txt
#   deleted:   file2.txt
#   deleted:   file3.txt
#
~
~
<docs/lessons/07_GIT/first-project/.git/COMMIT_EDITMSG[+] [unix] (02:48 25/02/2022)10,1 All
:wq

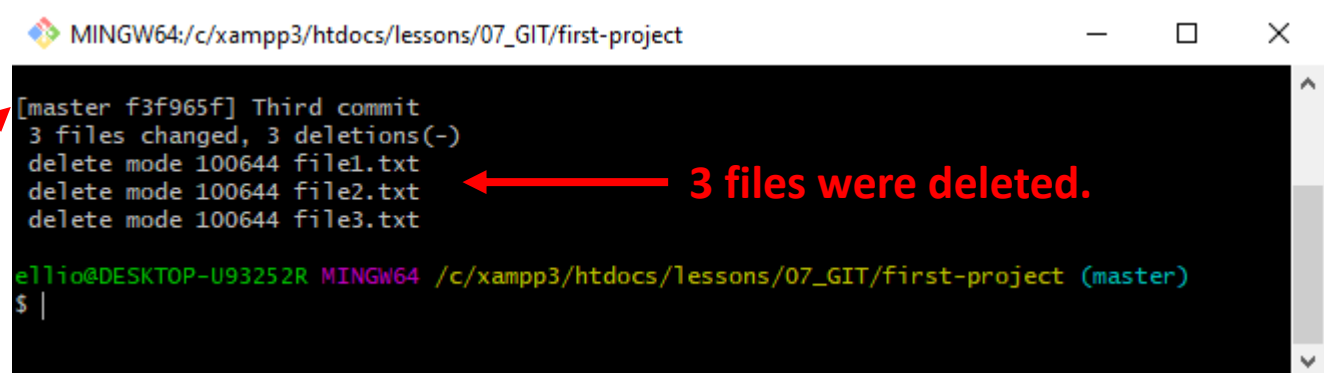
```

Here we see the output of the commit.

The commit was made on the **master branch**.

The hash of the commit is **f3f965f** and the comment is **Third commit**.

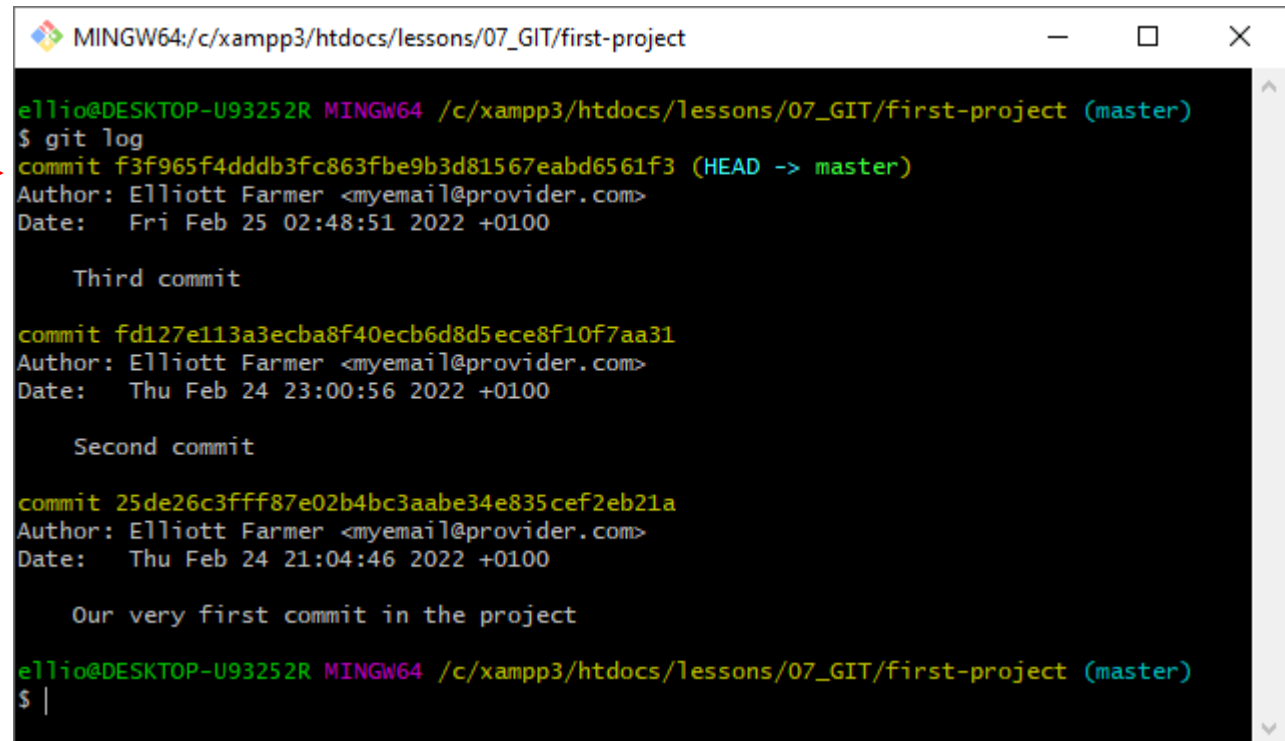
git log command shows the three commits. Note the usual information of author and datetime plus time zone. HEAD still points to master branch. The new commit also has its Hash.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

[master f3f965f] Third commit
3 files changed, 3 deletions(-)
delete mode 100644 file1.txt
delete mode 100644 file2.txt
delete mode 100644 file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit f3f965f4dddb3fc863f9b3d81567eabd6561f3 (HEAD -> master)
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 02:48:51 2022 +0100

    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

The heads/master file now points to the hash of the third commit.

The contents of this new commit hash file contain a pointer to new tree, a pointer to parent and author, committer and comment.

If we examine the contents of the tree we see it is empty and with file size of zero. This is because the last commit was to delete all three files.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/HEAD
ref: refs/heads/master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/master
f3f965f4dddb3fc863fbe9b3d81567eabd6561f3

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p f3f965
tree 4b825dc642cb6eb9a060e54bf8d69288fbee4904
parent fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
author Elliott Farmer <myemail@provider.com> 1645753731 +0100
committer Elliott Farmer <myemail@provider.com> 1645753731 +0100

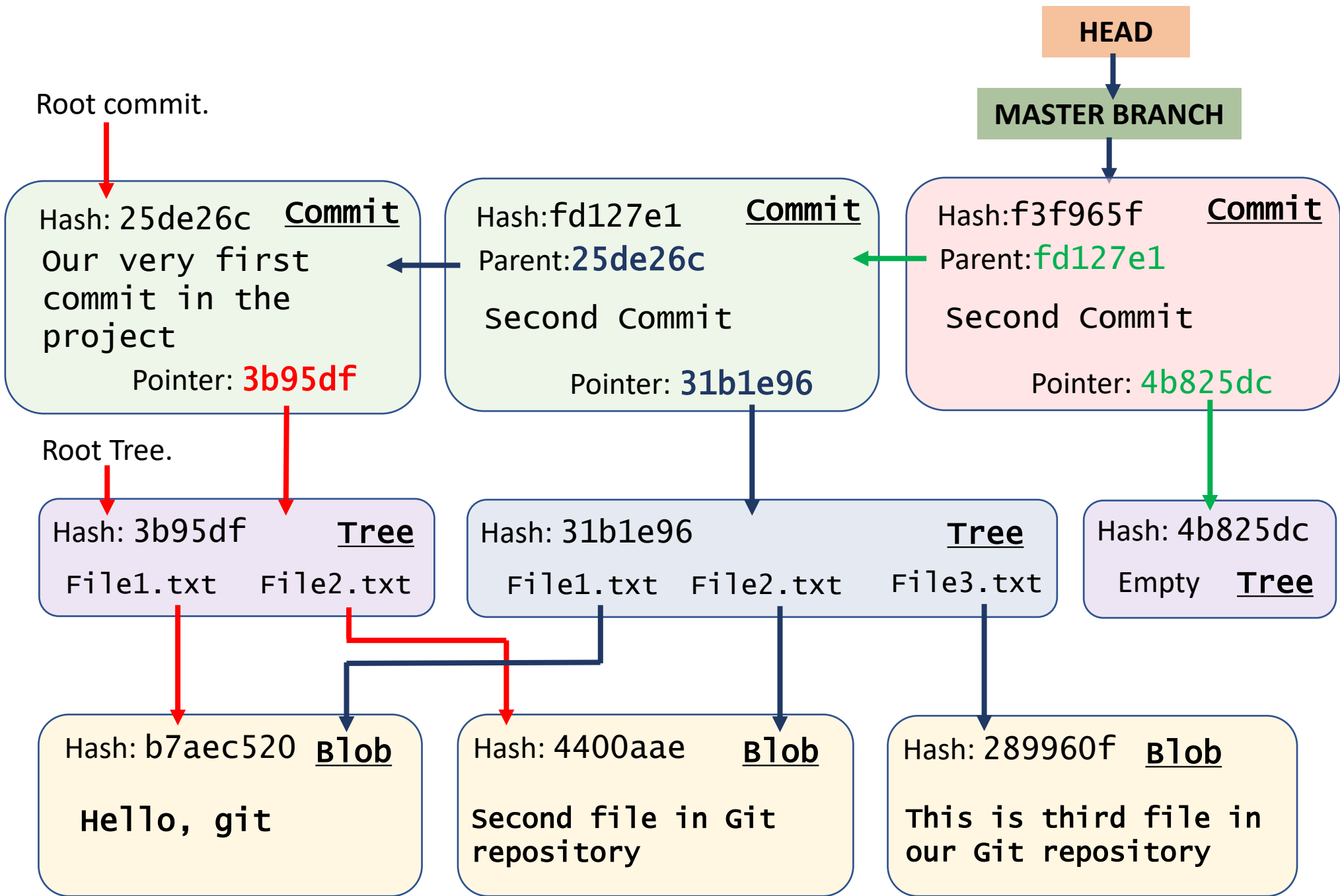
Third commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p 4b825dc

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -t 4b825dc
tree

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -s 4b825dc
0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

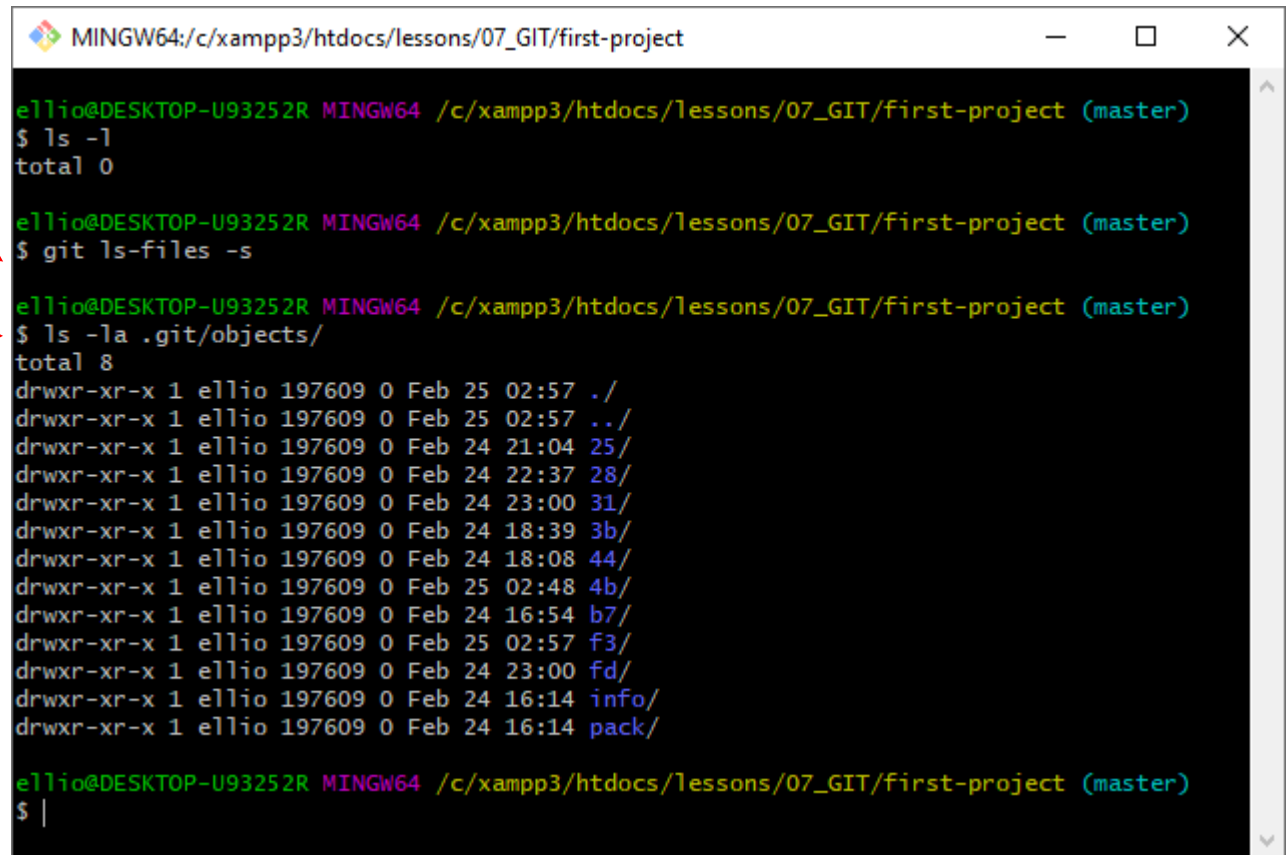


The working area is empty.

The staging area is empty.

But in the Git repository all the files are stored in the objects folder.

Git repository stores all previous files and commits.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 0

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -la .git/objects/
total 8
drwxr-xr-x 1 ellio 197609 0 Feb 25 02:57 ./
drwxr-xr-x 1 ellio 197609 0 Feb 25 02:57 ../
drwxr-xr-x 1 ellio 197609 0 Feb 24 21:04 25/
drwxr-xr-x 1 ellio 197609 0 Feb 24 22:37 28/
drwxr-xr-x 1 ellio 197609 0 Feb 24 23:00 31/
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:39 3b/
drwxr-xr-x 1 ellio 197609 0 Feb 24 18:08 44/
drwxr-xr-x 1 ellio 197609 0 Feb 25 02:48 4b/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:54 b7/
drwxr-xr-x 1 ellio 197609 0 Feb 25 02:57 f3/
drwxr-xr-x 1 ellio 197609 0 Feb 24 23:00 fd/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 info/
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:14 pack/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

The parent commit file hash can be found for the third commit.

git checkout <commit>

command for the parent hash of the third commit moves the Head pointer to the second commit.

Note how it gives a detached HEAD notice.

Now instead of master branch it says the hash of the second commit.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/HEAD
ref: refs/heads/master

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/master
f3f965f4dddb3fc863f9b3d81567eabd6561f3

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p f3f965
tree 4b825dc642cb6eb9a060e54bf8d69288fbee4904
parent fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
author Elliott Farmer <myemail@provider.com> 1645753731 +0100
committer Elliott Farmer <myemail@provider.com> 1645753731 +0100

Third commit

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout fd127e1
Note: switching to 'fd127e1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

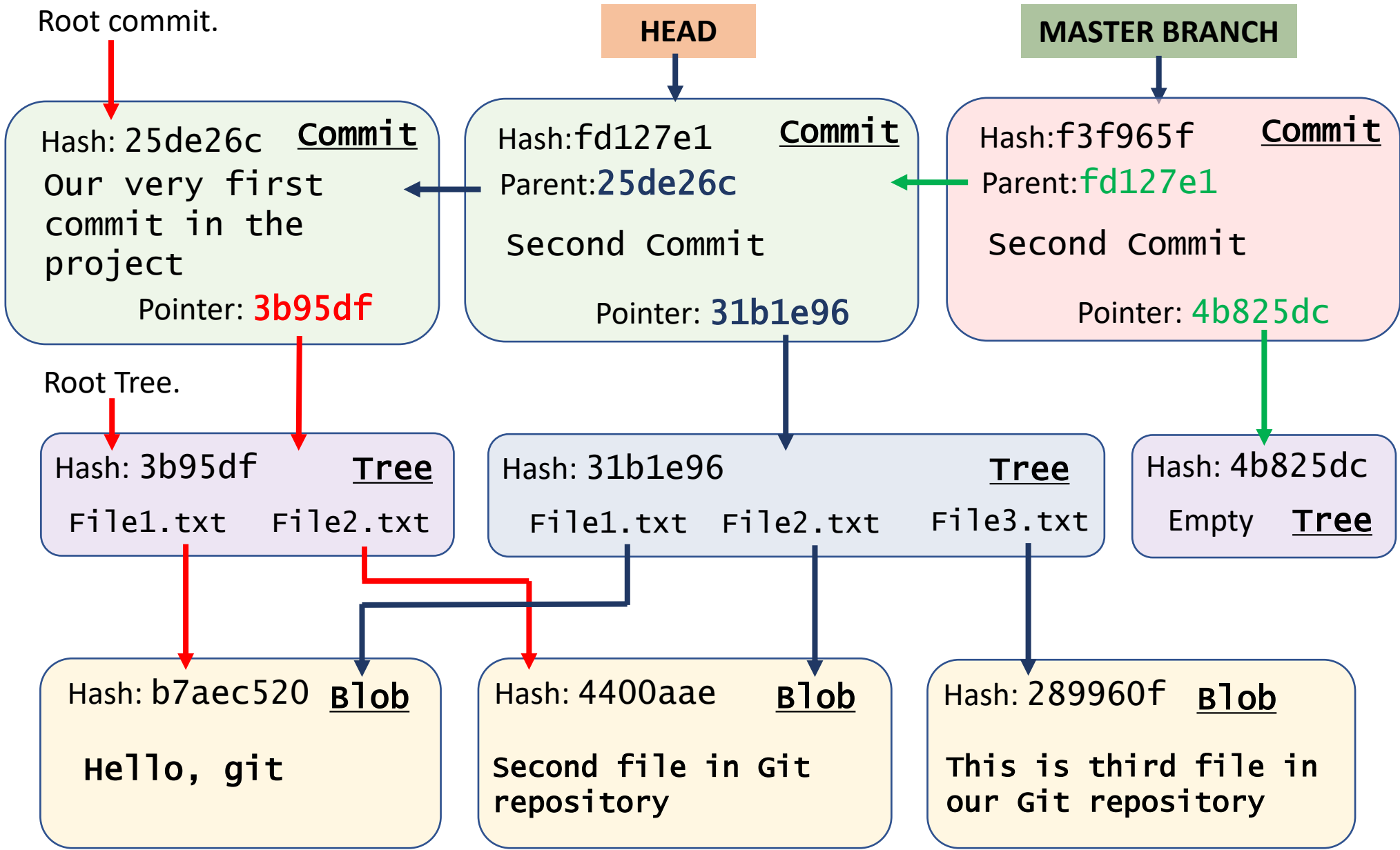
Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at fd127e1 Second commit
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ |
```

When head points to a specific commit rather than a branch it is referred to as **detached head state**.



If we look at the contents of the HEAD file it now just has a pointer to the second commit. →

The master still points → to the third commit.

The working directory now → contains all three files.

The Staging area also → contains all three files.

The tree is the second tree → and the parent points to the first commit.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ cat .git/refs/heads/master
f3f965f4dddb3fc863fbe9b3d81567eabd6561f3

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ ls -l
total 3
-rw-r--r-- 1 ellio 197609 12 Feb 25 19:53 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 25 19:53 file2.txt
-rw-r--r-- 1 ellio 197609 42 Feb 25 19:53 file3.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ git ls-files -s
git: 'ls-files-s' is not a git command. See 'git --help'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt
100644 289960fd4f172ef29a9bf30f9c49c1725ca253dc 0      file3.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ git cat-file -p fd127e1
tree 31b1e96ca47eb9a6c2a506d8a1d6530da7786ea7
parent 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
author Elliott Farmer <myemail@provider.com> 1645740056 +0100
committer Elliott Farmer <myemail@provider.com> 1645740056 +0100

Second commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$
```

The screenshot shows the GitHub Desktop application window. The top menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu, the 'Current repository' is 'first-project'. The 'Branch' section shows 'Detached HEAD On fd127e1', which is highlighted with a red rectangle. To the right is a 'Publish repository' button. The main area is divided into two tabs: 'Changes' and 'History'. The 'History' tab is active, showing a list of commits. The first commit is 'Second commit' by Elliott Farmer, 21h ago, with a message 'Our very first commit in the project'. The second commit is also by Elliott Farmer, 22h ago, with a message 'Our very first commit in the project'. Below the commit list, a diff view for 'file3.txt' is shown, indicating a change from an empty file to a file containing the text '+This is third file in our Git repository'.

File Edit View Repository Branch Help

Current repository: first-project

Detached HEAD On fd127e1

Publish repository
Publish this repository to GitHub

Changes History

Select branch to compare...

Second commit
Elliott Farmer • 21h

Our very first commit in the project
Elliott Farmer • 22h

file3.txt

@@ -0,0 +1 @@
1 +This is third file in our Git repository

In GitHub desktop history section we only see the first commit and the second commit. This is because the project is viewed from the perspective of the head pointer.

In the log we only see two commits with the second commit having HEAD only and not Master->HEAD as before.

Now if we checkout the first commit we get a message telling us the previous HEAD and current HEAD. The head file points to the first commit.

Note that the working directory and staging area also changes to reflect the two files present in the first commit.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ git log
commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31 (HEAD)
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((fd127e1...))
$ git checkout 25de26c
Previous HEAD position was fd127e1 Second commit
HEAD is now at 25de26c Our very first commit in the project

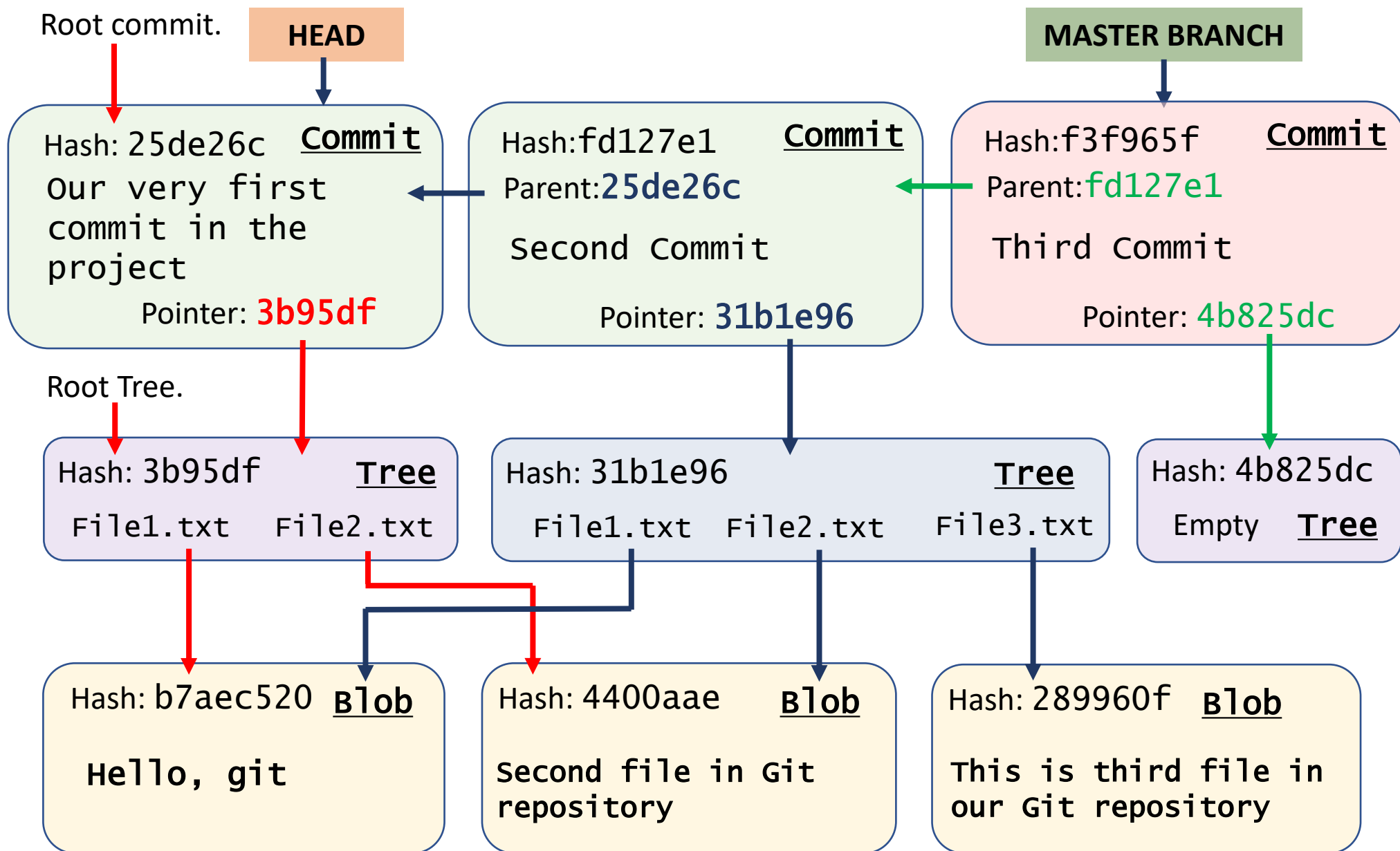
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((25de26c...))
$ cat .git/HEAD
25de26c3fff87e02b4bc3aabe34e835cef2eb21a

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((25de26c...))
$ ls -l
total 2
-rw-r--r-- 1 ellio 197609 12 Feb 25 19:53 file1.txt
-rw-r--r-- 1 ellio 197609 31 Feb 25 19:53 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((25de26c...))
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file1.txt
100644 4400aae52a27341314f423095846b1f215a7cf08 0      file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project ((25de26c...))
$ |
```

When head points to a specific commit rather than a branch it is referred to as **detached head state**.



git checkout master

command will switch the HEAD back to the master branch.

Now there are once again no files in the working area or the staging area.

Git log shows the 3 commits.

The HEAD file points to the refs/heads/master file.

The refs/heads/master file contains the pointer to the third commit.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((25de26c...))
$ git checkout master
Previous HEAD position was 25de26c Our very first commit in the project
Switched to branch 'master'

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 0

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git ls-files -s

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit f3f965f4dddb3fc863f9b3d81567eabd6561f3 (HEAD -> master)
Author: Elliott Farmer <myemail@provider.com>
Date:   Fri Feb 25 02:48:51 2022 +0100

    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date:   Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date:   Thu Feb 24 21:04:46 2022 +0100

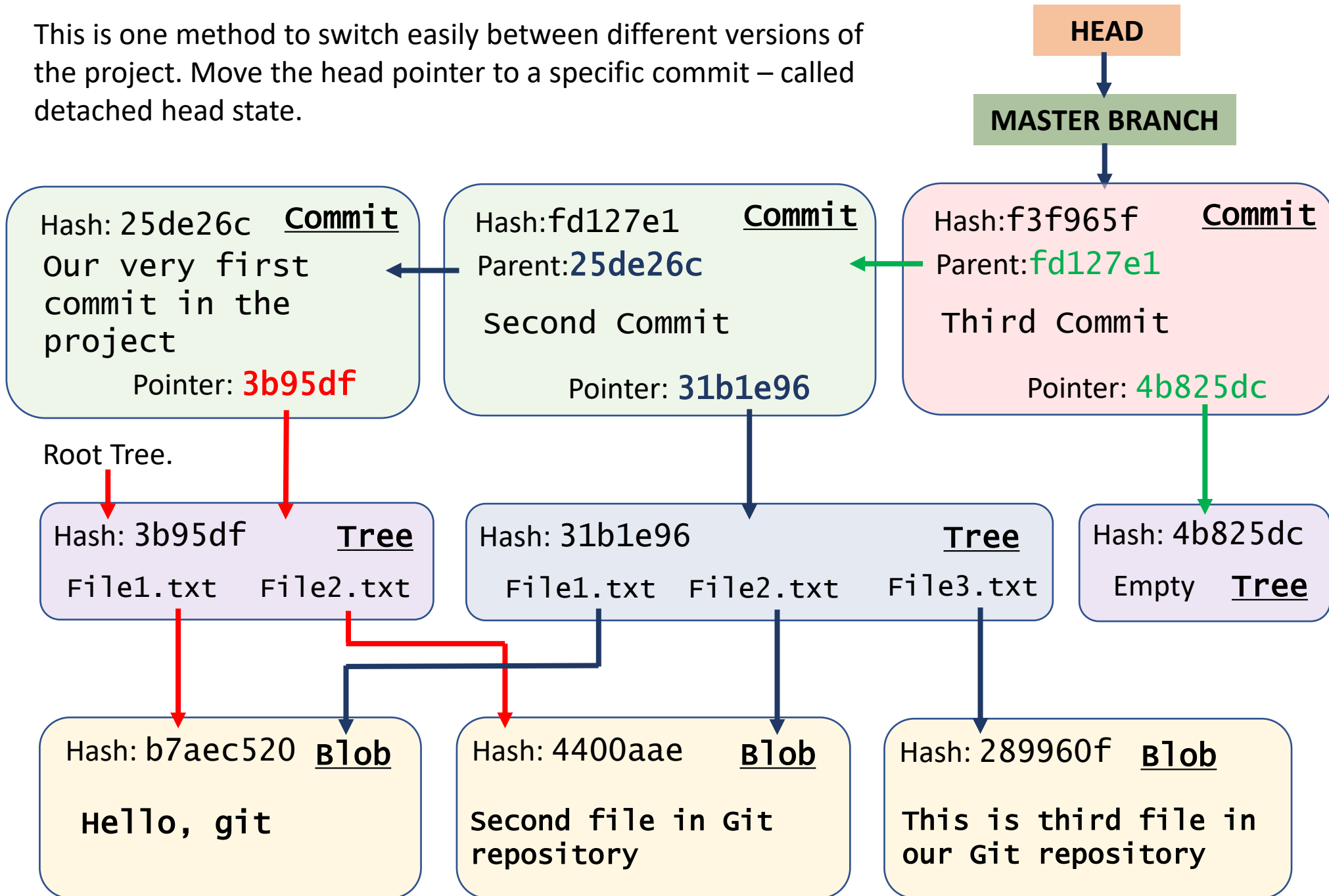
    Our very first commit in the project

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/HEAD
ref: refs/heads/master

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/master
f3f965f4dddb3fc863f9b3d81567eabd6561f3

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```


This is one method to switch easily between different versions of the project. Move the head pointer to a specific commit – called detached head state.



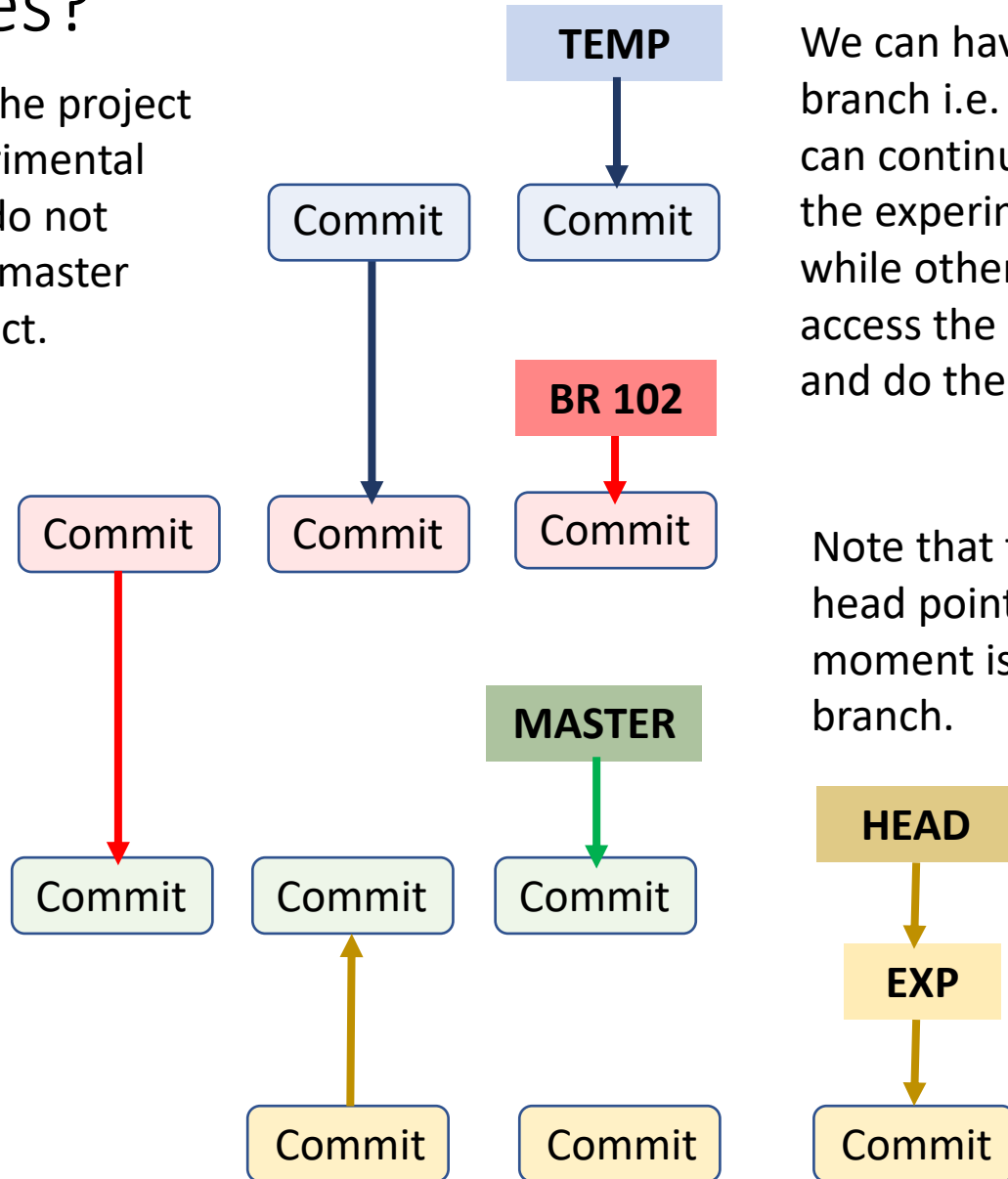
What are branches?

Lets say that at some point in the project we decide to work on an experimental version of the project and we do not want to mess around with the master branch which is the main project.

Branch is a wrapper for all the commits in that branch. It is a human readable name for a particular version of the project.

Root commit. → Commit

Branches are completely independent of each other.



We can have a separate branch i.e. BR102 where we can continue working on the experimental branch while other users can still access the master branch and do their work.

Note that there is only one head pointer which at the moment is pointing at EXP branch.

git branch command will list all branches.

git branch <name> will create a new branch, in this case called temp.

The git refs/heads file now has two pointers in it for each branch.

If we look at the master branch pointer it points to the third commit. As does the pointer for the new temp directory because we created it as a branch starting from the third commit.

If we checkout the new branch with **git checkout <branch name>** then the head pointer now goes to the temp branch.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master
  temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls .git/refs/heads
master temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/master
f3f965f4dddb3fc863fbe9b3d81567eabd6561f3

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/temp
f3f965f4dddb3fc863fbe9b3d81567eabd6561f3

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/HEAD
ref: refs/heads/master

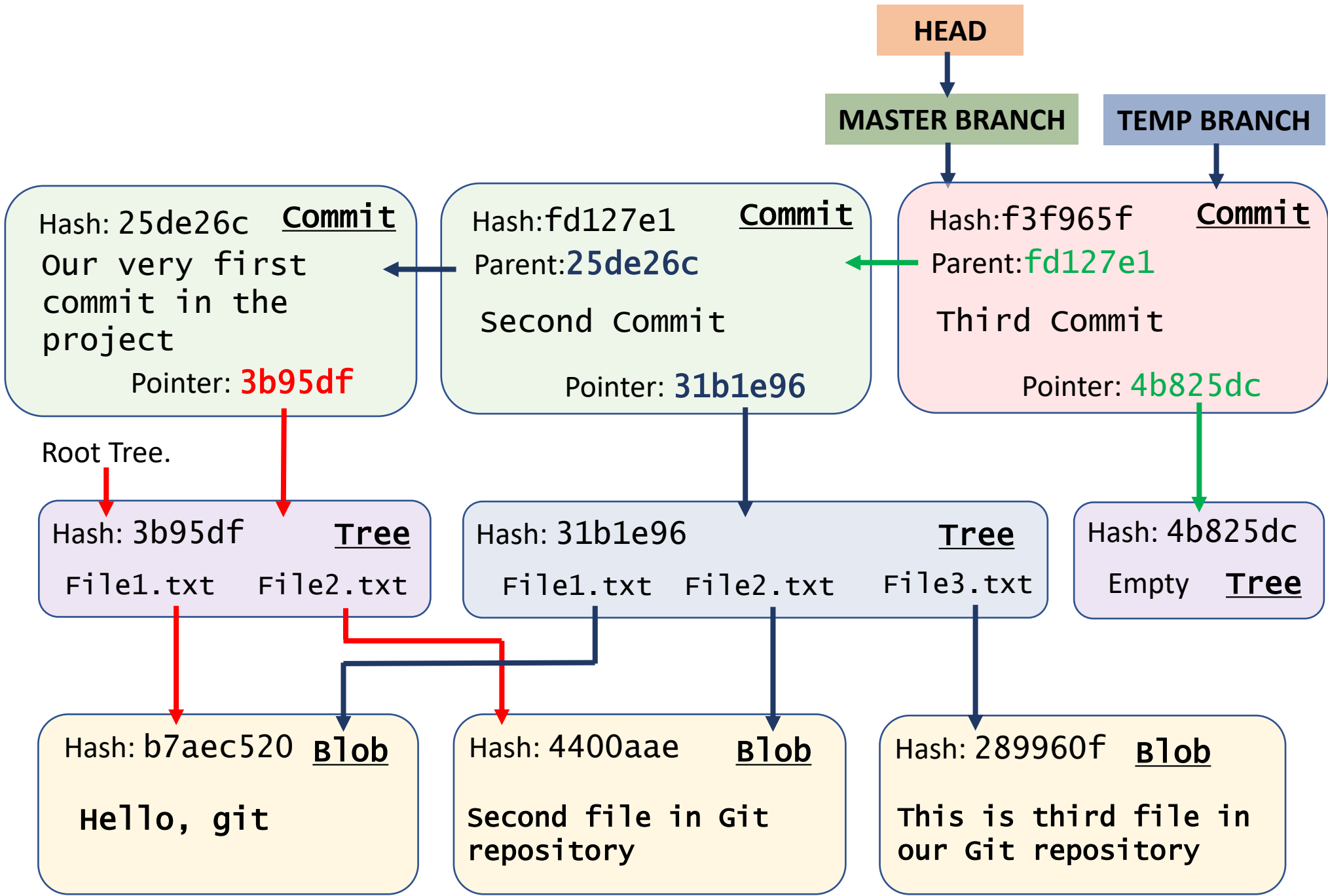
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout temp
Switched to branch 'temp'

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (temp)
$ cat .git/head
ref: refs/heads/temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (temp)
$ cat .git/refs/heads/temp
f3f965f4dddb3fc863fbe9b3d81567eabd6561f3

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (temp)
$ |
```

The head pointer of the temp branch points to the third commit because we have not created any new commits in temp branch..



git branch -m <old name>
<new name> command will
rename a branch. →

git checkout <branch
name> command will
switch the working
directory and staging
directory to new branch. →

git branch -d <branch →
name> command delete a
branch.

↗
git branch command to list
branches.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch -m temp new-temp

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master
  new-temp

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout new-temp
Switched to branch 'new-temp'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (new-temp)
$ git checkout master
Switched to branch 'master'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch -d new-temp
Deleted branch new-temp (was f3f965f).

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

Currently there is only the master branch in the project.

git checkout -b <branch name> command will create and checkout a new branch. In this case called BR-1.

The branch does not have any files in it.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout -b BR-1
Switched to a new branch 'BR-1'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git branch
* BR-1
  master

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ ls -l
total 0

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$
```

While in branch BR-1 We can create a new file called file4.txt with content "Hello, Git"

And we see the new file in the working directory.

git status shows that the file is ready to be tracked in the staged area.

git add file4.txt command will add file to tracking in the staged area.

git commit -m <comment> will add the new file to the repository in branch BR-1.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ echo "Hello, Git" > file4.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ ls -l
total 1
-rw-r--r-- 1 ellio 197609 11 Feb 25 22:25 file4.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ cat file4.txt
Hello, Git

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git status
On branch BR-1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file4.txt

nothing added to commit but untracked files present (use "git add" to track)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git add file4.txt
warning: LF will be replaced by CRLF in file4.txt.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git commit -m "First commit in the BR-1 branch"
[BR-1 8fa0cdb] First commit in the BR-1 branch
1 file changed, 1 insertion(+)
create mode 100644 file4.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$
```

git status command shows that we are synchronised across repositories. →

ls -la shows the file in the working directory. →

git ls-files -s shows the file in the staging area. →

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git status
On branch BR-1
nothing to commit, working tree clean

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ ls -la
total 5
drwxr-xr-x 1 ellio 197609  0 Feb 25 22:25 ./
drwxr-xr-x 1 ellio 197609  0 Feb 24 16:14 ../
drwxr-xr-x 1 ellio 197609  0 Feb 25 22:36 .git/
-rw-r--r-- 1 ellio 197609 11 Feb 25 22:25 file4.txt

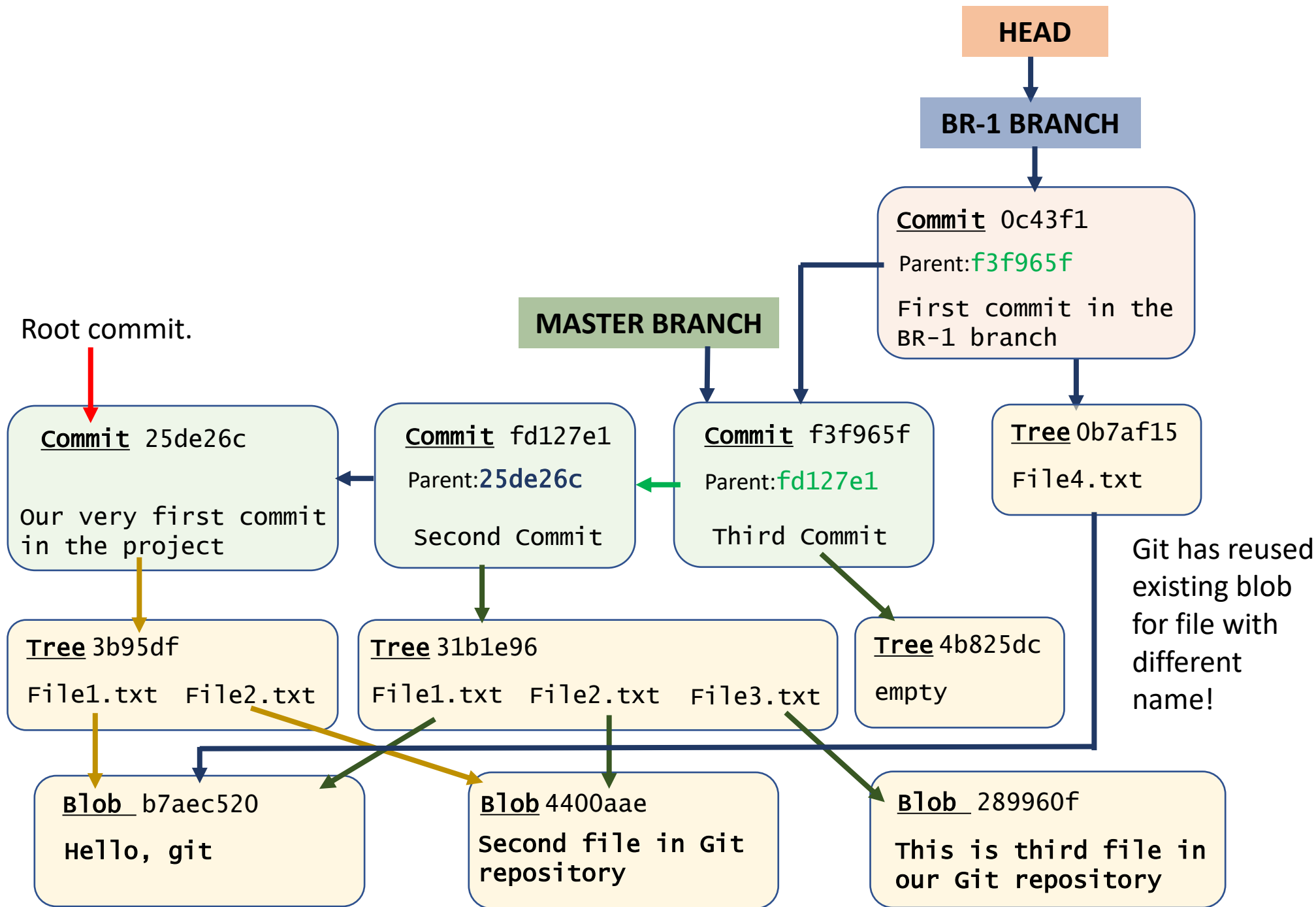
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file4.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$
```

b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e

Notice the hash. It is the same as the blob created in the first commit that had the same contents of “Hello, Git”

Git reused blobs where contents is the same, even if the filenames are not the same because the filename is stored in the tree.




```
elliott@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git log
commit 0c43f10850ead632a23702fefcadfca9ef9ec6da (HEAD -> BR-1)
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 22:28:00 2022 +0100

    First commit in the BR-1 branch

commit f3f965f4dddb3fc863fbe9b3d81567eabd6561f3 (master)
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 02:48:51 2022 +0100

    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project
```

— □ ×



If we look at the contents of the parent of the fourth commit it does indeed show as the third commit.

File Edit View Repository Branch Help

Current repository
first-project

Changes History

Select branch to compare...

First commit in the BR-1 branch
Elliott Farmer • 25m

Third commit
Elliott Farmer • 20h

Second commit
Elliott Farmer • 23h

Our very first commit in the project
Elliott Farmer • 1d

Current branch
BR-1

Filter

New branch

Recent branches

✓ BR-1	25 minutes ago
master	20 hours ago

Publish repository
Publish this repository to GitHub

+1 @@
git

Choose a branch to merge into **BR-1**

**In Branch BR-1 we
a history of
commits including
the first commit in
BR-1 branch.**

**Current branch is set to BR-1
and in the dropdown we can
switch to master branch.**

Summary of Shell Commands

Ls -l *List files in working area.*

git ls-files -s *List files in staging area*

Ls -la git/objects *List objects directories in the repository*

cat git/HEAD *See where the head pointer is indicating*

cat .git/refs/heads/master *See where the master pointer is indicating*

git checkout <commit> *Move the head pointer to a specific commit. Transfer files of that commit to the working and staged area.*

git checkout master *Move the head pointer to the master branch. Files in working and staged areas reflect those of that tree.*

Git branch *Gives a list of all the branches in the project*

Git branch <name> *Create new Branch*

Git checkout <branch name> *Checkout specific named branch*

Git checkout -b <branch name> *Create new branch and check it out at the same time.*

Git -d branch <branch name> *Delete specific named Branch*

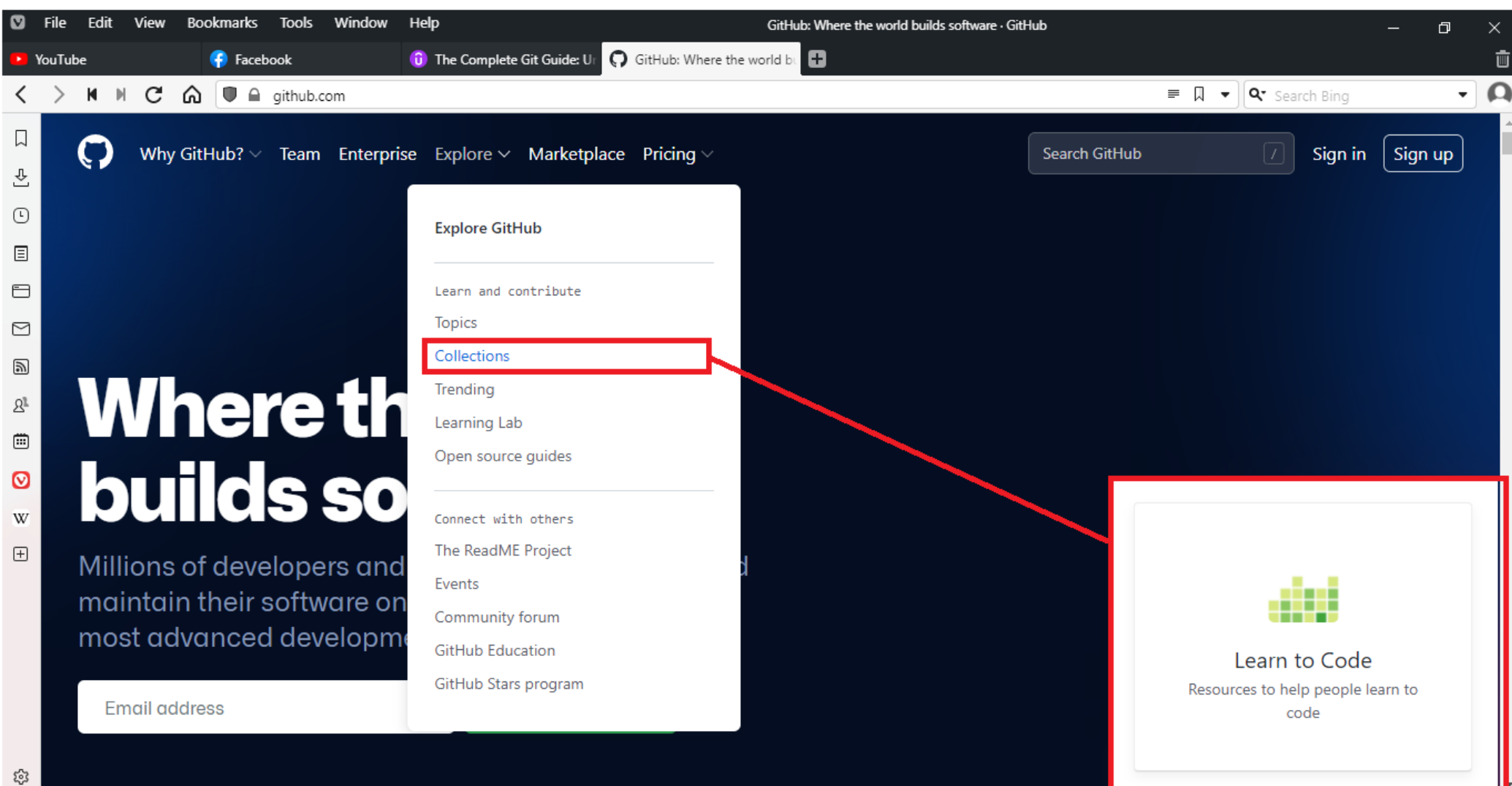
Git -m branch <old> <new> *Rename specific branch*

Git -m branch <current name> <new name> *Rename specific branch*

CLONING, EXPLORING AND MODIFYING PUBLIC REPOSITORIES

Clone GitHub repository

At github.com website select explore -> collections then chose learn to code.



From here we select the hello world link -> code button -> and copy the url
`git clone https://github.com/zz1686/hello-world.git`

leachim6 / hello-world

☆ Star

Hello world in every computer language. Thanks to everyone who contributes to this, make sure to see contributing.md for contribution instructions!

☆ 7887 👤 1569 Assembly

The screenshot shows the GitHub repository page for `leachim6/hello-world`. The repository is public and has 7887 stars and 1569 forks. The 'Code' button is highlighted with a red arrow. The 'Code' dropdown menu is open, showing the 'Clone' option with the HTTPS URL `https://github.com/leachim6/hello-world.git` highlighted. The 'About' section on the right provides a description of the repository and links to the README, MIT License, and Code of Conduct.

Why GitHub? Team Enterprise Explore Marketplace Pricing

leachim6 / hello-world (Public)

Code Issues 6 Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

MrBrain295 Expand .gitignore to include Vscode files

Add Latino, *><>, ><>, Charcoal,

.github Delete stale.yml

a Add Latino, *><>, ><>, Charcoal,

b Add Latino, *><>, ><>, Charcoal,

c Add Latino, *><>, ><>, Charcoal,

d Remove duplicate Db2 yesterday

Clone

HTTPS GitHub CLI

`https://github.com/leachim6/hello-world.git`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Hello world in every computer language. Thanks to everyone who contributes to this, make sure to see contributing.md for contribution instructions!

github.com/leachim6/hello-world/wikis

hello-world hacktoberfest esolangs

example-code

Readme

MIT License

Code of conduct

Git clone command can be used to clone the GitHub repository.

Now there are two projects in the directory.

If we **cd** into the directory we can see that it is under git control and puts us into the master branch.

We can also see the git directory.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/zz1686/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 3814, done.
remote: Total 3814 (delta 0), reused 0 (delta 0), pack-reused 3814
Receiving objects: 100% (3814/3814), 2.23 MiB | 2.20 MiB/s, done.
Resolving deltas: 100% (1812/1812), done.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
first-project/  hello-world/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd hello-world/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ ls -la
total 201
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 './'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 '../'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 './'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 './'
-rw-r--r-- 1 ellio 197609    22 Feb 26 01:42 '.git/'
-rw-r--r-- 1 ellio 197609  1076 Feb 26 01:42 '.gitignore'
-rw-r--r-- 1 ellio 197609  1101 Feb 26 01:42 'CONTRIBUTING.md'
-rw-r--r-- 1 ellio 197609  15262 Feb 26 01:42 'LICENSE'
-rw-r--r-- 1 ellio 197609  15262 Feb 26 01:42 'README.md'
-rw-r--r-- 1 ellio 197609    694 Feb 26 01:42 'README_nolist.md'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'a/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'b/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'c/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'd/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'e/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'f/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'g/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'h/'
drwxr-xr-x 1 ellio 197609    0 Feb 26 01:42 'i/'
```


If we check the HEAD we can see that it points to refs/head/main

Checking the contents of the HEAD we see that we have a hash pointer to a file

This pointer is the last commit of the main branch and references a tree.

Inside the tree we have multiple blobs and tree's

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ cat .git/head
ref: refs/heads/master
```

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ cat .git/refs/heads/master
48eb65bf27272eb7f0887be029b10e7b036b8df3
```

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git cat-file -p 48eb65
tree bbf4bff56a6b504ba211820614ac0fadd60ae361
parent 220bf4da45f6b18ad52823fd1e7ae1ea5c097eaf
parent bf32d839f72a34e58c4b3717f9fb32dcd5ed96d8
author Mike Donaghy <mike@mikedonaghy.org> 1536651169 -0500
committer GitHub <noreply@github.com> 1536651169 -0500
gpgsig -----BEGIN PGP SIGNATURE-----

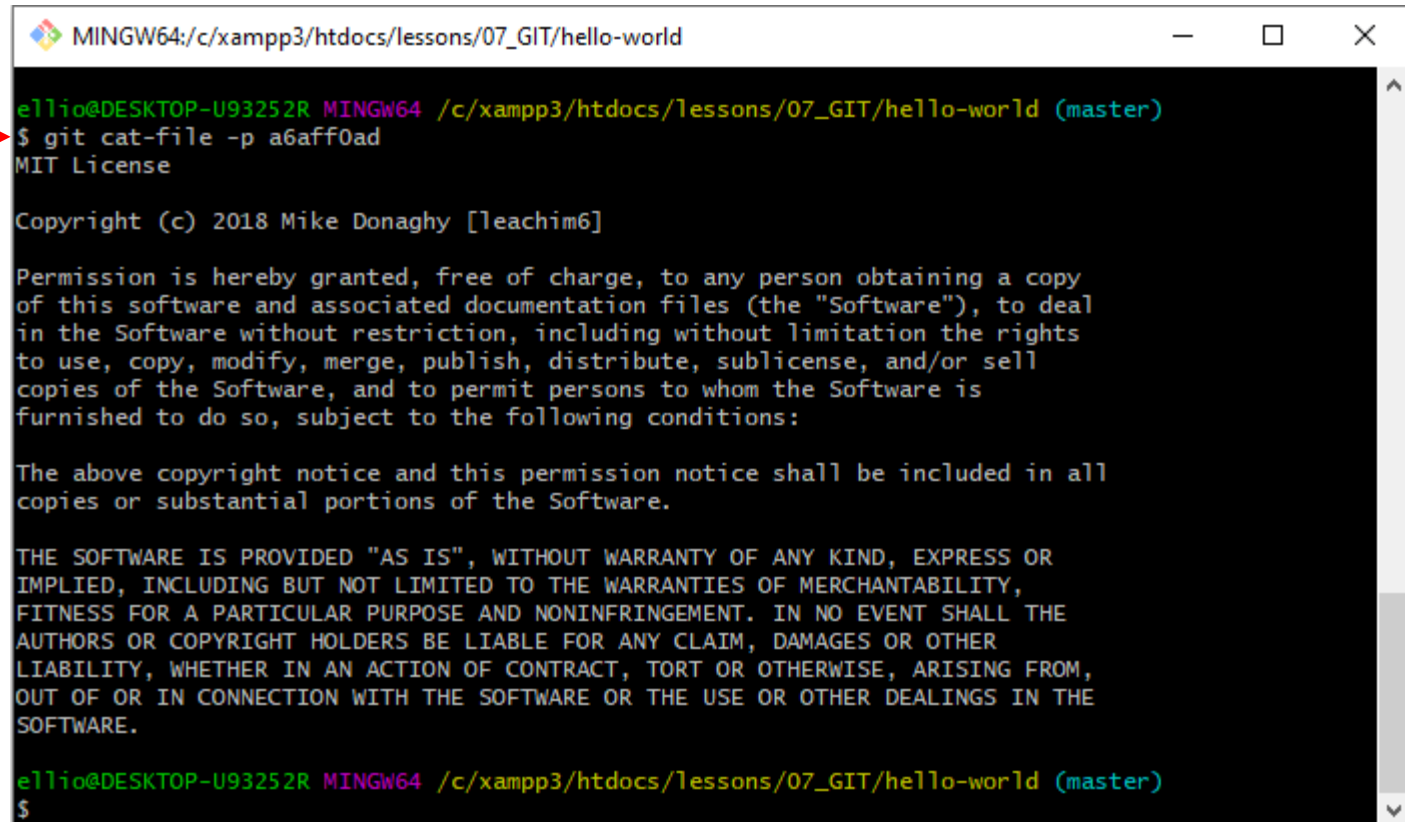
wsBcBAABCAAQBQJb12+hCRBK7hj40v3rIwAAdHIIAAgVtHJIKOs1pFnJ7RIgUNsZ
KqRkjKow11DLshBfdVP68IXM4zmI32DJnaiMIYl1wvj2NsXg2Y9bt6ByY//UL7N1
kzw7FJFAA2U41TX3MLarwXdnUoA6VMjdfQy7WohZ03MKI0iFW3Pw7nfTt6Bi52uj
Fsn06W7TrXG00aq1nD5hhae3615qSY/njxL5ksuPc15PcQp8CCZZ9gmMvXAiKKBi
IbFVs6Nkvi+i0CRDexPvtAPE1ZtNAEkqaGXAbsMr8vkezA45LPMOWvmO1cm6QPwH
1+QkKX5nIMANsBt/7Z5izXtc6UoTGccqZrPV1RaIu/FBT1bgYz6ewvj8HgGaQp0=
=niPk
-----END PGP SIGNATURE-----
```

Merge pull request #499 from x-t/master

Add HolyC RIP Terry Davis

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git cat-file -p bbf4bff
040000 tree d0fde2448221e5a93018ad76ae4ee1fc3d4fc178 #
100644 blob 239beea576fd63cb0e706678ce5fe0c5e08c8384 .gitignore
100644 blob f880ed6a46d5b73783af9d2224737ff6f8012c26 CONTRIBUTING.md
100644 blob a6aff0ad5ed8d1794d6024553a53166c803ad08b LICENSE
100644 blob 8f9fca1ce8adaf5b43b7d28fe9086caba95b85b6 README.md
100644 blob 9d63e6a58d8ebc888b6ff05780ea25f9584ffca9 README_nolist.md
040000 tree e5010e232cb8298b2efa3c80827ae5d5f7264a60 a
040000 tree db3d8ca493739aba4eb4a6e5a2faf34be6841b12 b
040000 tree 8239231919d1203560a91b4fa8704d2f9295602e c
040000 tree eafaca99833fc02b41da3435f9585b1af95d3ea8 d
040000 tree 3f9776fe6970b552b29b21d70a9dd24203f47478 e
040000 tree 930fff9f06479f02c61e5a2a147dab5e8fec0422 f
```

Contents of files can be viewed. In this case I am using the hash of the file called LICENSE.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git cat-file -p a6aff0ad
MIT License

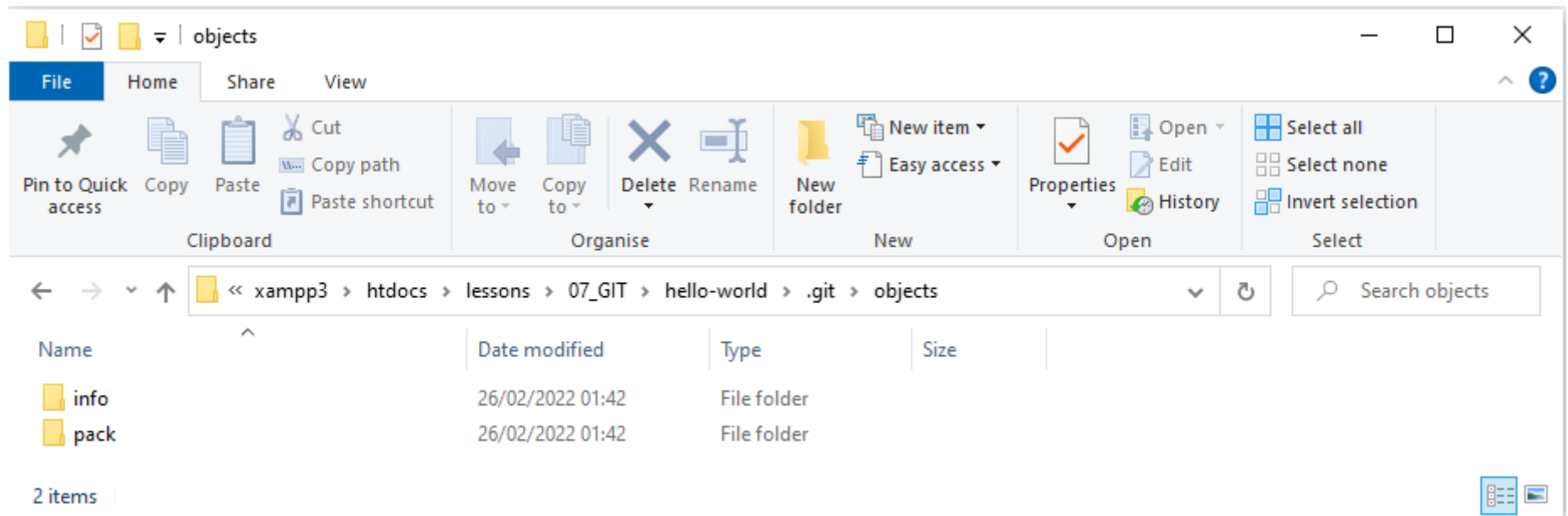
Copyright (c) 2018 Mike Donaghy [leachim6]

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

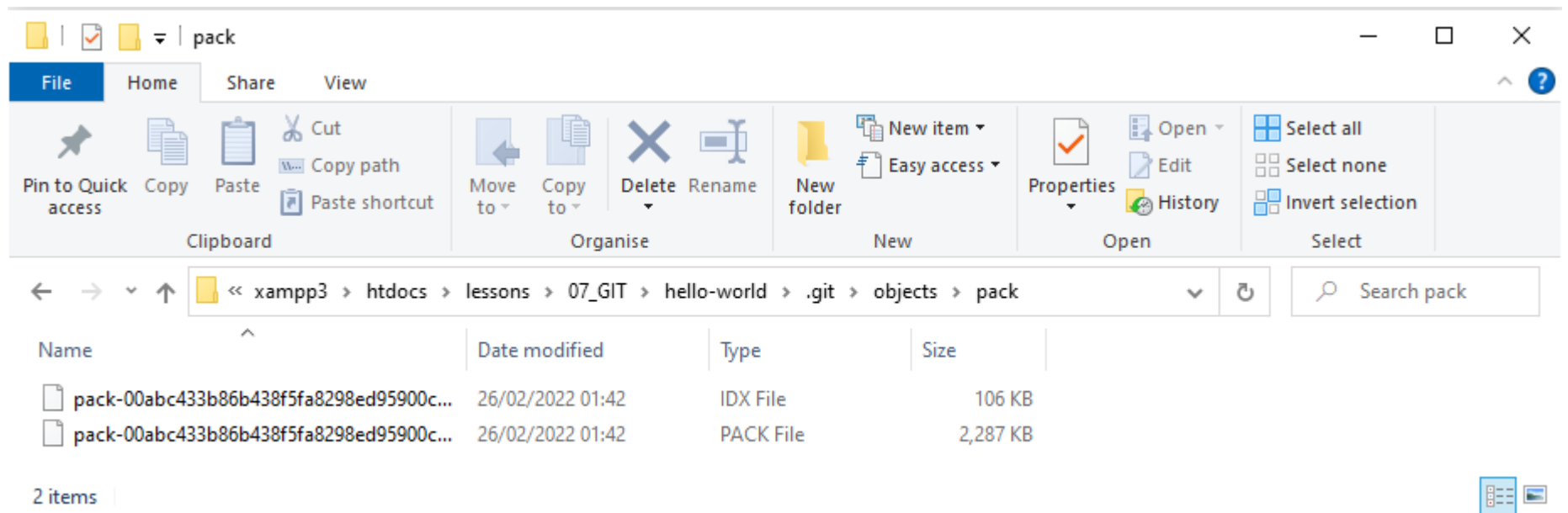
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$
```



In the git/objects folder there are no objects because it has been optimised in the pack folder. Note the 2 mb pack file. The IDX file is an index of this optimised content.



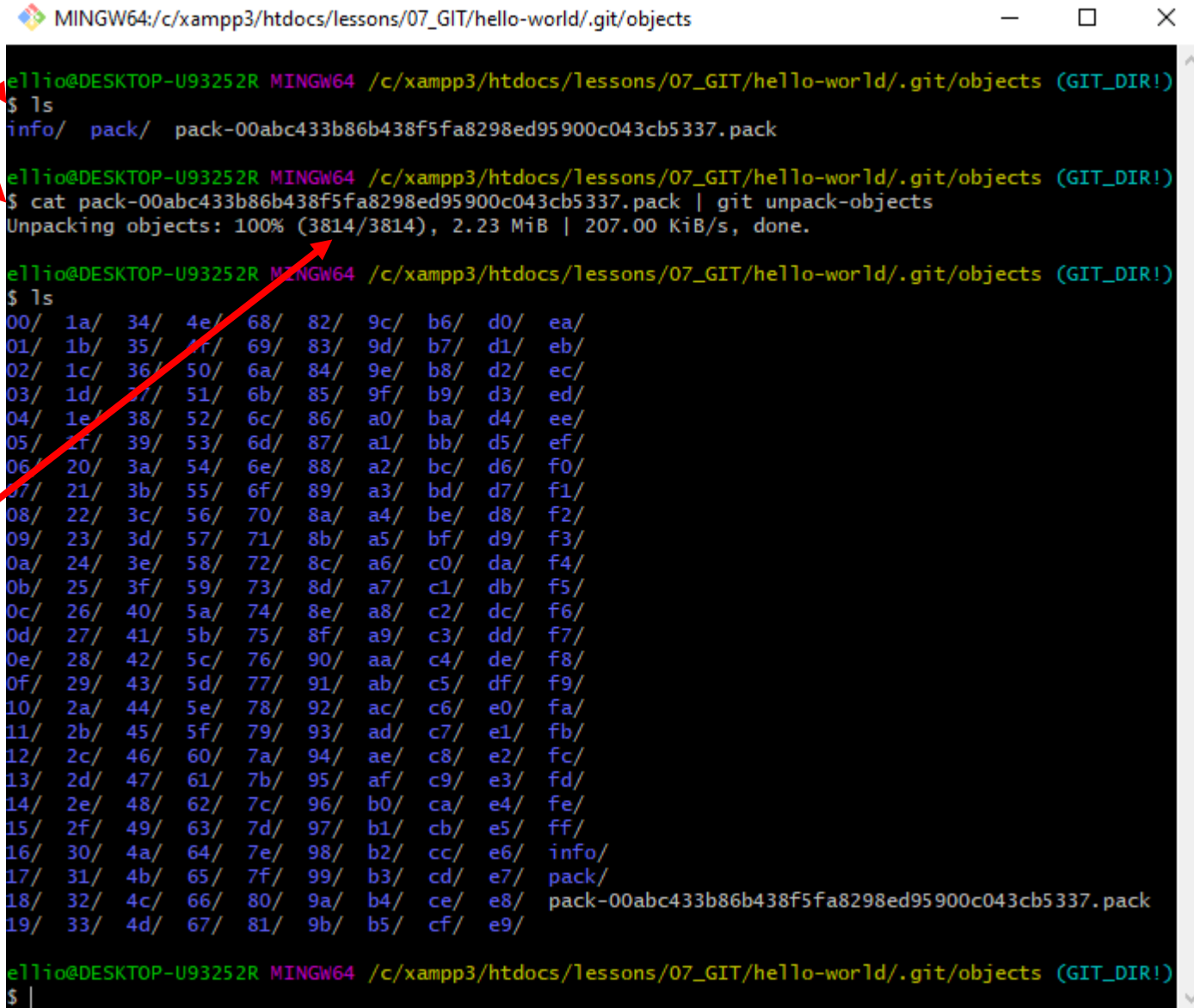
An ls of the .git/objects directory shows the pack file.

Using `cat <filename.pack> | git unpack-objects` command the contents of the pack file can be unpacked into the current folder, objects, where we can see all the folders with all the blobs, trees, commits and annotated tags.

Git has actually unpacked about 4000 files!

Notice how there are 256 directories. 16x16. this git repository is using the maximum number of objects directories.

In explorer the pack file can be moved manually to the objects folder.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world/.git/objects
$ ls
info/  pack/  pack-00abc433b86b438f5fa8298ed95900c043cb5337.pack

$ cat pack-00abc433b86b438f5fa8298ed95900c043cb5337.pack | git unpack-objects
Unpacking objects: 100% (3814/3814), 2.23 MiB | 207.00 KiB/s, done.

$ ls
00/  1a/  34/  4e/  68/  82/  9c/  b6/  d0/  ea/
01/  1b/  35/  4f/  69/  83/  9d/  b7/  d1/  eb/
02/  1c/  36/  50/  6a/  84/  9e/  b8/  d2/  ec/
03/  1d/  37/  51/  6b/  85/  9f/  b9/  d3/  ed/
04/  1e/  38/  52/  6c/  86/  a0/  ba/  d4/  ee/
05/  1f/  39/  53/  6d/  87/  a1/  bb/  d5/  ef/
06/  20/  3a/  54/  6e/  88/  a2/  bc/  d6/  f0/
07/  21/  3b/  55/  6f/  89/  a3/  bd/  d7/  f1/
08/  22/  3c/  56/  70/  8a/  a4/  be/  d8/  f2/
09/  23/  3d/  57/  71/  8b/  a5/  bf/  d9/  f3/
0a/  24/  3e/  58/  72/  8c/  a6/  c0/  da/  f4/
0b/  25/  3f/  59/  73/  8d/  a7/  c1/  db/  f5/
0c/  26/  40/  5a/  74/  8e/  a8/  c2/  dc/  f6/
0d/  27/  41/  5b/  75/  8f/  a9/  c3/  dd/  f7/
0e/  28/  42/  5c/  76/  90/  aa/  c4/  de/  f8/
0f/  29/  43/  5d/  77/  91/  ab/  c5/  df/  f9/
10/  2a/  44/  5e/  78/  92/  ac/  c6/  e0/  fa/
11/  2b/  45/  5f/  79/  93/  ad/  c7/  e1/  fb/
12/  2c/  46/  60/  7a/  94/  ae/  c8/  e2/  fc/
13/  2d/  47/  61/  7b/  95/  af/  c9/  e3/  fd/
14/  2e/  48/  62/  7c/  96/  b0/  ca/  e4/  fe/
15/  2f/  49/  63/  7d/  97/  b1/  cb/  e5/  ff/
16/  30/  4a/  64/  7e/  98/  b2/  cc/  e6/  info/
17/  31/  4b/  65/  7f/  99/  b3/  cd/  e7/  pack/
18/  32/  4c/  66/  80/  9a/  b4/  ce/  e8/  pack-00abc433b86b438f5fa8298ed95900c043cb5337.pack
19/  33/  4d/  67/  81/  9b/  b5/  cf/  e9/

$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world/.git/objects

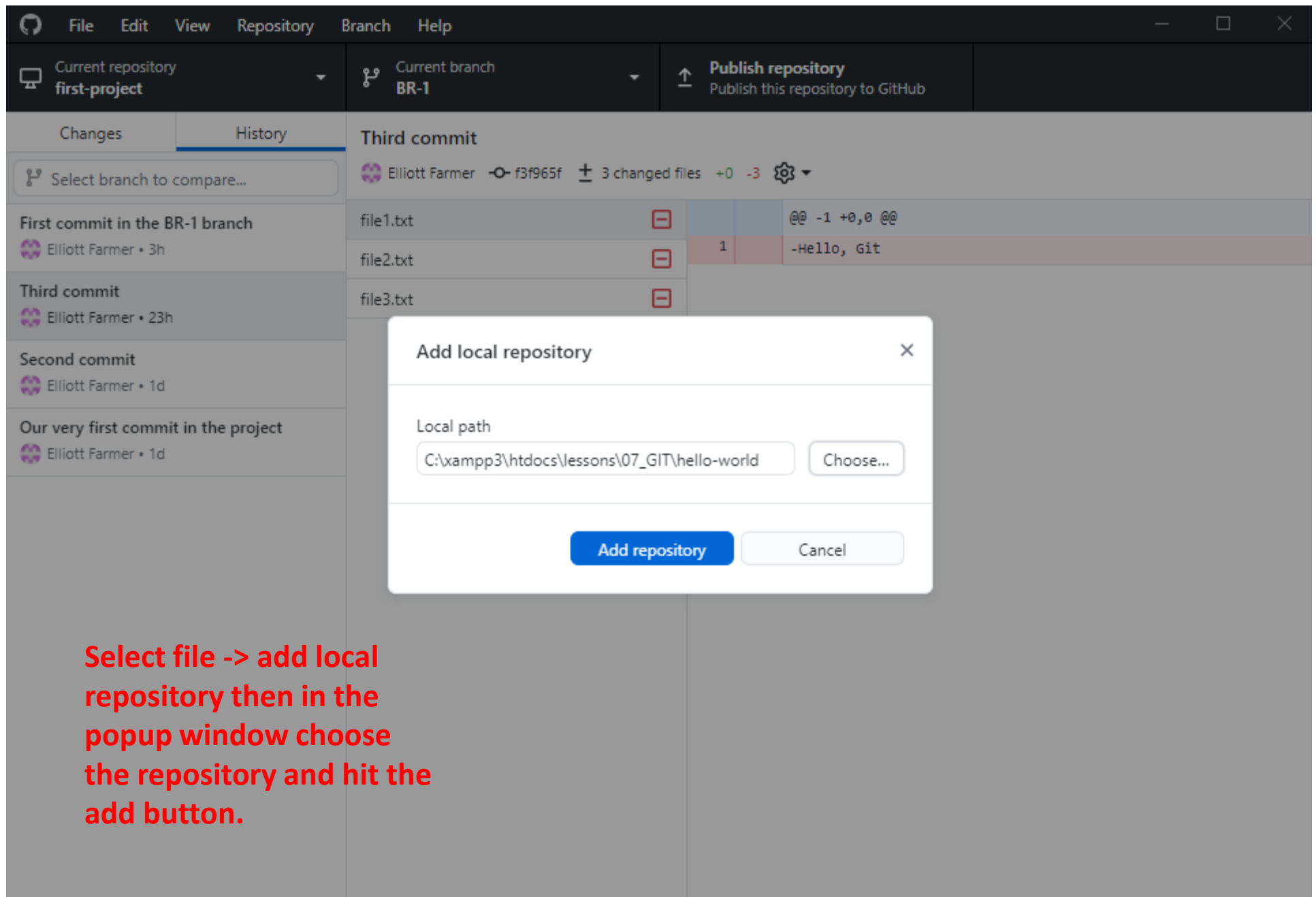
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world/.git/objects (GIT_DIR!)
$ rm pack-00abc433b86b438f5fa8298ed95900c043cb5337.pack
rm: remove write-protected regular file 'pack-00abc433b86b438f5fa8298ed95900c043cb5337.pack'? y

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world/.git/objects (GIT_DIR!)
$ ls
00/ 0e/ 1c/ 2a/ 38/ 46/ 54/ 62/ 70/ 7e/ 8c/ 9a/ a8/ b6/ c4/ d2/ e0/ ee/ fc/
01/ 0f/ 1d/ 2b/ 39/ 47/ 55/ 63/ 71/ 7f/ 8d/ 9b/ a9/ b7/ c5/ d3/ e1/ ef/ fd/
02/ 10/ 1e/ 2c/ 3a/ 48/ 56/ 64/ 72/ 80/ 8e/ 9c/ aa/ b8/ c6/ d4/ e2/ f0/ fe/
03/ 11/ 1f/ 2d/ 3b/ 49/ 57/ 65/ 73/ 81/ 8f/ 9d/ ab/ b9/ c7/ d5/ e3/ f1/ ff/
04/ 12/ 20/ 2e/ 3c/ 4a/ 58/ 66/ 74/ 82/ 90/ 9e/ ac/ ba/ c8/ d6/ e4/ f2/ info/
05/ 13/ 21/ 2f/ 3d/ 4b/ 59/ 67/ 75/ 83/ 91/ 9f/ ad/ bb/ c9/ d7/ e5/ f3/ pack/
06/ 14/ 22/ 30/ 3e/ 4c/ 5a/ 68/ 76/ 84/ 92/ a0/ ae/ bc/ ca/ d8/ e6/ f4/
07/ 15/ 23/ 31/ 3f/ 4d/ 5b/ 69/ 77/ 85/ 93/ a1/ af/ bd/ cb/ d9/ e7/ f5/
08/ 16/ 24/ 32/ 40/ 4e/ 5c/ 6a/ 78/ 86/ 94/ a2/ b0/ be/ cc/ da/ e8/ f6/
09/ 17/ 25/ 33/ 41/ 4f/ 5d/ 6b/ 79/ 87/ 95/ a3/ b1/ bf/ cd/ db/ e9/ f7/
0a/ 18/ 26/ 34/ 42/ 50/ 5e/ 6c/ 7a/ 88/ 96/ a4/ b2/ c0/ ce/ dc/ ea/ f8/
0b/ 19/ 27/ 35/ 43/ 51/ 5f/ 6d/ 7b/ 89/ 97/ a5/ b3/ c1/ cf/ dd/ eb/ f9/
0c/ 1a/ 28/ 36/ 44/ 52/ 60/ 6e/ 7c/ 8a/ 98/ a6/ b4/ c2/ d0/ de/ ec/ fa/
0d/ 1b/ 29/ 37/ 45/ 53/ 61/ 6f/ 7d/ 8b/ 99/ a7/ b5/ c3/ d1/ df/ ed/ fb/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world/.git/objects (GIT_DIR!)
$
```

As the pack file is now unpacked it is no longer needed and can be removed to clean up the objects directory.

If rm command does not work then try rm -f as there may be a permissions issue.



Select file -> add local repository then in the popup window choose the repository and hit the add button.

FileEditViewRepositoryBranchHelp

Current repository
hello-world

Current branch
master

Fetch origin
Last fetched 38 minutes ago

ChangesHistory

No branches to compare

Merge pull request #499 from x-t/mast...
Mike Donaghy, GitHub • Sep 11, 2018

Create holyc.HC
cmp, GitHub • Sep 7, 2018

Merge pull request #478 from chinsyo/...
Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #477 from cheezeb...
Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #476 from Pocable/...
Mike Donaghy, GitHub • Jul 23, 2018

add nasm macho64
Chinsyo • Jul 23, 2018

Added aspectj.aj
cheezeburger • Jul 22, 2018

Added MIPS
Thomas Roskewich • Jul 18, 2018

Merge pull request #418 from thathexa...
Mike Donaghy, GitHub • Jul 17, 2018

Merge pull request #467 from SLIB53/...
Mike Donaghy, GitHub • Jul 17, 2018

Merge pull request #499 from x-t/master

Mike Donaghy, GitHub 48eb65b ± 1 changed file +1 -0 ⚙

Add HolyC RIP Terry Davis

h\holyc.HC +

@@ -0,0 +1 @@

1 +"Hello, world!";

Fetch origin will download latest updates to the repository from the remote repository on GitHub

The history shows all the commits. On the highlighted commit we see various actions:

[.] this is a change to a file [-] this is a removal of file [+] this is addition of file

The screenshot displays a code editor interface with a dark theme. The top menu bar includes 'File', 'Edit', 'View', 'Repository', 'Branch', and 'Help'. Below the menu, the 'Current repository' is 'hello-world' and the 'Current branch' is 'master'. The 'Fetch origin' button shows 'Last fetched 3 minutes ago'.

The left sidebar shows a list of merge pull requests. The selected item is 'Merge pull request #418 from thathexa/master' by Mike Donaghy, GitHub, dated Jul 23, 2018. A tooltip for this item shows 'Merge pull request #418 from thathexa/master'.

The main area shows a diff view for the selected commit. The diff is titled 'Merge pull request #418 from thathexa/master' and shows changes to 'README.md', 'n\nnodejs', 'n\nnodejs.js', and 's\hello.st → s\smalltalk-gnu.st'. The diff is divided into sections by commit hashes. The first section, starting with '@@ -334,7 +334,6 @@', shows changes to 'Meet [FizzBuzz](https://github.com/zenwar e/FizzBuzz), the evolution of [hello-wo]'. The second section, starting with '@@ -419,7 +418,6 @@', shows changes to 'Meet [FizzBuzz](https://github.com/zenwar e/FizzBuzz), the evolution of [hello-wo]'. The third section, starting with '@@ -443,6 +441,7 @@', shows changes to 'Meet [FizzBuzz](https://github.com/zenwar e/FizzBuzz), the evolution of [hello-wo]'. The diff shows additions (blue background) and deletions (red background). Red arrows point from the text '[.] this is a change to a file' to a yellow square icon and from '[-] this is a removal of file' to a red square icon.

The bottom of the screen shows a Windows taskbar with various application icons and a system tray with the date '26/02/2022' and time '02:31'.

In visual studio code we can select file -> open folder -> then chose the project.

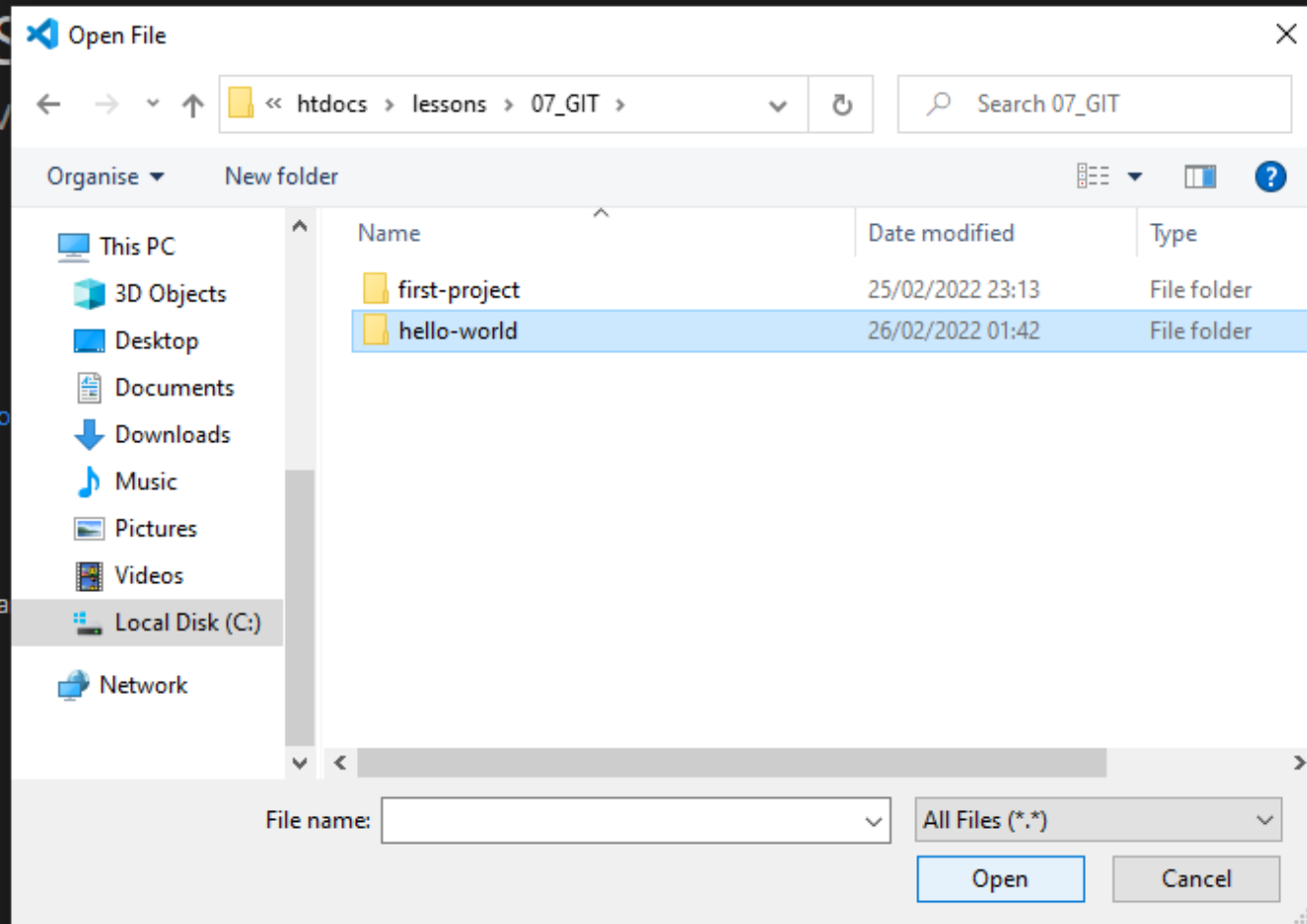
Visual S
Editing ev

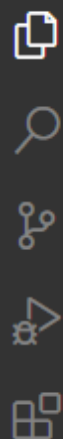
Start

- New File...
- Open File...
- Open Folder...
- Clone Git Repo

Recent

hello-world C:\xa





EXPLORER

HELLO-WORLD

> h
> i
> j
> k
> l
> m
> n
> o
> p
> q
> r
> s
> t
> u
> v
> w
> x
> y
> z

Here we see the contents of the folder and we can click on it to open a file in the editor pane.

.gitignore
CONTRIBUTING.md
LICENSE
list_langs.py
README_nolist.md
README.md> OUTLINE
> TIMELINE

README_nolist.md X

README_nolist.md > # Hello, World!

```
1 Hello, World!
2 =====
3
4 > Inspired by [The Hello World Collection](https://helloworldcollection.github.io/).
5
6 #### Hello world in every programming language.
7
8 As I watch the collection expand, this project has blown up more than I
9 ever thought possible.
10 Thanks to everyone who continues to contribute, new languages are
11 created every day!
12
13 Make sure to see CONTRIBUTING.md for instructions on contributing to
14 the project!
15
16 Spin-Off project smartly suggested and implemented by [@zenware](https://github.com/zenware):
17 Meet [FizzBuzz](https://github.com/zenware/FizzBuzz), the evolution of
18 [hello-world](https://github.com/leachim6/hello-world).
19
20 * [Hello Html](e/elm.elm)
```

git branch <branch name>

command can be used to create a new branch.

We can see the newly created branch with a **git branch** command.

The newly created branch can be checked out with **git checkout <branch name>** command.

ls command shows a list of directories and files in the working directory.

Now we can make changes to files in this new branch. For example we can open a nano editor for file README.md

The line **#### Hello world in every programming language** has been altered to **#### Hello world in all programming languages**

Also changed is the line: *Make sure to see CONTRIBUTING.md file for instructions on contributing to this project.*

Control X and **y** to exit and save changes.

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git branch new-feature

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git branch
* master
  new-feature

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git checkout new-feature
Switched to branch 'new-feature'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ ls
# /          README_nolist.md  d/  h/  l/          o/  s/  w/
CONTRIBUTING.md a/          e/  i/  list_langs.py* p/  t/  x/
LICENSE       b/          f/  j/  m/          q/  u/  y/
README.md     c/          g/  k/  n/          r/  v/  z/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ nano README.md
```

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world

```
GNU nano 5.9 README.md Modified
Hello, World!
=====
> Inspired by [The Hello World Collection](https://helloworldcollection.github.io/).

#### Hello world in all programming languages.

As I watch the collection expand, this project has blown up more than I ever thought possible.
Thanks to everyone who continues to contribute, new languages are created every day!

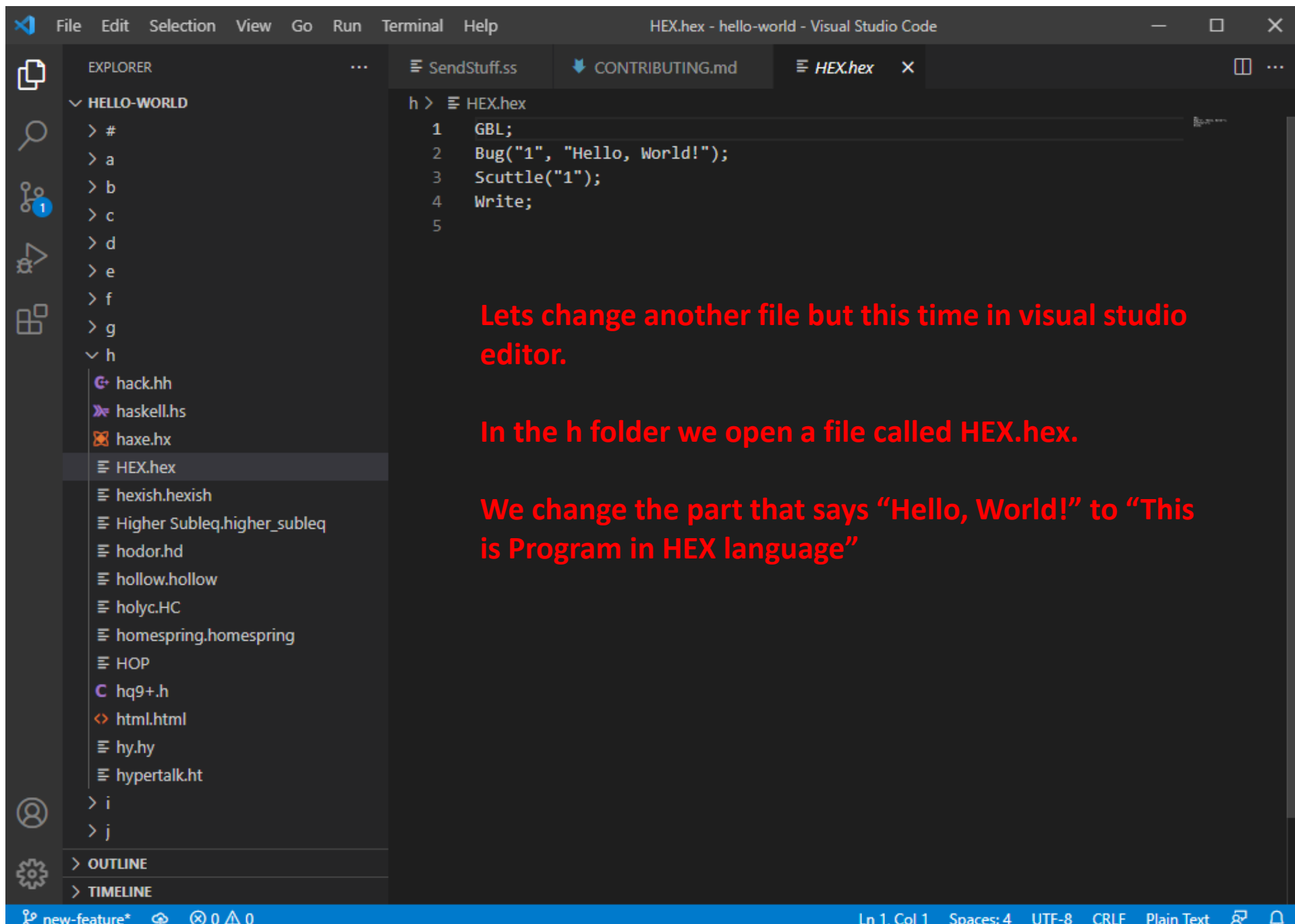
Make sure to see CONTRIBUTING.md file for instructions on contributing to the project!

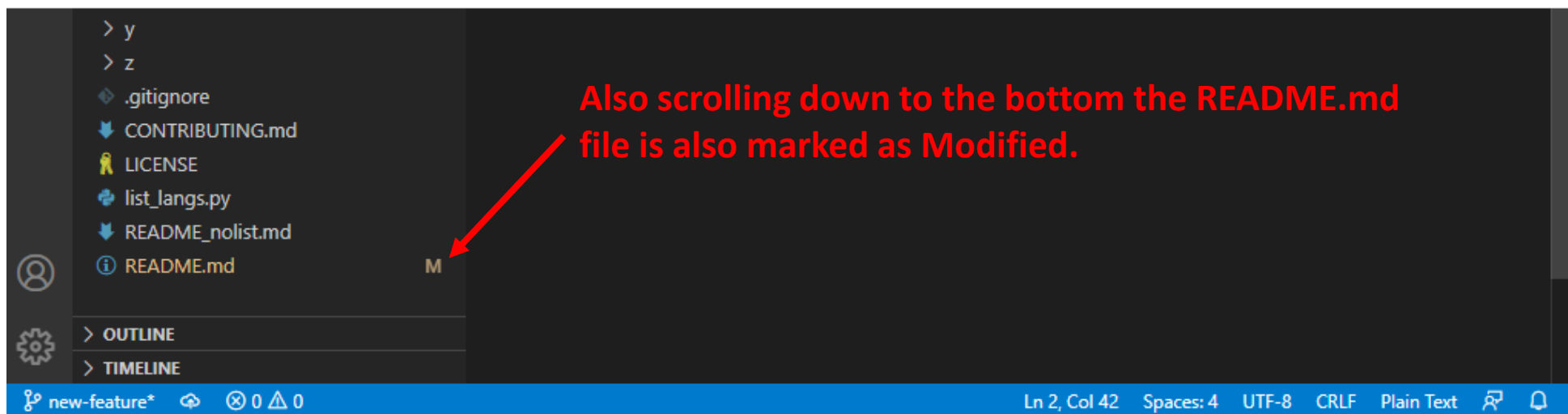
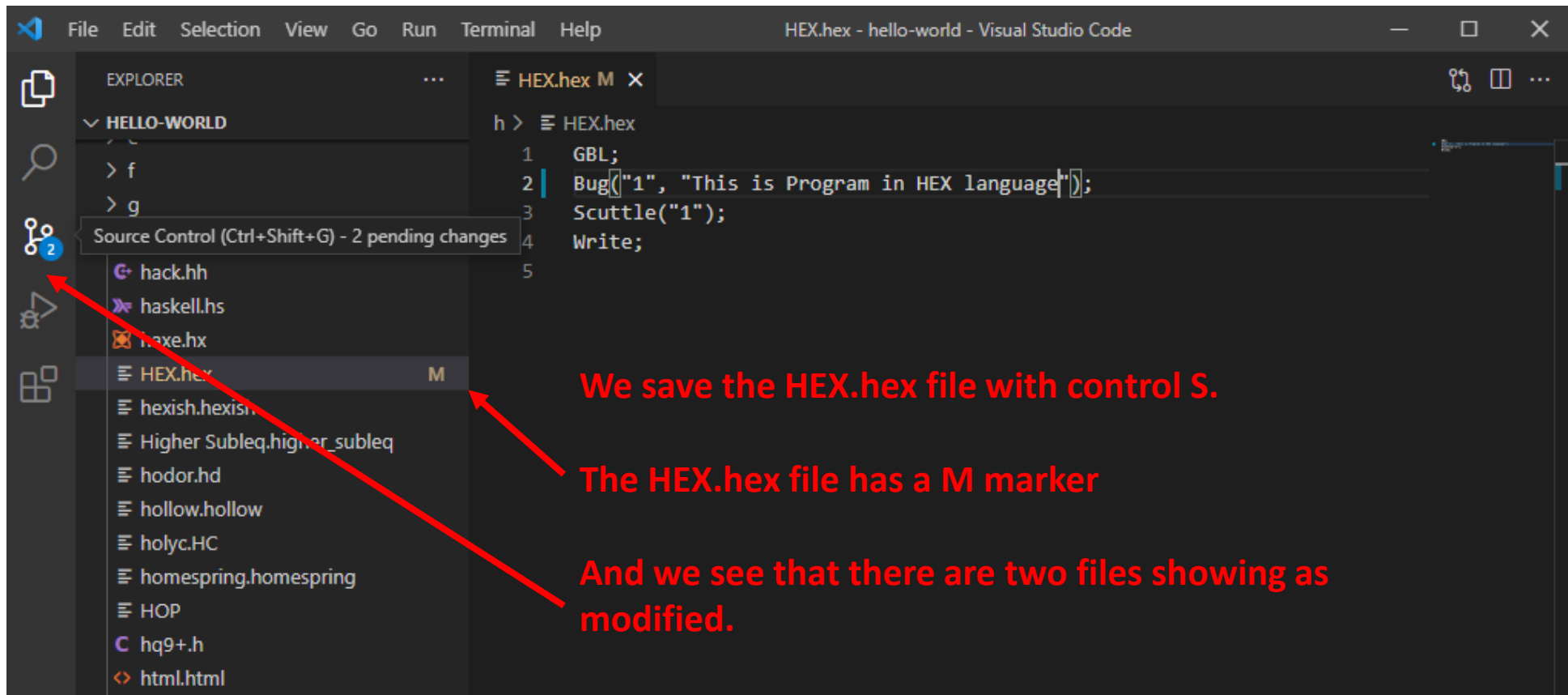
Spin-Off project smartly suggested and implemented by [@zenware](https://github.com/zenware),
Meet [FizzBuzz](https://github.com/zenware/FizzBuzz), the evolution of [hello-world](https://github.com/zenware/hello-world).

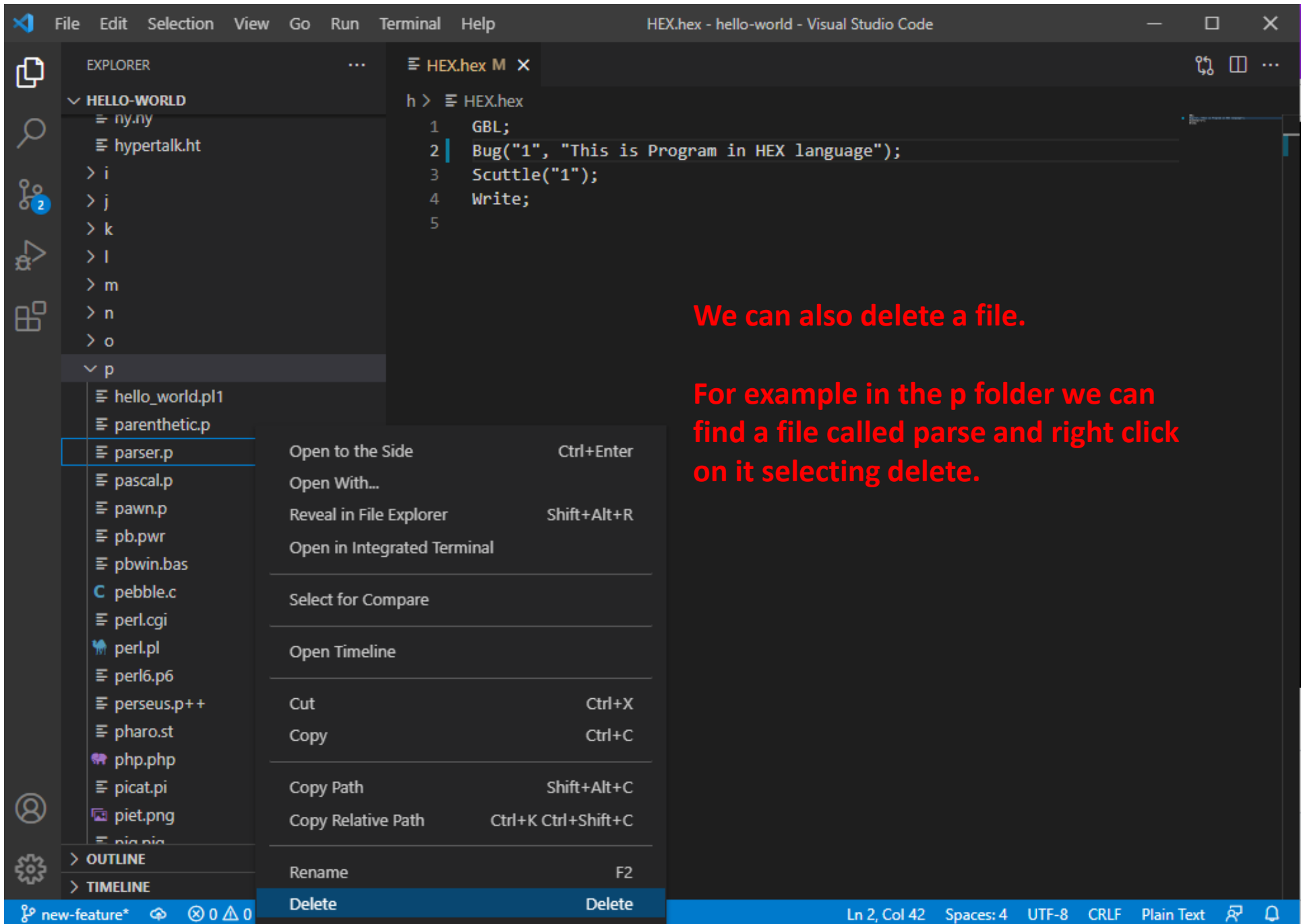
* [Hello Html](e/elm.elm)

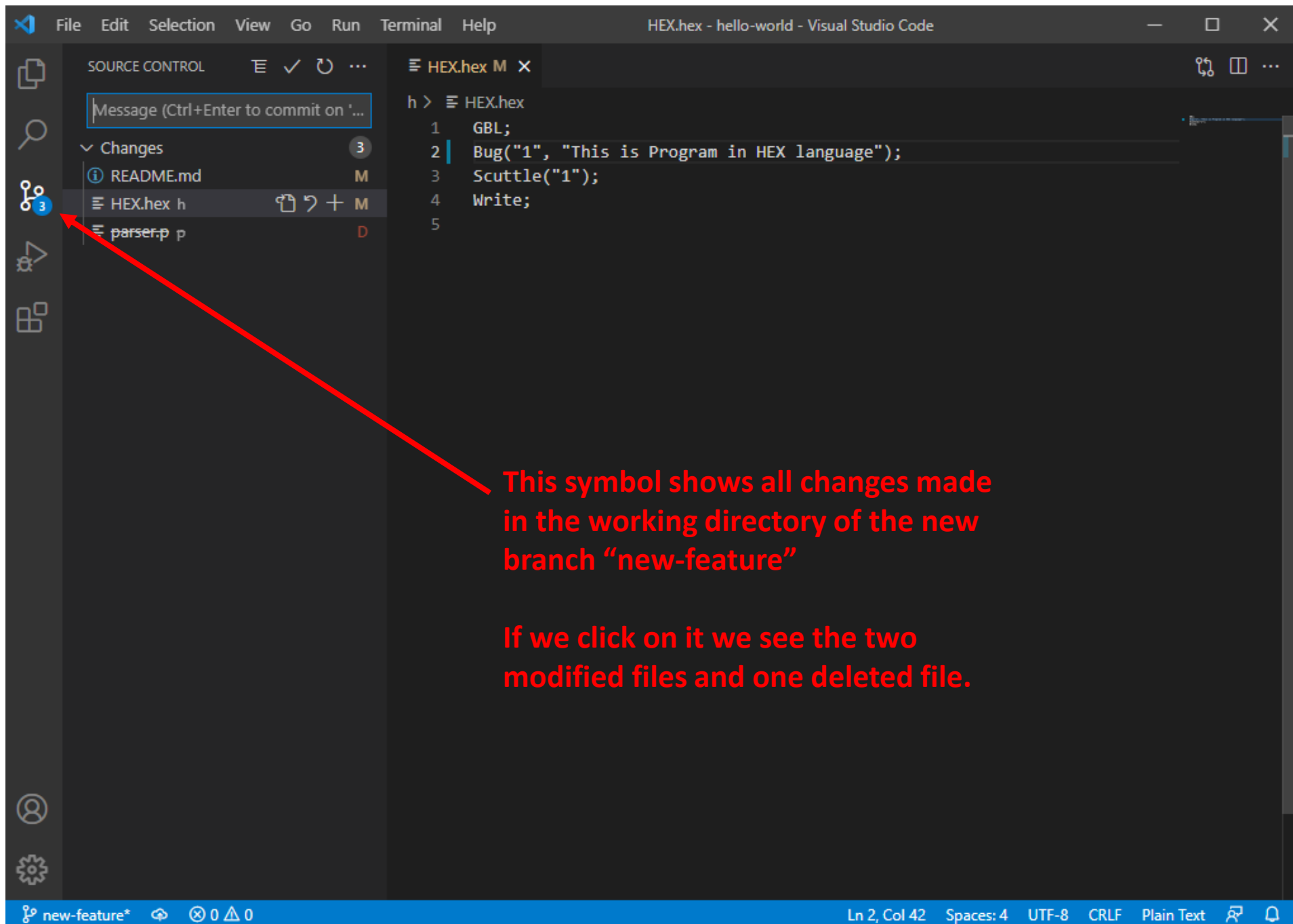
### This repository currently contains "Hello World" programs in the following languages:
* [0815](%23/0815.0815)
* [1C Enterprise](%23/1c-enterprise)

AG Help      AO Write Out  AW Where Is  AK Cut       AT Execute   AC Location
AX Exit      AR Read File  \ Replace   AU Paste     AJ Justify   ^/ Go To Line
```



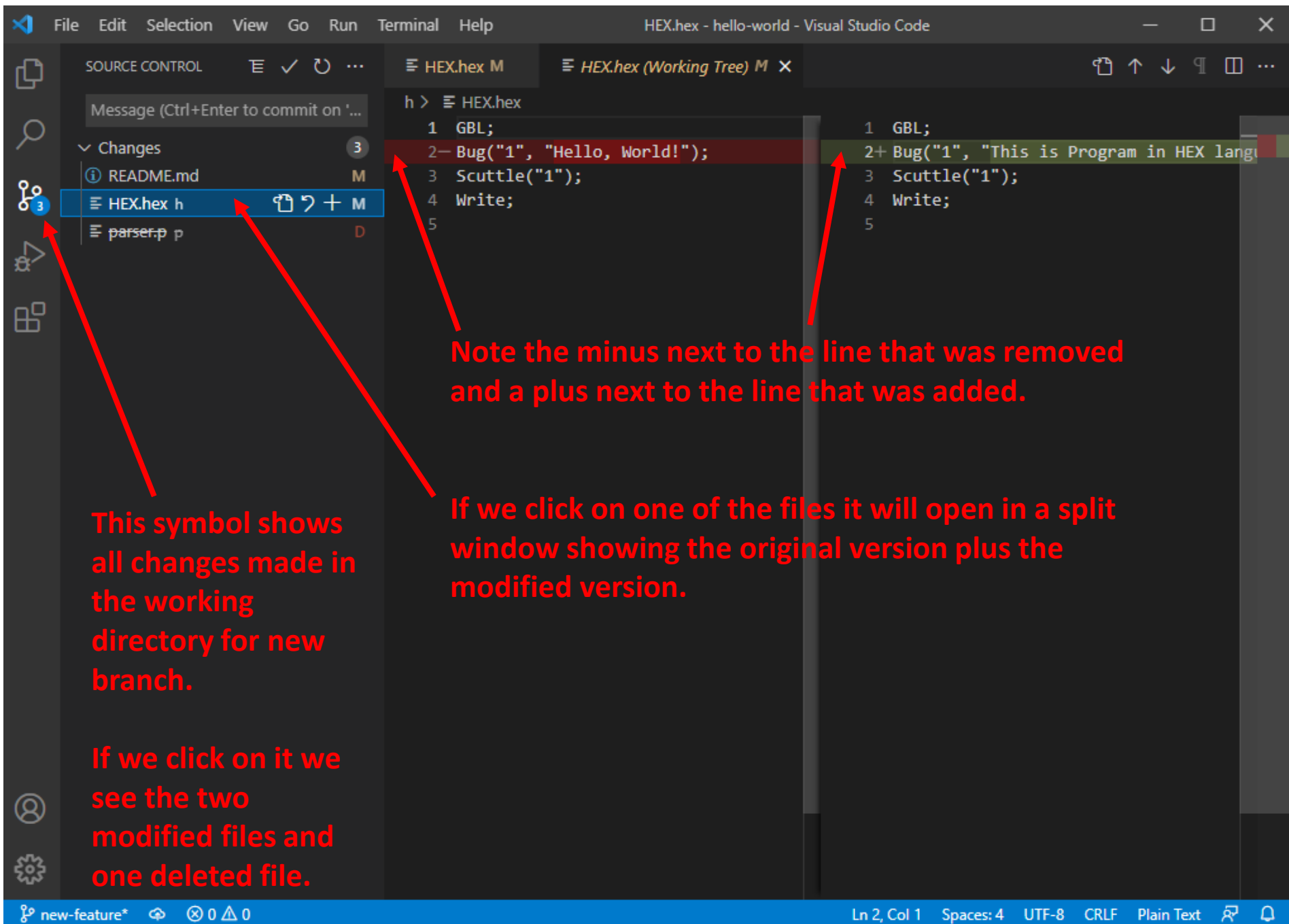






This symbol shows all changes made in the working directory of the new branch "new-feature"

If we click on it we see the two modified files and one deleted file.



File Edit Selection View Go Run Terminal Help

README.md - hello-world - Visual Studio Code

SOURCE CONTROL

Message (Ctrl+Enter to commit on '...')

Changes

- README.md
- HEX.hex h
- parser.p p

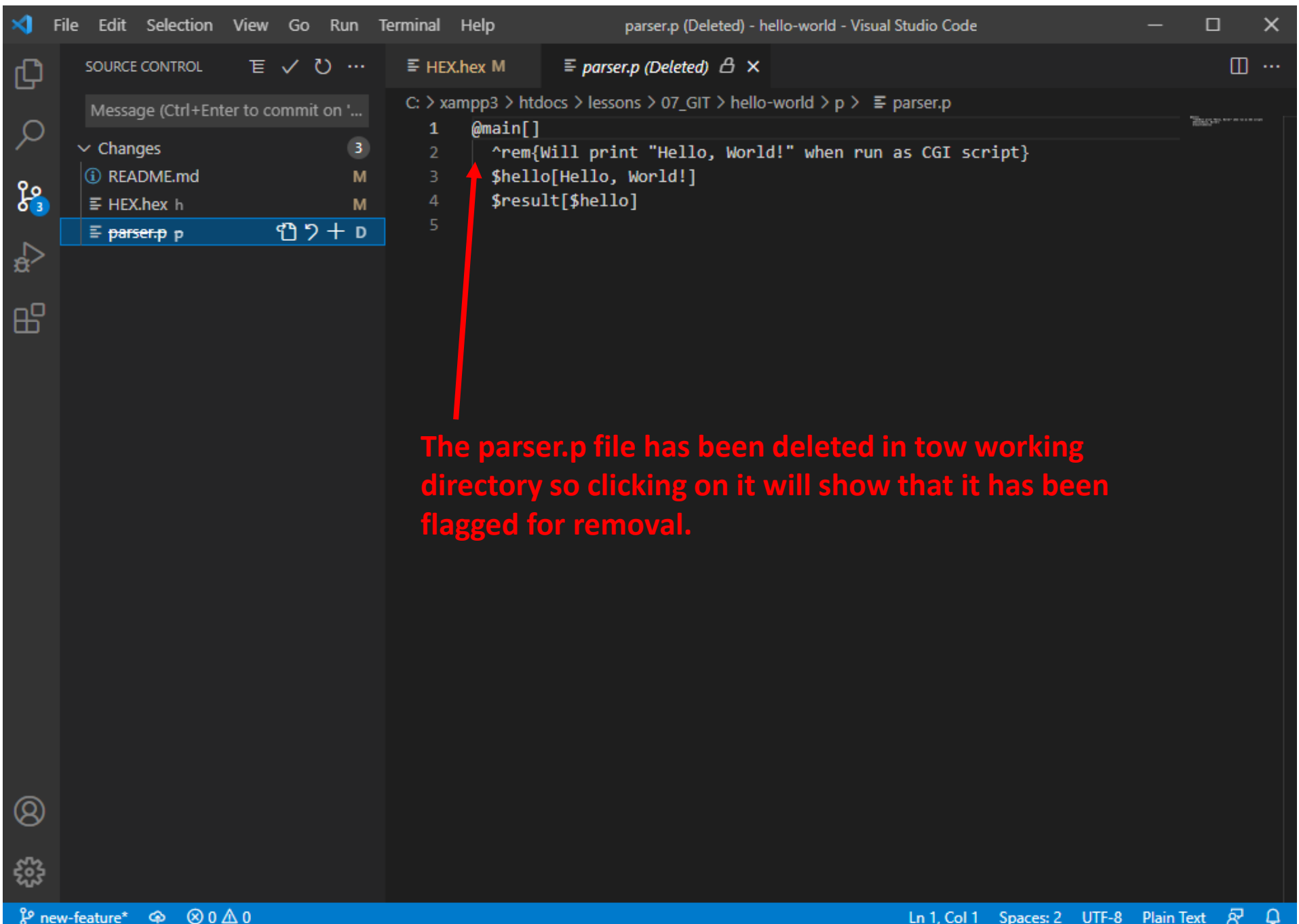
Clicking on the README.md file shows the changes.

① README.md > # Hello, World! > ##### Hello world in all programming languages.

```
1 Hello, World!
2 =====
3
4 > Inspired by [The Hello World
  Collection](https://
  helloworldcollection.github.io/).
5
6- ##### Hello world in every
  - programming language.
7
8 As I watch the collection expand,
  this project has blown up more
  than I ever thought possible.
9 Thanks to everyone who continues
  to contribute, new languages are
  created every day!
10
11- Make sure to see CONTRIBUTING.md
  - for instructions on contributing
  - to the project!
12
13 Spin-Off project smartly suggested
  and implemented by [@zenware]
  (https://github.com/zenware):
14 Meet [FizzBuzz](https://github.com/
  zenware/FizzBuzz), the evolution
  of [hello-world](https://github.
  com/leachim6/hello-world).
15
16 * [Hello Html](e/elm.elm)
17
18 ### This repository currently
  contains "Hello World" programs in
```

1 Hello, World!
2 =====
3
4 > Inspired by [The Hello World
 Collection](https://
 helloworldcollection.github.io/).
5
6+ ##### Hello world in all
 + programming languages.
7
8 As I watch the collection expand,
 this project has blown up more
 than I ever thought possible.
9 Thanks to everyone who continues
 to contribute, new languages are
 created every day!
10
11+ Make sure to see CONTRIBUTING.md
 + file for instructions on
 + contributing to the project!
12
13 Spin-Off project smartly suggested
 and implemented by [@zenware]
 (https://github.com/zenware):
14 Meet [FizzBuzz](https://github.com/
 zenware/FizzBuzz), the evolution
 of [hello-world](https://github.
 com/leachim6/hello-world).
15
16 * [Hello Html](e/elm.elm)
17
18 ### This repository currently
 contains "Hello World" programs in

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Markdown



git status command shows the three files that have been changed in the working directory. →

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git status
On branch new-feature
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
        modified:   h/HEX.hex
        deleted:    p/parser.p

no changes added to commit (use "git add" and/or "git commit -a")

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ |
```

FileEditViewRepositoryBranchHelp

Current repository
hello-world

Current branch
new-feature

↑ Publish branch

Publish this branch to the remote

Changes 3

History

Select branch to compare...

Merge pull request #499 from x-t/mast...
Mike Donaghy, GitHub • Sep 11, 2018

Create holyc.HC
cmp, GitHub • Sep 7, 2018

Merge pull request #478 from chinsyo/...
Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #477 from cheezeb...
Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #476 from Pocable/...
Mike Donaghy, GitHub • Jul 23, 2018

add nasm macho64
Chinsyo • Jul 23, 2018

Added aspectj.aj
cheezeburger • Jul 22, 2018

Added MIPS
Thomas Roskewich • Jul 18, 2018

Merge pull request #418 from thathexa...
Mike Donaghy, GitHub • Jul 17, 2018

Merge pull request #467 from SLIB53/...
Mike Donaghy, GitHub • Jul 17, 2018

Merge branch 'master' into add-clang

Merge pull request #499 from x-t/master

Mike Donaghy, GitHub 48eb65b ± 1 changed file +1 -0 ⚙

Add HolyC RIP Terry Davis

h\holyc.HC

@@ -0,0 +1 @@

1 + "Hello, world!";

The history tab of git hub desktop shows the last commit changes to be made to the local clone of the repository.

FileEditViewRepositoryBranchHelp

Current repository
hello-world

Current branch
new-feature

↑ Publish branch

Publish this branch to the remote

Changes 3

History

README.md

3 changed files

h\HEX.hex

p\parser.p

README.md

↑

@@ -3,12 +3,12 @@ Hello, World!

33

44

55

66

77

88

99

1010

1111

1212

1313

1414

↓

@@ -3,12 +3,12 @@ Hello, World!

> Inspired by [The Hello World Collection](https://helloworldcollection.github.io/).

-#### Hello world in every programming language.

+#### Hello world in all programming languages.

As I watch the collection expand, this project has blown up more than I ever thought possible.

Thanks to everyone who continues to contribute, new languages are created every day!

-Make sure to see CONTRIBUTING.md for instructions on contributing to the project!

+Make sure to see CONTRIBUTING.md file for instructions on contributing to the project!

Spin-Off project smartly suggested and implemented by [@zenware](https://github.com/zenware):

Meet [FizzBuzz](https://github.com/zenware/FizzBuzz), the evolution of [hello-world](https://github.com/leachim6/hello-world).

Summary (required)

Description

Commit to new-feature

The changes tab shows pending changes that have not yet been committed to the local clone of the repository.

git add . Command will add all changed files to the staging area.

If we list the contents of the staging area with a **git ls-files --s** command we see in the list the two modified files.

git status shows the files ready to be committed to branch *"new-feature"*

We can commit the changes to *"new feature"* branch with a **git commit -m "comment"** command.

The output of this command shows that 3 files have been changed, 3 insertions (this is 3 insertions of lines of text/code) And 7 deletions (7 lines of text/code removed). As well as delete mode for parser.p file.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git ls-files --s
100644 7414c30d49cf3c1b6040274c13e1803a7437ad86 0      #/0815.0815
100644 84019bea76a3720e8937544a30238e566457c5ce 0      #/1c-enterprise
100644 048379f638d0f531bd4af6e38a6a7e37023f3b96 0      #/2B.2b
100644 83af8e35665fc4e76635d58e455fe429cc521eaa 0      #/4gl.4gl
100644 c41502e5d641f3ba108cc01664e092aa4f5af444 0      #/4test
100644 4f84b79840f40040943a0d7ea83dfb869cd2fc91 0      #/4thdimension.4dd
100644 f9da04c3a34ab993bc3e5405ee37ced6d0cac31c 0      #/~English.ext
100644 239beea576fd63cb0e706678ce5fe0c5e08c8384 0      .gitignore
100644 f880ed6a46d5b73783af9d2224737ff6f8012c26 0      CONTRIBUTING.md
100644 a6aff0ad5ed8d1794d6024553a53166c803ad08b 0      LICENSE
100644 399953baadb4ea8a83dbb0bd6c1217b4828f9723 0      README.md
100644 9d63e6a58d8ebc888b6ff05780ea25f9584ffca9 0      README_nolist.md
100644 555d8acdf382026238622eb266096acfe98503d5 0      a/ACIDIC.acidic
100644 f23dad39656e6f60d05b18c223048101224e7760 0      g/gri.gri
100644 102b07fe0382aa2b220805ba8bfcaba6949d9438 0      g/grin.grin
100644 9510868692414624527f73ad85a555743ef1d5d7 0      g/groovy.groovy
100644 e51a861374385d0e4db38d9ef41c3519c39cf3eb 0      h/HEX.hex
100644 dea928968e219358ef18b1bb339f992ab90a0a46 0      h/HOP
100644 4e6f17d64bffd6cc83a50dc05358b4521580f7ee 0      h/Higher Subleq.higher_subleq

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git status
On branch new-feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        modified:   h/HEX.hex
        deleted:    p/parser.p

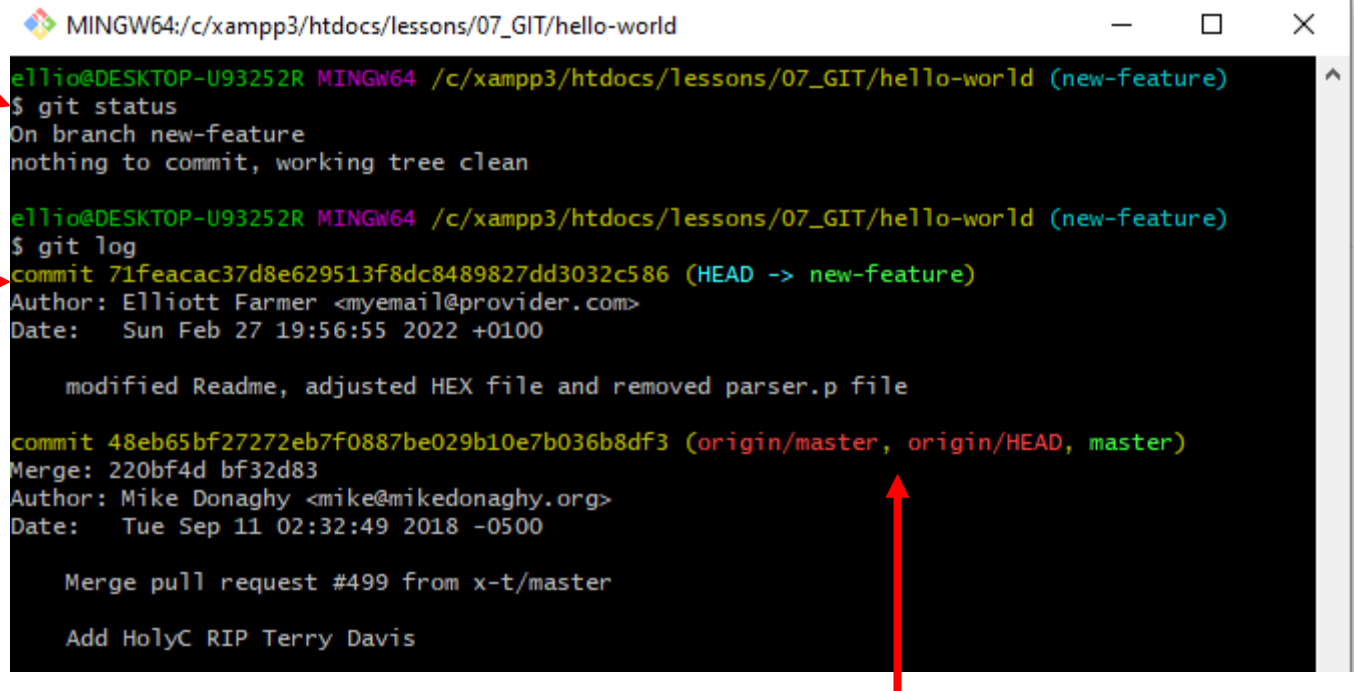
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git commit -m "modified Readme, adjusted HEX file and removed parser.p file"
[new-feature 71feaca] modified Readme, adjusted HEX file and removed parser.p file
3 files changed, 3 insertions(+), 7 deletions(-)
delete mode 100644 p/parser.p

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$
```

git status shows that there are no differences between working area, staged area and repository.

git log command shows last commit with its sha1 hash and commit comment.

Because our local repository is connected to a remote repository in git-hub the last commit is from the remote repository.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git status
On branch new-feature
nothing to commit, working tree clean

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git log
commit 71feacac37d8e629513f8dc8489827dd3032c586 (HEAD -> new-feature)
Author: Elliott Farmer <myemail@provider.com>
Date:   Sun Feb 27 19:56:55 2022 +0100

    modified Readme, adjusted HEX file and removed parser.p file

commit 48eb65bf27272eb7f0887be029b10e7b036b8df3 (origin/master, origin/HEAD, master)
Merge: 220bf4d bf32d83
Author: Mike Donaghy <mike@mikedonaghy.org>
Date:   Tue Sep 11 02:32:49 2018 -0500

    Merge pull request #499 from x-t/master

    Add HolyC RIP Terry Davis
```

The terminal window shows the output of `git status` and `git log`. A red arrow points from the text "no differences between working area, staged area and repository." to the `nothing to commit, working tree clean` line in the `git status` output. Another red arrow points from the text "git log command shows last commit with its sha1 hash and commit comment." to the first commit entry in the `git log` output. A third red arrow points from the text "Note on the penultimate commit it says origin/master. The origin refers to the remote repository. Similarly origin/HEAD refers to remote repository. This last commit is on master branch." to the `origin/master, origin/HEAD, master` line in the second commit entry of the `git log` output.

Note on the penultimate commit it says **origin/master**. The origin refers to the remote repository. Similarly **origin/HEAD** refers to remote repository. This last commit is on master branch.

File Edit View Repository Branch Help

Current repository: hello-world

Current branch: new-feature

Publish branch: Publish this branch to the remote

Changes History

Select branch to compare...

modified Readme, adjusted HE... Elliott Farmer • 29m

Merge pull request #499 from x-/mast... Mike Donaghy, GitHub • Sep 11, 2018

Create holyc.HC cmp, GitHub • Sep 7, 2018

Merge pull request #478 from chinsyo/... Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #477 from cheezeb... Mike Donaghy, GitHub • Jul 23, 2018

Merge pull request #476 from Pocable/... Mike Donaghy, GitHub • Jul 23, 2018

add nasm macho64 Chinsyo • Jul 23, 2018

Added aspectj.aj cheezeburger • Jul 22, 2018

Added MIPS Thomas Roskewich • Jul 18, 2018

Merge pull request #418 from thathexa... Mike Donaghy, GitHub • Jul 17, 2018

modified Readme, adjusted HEX file and removed parser.p file Elliott Farmer 71feaca 3 changed files +3 -7

README.md

h\HEX.hex

p\parser.p

@@ -3,12 +3,12 @@ Hello, World!

> Inspired by [The Hello World Collection](https://helloworldcollection.github.io/).

-#### Hello world in every programming language.

+#### Hello world in all programming languages.

As I watch the collection expand, this project has blown up more than I ever thought possible.

Thanks to everyone who continues to contribute, new languages are created every day!

-Make sure to see CONTRIBUTING.md for instructions on contributing to the project!

+Make sure to see CONTRIBUTING.md file for instructions on contributing to the project!

Spin-Off project smartly suggested and implemented by [zenware](https://github.com/zenware):

Meet [FizzBuzz](https://github.com/zenware/FizzBuzz), the evolution of [hello-world](https://github.com/leachim6/hello-world).

These changes were made on the new-feature branch.

In git hub desktop we can see the last change to the local repository was performed by me. The details of the last change were to modifications and one delete

File Edit Selection View Go Run Terminal Help list_langs.py - hello-world - Visual Studio Code

EXPLORER

HELLO-WORLD

- > h
- > i
- > j
- > k
- > l
- > m
- > n
- > o
- > p
- > q
- > r
- > s
- > t
- > u
- > v
- > w
- > x
- > y
- > z
- ◆ .gitignore
- ↓ CONTRIBUTING.md
- 🔒 LICENSE
- 🔌 list_langs.py M
- ↓ README_nolist.md
- 🔍 README.md

list_langs.py

```
1  #!/usr/bin/env python3
2
3  import os
4  from urllib.parse import quote
5
6
7
8  readme = open('README.md', 'w')
9
10 # Copy template to README
11 with open('README_nolist.md', 'r') as temp:
12     for line in temp:
13         readme.write(line)
14
15 # Write title
16 readme.write('\n### This repository contains "Hello World" programs in t
17
18 # List the available languages
19 for dirname in sorted(os.listdir('.')):
20     if not (dirname == '.' or dirname == '..' or dirname[0] == '.' or os.p
21         for filename in sorted(os.listdir(dirname), key=lambda s: s.lower()):
22             if os.path.isfile(os.path.join(dirname, filename)):
23                 lang = os.path.splitext(filename)[0].replace('-', ' ').replace('
24                 readme.write('* [{}]({})\n'.format(lang, quote(os.path.join(dirname
25
26 readme.close()
27
28 # End of the file
```

2 line space was added at lines 6 and 7

The word currently was removed at line 16.

Next we are modifying the list_lang.py file.

A comment was added at line 28

Control s to save the file

new-feature* 0 0 Ln 28, Col 18 Spaces: 2 UTF-8 CRLF Python

git status command will add all changed files to the staging area.

git diff command shows any differences between working area, staged area and repository.

Note how git has creates a sha1 hash d4aa133 for a/list-langs.py and sha1 ee48aa1 hash for a/list-langs.py.

What do these mean?
See slide 124 and 125.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git status
On branch new-feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list_langs.py

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git diff
diff --git a/list_langs.py b/list_langs.py
index d4aa133..ee48aa1 100755
--- a/list_langs.py
+++ b/list_langs.py
@@ -3,6 +3,8 @@
 import os
 from urllib.parse import quote

+
+
readme = open('README.md', 'w')


# Copy template to README
@@ -11,7 +13,7 @@ with open('README_nolist.md', 'r') as temp:
     readme.write(line)

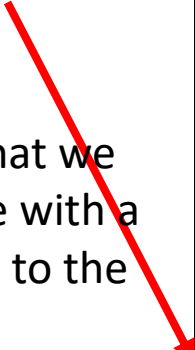
# Write title
-readme.write('\n### This repository currently contains "Hello World" programs in the followi
ng languages:\n')
+readme.write('\n### This repository contains "Hello World" programs in the following languag
es:\n')

# List the available languages
for dirname in sorted(os.listdir('.')):
@@ -22,3 +24,5 @@ for dirname in sorted(os.listdir('.')):
     readme.write('* [{}]({})\n'.format(lang, quote(os.path.join(dirname, filename))))

readme.close()
+
+# End of the file
\ No newline at end of file

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ ^C
```

If we look at the contents of the first file in the git diff command output using the hash we can see that it is the original unmodified file. 

If we try to look at the contents of the modified file using the hash from the git diff command we get a fatal error: Not a valid object name. 

The reason for this is that we have not staged the file with a git add or committed it to the repository.

What git is showing us is a provisional sha1 hash pending adding and committing.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git cat-file -p d4aa133
#!/usr/bin/env python3
import os
from urllib.parse import quote

readme = open('README.md', 'w')

# Copy template to README
with open('README_nolist.md', 'r') as temp:
    for line in temp:
        readme.write(line)

# Write title
readme.write('\n### This repository currently contains "Hello World" programs in the following languages:\n')

# List the available languages
for dirname in sorted(os.listdir('.')):
    if not (dirname == '.' or dirname == '..' or dirname[0] == '.' or os.path.isfile(dirname)):
        for filename in sorted(os.listdir(dirname), key=lambda s: s.lower()):
            if os.path.isfile(os.path.join(dirname, filename)):
                lang = os.path.splitext(filename)[0].replace('-', ' ').replace('_', ' ').title()
                readme.write('* [{}]({})\n'.format(lang, quote(os.path.join(dirname, filename))))

readme.close()

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git cat-file -p ee48aa1
fatal: Not a valid object name ee48aa1

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$
```

FileEditViewRepositoryBranchHelp

Current repository
hello-world

Current branch
new-feature

↑

Publish branch

Publish this branch to the remote

Changes 1

History

list_langs.py

@@ -3,6 +3,8 @@

3 3 import os

4 4 from urllib.parse import quote

5 5

6 +

7 +

6 8 readme = open('README.md', 'w')

7 9

8 10 # Copy template to README

@@ -11,7 +13,7 @@ with open('README_nolist.md', 'r') as temp:

11 13 readme.write(line)

12 14

13 15 # Write title

14 -readme.write('\n### This repository currently contains "Hello World" programs in the following language s:\n')

16 +readme.write('\n### This repository contains "Hello World" programs in the following languages:\n')

15 17

16 18 # List the available languages

17 19 for dirname in sorted(os.listdir('.')):

@@ -22,3 +24,5 @@ for dirname in sorted(os.listdir('.')):

22 24 readme.write('* {}\n'.format(lang, quote(os.path.join(dirname, filename))))

23 25

24 26 readme.close()

27 +

28 +# End of the file

1 changed file

list_langs.py

Update list_langs.py

Description

Commit to new-feature

Committed 1 hour ago
modified README, adjusted HEX file ...

Undo

Git is clever and it does not show us the whole file, only what has been changed. Here two blank lines were added.

Here the word currently was removed.

Here an additional comment was added.

FileEditViewRepositoryBranchHelp

Current repository
hello-world

Current branch
new-feature

↑

Publish branch

Publish this branch to the remote

Changes 1

History

list_langs.py

@@ -3,6 +3,8 @@

1 changed file

list_langs.py

3 3 import os

4 4 from urllib.parse import quote

5 5

6 6 +

7 7 +

6 8 readme = open('README.md', 'w')

7 9

8 10 # Copy template to README

@@ -11,7 +13,7 @@ with open('README_nolist.md', 'r') as temp:

11 13 readme.write(line)

12 14

13 15 # Write title

14 -readme.write('\n### This repository currently contains "Hello World" programs in the following language s:\n')

16 +readme.write('\n### This repository contains "Hello World" programs in the following languages:\n')

15 17

16 18 # List the available languages

17 19 for dirname in sorted(os.listdir('.')):

@@ -22,3 +24,5 @@ for dirname in sorted(os.listdir('.')):

22 24 readme.write('* {}\n'.format(lang, quote(os.path.join(dirname, filename))))

23 25

24 26 readme.close()

27 +

28 +# End of the file

Update list_langs.py

Description

Commit to new-feature

Committed 1 hour ago
modified README, adjusted HEX file ...

Undo

+3 represents start line of the addition from old file, 8 represents up to what line has been added in the new file.

-3 represents line on old file, 6 represents line in new file.

So we can now commit these changes. However it gives us an error because we have not staged the changes.

However the git commit command has the option to stage and commit with one line of code. We can just add an **-a** to the command which signifies add. **git commit -m <comment>** **-a**

Git diff is now empty because the changes have been committed.

We can also now view the file because it has changed from provisional to committed to the repository.

Git does not create new objects until you commit your changes.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git commit -m "Changed list-langs.py"
On branch new-feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list_langs.py

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git commit -m "Changed list-langs.py" -a
[new-feature 86ff7ad] Changed list-langs.py
1 file changed, 5 insertions(+), 1 deletion(-)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git diff
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git cat-file -p ee48aa1
#!/usr/bin/env python3

import os
from urllib.parse import quote

readme = open('README.md', 'w')

# Copy template to README
with open('README_nolist.md', 'r') as temp:
    for line in temp:
        readme.write(line)

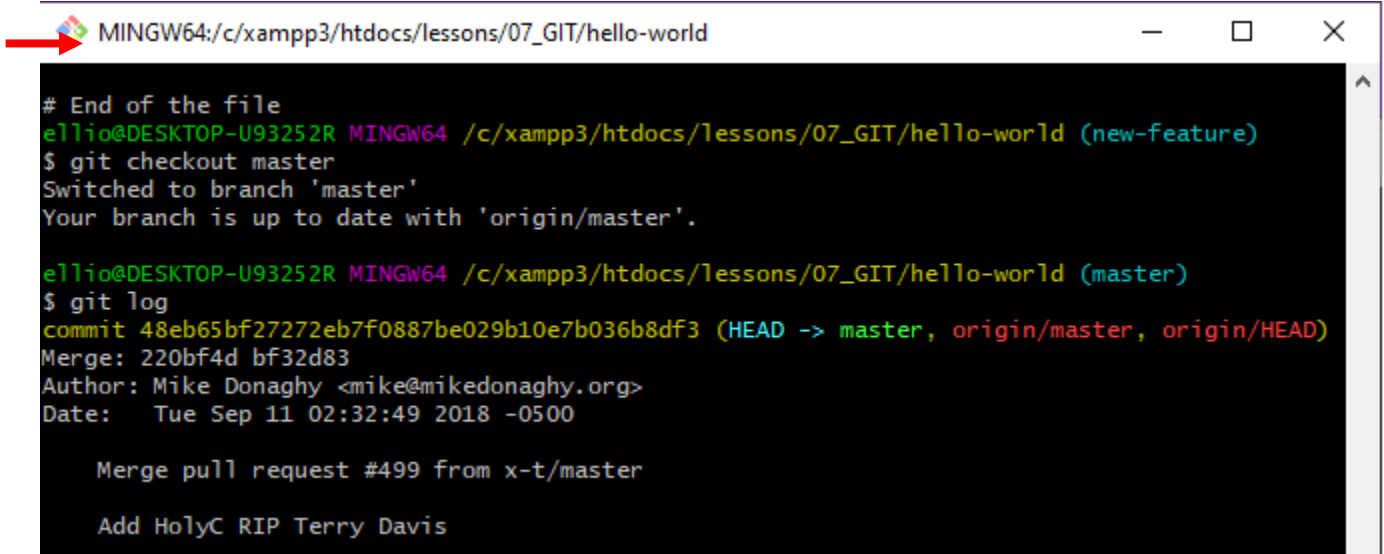
# Write title
readme.write('\n### This repository contains "Hello World" programs in the following languages:\n')

# List the available languages
for dirname in sorted(os.listdir('.')):
    if not (dirname == '.' or dirname == '..' or dirname[0] == '.' or os.path.isfile(dirname)):
        for filename in sorted(os.listdir(dirname), key=lambda s: s.lower()):
            if os.path.isfile(os.path.join(dirname, filename)):
                lang = os.path.splitext(filename)[0].replace('-', ' ').replace('_', ' ').title()
                readme.write('* {}({})\n'.format(lang, quote(os.path.join(dirname, filename))))

readme.close()

# End of the file
```

If we checkout the master branch and check the log the two commits made under the “*new feature*” branch are not shown.

A terminal window titled 'MINGW64:/c/xampp3/htdocs/lessons/07_GIT/hello-world' with standard window controls. A red arrow points to the terminal icon in the title bar. The terminal shows the following commands and output:

```
# End of the file
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (new-feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git log
commit 48eb65bf27272eb7f0887be029b10e7b036b8df3 (HEAD -> master, origin/master, origin/HEAD)
Merge: 220bf4d bf32d83
Author: Mike Donaghy <mike@mikedonaghy.org>
Date: Tue Sep 11 02:32:49 2018 -0500

    Merge pull request #499 from x-t/master

    Add HolyC RIP Terry Davis
```

Summary of Shell Commands

git clone

Will clone a git repository.

cat <filename. Pack> | git unpack-objects

Unpack pack file

git diff

Check differences between repository and working area

git commit -m <comment> -a

Combined add and commit command.

MERGING BRANCHES

Merge branches

Here we have three branches, Master Branch. This is the main Branch, also known as master branch or release branch.

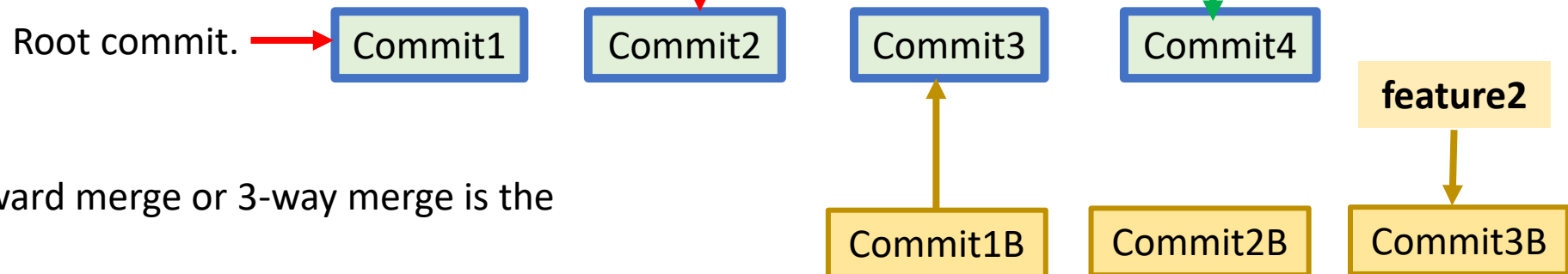
Different Branches spur off of main Branches to develop new features to the project.

For example with Branch feature 1 checked out we will see commits 1, 2, 1A, 2A and 3A.

Because feature1 Branch is checked out the HEAD only sees commits that relate to that branch.

But what if we want to incorporate the changes made in branch 1 into the main project or master Branch?

Root commit. →

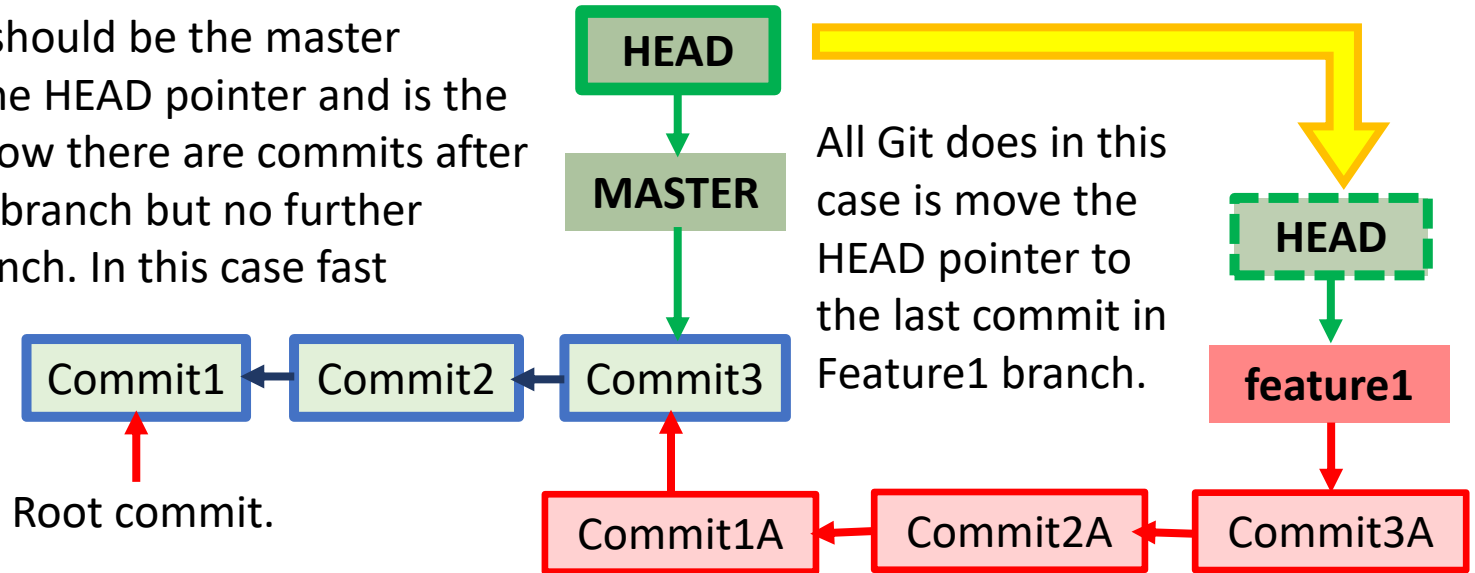


Fast forward merge or 3-way merge is the solution.

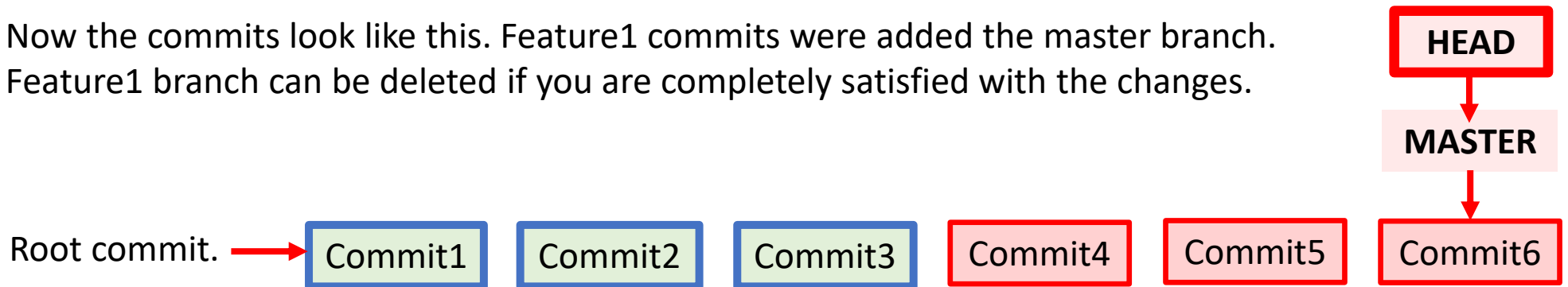
Fast forward merge

Here the master Branch is the main Branch with three commits. At this moment in time there is another feature1 branch with three commits. Lets say at this moment in time we are happy with the changes in feature1 branch and we want to merge these changes into master branch.

First the checkout branch should be the master branch so it should have the HEAD pointer and is the **receiving branch**. Notice how there are commits after master branch in feature1 branch but no further commits in the master branch. In this case fast forward merge applies.

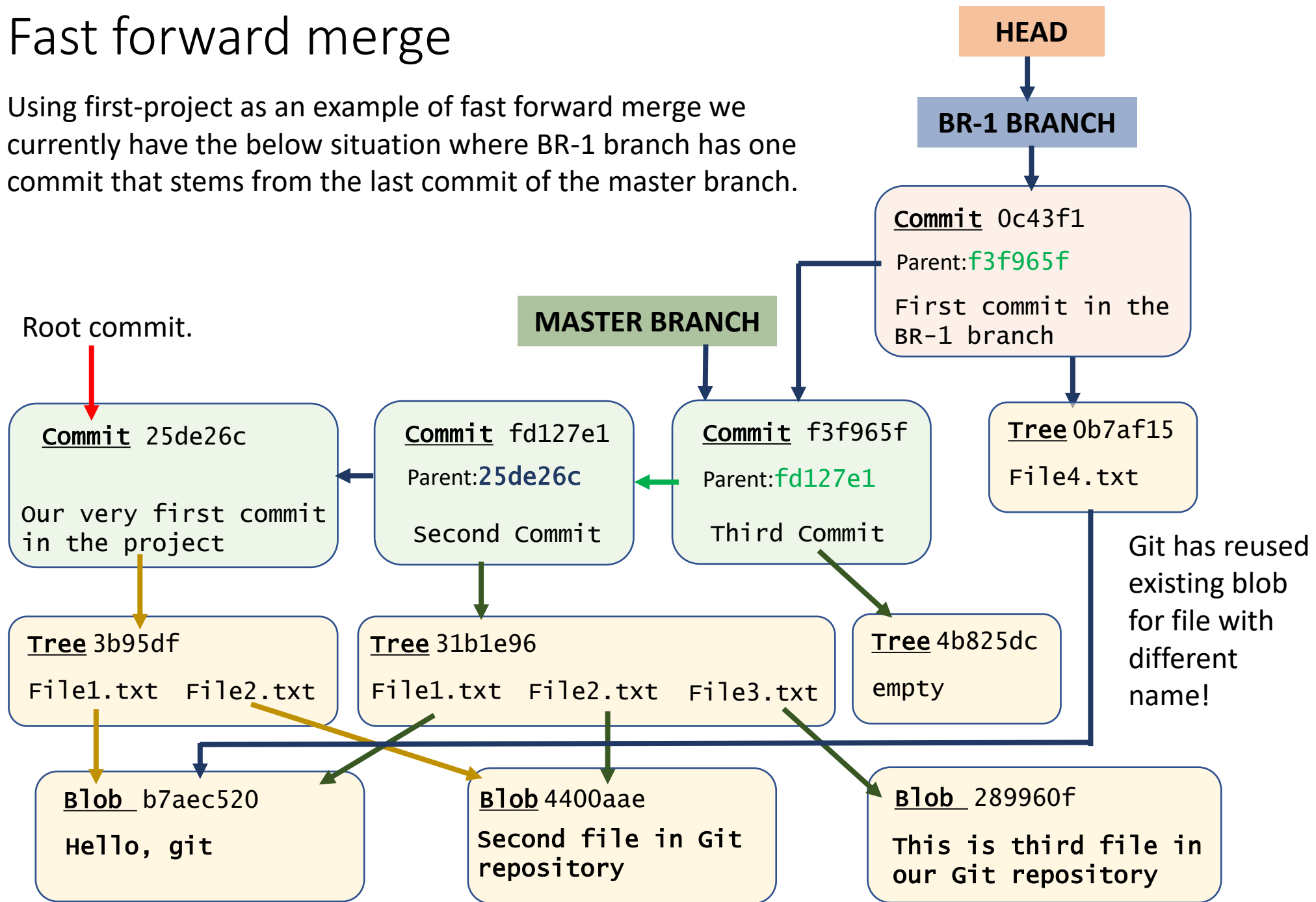


Now the commits look like this. Feature1 commits were added the master branch. Feature1 branch can be deleted if you are completely satisfied with the changes.



Fast forward merge

Using first-project as an example of fast forward merge we currently have the below situation where BR-1 branch has one commit that stems from the last commit of the master branch.



If we checkout the master branch a **ls -l** command shows no files in the master branch.

If we checkout the BR-1 branch we can see one file in it-

Git log shows our three commits to the project. Note that in the third commit previously we removed all files from the master branch.

MINGW64: c:/xampp3/htdocs/lessons/07_GIT/first-project

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git checkout master
Switched to branch 'master'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
  BR-1
* master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout BR-1
Switched to branch 'BR-1'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ ls -l
total 1
-rw-r--r-- 1 ellio 197609 12 Feb 28 17:46 file4.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git log
commit 8fa0cdb8b1f83909b42278c993d037c76951db18 (HEAD -> BR-1)
Author: Elliott Farmer <myemail@provider.com>
Date:   Fri Feb 25 23:18:08 2022 +0100

    First commit in the BR-1 branch

commit f3f965f4dddb3fc863f9b3d81567eabd6561f3 (master)
Author: Elliott Farmer <myemail@provider.com>
Date:   Fri Feb 25 02:48:51 2022 +0100

    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date:   Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date:   Thu Feb 24 21:04:46 2022 +0100

    Our very first commit in the project
```

First we checkout master branch.

git merge <branch name>

command is used to fast forward merge BR-1 branch into master branch.

Notice the command output

“updating f3f965f..8fa0cdb”

What does this mean?

ls -l command shows that file4 from branch BR-1 is now in the master branch.

Git log shows the commit hash of 8fa0cdb “first commit in the BR-1 branch” as appearing as the HEAD and master and BR-1.

The third commit with hash f3f965f was previously the HEAD of the master branch.

“updating f3f965f..8fa0cdb” The first hash is the previous head before the merge. The second the new head after the merge.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-1)
$ git checkout master
Switched to branch 'master'

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git merge BR-1
Updating f3f965f..8fa0cdb
Fast-forward
 file4.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file4.txt

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls -l
total 1
-rw-r--r-- 1 elliott 197609 12 Feb 28 17:59 file4.txt

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit 8fa0cdb8b1f83909b42278c993d037c76951db18 (HEAD -> master, BR-1)
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 23:18:08 2022 +0100

    First commit in the BR-1 branch

commit f3f965f4dddb3fc863f9b3d81567eabd6561f3
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 02:48:51 2022 +0100

    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 23:00:56 2022 +0100

    Second commit

commit 25de26c3fff87e02b4bc3aabe34e835cef2eb21a
Author: Elliott Farmer <myemail@provider.com>
Date: Thu Feb 24 21:04:46 2022 +0100

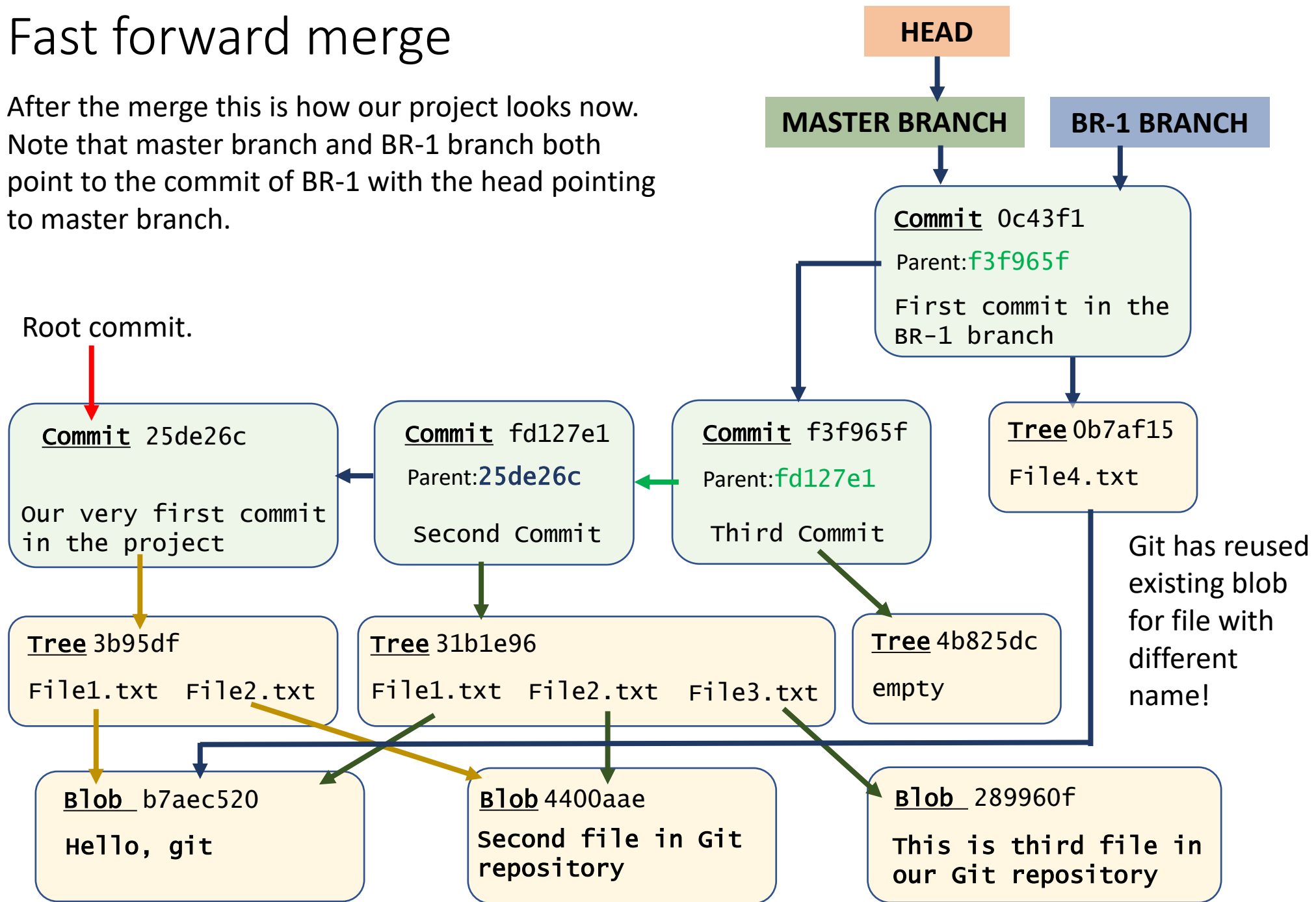
    Our very first commit in the project

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cat .git/refs/heads/master
8fa0cdb8b1f83909b42278c993d037c76951db18

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

Fast forward merge

After the merge this is how our project looks now.
Note that master branch and BR-1 branch both point to the commit of BR-1 with the head pointing to master branch.

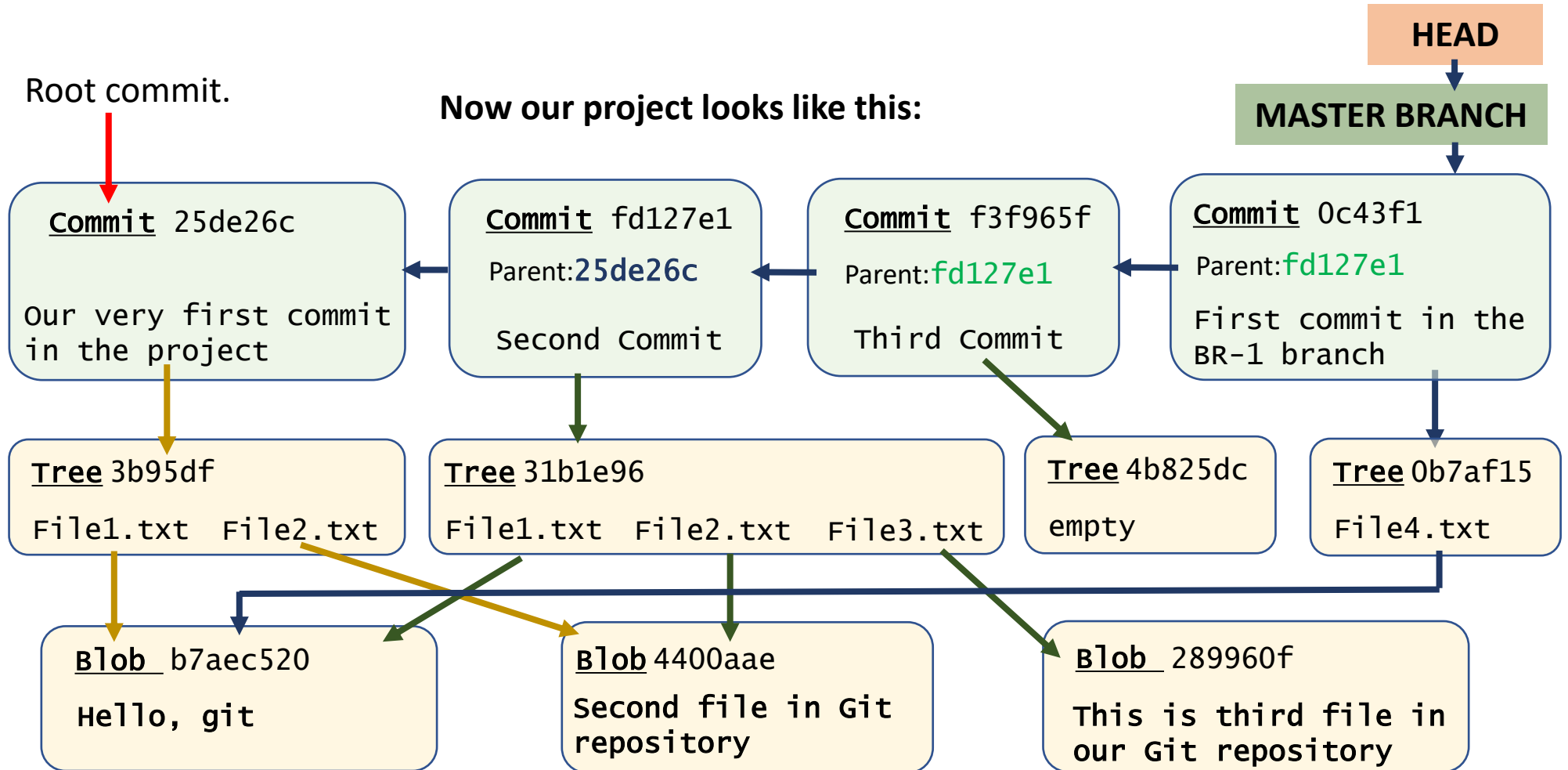


git branch -d <branch name>
command can be used to delete
BR-1 branch so only the master
branch remains.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch -d BR-1
Deleted branch BR-1 (was 8fa0cdb).

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

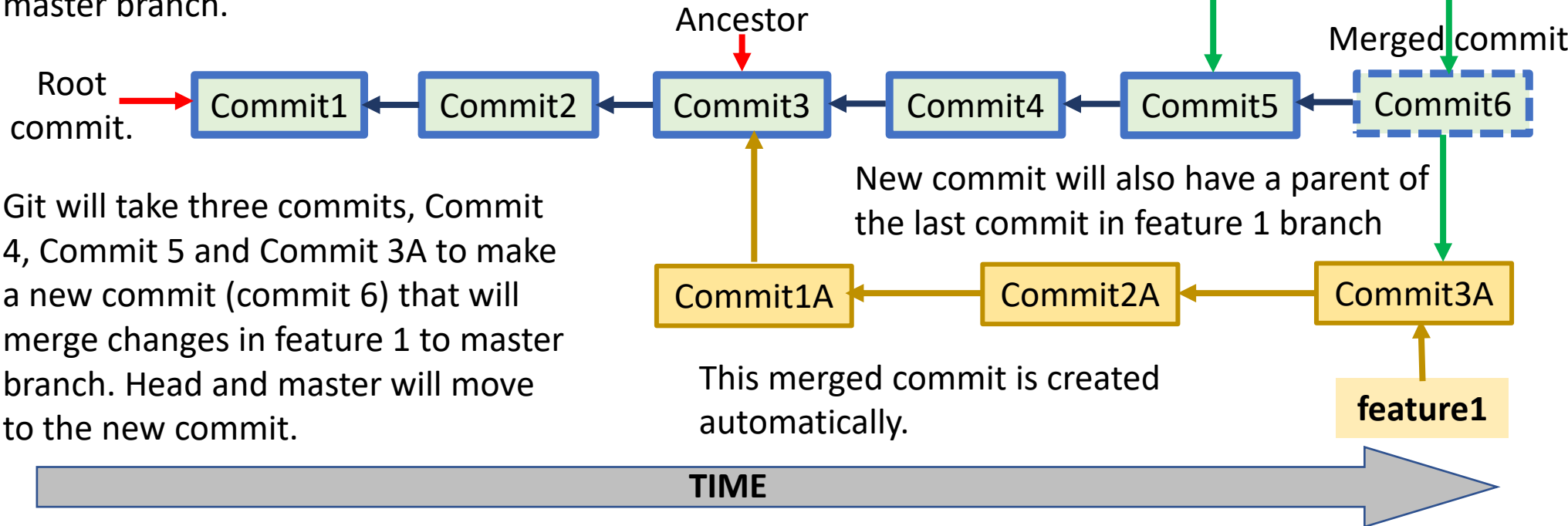


Three-Way merge

Here the master Branch is the main Branch with five commits. At this moment in time there is another feature1 branch with three commits starting from commit 3 of the master branch. Lets say at this moment in time we are happy with the changes in feature1 branch and we want to merge these changes into master branch. We cannot use a fast forward merge because there are commits in the master branch after the divergence of feature1 branch.

Commit 3 is common for master branch and feature 1 branch and is called ancestor. Of course commit 2 and commit 1 are also ancestors but git will try to find nearest commit to both branches. This is the ancestor.

The branch we want to merge to must be checked out, in this case the master branch.

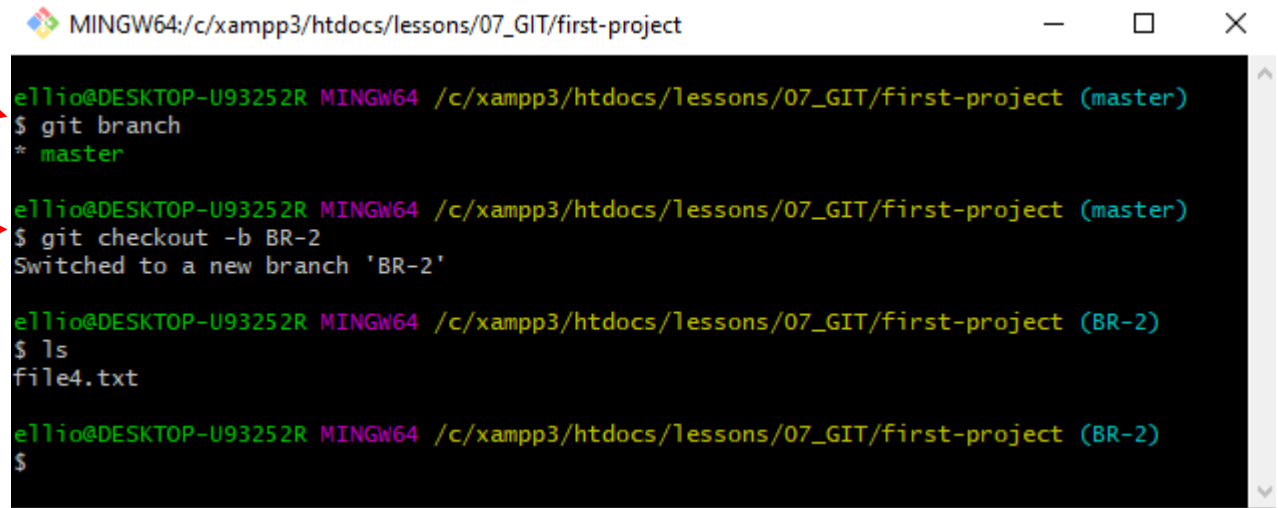


Git will take three commits, Commit 4, Commit 5 and Commit 3A to make a new commit (commit 6) that will merge changes in feature 1 to master branch. Head and master will move to the new commit.

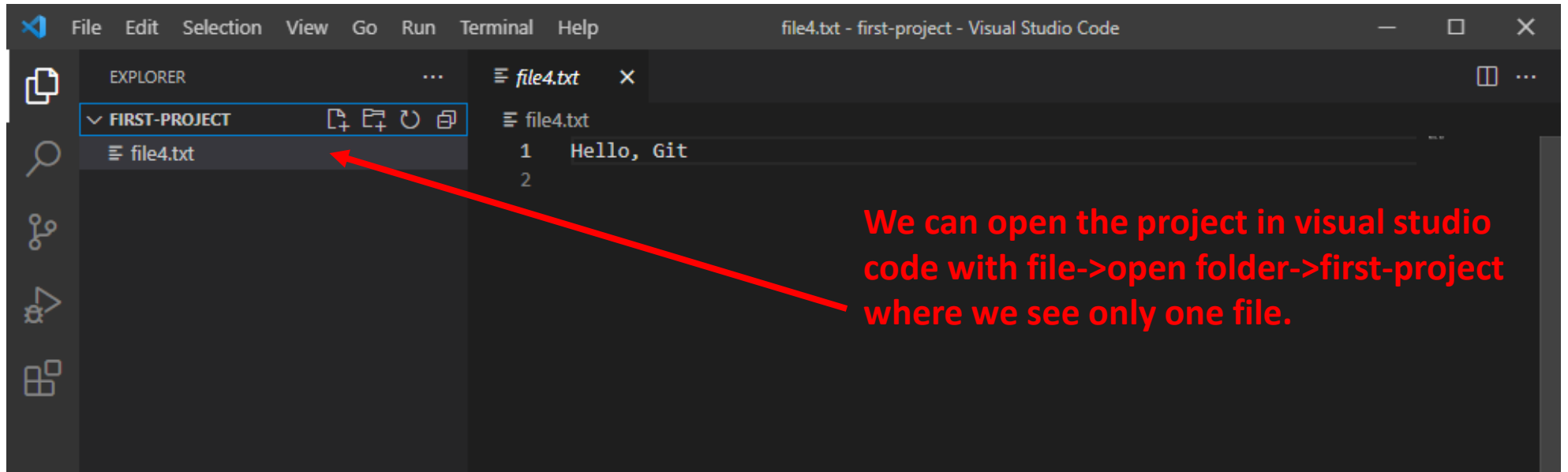
At the moment there is only one branch, the master branch.

We will create a new branch, BR-2 with the command git checkout -b <branch name>

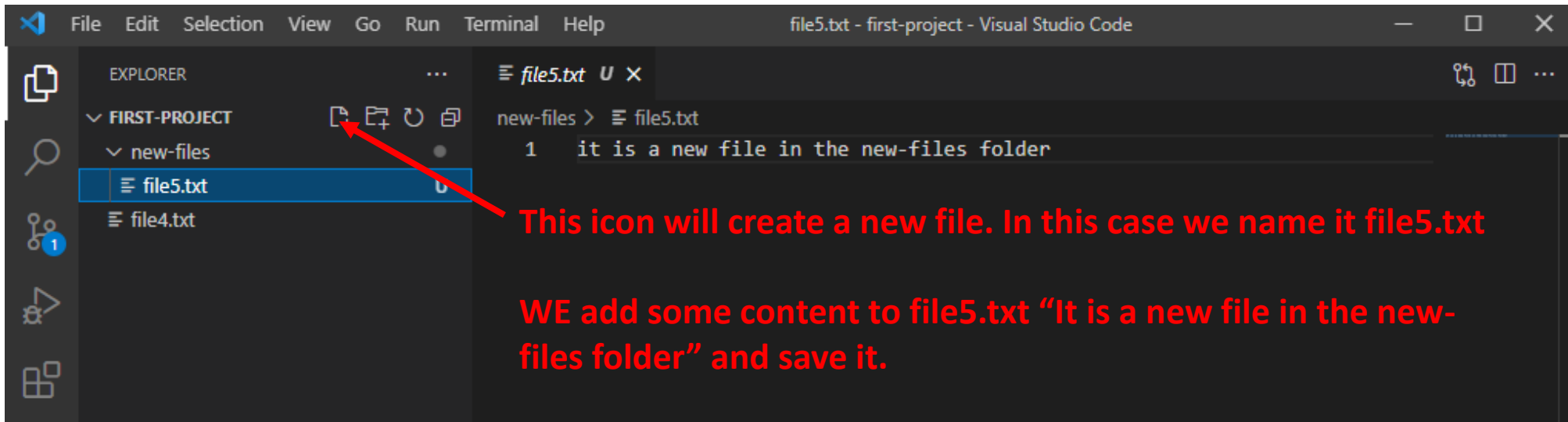
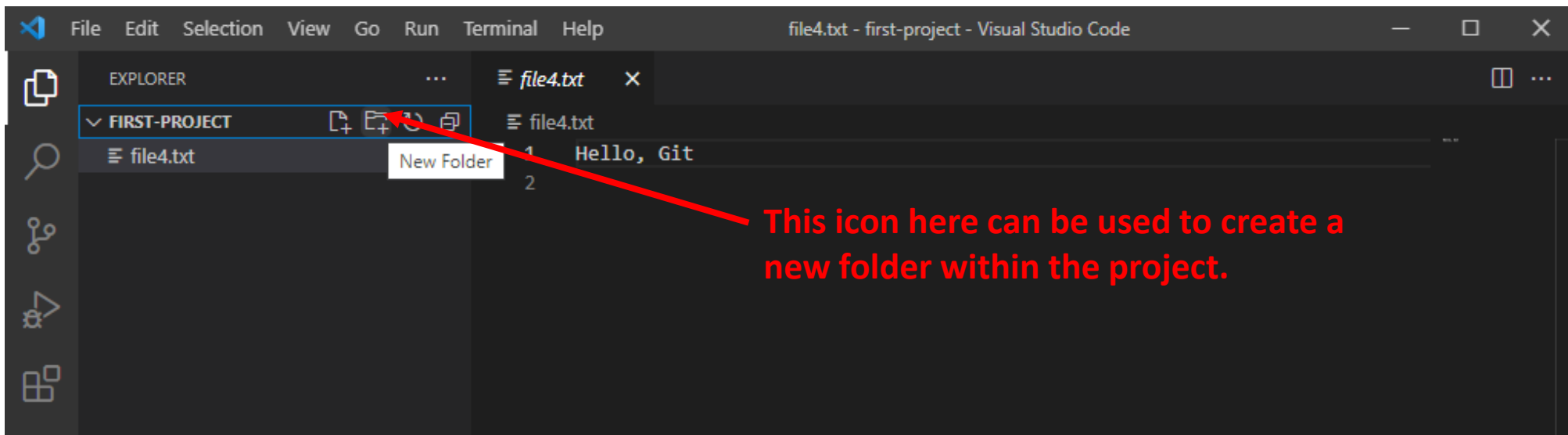
In branch BR-2 there is only one file.

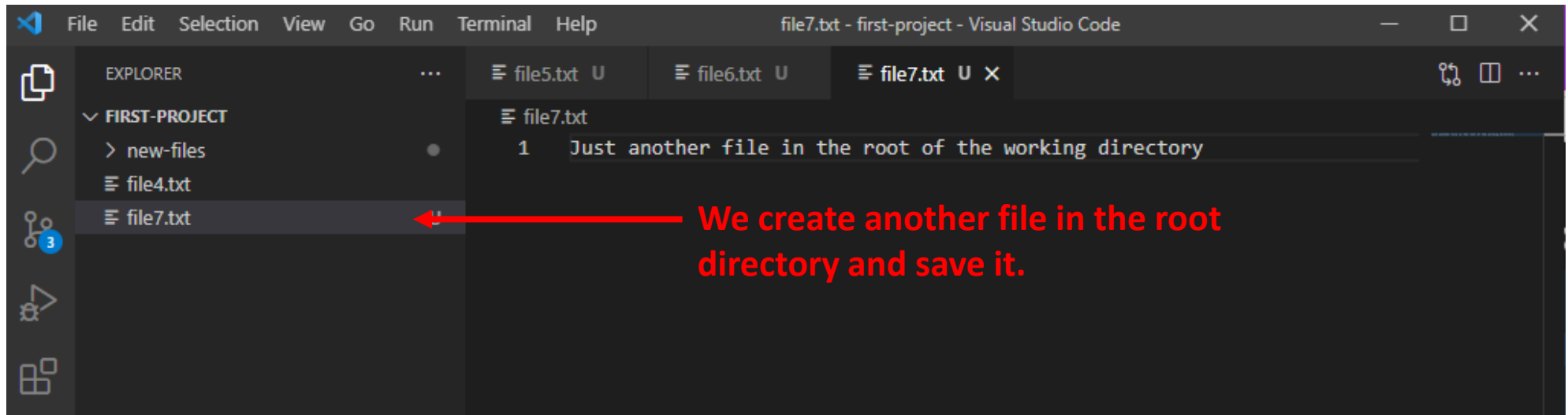
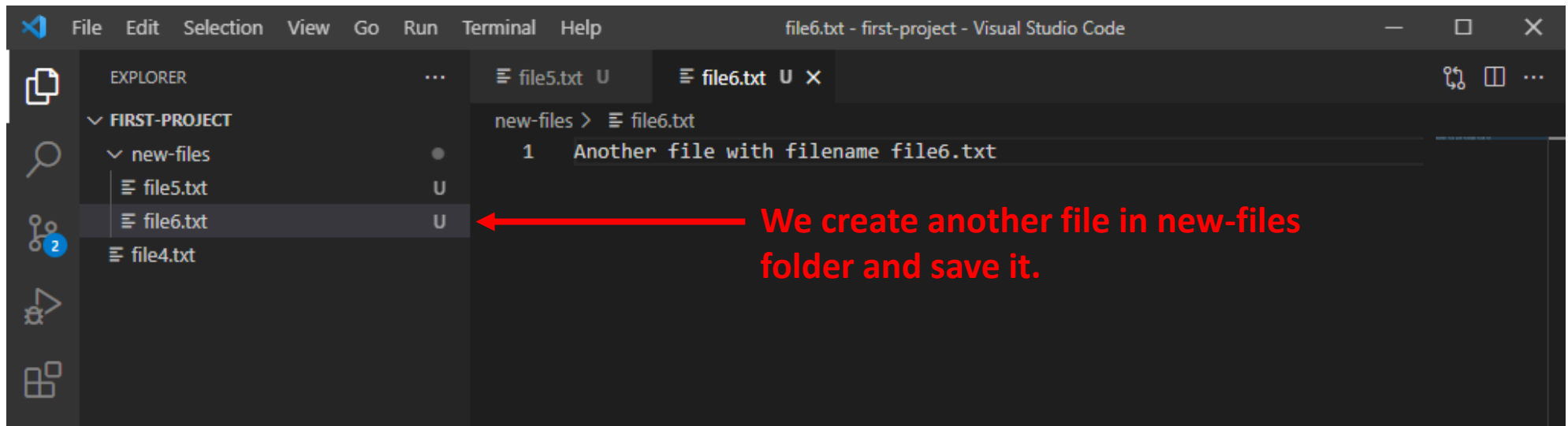


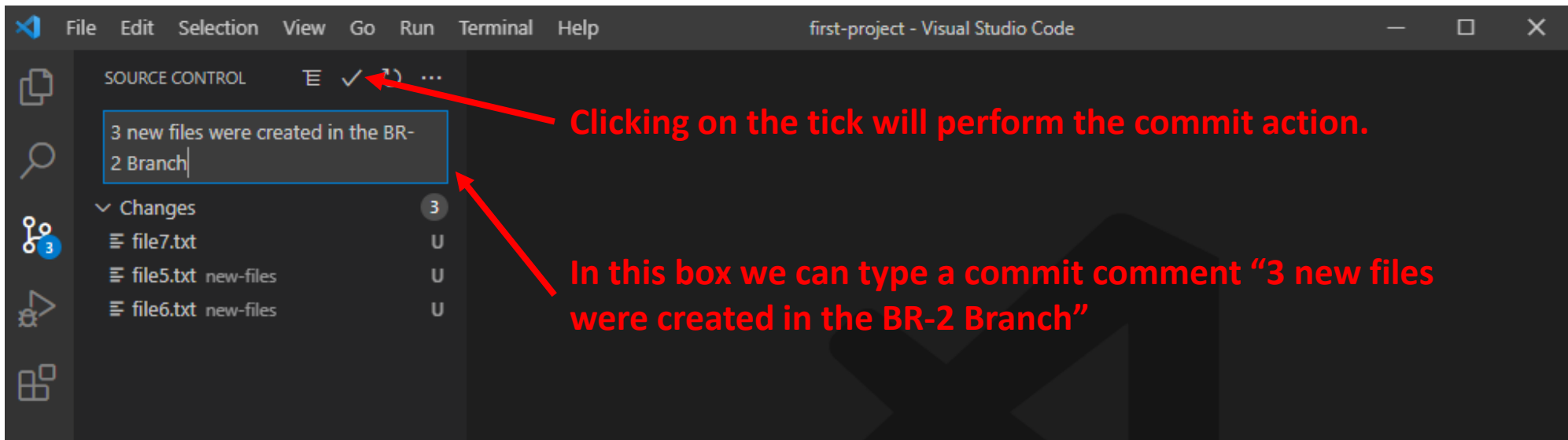
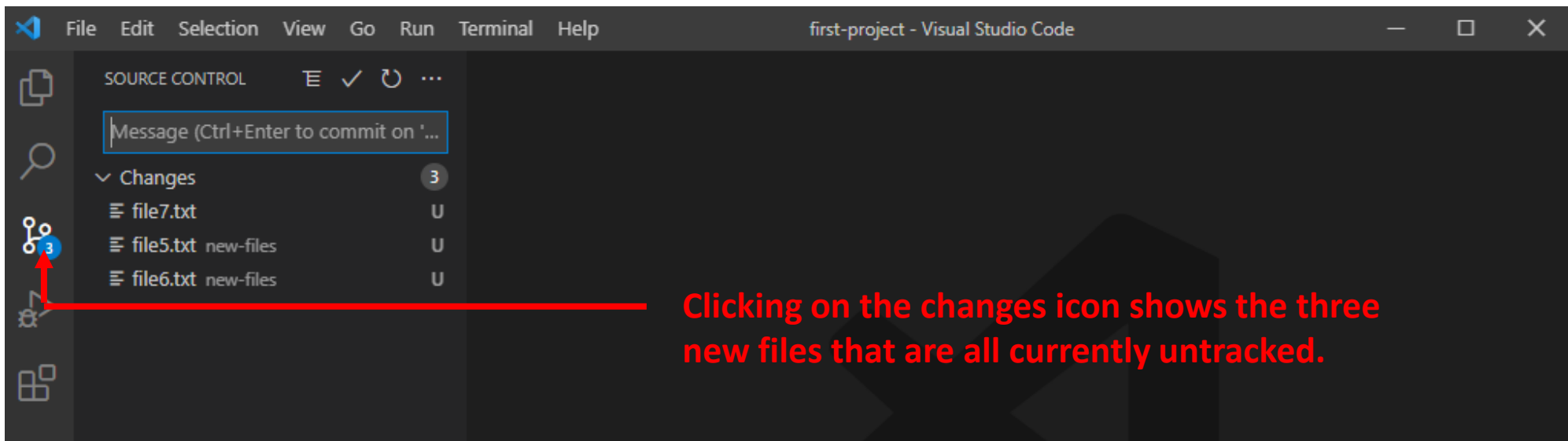
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project  
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)  
$ git branch  
* master  
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)  
$ git checkout -b BR-2  
Switched to a new branch 'BR-2'  
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)  
$ ls  
file4.txt  
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)  
$
```

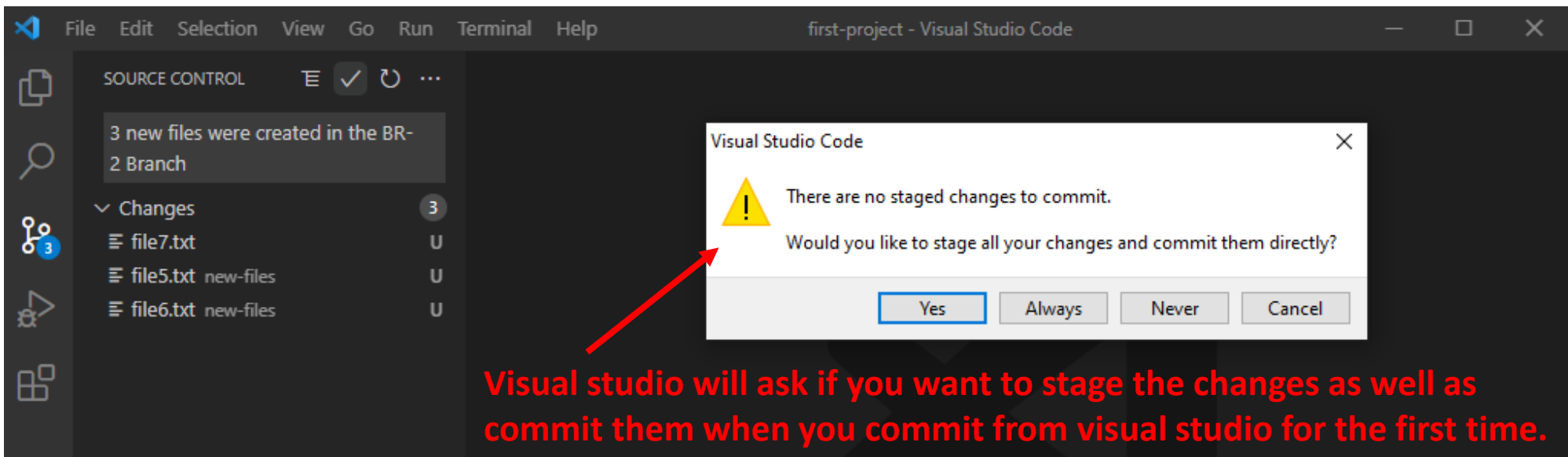


We can open the project in visual studio code with file->open folder->first-project where we see only one file.









Visual studio will ask if you want to stage the changes as well as commit them when you commit from visual studio for the first time.

git status shows no pending commits or unsynchronised files between working directory or staged area or repository because we have just committed them via visual studio code.

git log shows the new commit with the head pointing to BR-2 branch.

The penultimate commit shows in the master branch.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

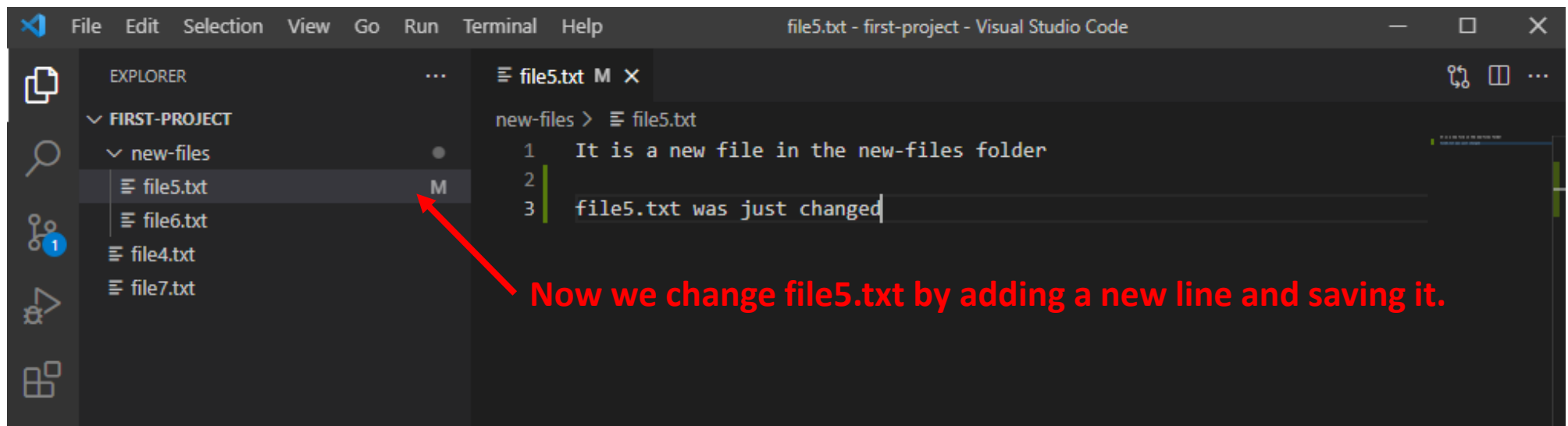
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$ git status
On branch BR-2
nothing to commit, working tree clean

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$ git log
commit f6695ebe499ee1e122343821b60a29ee668ba84b (HEAD -> BR-2)
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 21:39:50 2022 +0100

    3 new files were created in the BR-2 Branch

commit 8fa0cdb8b1f83909b42278c993d037c76951db18 (master)
Author: Elliott Farmer <myemail@provider.com>
Date: Fri Feb 25 23:18:08 2022 +0100

    First commit in the BR-1 branch
```



Git status show file5.tx is pending to be staged and committed.

We can use 1 line command to stage and commit this file.

git commit -a -m <comment>

The output shows one file changed, 3 insertions and one deletion. File5.txt has three lines. The one deletion refers to line one which was overwritten.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$ git status
On branch BR-2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   new-files/file5.txt

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$ git commit -a -m "changed file5.txt"
[BR-2 d46ea50] changed file5.txt
1 file changed, 3 insertions(+), 1 deletion(-)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$
```

The screenshot shows the GitHub Desktop application interface. The top bar includes menus for File, Edit, View, Repository, Branch, and Help. Below this, the 'Current repository' is 'first-project', the 'Current branch' is 'BR-2', and there is a 'Publish repository' button. The main area is divided into 'Changes' and 'History' tabs. The 'History' tab is active, showing a list of commits. The most recent commit, 'changed file5.txt' by Elliott Farmer 4m ago, is selected. To the right of the commit list, a diff view for 'new-files\file5.txt' is shown. The diff indicates a deletion of one line (marked with '-') and the addition of three lines (marked with '+'). The lines are: '-It is a new file in the new-files folder', '+It is a new file in the new-files folder', and '+file5.txt was just changed'. Red arrows originate from the explanatory text at the bottom and point to the selected commit in the history list, the diff view, and the specific lines of code that were deleted and added.

File Edit View Repository Branch Help

Current repository
first-project

Current branch
BR-2

Publish repository
Publish this repository to GitHub

Changes History

Select branch to compare...

changed file5.txt
Elliott Farmer • 4m

3 new files were created in the BR-2 Br...
Elliott Farmer • 15m

First commit in the BR-1 branch
Elliott Farmer • 2d

Third commit
Elliott Farmer • 3d

Second commit
Elliott Farmer • 3d

Our very first commit in the project
Elliott Farmer • 4d

changed file5.txt
new-files\file5.txt

@@ -1 +1,3 @@

1 -It is a new file in the new-files folder

1 +It is a new file in the new-files folder

2 +

3 +file5.txt was just changed

In git hub desktop in the history the ultimate commit shows details about file5.txt.

Note that the first line was deleted (-) and three lines were added (+).

The reason 3 lines were added and one line was deleted is that the first line is different because now it has a return after it whereas before it was just a single line.

Now if we checkout the master branch and check the log we see that the commits withing BR-2 are not in this branch.

The master branch contains the last commit made from within master branch.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-2)
$ git checkout master
Switched to branch 'master'

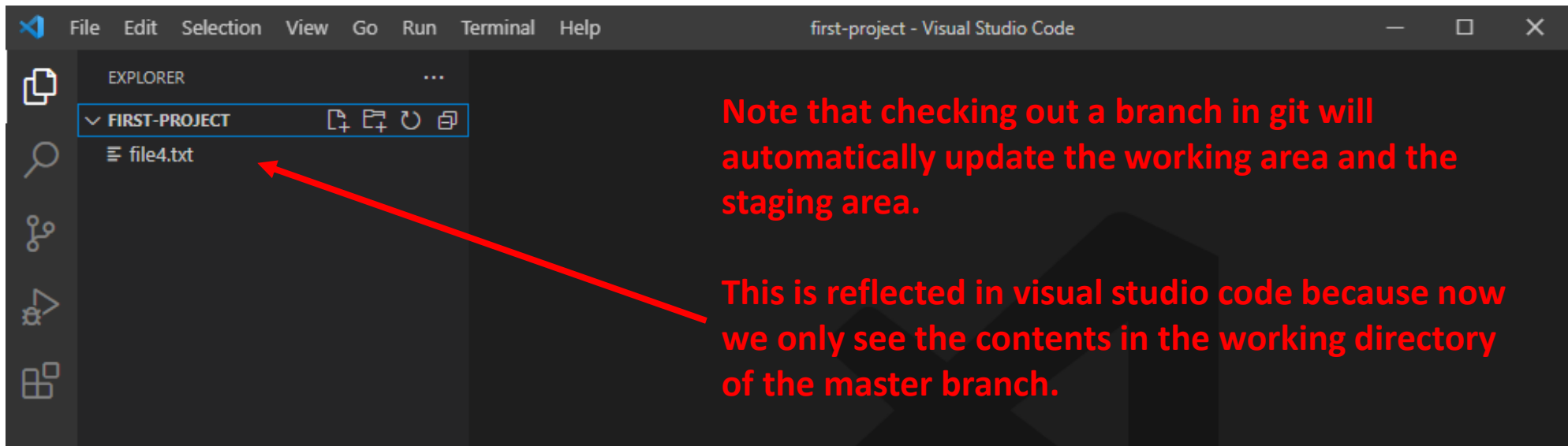
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit 8fa0cdb8b1f83909b42278c993d037c76951db18 (HEAD -> master)
Author: Elliott Farmer <myemail@provider.com>
Date:   Fri Feb 25 23:18:08 2022 +0100

    First commit in the BR-1 branch

commit f3f965f4dddb3fc863fbe9b3d81567eabd6561f3
Author: Elliott Farmer <myemail@provider.com>
Date:   Fri Feb 25 02:48:51 2022 +0100

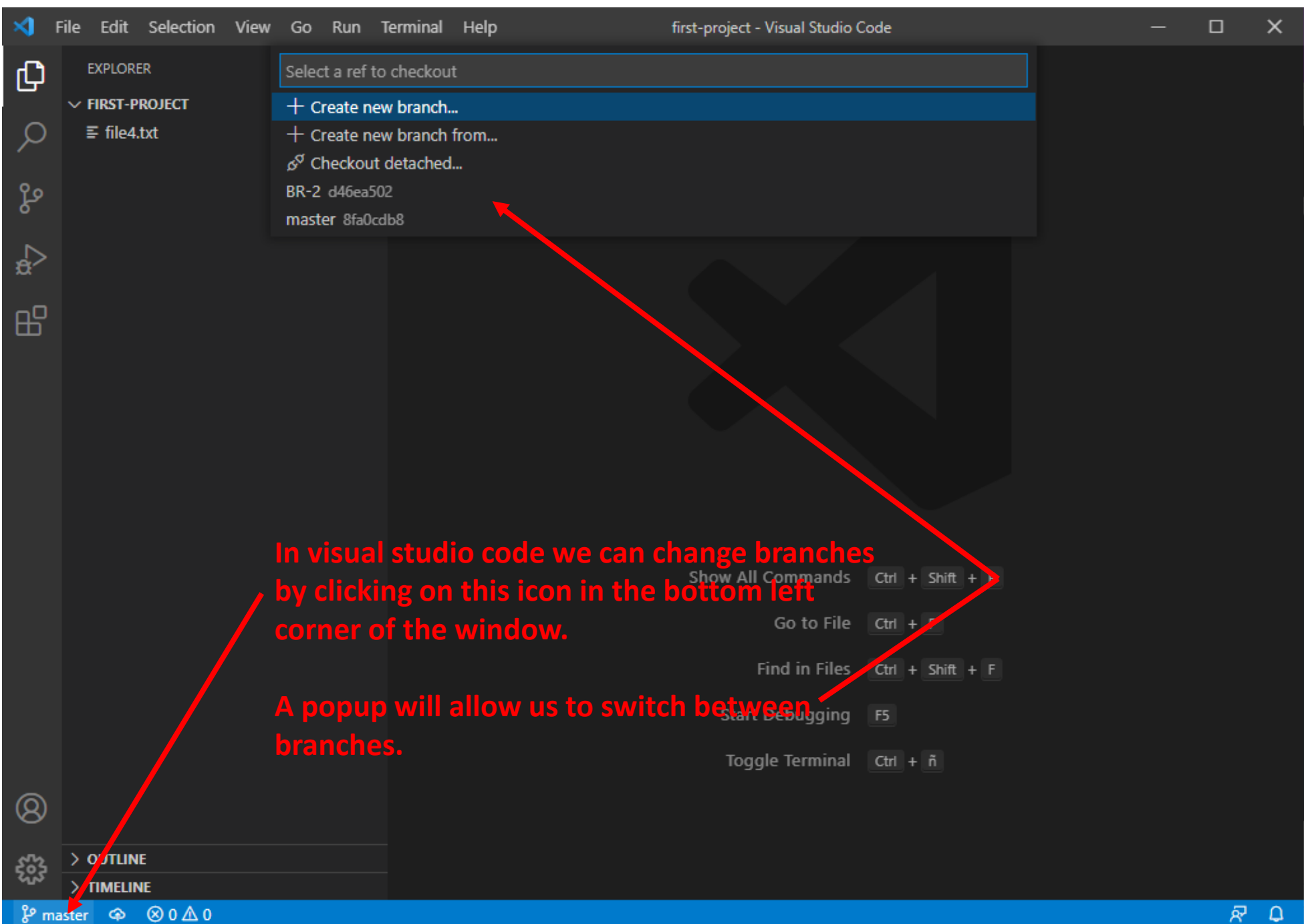
    Third commit

commit fd127e113a3ecba8f40ecb6d8d5ece8f10f7aa31
Author: Elliott Farmer <myemail@provider.com>
Date:   Thu Feb 24 23:00:56 2022 +0100
```



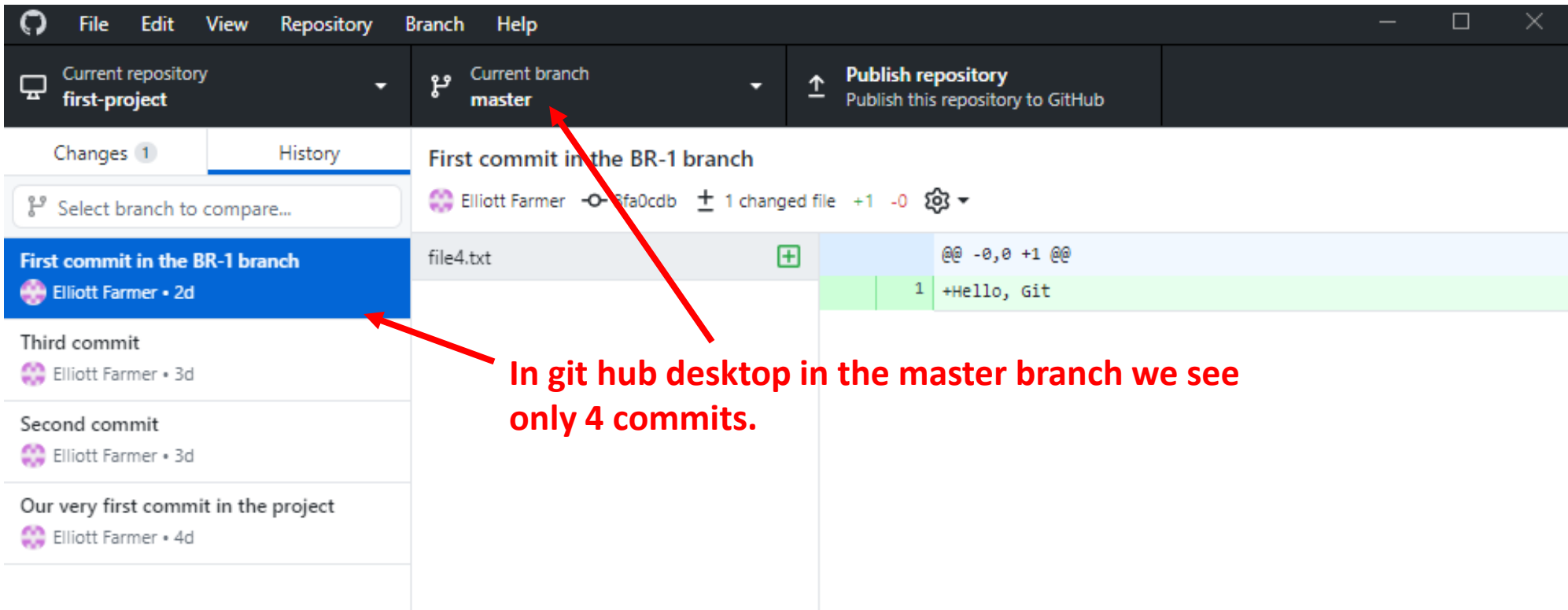
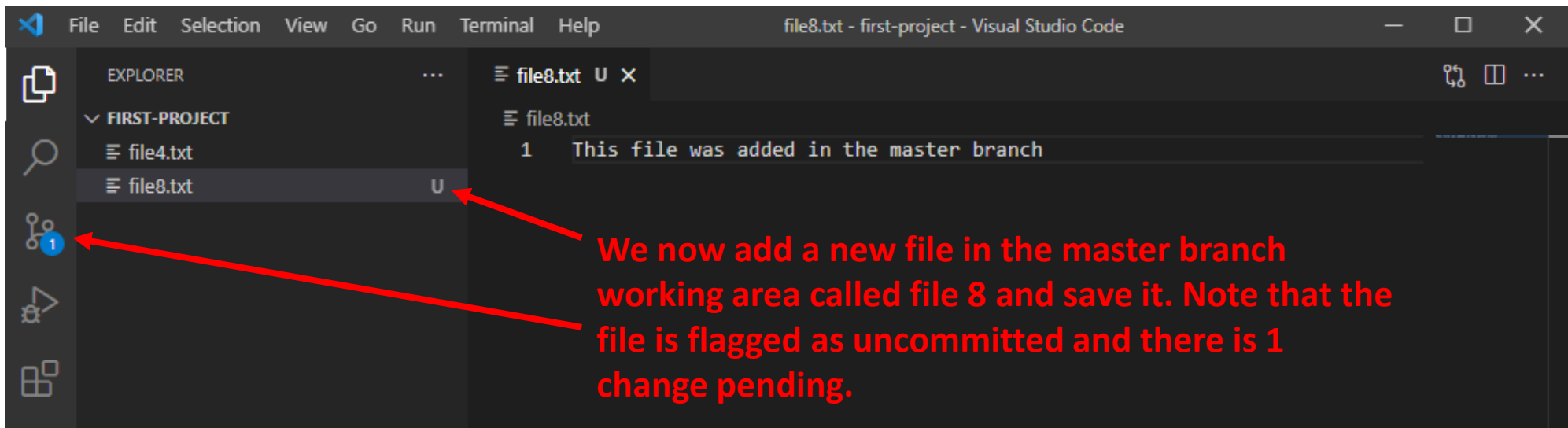
Note that checking out a branch in git will automatically update the working area and the staging area.

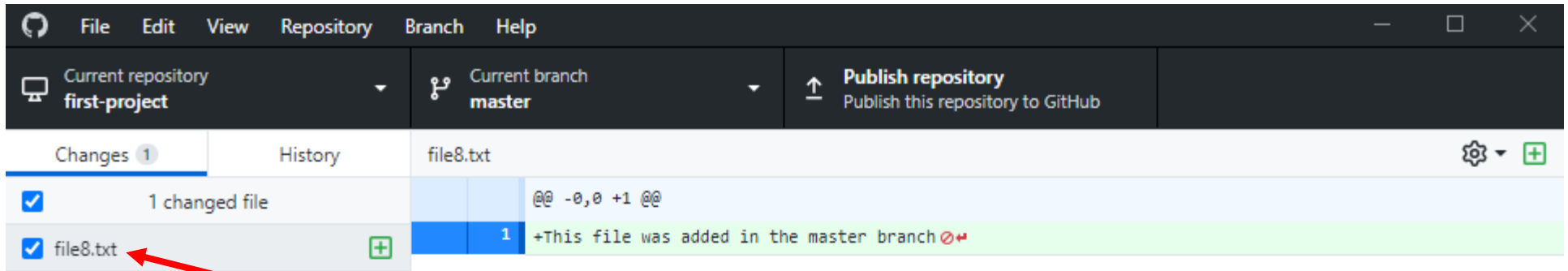
This is reflected in visual studio code because now we only see the contents in the working directory of the master branch.



In visual studio code we can change branches by clicking on this icon in the bottom left corner of the window.

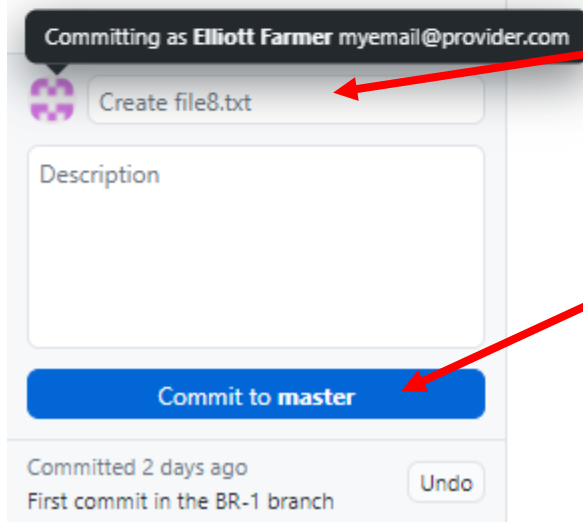
A popup will allow us to switch between branches.



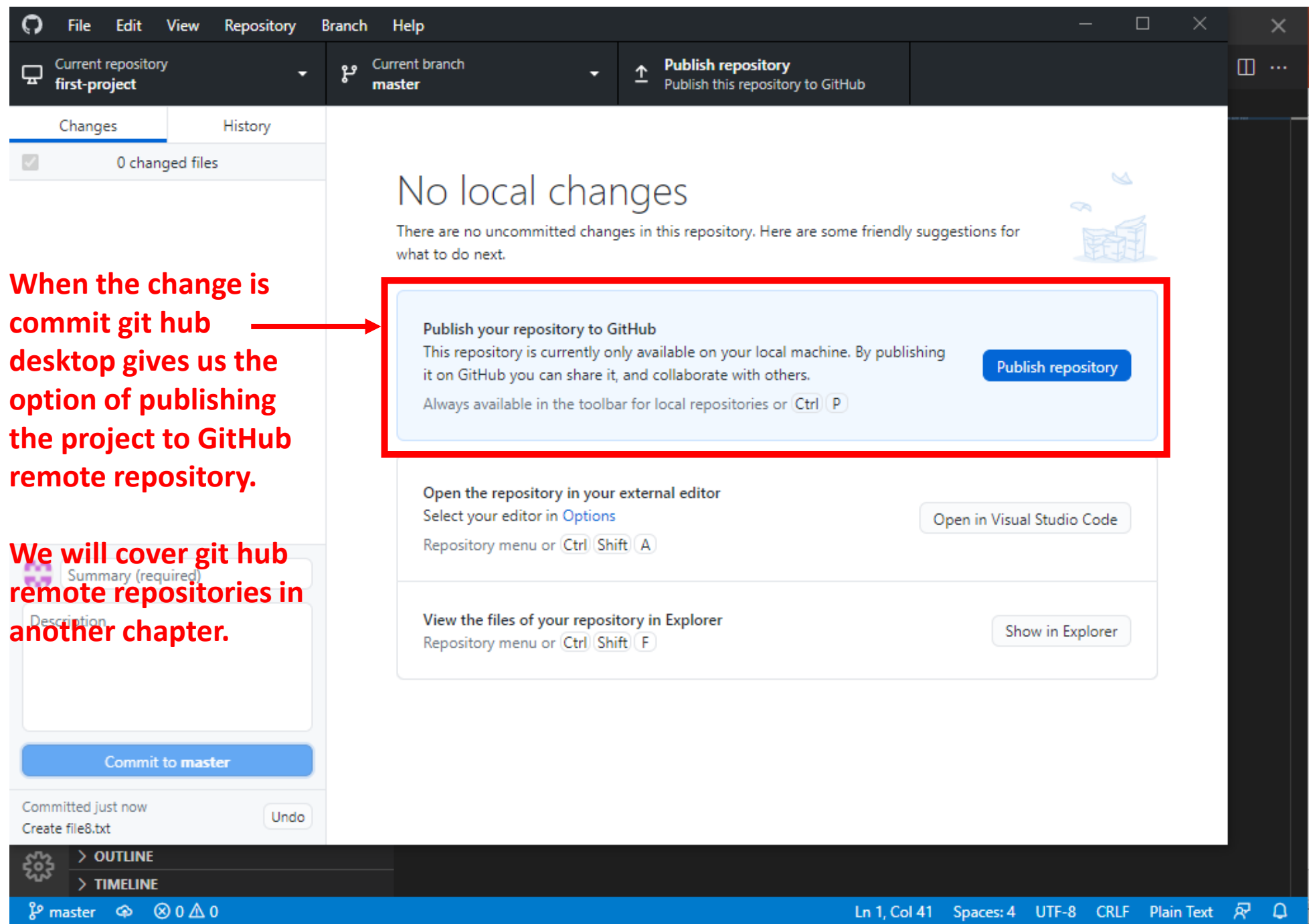


In the changes tab of git hub desktop we see that file 8 is ready to be added (staged) and committed to the repository.

We can do this from git hub desktop. The commit button by default gives us the commit message "Create file8.txt".



Clicking the commit button will commit this change with the default message as user Elliott Farmer with email myemail@provider.com



File Edit View Repository Branch Help

Current repository
first-project

Current branch
master

Publish repository
Publish this repository to GitHub

Changes History

Select branch to compare...

Create file8.txt
Elliott Farmer • 4m

First commit in the BR-1 branch
Elliott Farmer • 2d

Third commit
Elliott Farmer • 3d

Second commit
Elliott Farmer • 3d

Our very first commit in the project
Elliott Farmer • 4d

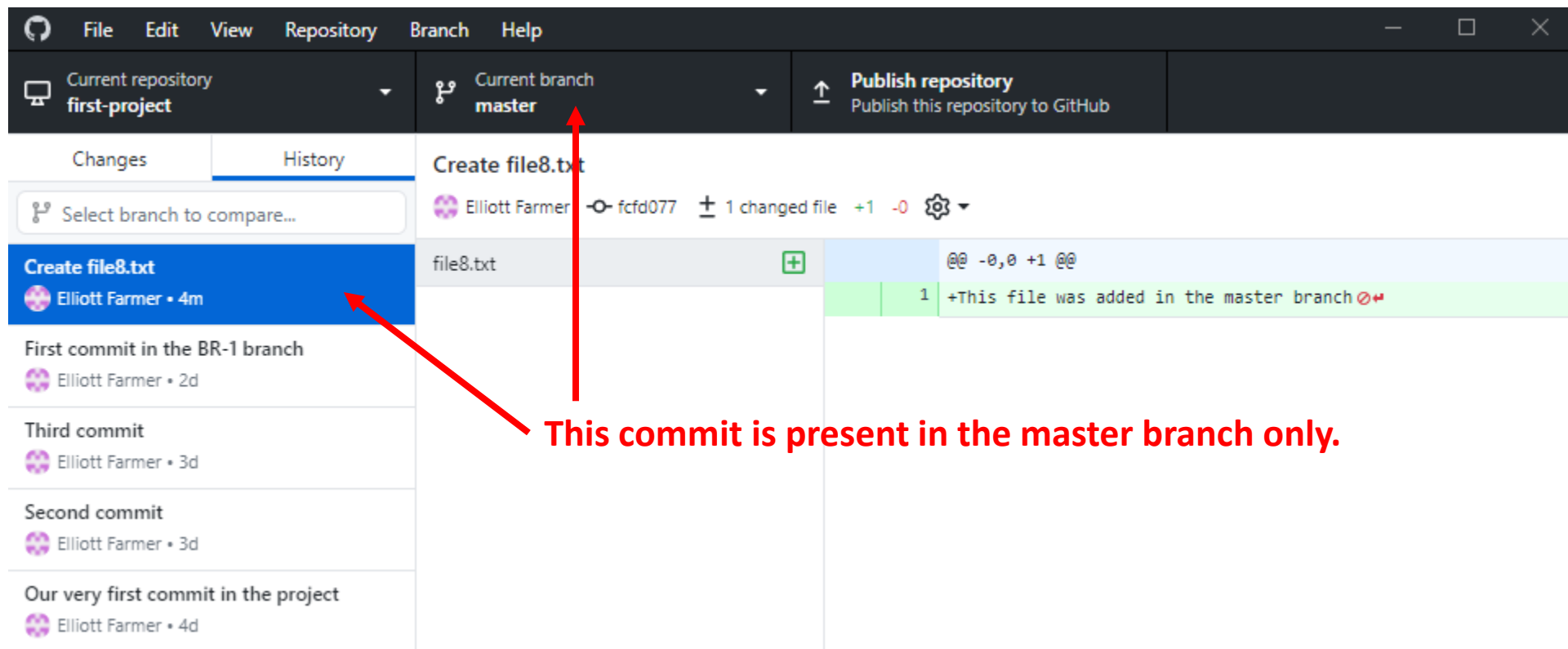
Create file8.txt

Elliott Farmer • fcf0077 ± 1 changed file +1 -0 ⚙

file8.txt

@@ -0,0 +1 @@

1 +This file was added in the master branch ↗



This commit is present in the master branch only.

File Edit View Repository Branch Help

Current repository **first-project** Current branch **BR-2** Publish repository
Publish this repository to GitHub

Changes History

Select branch to compare...

changed file5.txt
Elliott Farmer • 37m

3 new files were created in the BR-2 Br...
Elliott Farmer • 48m

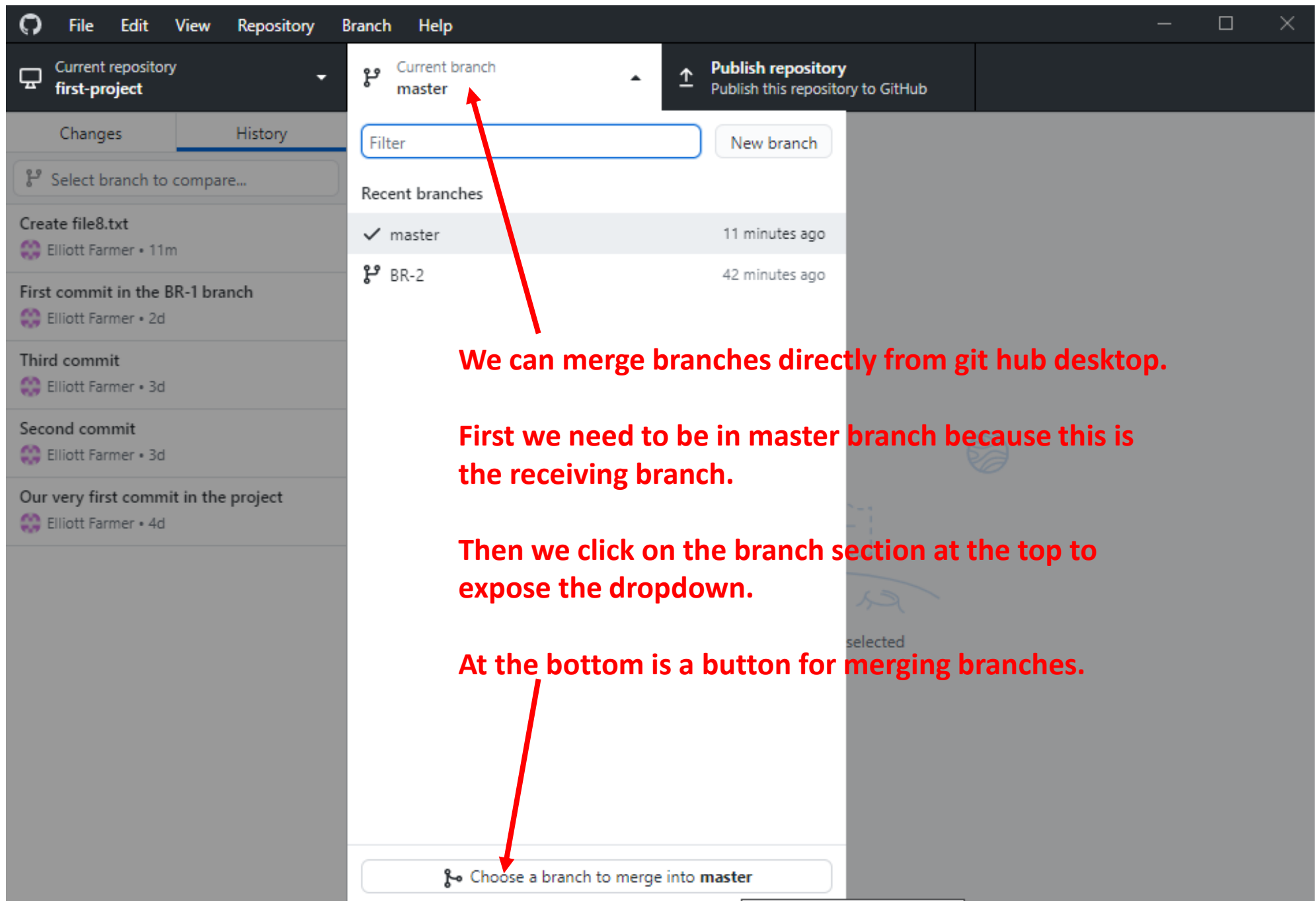
First commit in the BR-1 branch
Elliott Farmer • 2d

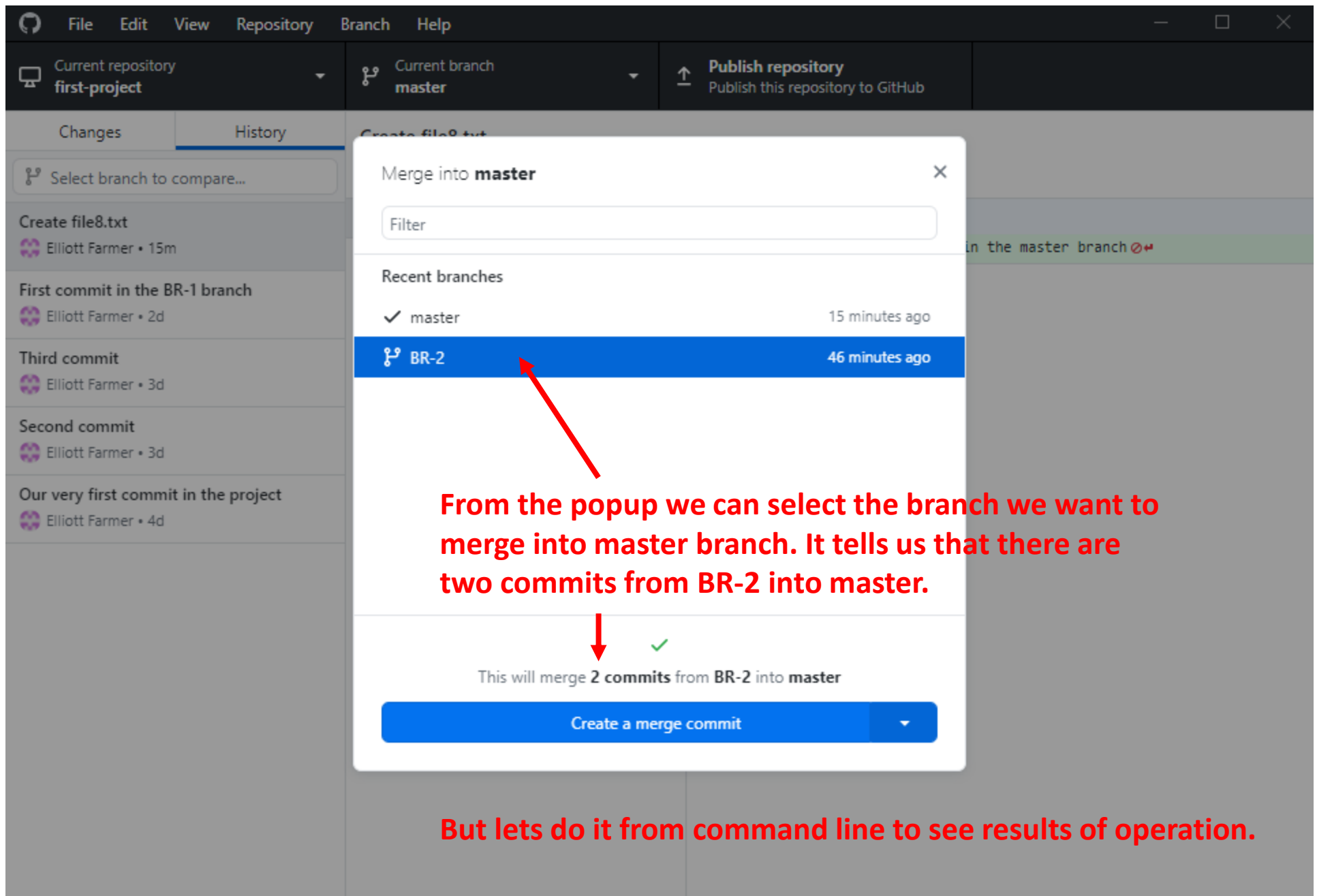
Third commit
Elliott Farmer • 3d

Second commit
Elliott Farmer • 3d

Our very first commit in the project
Elliott Farmer • 4d

In branch BR-2 we only see the commits of files 5, 6, and 7. These commits are not present in master branch.





From the popup we can select the branch we want to merge into master branch. It tells us that there are two commits from BR-2 into master.

But lets do it from command line to see results of operation.

git merge <branch name> -m <comment> command is used to merge BR-2 into master branch.

The output of this command tells us that 'ort' strategy was used.

It lists the three files that were changed with the number of lines that were added. This makes five lines in total or five insertions.

In the master branch we can see that files and new-files folder were added after performing the merge.

In git log we see the merge action commit **ec4520f9** to master branch.

And the last commit action in branch BR-2

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git merge BR-2 -m "merge branch BR-2 into master branch"
Merge made by the 'ort' strategy.
 file7.txt | 1 +
new-files/file5.txt | 3 +++
new-files/file6.txt | 1 +
3 files changed, 5 insertions(+)
create mode 100644 file7.txt
create mode 100644 new-files/file5.txt
create mode 100644 new-files/file6.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ ls
File4.txt file7.txt file8.txt new-files/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc (HEAD -> master)
Merge: fcf0d77 d46ea50
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 22:40:44 2022 +0100

    merge branch BR-2 into master branch

commit fcf0d7738d22c88c9bd9c7b881417ed517ebb9b6
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 22:21:44 2022 +0100

    Create file8.txt

commit d46ea502ce487a3b37843442f36a7289ea72e582 (BR-2)
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 21:50:53 2022 +0100

    changed file5.txt

commit f6695ebe499ee1e122343821b60a29ee668ba84b
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 21:39:50 2022 +0100

    3 new files were created in the BR-2 Branch
```

git cat-file -p <commit hash>

of the merge hash shows that the merge hash contains two parents, and a tree.

Checking the contents of the tree we see three blobs for the tree files and a tree for the new-files folder.

git cat-file -p <commit> of the first parent shows that this is for the creation of file8.txt.

git cat-file -p <commit> of the second parent shows that this is for the change of file4.txt

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p ec4520f9
tree a037a976b992f14e6bdec4e164b9068825b302da
parent fcfd07738d22c88c9bd9c7b881417ed517ebb9b6
parent d46ea502ce487a3b37843442f36a7289ea72e582
author Elliott Farmer <myemail@provider.com> 1646084444 +0100
committer Elliott Farmer <myemail@provider.com> 1646084444 +0100

merge branch BR-2 into master branch

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p a037a976
100644 blob b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e    file4.txt
100644 blob 2107281b79ed03e0147ea53fb62eb5f9a47aba2a    file7.txt
100644 blob b917051ef02bbb16e194c4ed120f8df70271aa3c    file8.txt
040000 tree 0e722963f708e7c3bbb285f0de0bff5db252d5bd    new-files

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p fcfd077
tree 30fe88853ae60b07786b2eccf80ada0f3fd3a42
parent 8fa0cdb8b1f83909b42278c993d037c76951db18
author Elliott Farmer <myemail@provider.com> 1646083304 +0100
committer Elliott Farmer <myemail@provider.com> 1646083304 +0100

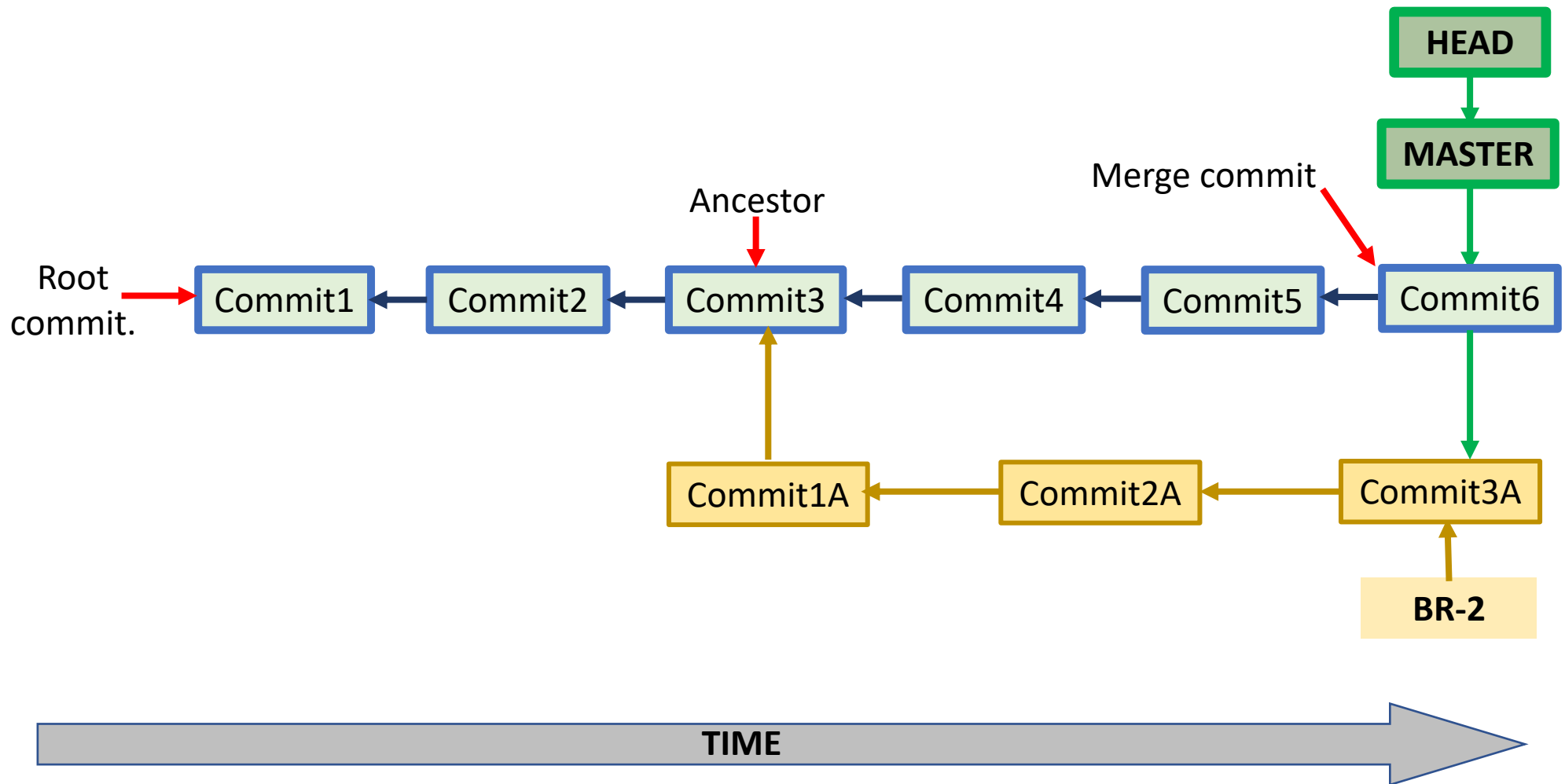
Create file8.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git cat-file -p d46ea502
tree 06f79e7659100d88e8124361abf1e5f290aa4955
parent f6695ebe499ee1e122343821b60a29ee668ba84b
author Elliott Farmer <myemail@provider.com> 1646081453 +0100
committer Elliott Farmer <myemail@provider.com> 1646081453 +0100

changed file5.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ |
```

As a diagram, commit 6 is the merge commit which has two parents, one pointing to the last commit in the master branch and one pointing to the last commit in the BR-2 branch.



File Edit View Repository Branch Help

Current repository: first-project | Current branch: master | Publish repository: Publish this repository to GitHub

Changes | History

Select branch to compare...

merge branch BR-2 into master branch
Elliott Farmer • 48m

Create file8.txt
Elliott Farmer • 1h

changed file5.txt
Elliott Farmer • 1h

3 new files were created in the BR-2 Br...
Elliott Farmer • 1h

First commit in the BR-1 branch
Elliott Farmer • 3d

Third commit
Elliott Farmer • 3d

Second commit
Elliott Farmer • 4d

Our very first commit in the project
Elliott Farmer • 4d

merge branch BR-2 into master branch

Elliott Farmer ec4520f ± 3 changed files +5 -0 ⚙

file7.txt	+	@@ -0,0 +1 @@
new-files\file5.txt	+	1 +Just another file in the root of the working directory
new-files\file6.txt	+	

Git hub desktop history shows the last merge of BR-2 into master branch.

This is the last commit in master branch before the merge.

These commits were performed in branch BR-2 but are now visible in master branch after the merge.

git branch -d <branch name>

Command can be used to delete BR-2 branch. Notice that it does not give any warning message because we have already merged the contents of BR-2 into the master branch. Nor does it create a new commit action. The last commit is still **ec4520f9**

git log Command shows no new commit actions for this delete. However notice that the commit action that was labelled BR-2->Master has lost its label of BR-2-> master.

However the merge commit retains two parents even after deleting the BR-2 branch.

This is because these commits made under BR-2 are still in the repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch -d BR-2
Deleted branch BR-2 (was d46ea50).

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc (HEAD -> master)
Merge: fcf0d77 d46ea50
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 22:40:44 2022 +0100

    merge branch BR-2 into master branch

commit fcf0d7738d22c88c9bd9c7b881417ed517ebb9b6
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 22:21:44 2022 +0100

    Create file8.txt

commit d46ea502ce487a3b37843442f36a7289ea72e582
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 21:50:53 2022 +0100

    changed file5.txt

commit f6695ebe499ee1e122343821b60a29ee668ba84b
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 21:39:50 2022 +0100

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project
$ git cat-file -p ec4520f9
tree a037a976b992f14e6bdec4e164b9068825b302da
parent fcf0d7738d22c88c9bd9c7b881417ed517ebb9b6
parent d46ea502ce487a3b37843442f36a7289ea72e582
author Elliott Farmer <myemail@provider.com> 1646084444 +0100
committer Elliott Farmer <myemail@provider.com> 1646084444 +0100

merge branch BR-2 into master branch

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$
```

Git hub repository has some limitations when viewing commits because it does not show from what tree commits were merged. This is why we are going to install another free graphical user interface from www.sourcetreeapp.com

The image shows a web browser displaying the Sourcetree website. The website has a navigation bar with the Sourcetree logo and a 'Download free' button. The main content area features the text 'Simplicity and power in a beautiful Git GUI' and a 'Download for Windows' button, which is highlighted by a red arrow. Below this, it says 'Also available for Mac OS X'. On the right side of the website, there is a preview of the Sourcetree application interface. The application window shows a sidebar with 'WORKSPACE' and 'History' selected. The main area displays a commit history table with columns for Commit, Author, Description, and Date. The commit history table lists several commits, including one by Rahul Chhabra on Mar 3, 2016, and another by Michael Minns on Nov 23, 2015. The application interface also shows a 'Graph' view of the commit history and a 'Commit' button.

www.sourcetreeapp.com

Sourcetree

Download free

Simplicity and power in a beautiful Git GUI

Download for Windows

Also available for Mac OS X

source-tree-website (Git)

Commit Pull Push Branch Merge Shelf Show in Finder Terminal Settings

WORKSPACE

File status

History

Search

BRANCHES

BOOKMARKS

TAGS

REMOTES

SHELVED

SUBREPOSITORIES

Graph

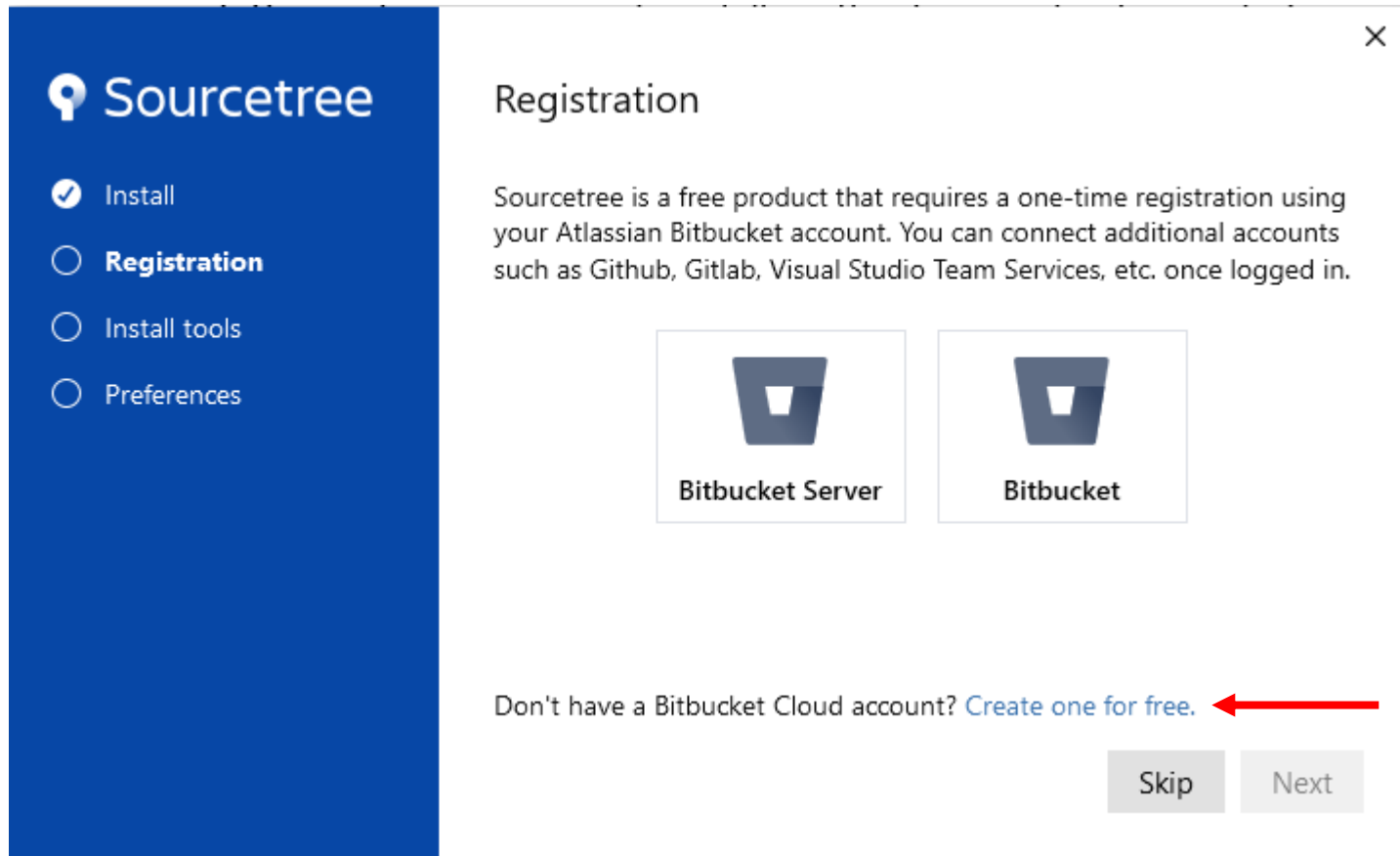
Commit	Author	Description	Date
b7358c7	Rahul Chha...	master origin/master origin/HEAD Removing ol...	Mar 3, 2016, 11:...
bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
dfe975d	Tyler Tadej...	origin/update-google-verification Update google verificati...	Feb 11, 2016, 2:2...
3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
dba4719	Tyler Tadej...	Add gitignore	Feb 11, 2016, 1:3...
ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
246c4ff	Joel Unger...	origin/hero_images hero_images Used TinyPNG to c...	Feb 11, 2016, 3:3...
9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
85367bb	Patrick Tho...	origin/bug/date-https fixed date and https errors	Jan 7, 2016, 12:2...
4f9b557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
384e6d5	Rahul Chhab...	origin/search-console-access search console google ver...	Feb 3, 2016, 2:09...
6fa47a9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
0cdf9e6	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
a723bc2	Mike Minns...	Merge branch 'au2gex'	Nov 23, 2015, 1:5...
65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
500a892	Michael Min...	Merged in au2gex (pull request #11)	Nov 23, 2015, 1:0...

Activate Windows
Go to Settings to activate Windows.

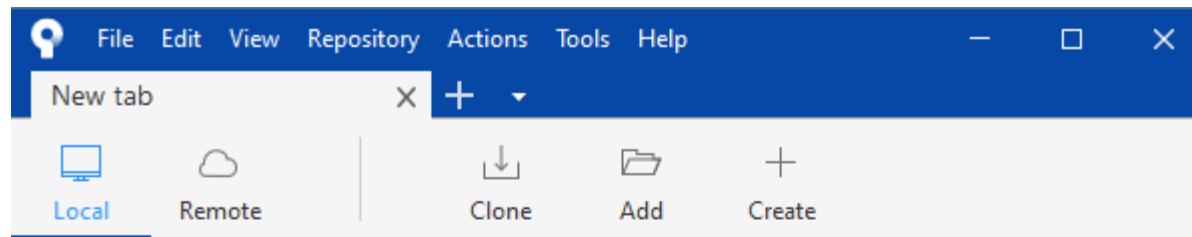
Waiting for wac-cdn.atlassian.com...

Reset 100% 23:46


When we install SourceTree it requires us to have a bitbucket account. Bit bucket is another remote repository like git hub.



Once the registration and installation of SourceTree is complete we can drag and drop our first-project into the local repository tab.



Local repositories

 first-project
 master C:\xampp3\htdocs\lessons\07_GIT\first-project

File Edit View Repository Actions Tools Help

first-project

Commit

Pull

Push

Fetch

Branch

Merge

Stash

Discard

Tag

Git-flow

Remote

Terminal

Explorer

Settings

WORKSPACE

File Status

History

Search

BRANCHES

master

TAGS

REMOTES

STASHES

All Branches Show Remote Branches Date Order

Graph

merge branch BR-2 into master branch

Create file8.txt

changed file5.txt

3 new files were created in the BR-2 Branch

First commit in the BR-1 branch

Third commit

Second commit

Our very first commit in the project

Date

28 Feb 2022 22:40

28 Feb 2022 22:21

28 Feb 2022 21:50

28 Feb 2022 21:39

25 Feb 2022 23:18

25 Feb 2022 2:48

24 Feb 2022 23:00

24 Feb 2022 21:04

Author

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Elliott Farmer <my

Commit

ec4520f

fcfd077

d46ea50

f6695eb

8fa0cdb

f3f965f

fd127e1

25de26c

Sorted by file status

file7.txt

Commit: ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc [ec4520f]

Parents: fcfd07738d, d46ea502ce

Author: Elliott Farmer <myemail@provider.com>

Date: 28 February 2022 22:40:44

Committer: Elliott Farmer

merge branch BR-2 into master branch

file7.txt

new-files/file5.txt

new-files/file6.txt

File Contents

Reverse hunk

Just another file in the root of the working directory

No newline at end of file

If we double click on the project it opens in source tree and we can see a complete history of the project including all the branches. This is a much better interface than git hub desktop.

first-project

Commit

Pull

Push

Fetch

Branch

Merge

Stash

Discard

Tag

Git-flow

Remote

Terminal

Explorer

Settings

WORKSPACE

File Status

History

Search

BRANCHES

master

TAGS

REMOTES

STASHES

All Branches

Show Remote Branches

Date Order

Jump to:

Graph	Description	Date	Author	Commit
	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <myemail@provider.com>	ec4520f
	Create file8.txt	28 Feb 2022 22:21	Elliott Farmer <myemail@provider.com>	fcfd077
	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <myemail@provider.com>	d46ea50
	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <myemail@provider.com>	f6695eb
	First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <myemail@provider.com>	8fa0cdb
	Third commit	25 Feb 2022 2:48	Elliott Farmer <myemail@provider.com>	f3f965f
	Second commit	24 Feb 2022 23:00	Elliott Farmer <myemail@provider.com>	fd127e1
	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <myemail@provider.com>	25de26c

Sorted by file status

Commit: ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc [ec4520f]

Parents: fcfd07738d, d46ea502ce

Author: Elliott Farmer <myemail@provider.com>

Date: 28 February 2022 22:40:44

Committer: Elliott Farmer

merge branch BR-2 into master branch

file7.txt

new-files/file5.txt

new-files/file6.txt

file7.txt

File Contents

Reverse hunk

0

Just another file in the root of the working directory

No newline at end of file

In the top pane we have a history of the commits with description, date, author and commit SHA1

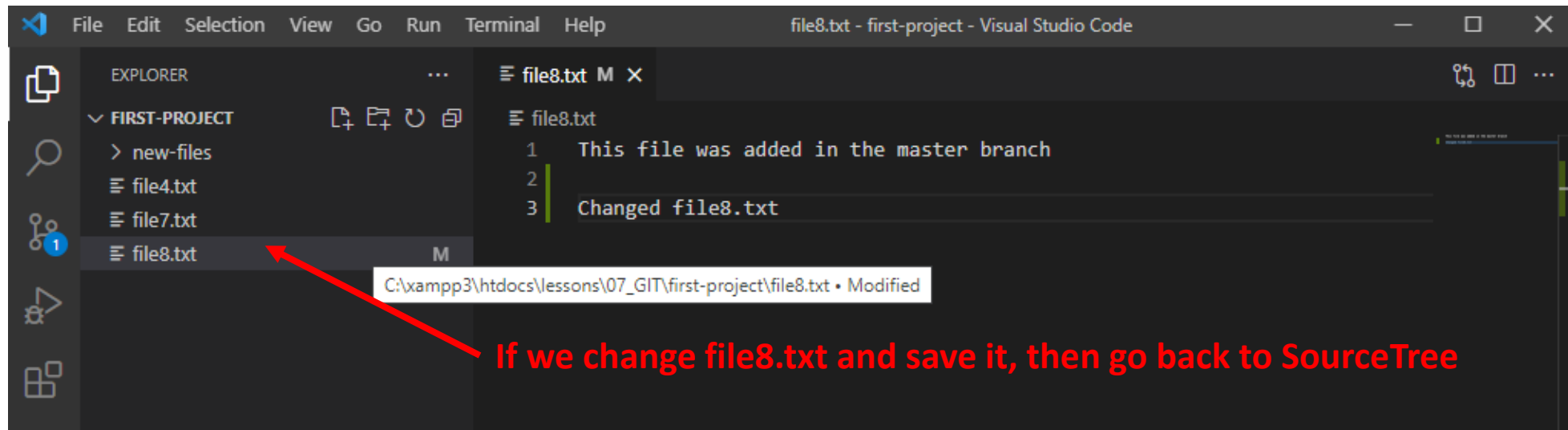
In this pane we see all branches which right now is only the master branch.

This pane shows the contents of the commit file like a git cat-file -p <commit> command

This pane shows contents of the selected file like a cat-file -p command.

This pane shows the list of files like a ls command

Activate Windows
Go to Settings to activate Wi



first-project

Commit Pull Push Fetch Branch Merge Stash Discard Tag

Git-flow Remote Terminal Explorer Settings

WORKSPACE

File Status History Search

BRANCHES

- master

TAGS

REMOTES

STASHES

All Branches Show Remote Branches Date Order

Graph Description Date Author Commit

Uncommitted changes 1 Mar 2022 0:22 *

merge branch BR-2 into master branch 28 Feb 2022 22:40 Elliott Farmer <myemail@provider.com> ec4520f

Create file8.txt 28 Feb 2022 22:21 Elliott Farmer <myemail@provider.com> fcf077

changed file5.txt 28 Feb 2022 21:50 Elliott Farmer <myemail@provider.com> d46ea50

3 new files were created in the BR-2 Branch 28 Feb 2022 21:39 Elliott Farmer <myemail@provider.com> f6695eb

First commit in the BR-1 branch 25 Feb 2022 23:18 Elliott Farmer <myemail@provider.com> 8fa0cdb

Third commit 25 Feb 2022 2:48 Elliott Farmer <myemail@provider.com> f3f965f

Second commit 24 Feb 2022 23:00 Elliott Farmer <myemail@provider.com> fd127e1

Our very first commit in the project 24 Feb 2022 21:04 Elliott Farmer <myemail@provider.com> 25de26c

Pending files, sorted by file status

Staged files Unstage All Unstage Selected file8.txt

Unstaged files Stage All Stage Selected file8.txt

Hunk 1: Lines 1-3

Stage hunk Discard hunk

1 - This file was added in the master branch

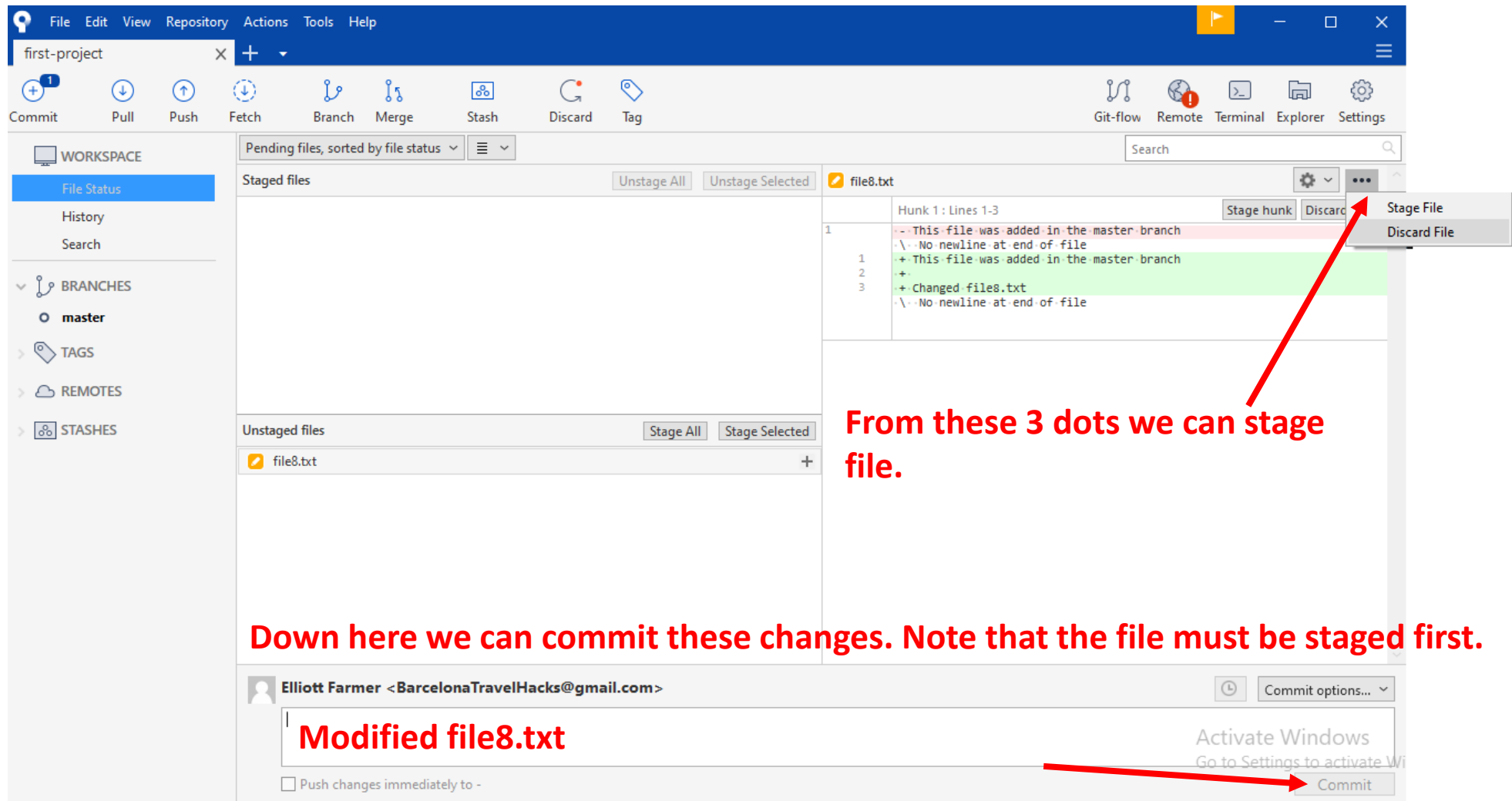
2 + This file was added in the master branch

3 + Changed file8.txt

No newline at end of file

Activate Windows Go to Settings to activate Windows

We see immediately in SourceTree that file8.txt has been changed and what changes have been made.



From this button a dropdown for new branches appears

From this button merges can be performed

The screenshot displays the GitHub Desktop application interface. The top menu bar includes File, Edit, View, Repository, Actions, Tools, and Help. Below the menu is a toolbar with buttons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag. The main workspace is divided into three sections: a left sidebar with 'WORKSPACE' (File Status, History, Search), 'BRANCHES' (master), 'TAGS', 'REMOTES', and 'STASHES'; a central 'Commit History' table; and a bottom section for file diffing.

Commit History Table:

Graph	Description	Date	Author	Commit
	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <BarcelonaTravelHacks@gmail.com>	1c25cd3
	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <myemail@provider.com>	ec4520f
	Create file8.txt	28 Feb 2022 22:21	Elliott Farmer <myemail@provider.com>	fcfd077
	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <myemail@provider.com>	d46ea50
	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <myemail@provider.com>	f6695eb
	First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <myemail@provider.com>	8fa0cdb
	Third commit	25 Feb 2022 2:48	Elliott Farmer <myemail@provider.com>	f3f965f
	Second commit	24 Feb 2022 23:00	Elliott Farmer <myemail@provider.com>	fd127e1
	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <myemail@provider.com>	25de26c

File Diff Section (file8.txt):

Commit: 1c25cd31a73d58e8ef6fc39e0f20012ccdecdb8 [1c25cd3]
Parents: ec4520f9a4
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: 01 March 2022 00:30:33
Committer: Elliott Farmer
Modified file8.txt

Hunk 1: Lines 1-3

```
1  This file was added in the master branch
2  No newline at end of file
3  This file was added in the master branch
+  Changed file8.txt
+  No newline at end of file
```

The commit shows up in the branch diagram history section

SourceTree has one more feature. If I double click on any commit then a dialogue window pops up. This is to confirm a checkout action to checkout the repository at that particular commit.

The screenshot shows the SourceTree application window for a project named 'first-project'. The interface includes a menu bar (File, Edit, View, Repository, Actions, Tools, Help), a toolbar with icons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag, and a sidebar with sections for WORKSPACE (File Status, History, Search), BRANCHES (master), TAGS, REMOTES, and STASHES. The main area displays a commit history table with columns for Graph, Description, Date, Author, and Commit. A commit with ID '8fa0cdb' is selected and highlighted in blue. A dialog box titled 'Confirm change working copy' is overlaid on the interface, containing a warning icon and the following text: 'Checking out this commit will create a detached HEAD, and you will no longer be on any branch (unless you subsequently create a new one). Are you sure that's what you want? If not, press Cancel and pick another commit, or if this is a remote branch you can use the 'Checkout New Branch' tab instead to create a new local branch.' The dialog has an 'OK' button and a 'Cancel' button. A red arrow points from the text above to the 'OK' button. Another red arrow points from the 'OK' button to the text below. The bottom of the window shows the commit details for the selected commit, including the commit hash, parent hash, author, date, and committer. The file 'file4.txt' is shown in the workspace with its contents 'Hello, Git'.

Graph	Description	Date	Author	Commit
○	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <BarcelonaTravelHacks@gmail.com>	1c25cd3
○	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <myemail@provider.com>	ec4520f
○	Create file8.txt			fcd077
○	changed file5.txt			d46ea50
○	3 new files were created in the BR-2 Branch			f6695eb
○	First commit in the BR-1 branch			8fa0cdb
○	Third commit			f3f965f
○	Second commit			fd127e1
○	Our very first commit in the project			25de26c

Sorted by file status

Commit: 8fa0cdb8b1f83909b42278c993d037c76951db18 [8fa0cdb]
Parents: f3f965f4dd
Author: Elliott Farmer <myemail@provider.com>
Date: 25 February 2022 23:18:08
Committer: Elliott Farmer

First commit in the BR-1 branch

file4.txt

File Contents

0
+Hello, Git

Reverse hunk

This will create a detached head state at that particular commit.

Now we see the head pointer is on that commit but the master remains at the last commit (detached HEAD).

The screenshot displays the Git GUI interface for a repository named 'first-project'. The interface is divided into several sections:

- Top Bar:** Contains menu items (File, Edit, View, Repository, Action, Tools, Help) and icons for common Git actions (Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, Tag).
- Left Sidebar:** Shows the 'WORKSPACE' section with 'File Status', 'History' (selected), and 'Search'. Below it, the 'BRANCHES' section lists 'master' and 'HEAD' (detached).
- Main Panel:** Displays a commit history table with columns: Graph, Description, Date, Author, and Commit. The 'HEAD' pointer is highlighted on the commit 'First commit in the BR-1 branch' (commit hash 8fa0cdb).
- Bottom Panel:** Shows the 'file4.txt' diff view, indicating changes made in the current commit.

The commit history table is as follows:

Graph	Description	Date	Author	Commit
●	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <BarcelonaTravelHacks@gmail.c	1c25cd3
●	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <myemail@provider.com>	ec4520f
●	create file8.txt	28 Feb 2022 22:21	Elliott Farmer <myemail@provider.com>	fcfd077
●	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <myemail@provider.com>	d46ea50
●	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <myemail@provider.com>	f6695eb
●	HEAD First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <myemail@provider.com>	8fa0cdb
●	Third commit	25 Feb 2022 2:48	Elliott Farmer <myemail@provider.com>	f3f965f
●	Second commit	24 Feb 2022 23:00	Elliott Farmer <myemail@provider.com>	fd127e1
●	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <myemail@provider.com>	25de26c

The diff view for 'file4.txt' shows the following changes:

```
File Contents
0
+ Hello, Git
```

In terminal view we can see we are in detached head state and that files in working directory are those up to that commit.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project ((8fa0cdb...))
$ ls
file4.txt
```

The screenshot shows the Git GUI interface with the following components:

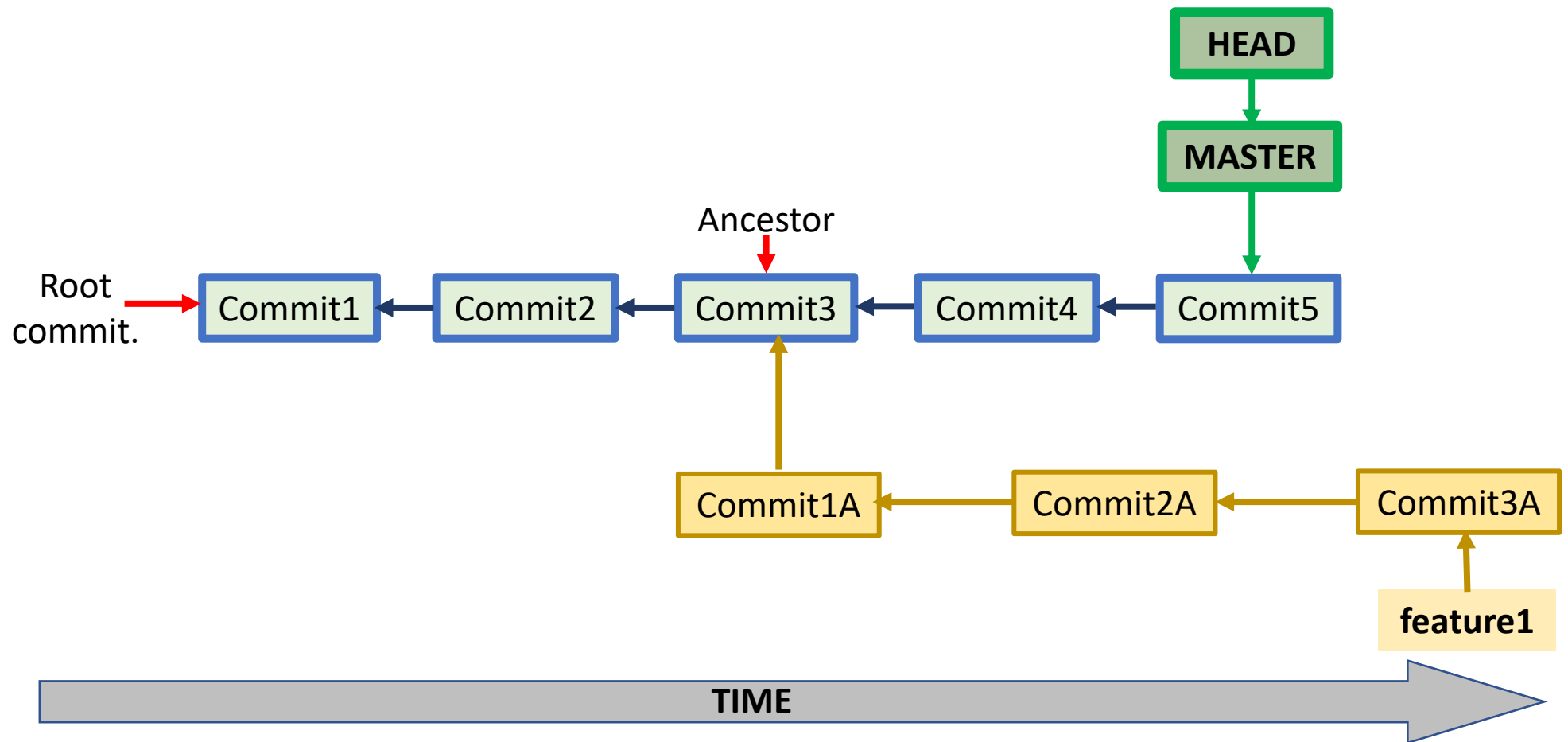
- Toolbar:** Includes icons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag.
- Left Sidebar:** Contains sections for WORKSPACE (File Status, History, Search), BRANCHES (master), TAGS, REMOTES, and STASHES.
- Commit History Table:**

Graph	Description	Date	Author	Commit
●	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <BarcelonaTravelHacks@gmail.com>	1c25cd3
●	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <myemail@provider.com>	ec4520f
●	Create file8.txt	28 Feb 2022 22:21	Elliott Farmer <myemail@provider.com>	fcfd077
●	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <myemail@provider.com>	d46ea50
●	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <myemail@provider.com>	f6695eb
●	First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <myemail@provider.com>	8fa0cdb
●	Third commit	25 Feb 2022 2:48	Elliott Farmer <myemail@provider.com>	f3f965f
●	Second commit	24 Feb 2022 23:00	Elliott Farmer <myemail@provider.com>	fd127e1
●	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <myemail@provider.com>	25de26c
- Commit Details Panel:** Shows details for commit 1c25cd3, including parents (ec4520f9a4), author (Elliott Farmer), date (01 March 2022 00:30:33), and committer (Elliott Farmer). It also lists the modified file8.txt.
- File Diff Panel:** Shows the diff for file8.txt, highlighting changes in the master branch.

Double clicking on the last commit will return the HEAD to this commit

Merge Conflicts

Merge conflicts can occur when the same file has different edits in two or more branches. As a result merge conflicts cannot occur in fast forward commit because there are no other commits after the divergence into the new branch. However, merge conflicts can occur when there are commits after ancestor, i.e. a three-way merge.



Current situation:

There is only one branch,
master branch.

We have created a new
branch called BR-3.

We have checked out BR-3.
It contains several files and
a folder.

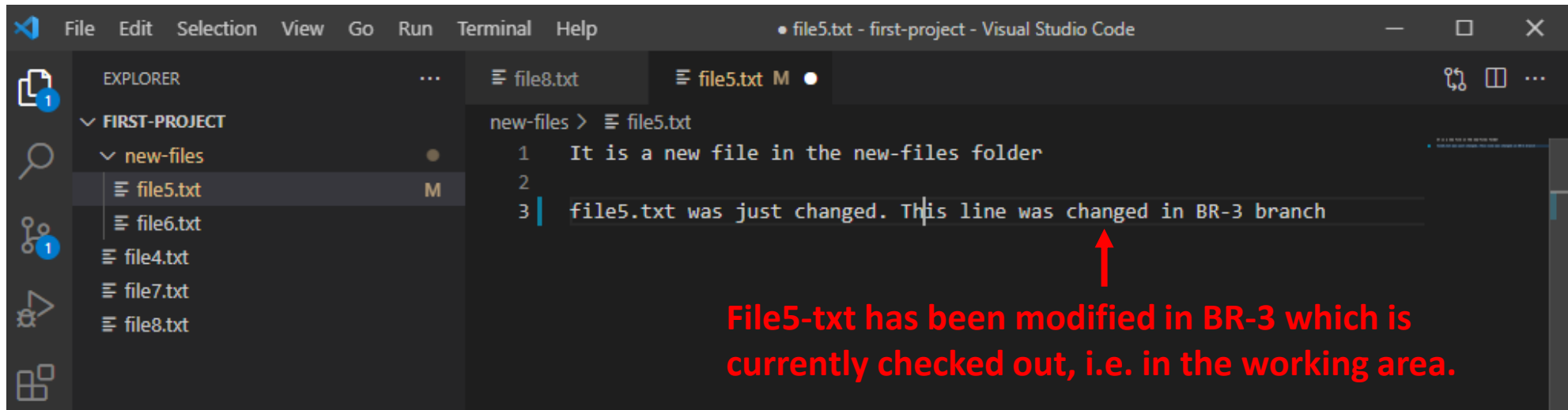
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch
* master

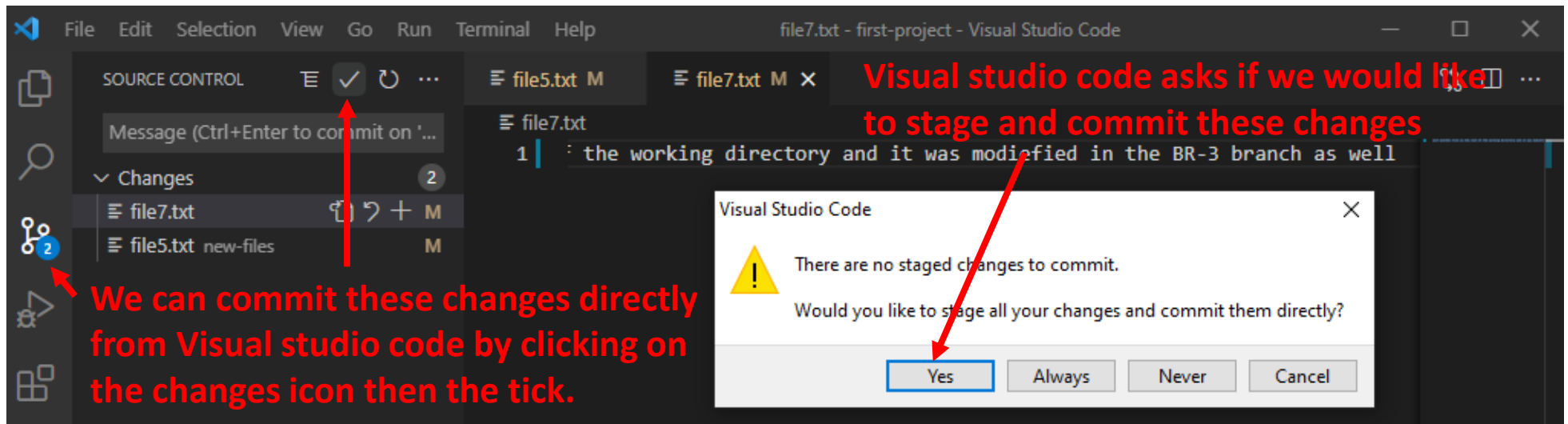
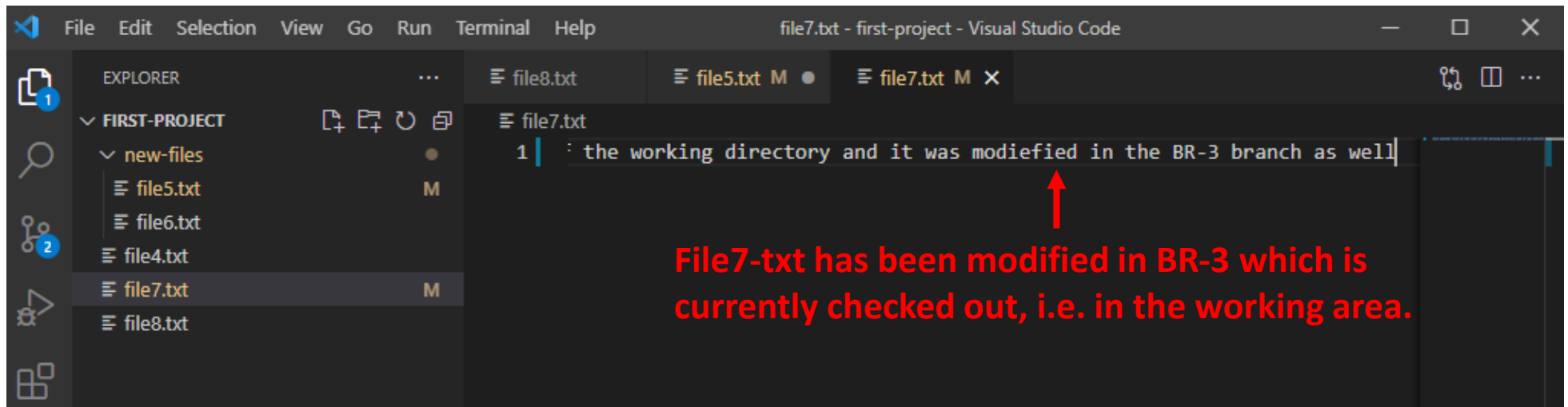
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git branch BR-3

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout BR-3
Switched to branch 'BR-3'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-3)
$ ls
file4.txt  file7.txt  file8.txt  new-files/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (BR-3)
$ |
```





The commit in branch BR-3 shows in SourceTree.

The screenshot displays the SourceTree application interface. The top menu bar includes File, Edit, View, Repository, Actions, Tools, and Help. Below the menu is a toolbar with icons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag. The left sidebar shows the Workspace panel with File Status, History, and Search options. The BRANCHES section lists BR-3 and master. The main panel shows the commit history table, with the top commit in BR-3 highlighted. A red arrow points from the text above to this commit. The bottom panel shows the diff for file8.txt, with a hunk of changes highlighted in green.

Graph	Description	Date	Author	Commit
BR-3	Changes made in BR-3 branch	1 Mar 2022 20:52	Elliott Farmer <Ba	143b205
master	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <Bar	1c25cd3
	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <my	ec4520f
	Create file8.txt	28 Feb 2022 22:21	Elliott Farmer <my	fcfd077
	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <my	d46ea50
	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <my	f6695eb
	First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <my	8fa0cdb
	Third commit	25 Feb 2022 2:48	Elliott Farmer <my	f3f965f
	Second commit	24 Feb 2022 23:00	Elliott Farmer <my	fd127e1
	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <my	25de26c

Sorted by file status

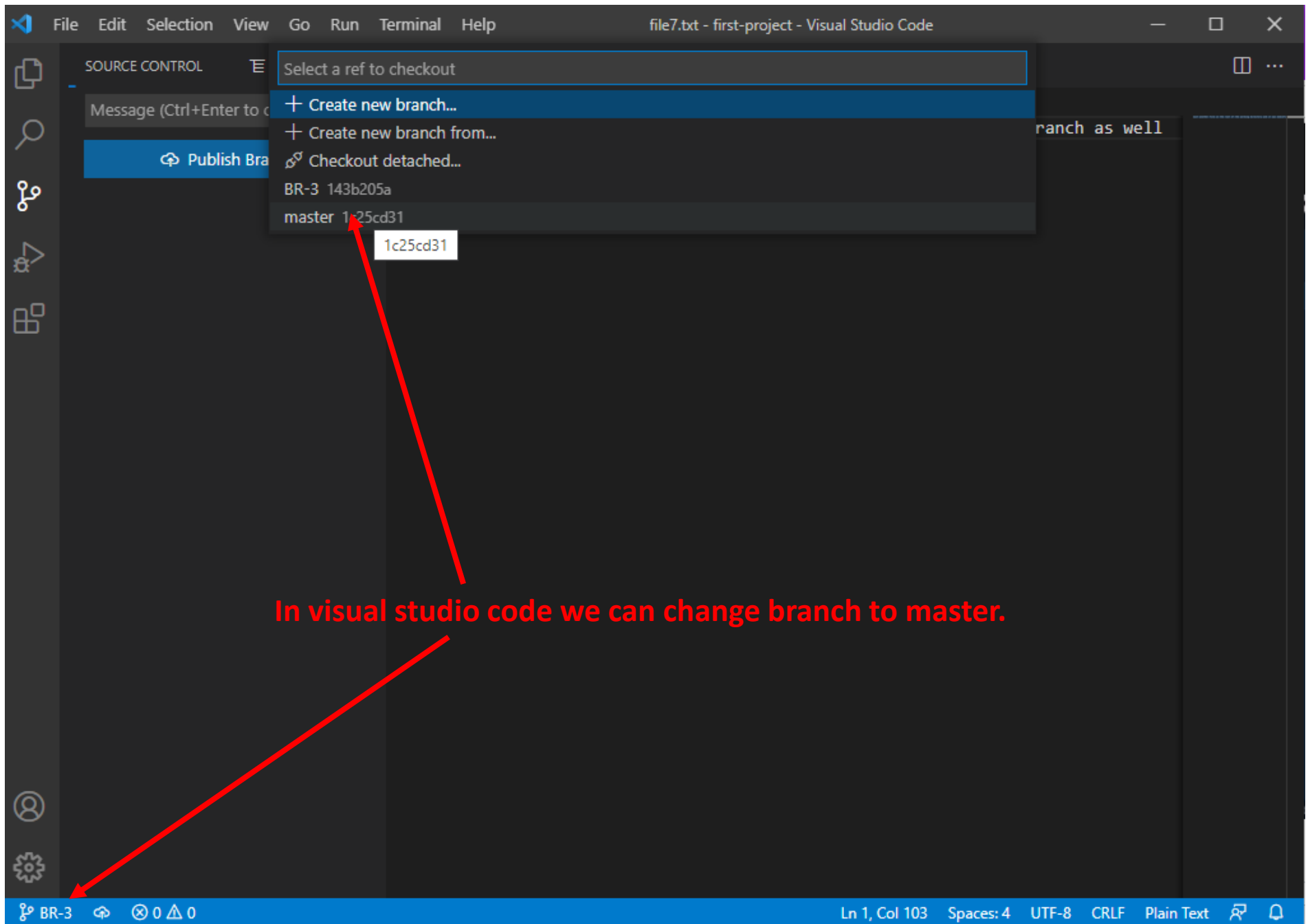
Commit: 1c25cd31a73d58e8ef6fc39e0f20012ccdecd1b8 [1c25cd3]
Parents: ec4520f9a4
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: 01 March 2022 00:30:33
Committer: Elliott Farmer

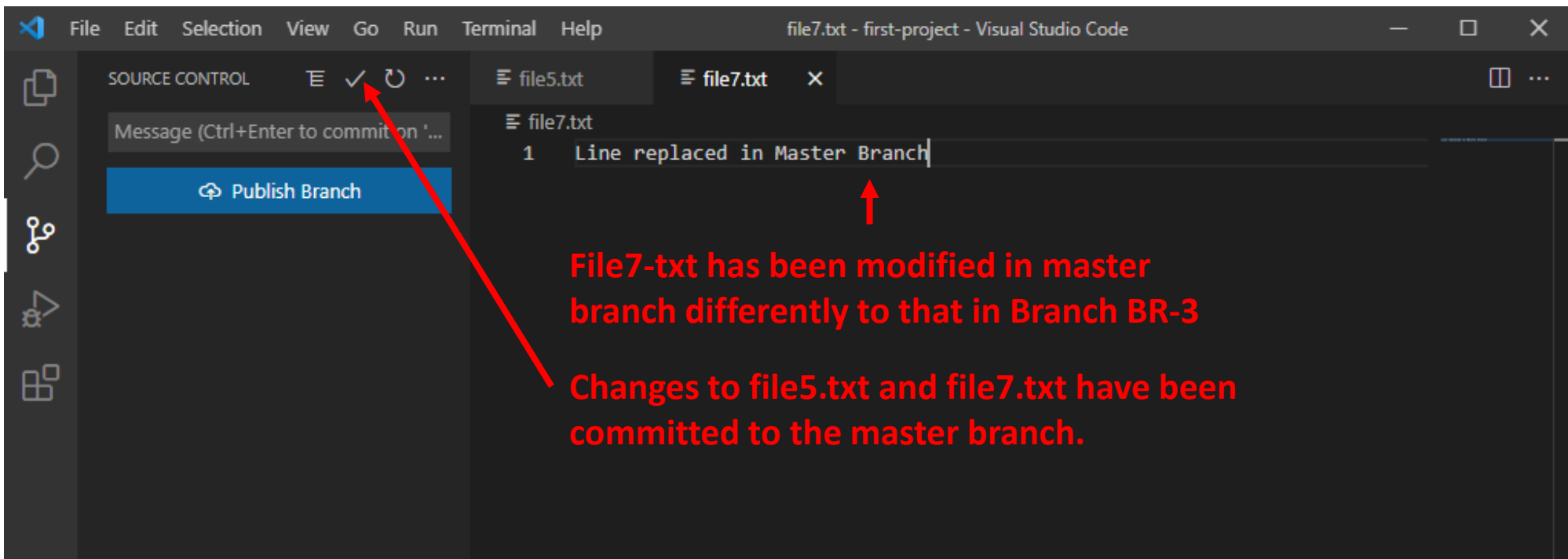
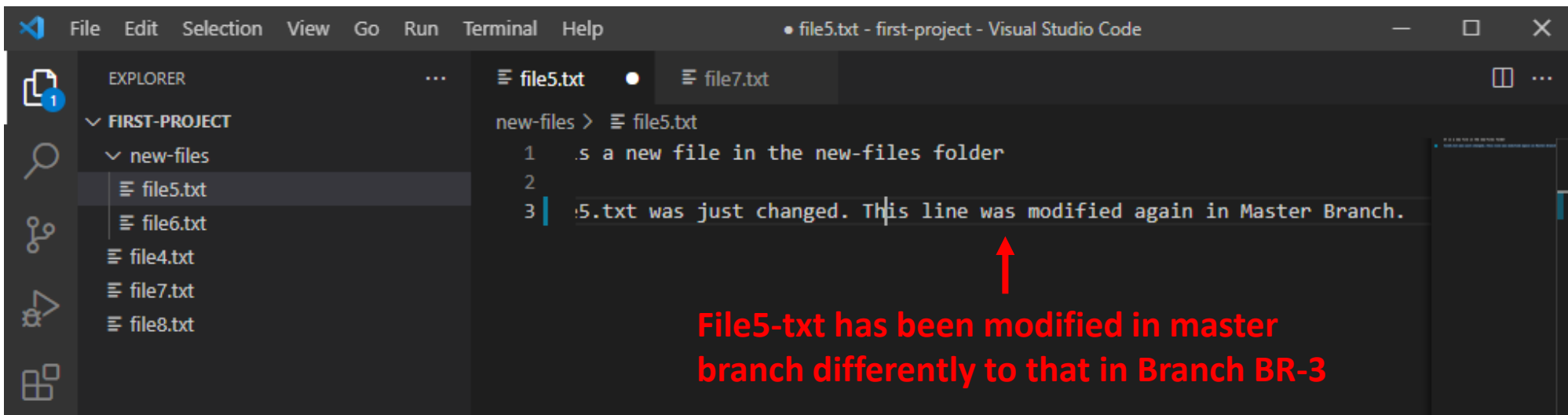
Modified file8.txt

file8.txt

Hunk 1: Lines 1-3

```
1 - - This file was added in the master branch
2 - \ No newline at end of file
3 + + This file was added in the master branch
4 + +
5 + + Changed file8.txt
6 - \ No newline at end of file
```





Git log shows the this commit to the master branch.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git log
commit 54e9e32d00e742941f07584333c2ff204a8ad646 (HEAD -> master)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Tue Mar 1 21:07:44 2022 +0100

    file7.txt and file5.txt modified in master branch

commit 1c25cd31a73d58e8ef6fc39e0f20012ccdecd1b8
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Tue Mar 1 00:30:33 2022 +0100

    Modified file8.txt

commit ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc
Merge: fcf077 d46ea50
Author: Elliott Farmer <myemail@provider.com>
Date: Mon Feb 28 22:40:44 2022 +0100
```

However if we look in the BR-3 branch we do not see this commit.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git checkout BR-3
Switched to branch 'BR-3'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-3)
$ git log
commit 143b205a729e3e7aa62bc26862e856e52657b321 (HEAD -> BR-3)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Tue Mar 1 20:52:10 2022 +0100

    Changes made in BR-3 branch

commit 1c25cd31a73d58e8ef6fc39e0f20012ccdecd1b8
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Tue Mar 1 00:30:33 2022 +0100

    Modified file8.txt

commit ec4520f9a4ec9b4b012d6233511f9d6e62c3fdcc
```

We want to merge BR-3 into master branch so we first checkout the receiving branch, master branch.

Then we git merge, however, it throws a merge conflict error because the file5.txt and file7.txt are different in the two branches.

Git status shows we have unmerged branches and it lists the conflicting files.

We can also abort a merge with **git merge --abort**.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (BR-3)
$ git checkout master
Switched to branch 'master'

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git merge BR-3
Auto-merging file7.txt
CONFLICT (content): Merge conflict in file7.txt
Auto-merging new-files/file5.txt
CONFLICT (content): Merge conflict in new-files/file5.txt
Automatic merge failed; fix conflicts and then commit the result.

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```

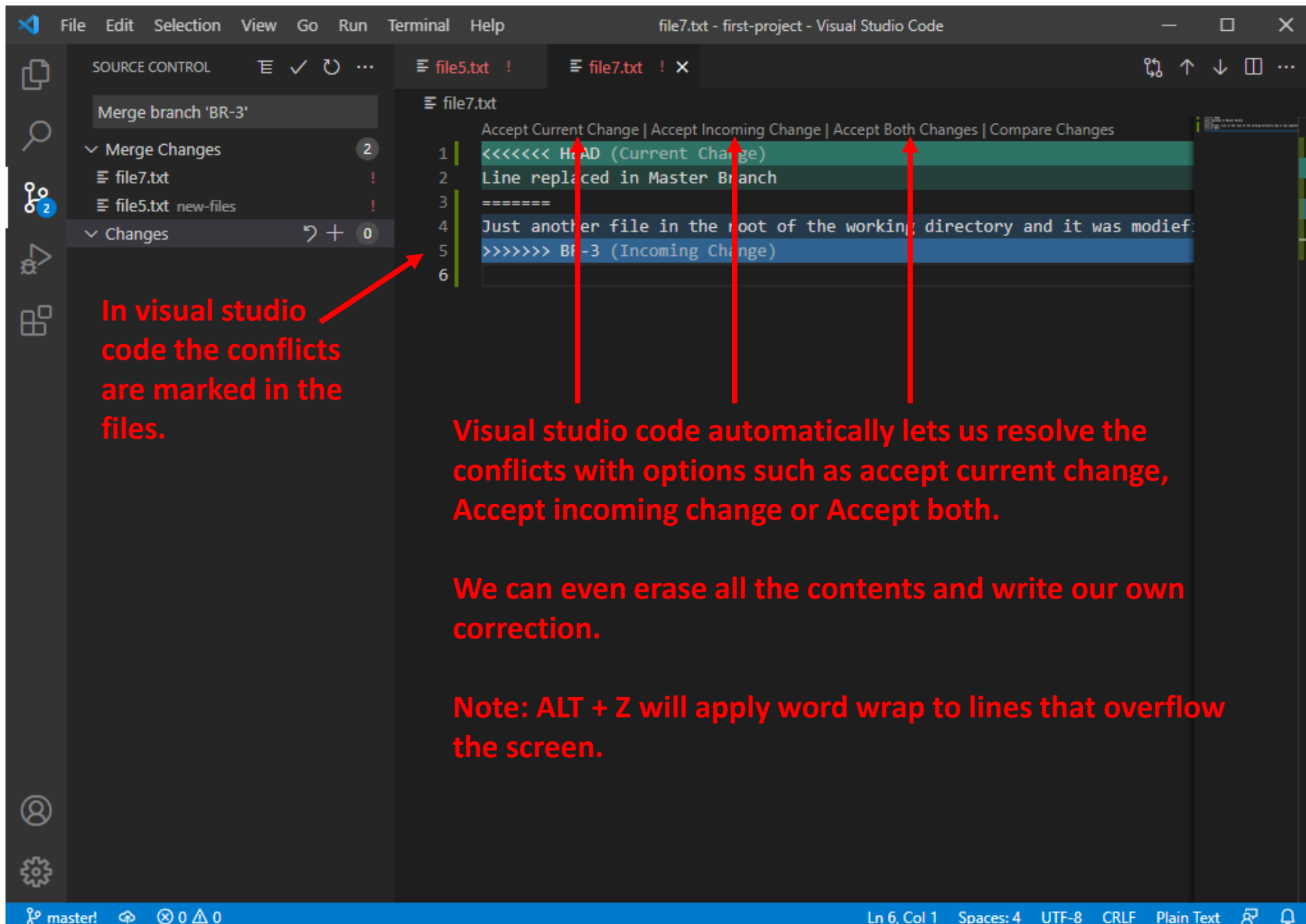
We need to fix the conflicts and commit results.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

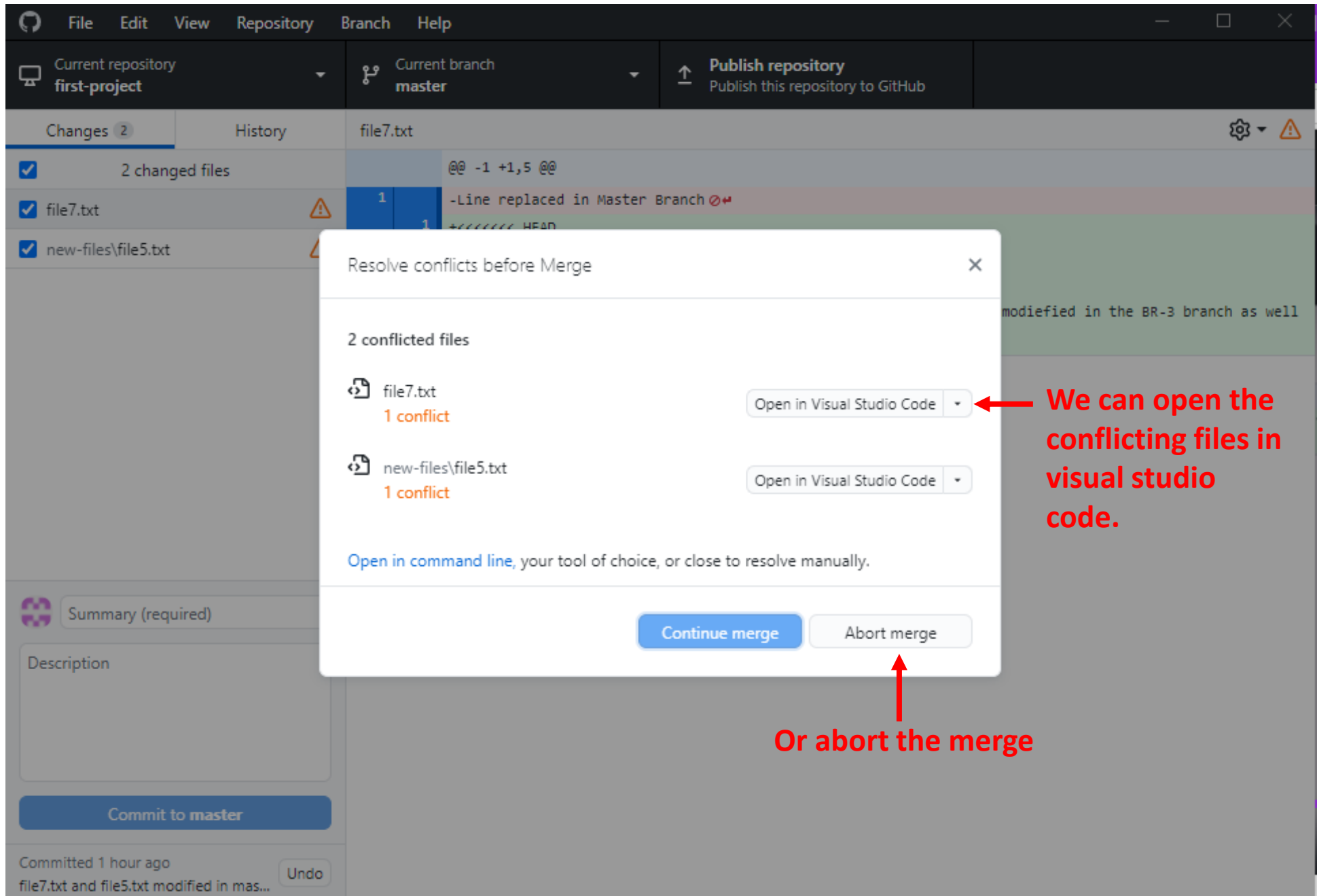
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   file7.txt
    both modified:   new-files/file5.txt

no changes added to commit (use "git add" and/or "git commit -a")

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```



Git hub desktop shows merge conflict changes as a pop up window



first-project

Commit Pull Push Fetch Branch Merge Stash Discard Tag

Git-flow Remote Terminal Explorer Settings

WORKSPACE

- File Status
- History
- Search

BRANCHES

- BR-3
- master

TAGS

REMOTES

STASHES

Pending files, sorted by file status

Staged files

- file7.txt
- new-files/file5.txt

Unstaged files

- file7.txt
- new-files/file5.txt

file7.txt

Hunk 1 : Lines 1-5

Stage hunk Discard hunk

```
-- Line replaced in Master Branch
\..No newline at end of file
1 + <<<<<< HEAD
2 + Line replaced in Master Branch
3 + =====
4 + Just another file in the root of the working directory and it w
5 + >>>>>> BR-3
```

Elliott Farmer <BarcelonaTravelHacks@gmail.com>

Merge branch 'BR-3'

Conflicts:

- # file7.txt
- # new-files/file5.txt

☐ Push changes immediately to -

Commit

In SourceTree we see the same conflict and information as visual studio code

If we look at the staging area with a **git ls-files -s** command we see something strange.

There are three versions of each of the two conflicting files and they are numbered 1, 2 and 3.

We can look in each of these files with a **git cat-file -p <Sha1>** command

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file4.txt
100644 2107281b79ed03e0147ea53fb62eb5f9a47aba2a 1      file7.txt
100644 391a9560c041b7388d7a3388e1fe0de75949bc24 2      file7.txt
100644 1590429c15eea70a9090e20b5a0ba3f2b1fa9d4c 3      file7.txt
100644 a529592a90242ad5002712bd5ba3b7ae97496666 0      file8.txt
100644 ac30be19d89751ac23fc50287f755a4069550daa 1      new-files/file5.txt
100644 bed5481a952a2d516df24ae29f8b7d4fc839fefa 2      new-files/file5.txt
100644 6062717e0b0039e54993e1cche0e13d7e463b84f 3      new-files/file5.txt
100644 cf9ca3f4f20ab449121b916d6387328a789df46f 0      new-files/file6.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git cat-file -p 2107281
Just another file in the root of the working directory
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git cat-file -p 391a956
Line replaced in Master Branch
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git cat-file -p 1590429c
Just another file in the root of the working directory and it was modiefied in the BR-3 branch as
well
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```

The file labelled as 1 is the common version, i.e. the ancestor version that is common in both branches. The file labelled as 2 is the version from the master branch. The file labelled as 3 is the version that has been modified in the BR-3 Branch.

If we check the conflicting file in the working directory we get the same output highlighting the conflicts as in visual studio code.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ cat file7.txt
<<<<<<< HEAD
Line replaced in Master Branch
=====
Just another file in the root of the working directory and it was modiefied in the BR-3 branch as
well
>>>>>>> BR-3
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```

Cat file7.txt shows us the conflicts in this file in the working directory.

We need to manually resolve them. Option is to use nano editor.

We need to decide which line to keep. In this example we will keep the top line, the version from the master branch.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
ello@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ cat file7.txt
<<<<<<< HEAD
Line replaced in Master Branch
=====
Just another file in the root of the working directory and it was modiefied in the BR-3 branch as
well
>>>>>>> BR-3
```

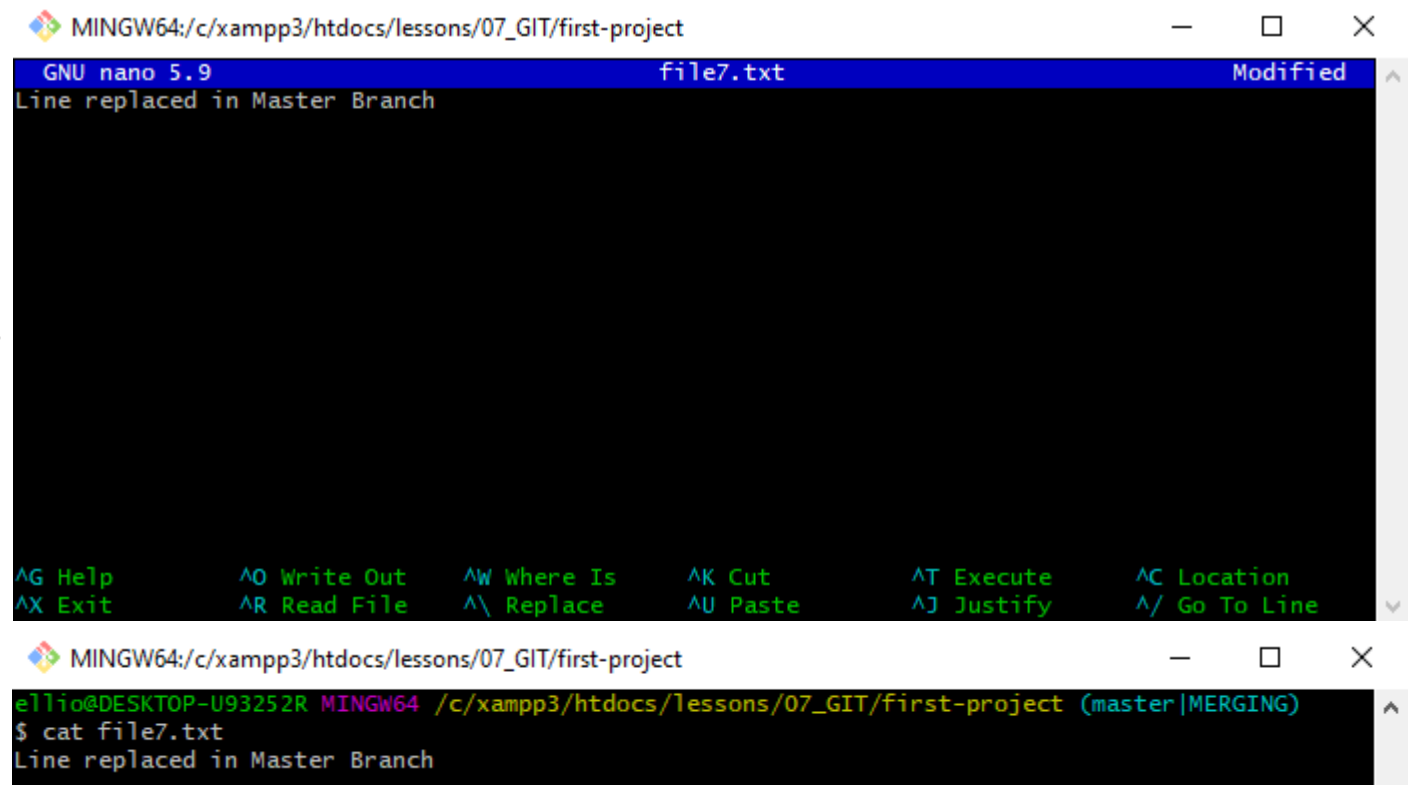
```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
GNU nano 5.9 file7.txt
<<<<<<< HEAD
Line replaced in Master Branch
=====
Just another file in the root of the working directory and it was modiefied in the BR-3 branch as
>>>>>>> BR-3

[ Read 5 lines (Converted from DOS format) ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

With the cursor at the beginning of the line we want to remove control + K will cut that line. I have repeated it for all the other undesired lines.

Control s and Y to save and exit nano editor.

Cat file7.txt shows no conflicts now in the working directory.

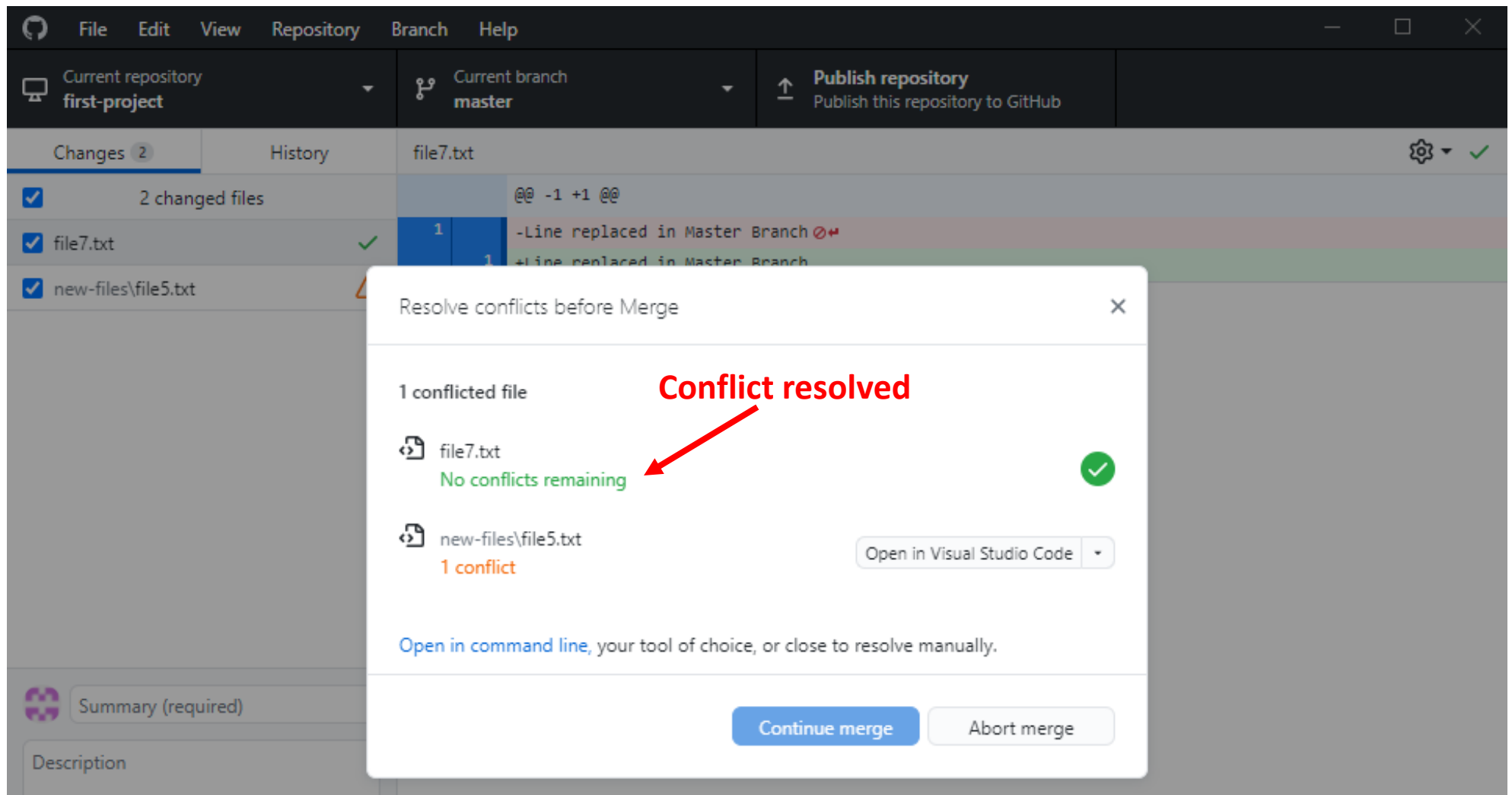


The image shows two screenshots of a terminal window. The top screenshot shows the nano editor interface with the file 'file7.txt' open. The status bar at the bottom indicates 'Line replaced in Master Branch'. The bottom screenshot shows the terminal prompt 'e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)' followed by the command '\$ cat file7.txt' and the output 'Line replaced in Master Branch'.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project
GNU nano 5.9 file7.txt Modified
Line replaced in Master Branch

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ cat file7.txt
Line replaced in Master Branch
```

Git status still shows both files as conflicting but if we add the recently corrected file7.txt then it shows as modified in git status.

Now in the staging area we do not see three versions of file7.txt, only the one because the conflict has been resolved.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/first-project

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   file7.txt
    both modified:   new-files/file5.txt

no changes added to commit (use "git add" and/or "git commit -a")

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git add file7.txt

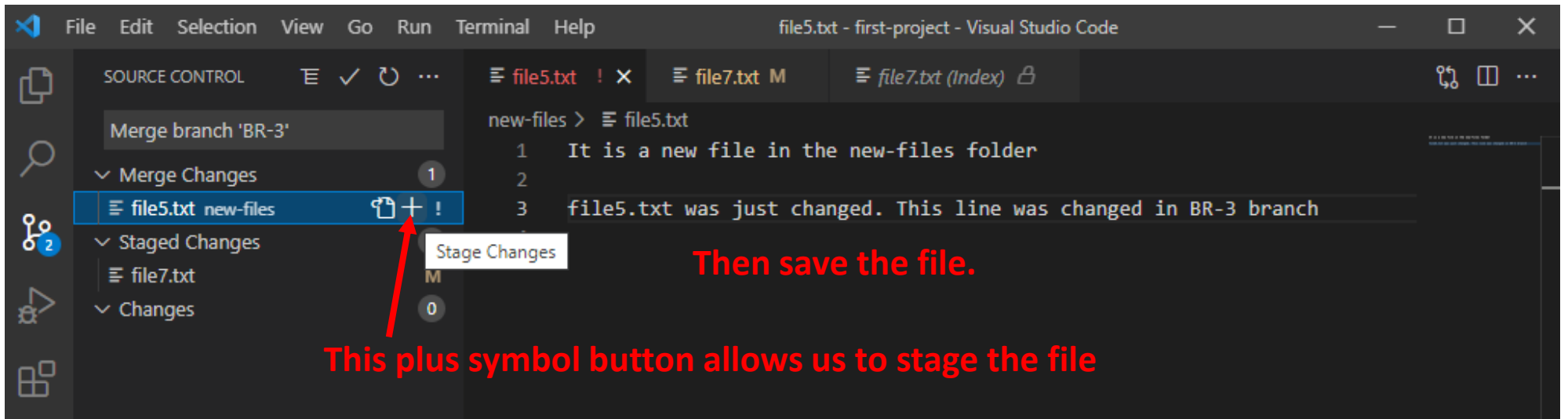
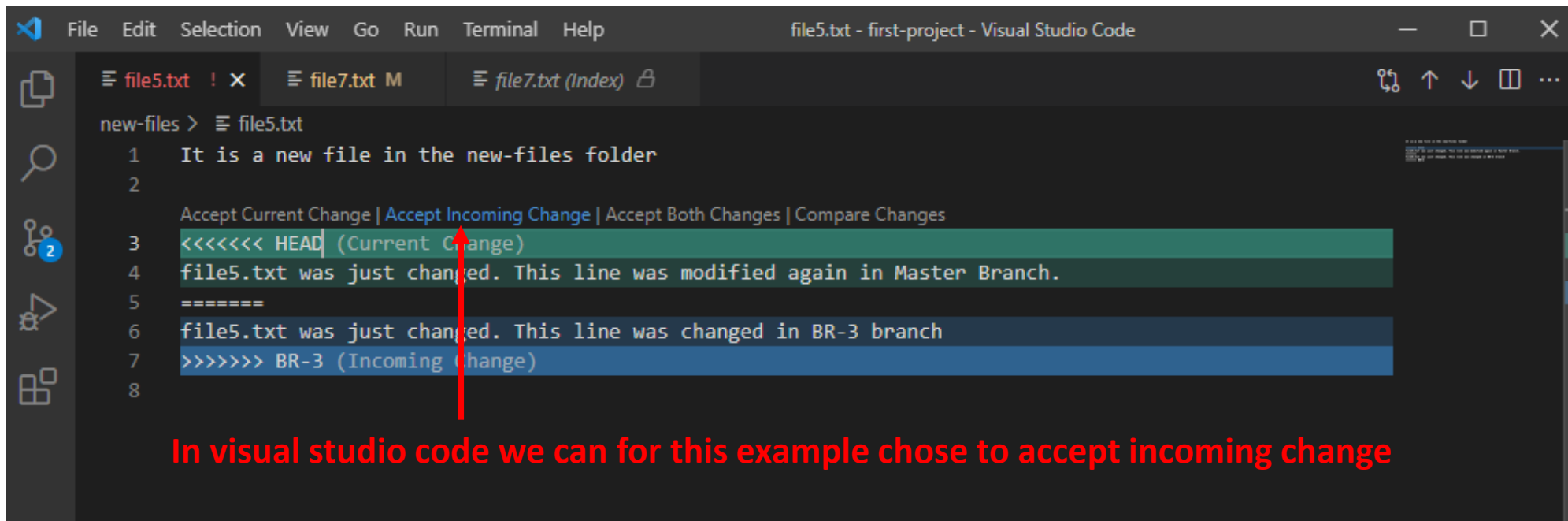
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   file7.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   new-files/file5.txt

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file4.txt
100644 b054618e0d78b2f78976222d748e3b059ed3baeb 0      file7.txt
100644 a529592a90242ad5002712bd5ba3b7ae97496666 0      file8.txt
100644 ac30be19d89751ac23fc50287f755a4069550daa 1      new-files/file5.txt
100644 bed5481a952a2d516df24ae29f8b7d4fc839fefaf 2      new-files/file5.txt
100644 6062717e0b0039e54993e1ccbe0e13d7e463b84f 3      new-files/file5.txt
100644 cf9ca3f4f20ab449121b916d6387328a789df46f 0      new-files/file6.txt

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```



The top command shows the three versions of file7.txt in the staging area before conflict resolution.

The bottom command shows only one version of file7.txt in the staging area after conflict resolution.

Now we can type git commit to merge branches.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/first-project

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file4.txt
100644 b054618e0d78b2f78976222d748e3b059ed3baeb 0      file7.txt
100644 a529592a90242ad5002712bd5ba3b7ae97496666 0      file8.txt
100644 ac30be19d89751ac23fc50287f755a4069550daa 1      new-files/file5.txt
100644 bed5481a952a2d516df24ae29f8b7d4fc839fefaf 2      new-files/file5.txt
100644 6062717e0b0039e54993e1ccbe0e13d7e463b84f 3      new-files/file5.txt
100644 cf9ca3f4f20ab449121b916d6387328a789df46f 0      new-files/file6.txt

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$ git ls-files -s
100644 b7aec520dec0a7516c18eb4c68b64ae1eb9b5a5e 0      file4.txt
100644 b054618e0d78b2f78976222d748e3b059ed3baeb 0      file7.txt
100644 a529592a90242ad5002712bd5ba3b7ae97496666 0      file8.txt
100644 321e737afbbba488051141287b6090383cabfd1bd 0      new-files/file5.txt
100644 cf9ca3f4f20ab449121b916d6387328a789df46f 0      new-files/file6.txt

elliott@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master|MERGING)
$
```

The screenshot shows the Visual Studio Code interface with the Git extension. The left sidebar shows the 'History' view. The main area displays a commit history table. A red arrow points to the 'master' branch in the 'All Branches' dropdown. Another red arrow points to the 'Merge branch 'BR-3'' commit in the history table. A red text box with an arrow points to the two commits immediately preceding the merge.

Graph	Description	Date	Author	Commit
● master	Merge branch 'BR-3'	1 Mar 2022 23:44	Elliott Farmer <Ba	9f516a5
●	file7.txt and file5.txt modified in master branch	1 Mar 2022 21:07	Elliott Farmer <Bar	54e9e32
●	Changes made in BR-3 branch	1 Mar 2022 20:52	Elliott Farmer <Bar	143b205
●	Modified file8.txt	1 Mar 2022 0:30	Elliott Farmer <Bar	1c25cd3
●	merge branch BR-2 into master branch	28 Feb 2022 22:40	Elliott Farmer <my	ec4520f
●	Create file8.txt	28 Feb 2022 22:21	Elliott Farmer <my	fcfd077
●	changed file5.txt	28 Feb 2022 21:50	Elliott Farmer <my	d46ea50
●	3 new files were created in the BR-2 Branch	28 Feb 2022 21:39	Elliott Farmer <my	f6695eb
●	First commit in the BR-1 branch	25 Feb 2022 23:18	Elliott Farmer <my	8fa0cdb
●	Third commit	25 Feb 2022 2:48	Elliott Farmer <my	f3f965f
●	Second commit	24 Feb 2022 23:00	Elliott Farmer <my	fd127e1
●	Our very first commit in the project	24 Feb 2022 21:04	Elliott Farmer <my	25de26c

Summary of Shell Commands

git merge <feature-branch>

Will merge feature branch into master branch.

git merge <branch name> -m <comment>

Will merge branch into current branch with comment

git merge --abort

Abort merging of branches

GIT HUB REMOTE REPOSITORIES



Git

vs

GitHub



Distributed version-control
system

Repository hosting
service

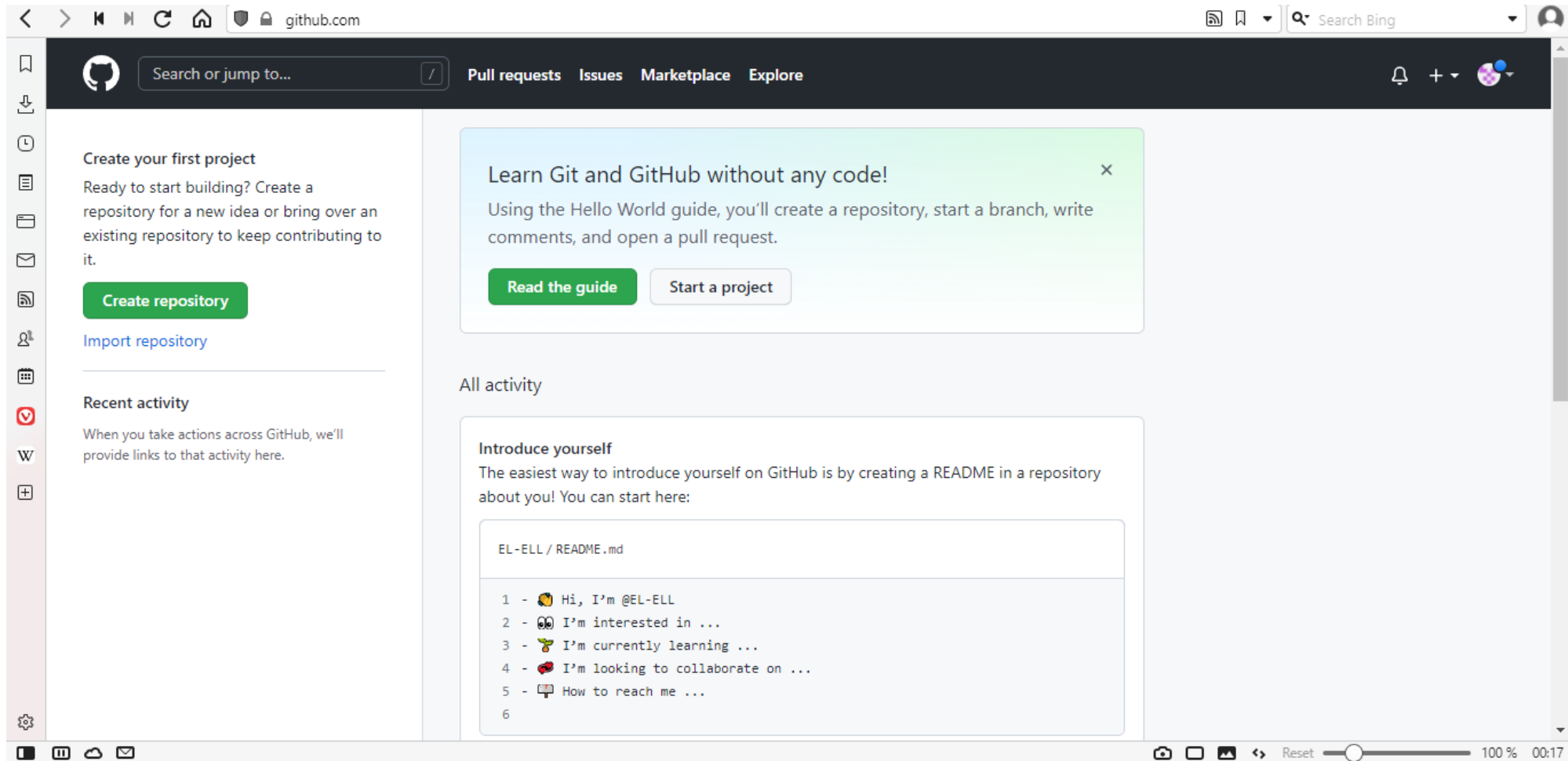
These reside locally on computer.

This is a remote repository for sharing the repository for collaboration. There are various settings such as public or private.

This is also like a backup of local repository and can be cloned in case of loss or corruption of local repository.

Various settings such as private or public or specified users

At <https://www.github.com> create an account by following the onscreen prompts. Once done landing page should look something like this.



In git hub the master branch has now been renamed to main branch.

The reason for this is after the black lives protests, terms such as master, slave, blacklist, whitelist were removed from the vernacular of the tool.

This may cause some issues with integrating project from local repository to remote repository.

For more information see: <https://www.youtube.com/watch?v=hZznWbEGv1U>

github.com/new

Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

EL-ELL / my-first-git-hub-repository ✓

Great repository names are short and memorable. Need inspiration? [How about supreme-winner?](#)

Description (optional)

Here is my first GitHub repository. Let's get started with GitHub.

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

80 % 00:50

Convention is when naming a repository that it contains lowercase letters with hyphens between words.

**Equally valid would be MyFirstGitHubRepository
Spaces in repository names are not permitted.**

Public means that anyone can download and view the repository.

Private means that only I can see it and people who I grant permission to.

README file is advised for a repository and allows cloning to local machine

gitignore – covered in future lesson

License – this is regarding usage rights and reproduction of the contents of a repository and is recommended for public repositories

CREATE REPOSITORY button at bottom of screen

Once created the URL shows a permanent link to this repository. This link can be used for sharing with anybody in the world.



The screenshot shows a web browser displaying the GitHub repository page for 'EL-ELL/my-first-git-hub-repository'. The URL bar shows 'https://github.com/EL-ELL/my-first-git-hub-repository'. The repository is private and has 1 branch (main) and 0 tags. The commit history shows 'EL-ELL Initial commit' with a SHA of 'd67d9ce' and '1 commit'. The README.md file is listed with the text 'my-first-git-hub-repository' and 'Here is my first GitHub repository. Let's get started with GitHub.' The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. The bottom of the page shows the GitHub footer with copyright information and various links.

The repository only has one commit. We can click on this to see more information



This page shows the working directory of the repository. At the moment it only has the one README file and one commit.

This panel shows contents of README.md file and can be clicked on the be edited.

Clicking on the commit the URL changes. Here we see that it is in the commit/main



The screenshot shows the GitHub interface for the repository 'EL-ELL / my-first-git-hub-repository'. The browser address bar displays the URL 'github.com/EL-ELL/my-first-git-hub-repository/commits/main'. The repository is marked as 'Private'. The 'main' branch is selected. The commit history shows one commit: 'Initial commit' by 'EL-ELL' committed 9 minutes ago. A tooltip is visible over the 'Verified' button, stating: 'This commit was created on GitHub.com and signed with GitHub's verified signature. GPG key ID: 4AEE18F83AFDEB23. Learn about vigilant mode.' The footer shows the GitHub logo, copyright notice '© 2022 GitHub, Inc.', and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

github.com/EL-ELL/my-first-git-hub-repository/commits/main

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

main

Commits on Mar 2, 2022

Initial commit
EL-ELL committed 9 minutes ago

Verified d67d9ce <>

Newer Older

This commit was created on GitHub.com and signed with GitHub's verified signature.
GPG key ID: 4AEE18F83AFDEB23
Learn about vigilant mode.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

<> Button shows
Verified SHA1 commit history

< > ⏮ ⏭ ↺ 🏠 🔒 github.com/EL-ELL/my-first-git-hub-repository/edit/main/README.md 🔖 🔍 Search Bing

🐙 Search or jump to... Pull requests Issues Marketplace Explore 🔔 + 👤

🔒 EL-ELL / my-first-git-hub-repository Private Unwatch 1 Fork 0 Star 0

<> Code ⌚ Issues 🔗 Pull requests ⚙️ Actions 📁 Projects 🛡️ Security 📊 Insights ⚙️ Settings


my-first-git-hub-repository / README.md in main Cancel changes

<> Edit file Preview Spaces 2 Soft wrap

```
1 # my-first-git-hub-repository
2 Here is my first GitHub repository. Let's get started with GitHub.
3
4 README.md file was just eddited manually
```

Line of text added.

Attach files by dragging & dropping, selecting or pasting them.

 Commit changes

Update README.md Default commit comment

Add an optional extended description... Or manually add commit text

☒ Commit directly to the main branch.
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

🐙 © 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

github.com/EL-ELL/my-first-git-hub-repository/commits/main

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

main

Commits on Mar 2, 2022

Update README.md
EL-ELL committed 22 seconds ago Verified bddce3f

Initial commit
EL-ELL committed 20 minutes ago Verified d67d9ce

Newer Older

Note that these commits are direct to remote repository.

Now we have two commits

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

github.com/EL-ELL/my-first-git-hub-repository/commit/bddce3f409d57ad465027ce5c33bc67ba8e30f08

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

Update README.md
main
EL-ELL committed 3 minutes ago Verified 1 parent d67d9ce commit bddce3f409d57ad465027ce5c33bc67ba8e30f08 Browse files

Showing 1 changed file with 2 additions and 0 deletions. Split Unified

2 README.md

```
... @@ -1,2 +1,4 @@
1 1 # my-first-git-hub-repository
2 2 Here is my first GitHub repository. Let's get started with GitHub.
3 +
4 + README.md file was just edited manually
```

0 comments on commit bddce3f Lock conversation

Detailed commit information

I have added a new file called hello-github.txt and committed it to the repository using default commit comment.

The screenshot shows the GitHub interface for a repository named 'my-first-git-hub-repository' by user 'EL-ELL'. The repository is private and has 1 branch (main) and 0 tags. The commit history table shows three commits: 'EL-ELL Create hello-github.txt' (185aed4, now, 3 commits), 'Update README.md' (10 minutes ago), and 'Create hello-github.txt' (now). The README file is displayed below the table, containing the text 'my-first-git-hub-repository', 'Here is my first GitHub repository. Let's get started with GitHub.', and 'README.md file was just edited manually'. The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. The bottom of the page features a footer with GitHub's terms, privacy, and security information, as well as a Windows activation watermark.

github.com/EL-ELL/my-first-git-hub-repository/tree/main

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

EL-ELL Create hello-github.txt	185aed4 now	3 commits
README.md	Update README.md	10 minutes ago
hello-github.txt	Create hello-github.txt	now

README.md

my-first-git-hub-repository

Here is my first GitHub repository. Let's get started with GitHub.

README.md file was just edited manually

About

Here is my first GitHub repository. Let's get started with GitHub.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

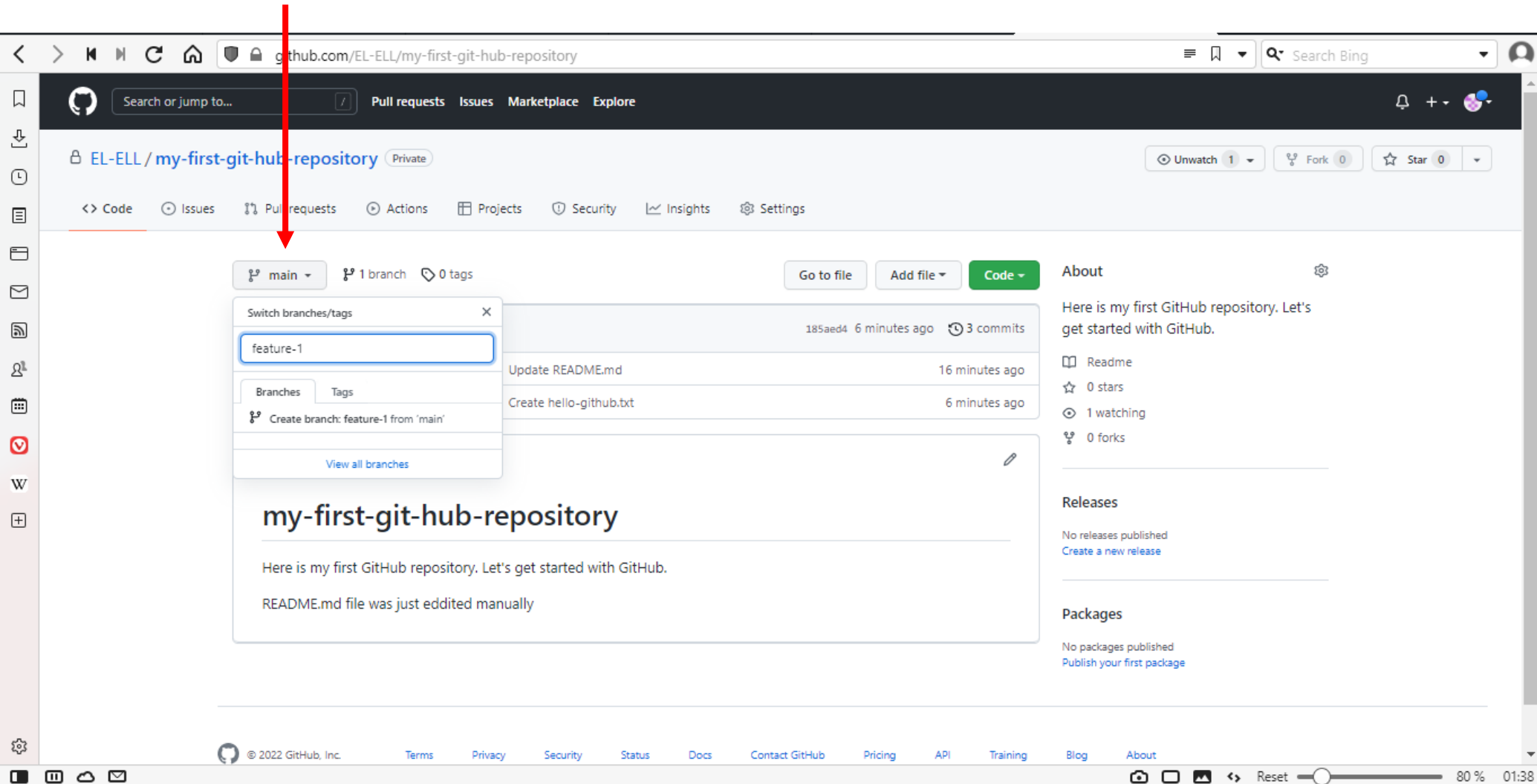
Activate Windows

Go to Settings to activate Windows.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Reset 80 % 01:32

Clicking this button will allow us to create another branch. When we click on create branch.... It automatically checks out this branch.



The screenshot shows the GitHub interface for a repository named 'my-first-git-hub-repository'. A red arrow points to the 'main' branch dropdown menu, which is open, showing the option to 'Create branch: feature-1 from 'main''. The repository page includes a search bar, navigation links (Pull requests, Issues, Marketplace, Explore), and a sidebar with repository details (About, Releases, Packages). The main content area displays the repository name, a description, and a list of commits.

github.com/EL-ELL/my-first-git-hub-repository

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

Switch branches/tags

feature-1

Branches Tags

Create branch: feature-1 from 'main'

View all branches

185aed4 6 minutes ago 3 commits

Update README.md 16 minutes ago

Create hello-github.txt 6 minutes ago

my-first-git-hub-repository

Here is my first GitHub repository. Let's get started with GitHub.

README.md file was just eddited manually

About

Here is my first GitHub repository. Let's get started with GitHub.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Reset 80 % 01:38

The branches button shows us this window where we can see an overview of branches, all branches created by me, Active branches which contain contributions by other users, Stale branches which have not been worked on for a long time and all branches.

The screenshot displays the GitHub interface for a repository named 'my-first-git-hub-repository'. The 'branches' tab is selected, showing a table with columns: Overview, Yours, Active, Stale, and All branches. The 'Overview' tab is active, displaying a search bar for branches. Below the search bar, there are three sections: 'Default branch' (main), 'Your branches' (feature-1), and 'Active branches' (feature-1). Each section shows the branch name, update time, and a 'New pull request' button.

Overview	Yours	Active	Stale	All branches
Default branch				
main Updated 11 minutes ago by EL-ELL Default				
Your branches				
feature-1 Updated 11 minutes ago by EL-ELL 0 0 New pull request				
Active branches				
feature-1 Updated 11 minutes ago by EL-ELL 0 0 New pull request				

We can create a new file in the feature-1 branch. If we add a slash after the file name the GitHub will make it a folder and after that we can enter the filename.

File Edit View Bookmarks Tools Window Help New File

YouTube (8 unread) - elliot_w_farme Yahoo UK | News, email and Facebook The Complete Git Guide: U New File

github.com/EL-ELL/my-first-git-hub-repository/new/feature-1 Search Bing

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

my-first-git-hub-repository / feature-1 / list.txt in feature-1 Cancel changes

<> Edit new file Preview Spaces 2 No wrap

```
1 1. one
2 2. two
3 3. three
```

Commit new file

Create list.txt

Add an optional extended description...

☒ Commit directly to the feature-1 branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

At the bottom of the page we can commit this with custom comment. Note how it says commit directly to feature-1 branch.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About Activate Windows Go to Settings to activate Windows.

In feature-1 branch we can see the new folder that was created with the commit comment. Note the yellow box notifying us of differences between branches.

github.com/EL-ELL/my-first-git-hub-repository/tree/feature-1

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

feature-1 had recent pushes 1 minute ago [Compare & pull request](#)

feature-1 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

This branch is 1 commit ahead of main. [Contribute](#)

EL-ELL	file list.txt was created in the feature-1 branch	7422106 1 minute ago 4 commits
feature	file list.txt was created in the feature-1 branch	1 minute ago
README.md	Update README.md	31 minutes ago
hello-github.txt	Create hello-github.txt	22 minutes ago

README.md

my-first-git-hub-repository

Here is my first GitHub repository. Let's get started with GitHub.

About

Here is my first GitHub repository. Let's get started with GitHub.

Readme 0 stars 1 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Activate Windows
Go to Settings to activate Windows.

Reset 80 % 01:53

WE can find the URL of this repository under the green code button and copy it.

The screenshot shows a web browser displaying the GitHub repository page for `EL-ELL / my-first-git-hub-repository`. The repository is marked as `Private`. The top navigation bar includes links for `Pull requests`, `Issues`, `Marketplace`, and `Explore`. Below the repository name, there are buttons for `Unwatch` (1), `Fork` (0), and `Star` (0). The main navigation bar shows tabs for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Security`, `Insights`, and `Settings`. A yellow banner at the top of the repository content area states: `feature-1` had recent pushes 6 minutes ago, with a `Compare & pull request` button. Below this, there are buttons for `Go to file`, `Add file`, and `Code`. The `Code` button is open, showing a dropdown menu with options: `Clone`, `Open with GitHub Desktop`, and `Download ZIP`. Under the `Clone` option, there are three methods: `HTTPS`, `SSH`, and `GitHub CLI` (marked as `New`). The `HTTPS` method is selected, and the URL `https://github.com/EL-ELL/my-first-git-hu` is highlighted in a blue box. Below the URL, it says `Use Git or checkout with SVN using the web URL.` The repository content area shows a list of files: `README.md` (Update README.md) and `hello-github.txt` (Create hello-github.txt). Below the file list, there is a section for `README.md` with the title `my-first-git-hub-repository` and the text: `Here is my first GitHub repository. Let's get started with GitHub.` and `README.md file was just edited manually`. On the right side of the page, there is an `About` section with a gear icon, stating: `Here is my first GitHub repository. Let's get started with GitHub.` Below this, there are statistics: `Readme`, `0 stars`, `1 watching`, and `0 forks`. There is also a `Releases` section stating `No releases published` with a link `Create a new release`, and a `Packages` section stating `No packages published` with a link `Publish your first package`. The bottom of the page shows a taskbar with various icons and the time `11:58`.

We can check the contents of our folder of git repositories.

With the URL of the repository we can clone it to our local machine with **git clone** command.

And now we see it in the directory of our local machine.

In that directory we see the contents of the repository.

Note how we see hidden folder `.git` and that the prompt says (main) so we know that the directory is under git control.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ ls -l
total 8
drwxr-xr-x 1 ellio 197609 0 Mar  1 21:14 first-project/
drwxr-xr-x 1 ellio 197609 0 Feb 27 22:18 hello-world/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-ELL/my-first-git-hub-repository.git
Cloning into 'my-first-git-hub-repository'...
EL_remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 13 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (1/1), done.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ ls -l
total 8
drwxr-xr-x 1 ellio 197609 0 Mar  1 21:14 first-project/
drwxr-xr-x 1 ellio 197609 0 Feb 27 22:18 hello-world/
drwxr-xr-x 1 ellio 197609 0 Mar  2 02:02 my-first-git-hub-repository/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd my-first-git-hub-repository/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ ls -la
total 10
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:02 ./
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:01 ../
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:02 .git/
-rw-r--r-- 1 ellio 197609 143 Mar  2 02:02 README.md
-rw-r--r-- 1 ellio 197609  14 Mar  2 02:02 hello-github.txt

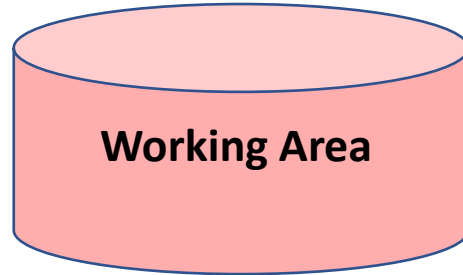
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

Remote repository and local repository have similar structure.

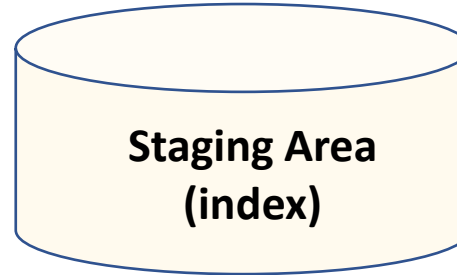
Does Local repository automatically update to remote repository - no we need to **push**.

Does remote repository automatically update to local repository – no we have to **pull**.

Remote Repository



Working Area



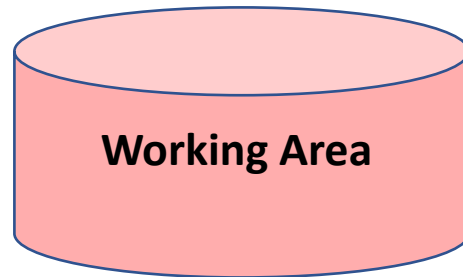
Staging Area
(index)



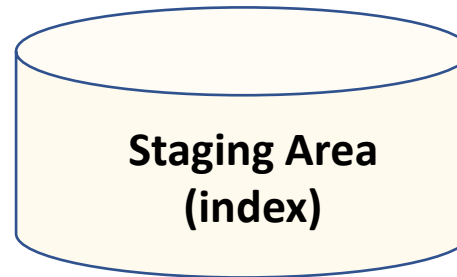
Git Repository



Local Repository



Working Area



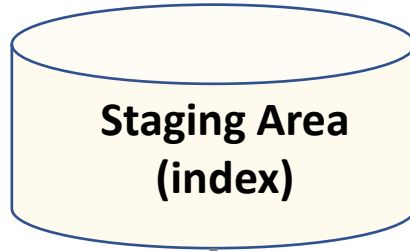
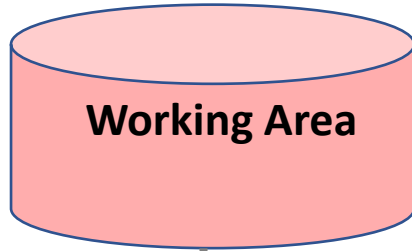
Staging Area
(index)



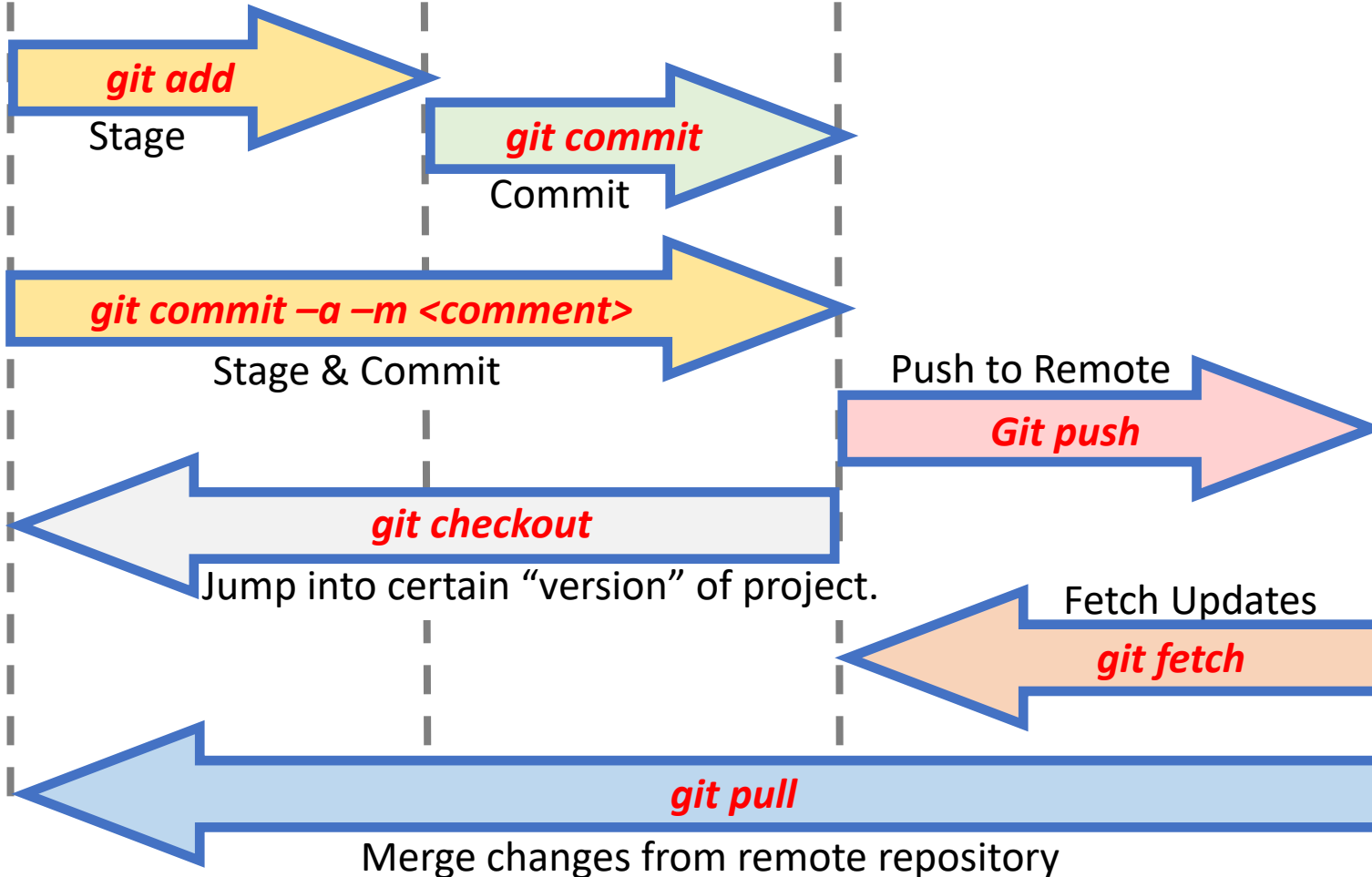
Git Repository

**GIT PUSH, PULL AND
FETCH**

Local Repository

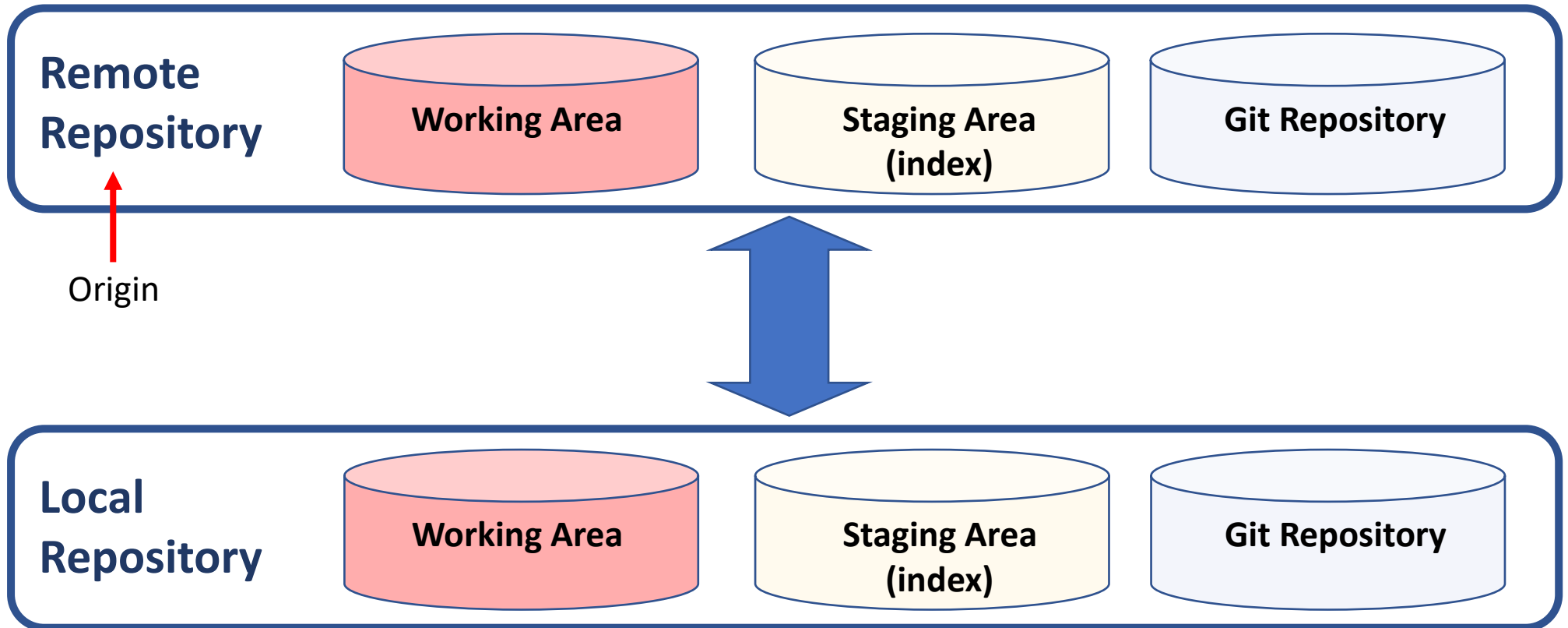


Remote Repository



What is Origin?

When you clone remote repository to local repository git automatically creates a binding between remote and local repositories. The default name of Remote repository is **Origin**. A local repository can be connected to multiple remote repositories.



git remote command shows a remote repository of origin.

git remote -v command shows the remote repository URL.

First-project repository does not have a remote repository.

hello-world repository has a remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/hello-world

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git remote
origin

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git remote -v
origin https://github.com/EL-ELL/my-first-git-hub-repository.git (fetch)
origin https://github.com/EL-ELL/my-first-git-hub-repository.git (push)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ cd ..

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd first-project/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git remote

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ git remote -v

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/first-project (master)
$ cd ..

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd hello-world/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git remote
origin

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$ git remote -v
origin https://github.com/zzl686/hello-world.git (fetch)
origin https://github.com/zzl686/hello-world.git (push)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/hello-world (master)
$
```

In my-first-git-hub-repository, locally we only have one branch, main.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd my-first-git-hub-repository/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch
* main

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

In remote repository there is a main branch and a feature-1 branch.

The screenshot shows the GitHub repository page for 'EL-ELL / my-first-git-hub-repository'. The repository is private. The 'Code' tab is selected. The 'Overview' tab is active, showing a table of branches. The 'main' branch is the default branch, updated 16 hours ago by EL-ELL. The 'feature-1' branch is also listed, updated 16 hours ago by EL-ELL, with 0 commits and 1 pull request. The 'feature-1' branch is also listed under 'Active branches'.

git branch command only shows local repository branches.

git branch -a command shows all branches. Notice how the remote repositories are in red with the first line showing which branch is the HEAD.

git branch -r command shows only remote repositories.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch
* main

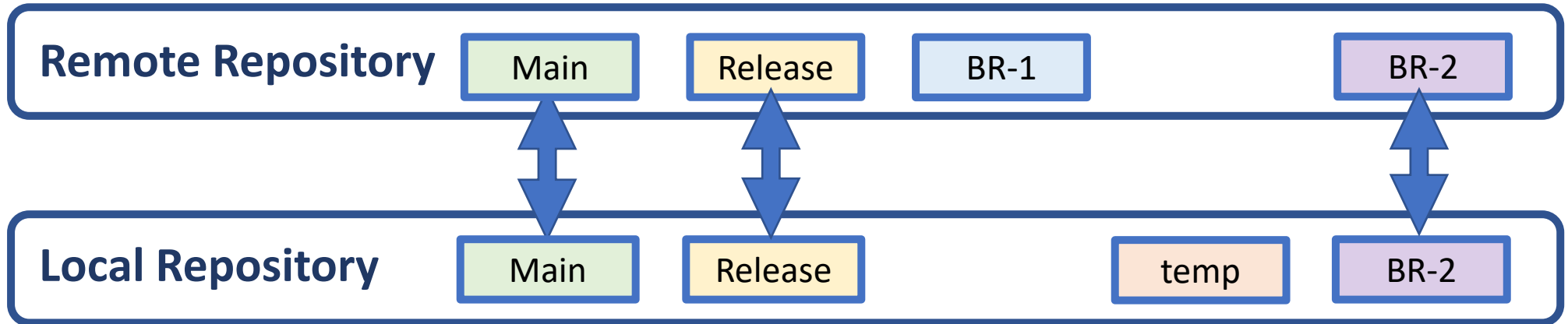
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -a
* main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -r
origin/HEAD -> origin/main
origin/feature-1
origin/main

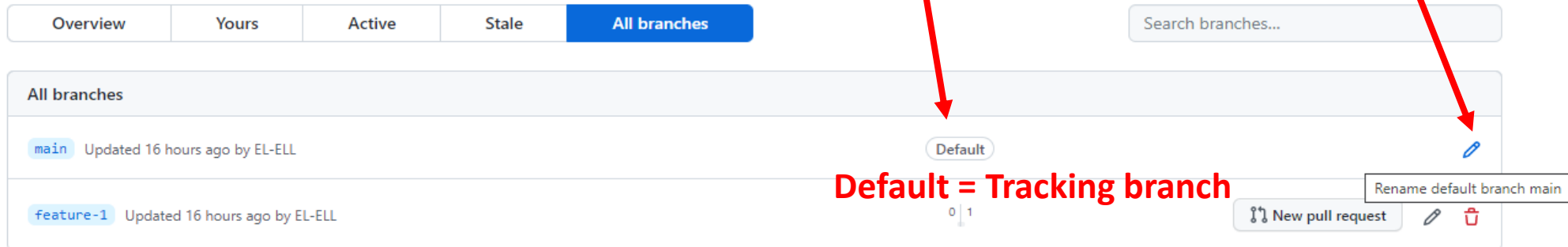
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

What is Tracking Branch?

Tracking branch is local repository branch that is linked to a remote repository branch. By default git creates tracking branch in remote repository which is always main branch.



We can see the tracking branch in git hub because it is labelled as default. We can change the tracking branch.



When remote repository is cloned Git will automatically checkout the default branch to the remote repository. By checking out remote repository any additional branches not present in local repository will also be cloned and track their corresponding branch in the remote repository.

git branch -vv

command shows which branches are tracked in the local repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -vv
* main 185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

At the moment only main branch is a tracking branch because it has label origin/main. It indicates SHA1 hash of the last commit it received from the remote repository as well as the commit comment.

git branch -a

command shows all local branches in the repositories.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/feature-1
  remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git checkout feature-1
Switched to a new branch 'feature-1'
branch 'feature-1' set up to track 'origin/feature-1'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -a
* feature-1
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/feature-1
  remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main 185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

git checkout <branch>

command or the feature-1 branch in the remote repository will check it out to the local repository.

git branch -a shows that feature-1 branch is now the head branch in local repository.

Git branch -vv shows that there are now two tracking branches.

git branch -vv command shows which branches are tracked in the local repository. →

Git will not allow us to delete feature-1 branch in local repository because it is the head branch. →

Lets make the main branch the head branch.

NOTE: normal behaviour of git does not allow us to delete unmerged branches (not merged to head) because we would loose files.

However, because feature-1 branch is tracked in the remote repository we can now delete in the local repository an unmerged branch.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -d feature-1
error: Cannot delete branch 'feature-1' checked out at 'C:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -d feature-1
warning: deleting branch 'feature-1' that has been merged to
        'refs/remotes/origin/feature-1', but not yet merged to HEAD.
Deleted branch feature-1 (was 7422106).

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ |
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch
* main

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -vv
* main 185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git checkout feature-1
Switched to a new branch 'feature-1'
branch 'feature-1' set up to track 'origin/feature-1'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

Feature-1 branch checked out from remote repository.

git remote show origin

Command gives much more detail about the remote repository connections.

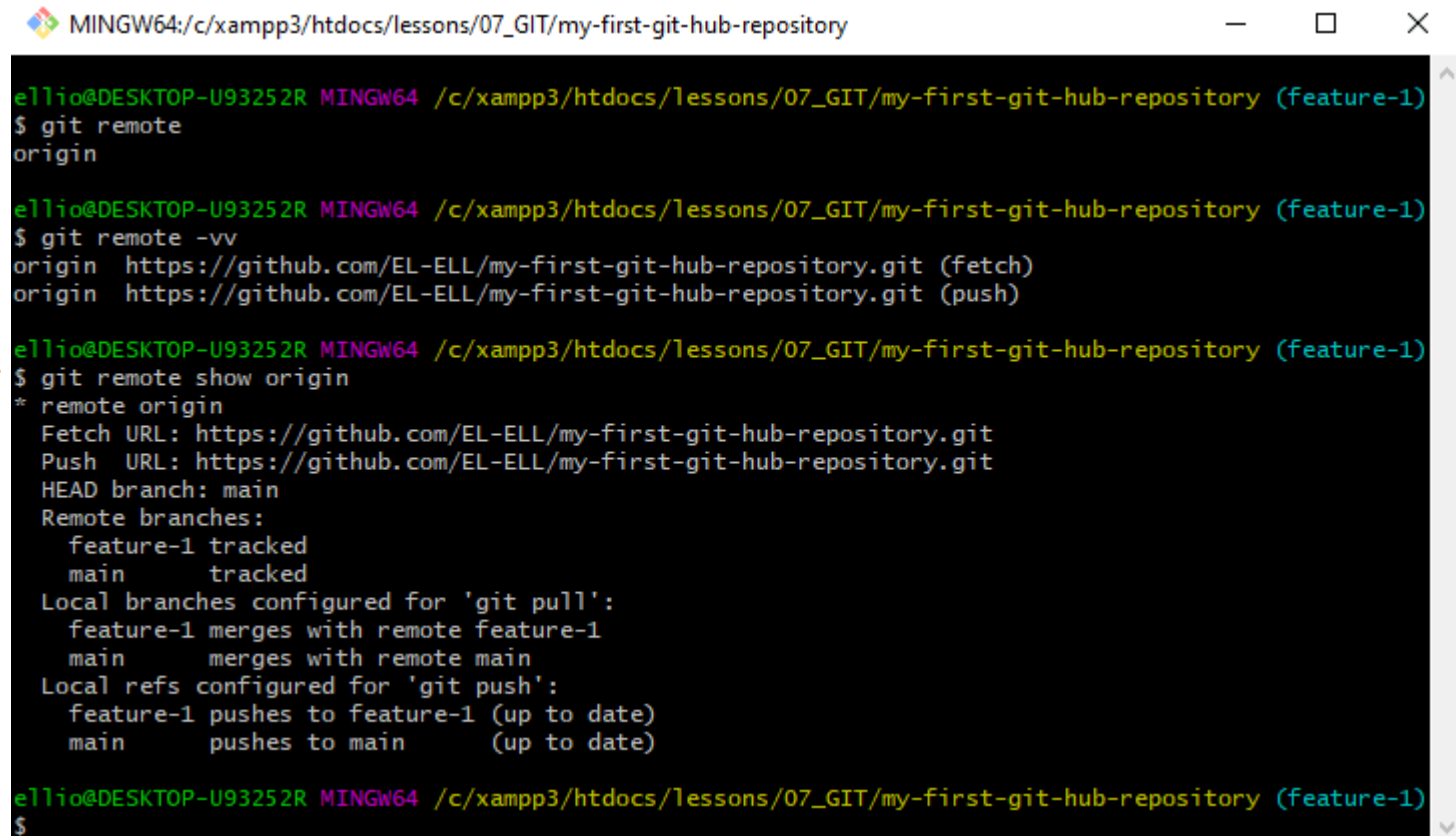
Fetch and push URLs

Head branch

Remote branches and tracked status

Local branches configured for pull

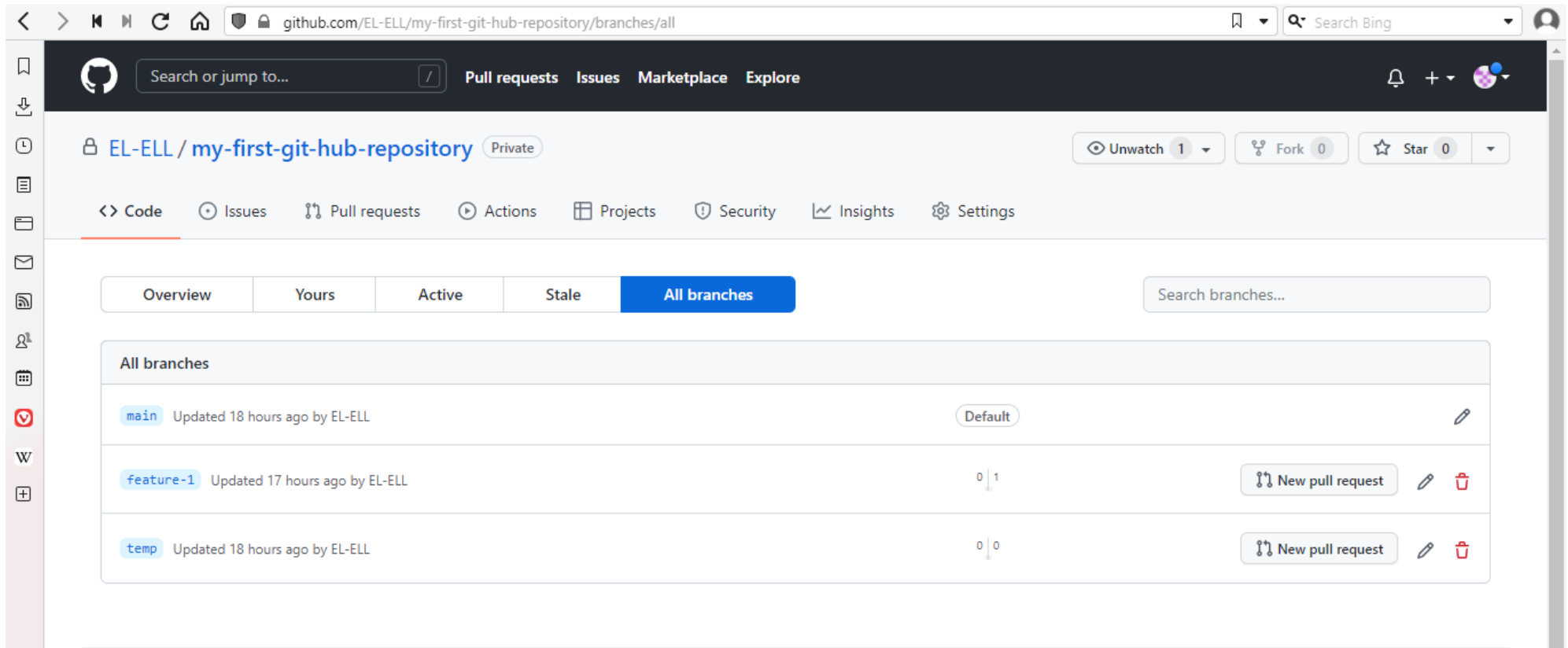
Local refs configured for push

A terminal window titled 'MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository' showing the output of the 'git remote show origin' command. The output details the fetch and push URLs, the head branch (main), and the status of remote branches (feature-1 and main) as tracked. It also shows the local branches configured for pull and push.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git remote
origin
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git remote -vv
origin https://github.com/EL-ELL/my-first-git-hub-repository.git (fetch)
origin https://github.com/EL-ELL/my-first-git-hub-repository.git (push)
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  Push URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  HEAD branch: main
  Remote branches:
    feature-1 tracked
    main tracked
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main merges with remote main
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main pushes to main (up to date)
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

Because both the main and feature-1 branches are tracked we can perform push and pull between remote and local repositories.

I have created a new branch in the remote repository called temp.



github.com/EL-ELL/my-first-git-hub-repository/branches/all

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

Overview Yours Active Stale All branches Search branches...

All branches

main	Updated 18 hours ago by EL-ELL	Default	
feature-1	Updated 17 hours ago by EL-ELL	0 1	New pull request
temp	Updated 18 hours ago by EL-ELL	0 0	New pull request

The newly create temp branch does not show in remote repository despite being created there.

git fetch command can be used to copy temp branch from remote repository to local repository.

But this new temp branch is still not showing?

Git branch -r now shows the new branch which can be checked out. Note that by default checking out a branch switches the HEAD to this branch.

We now see this branch and it is tracked.

git remote show origin Shows more detail about the three branches in the local and remote repositories.

```
MINGW64: /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -r
  origin/HEAD -> origin/main
  origin/feature-1
  origin/main

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git fetch
From https://github.com/EL-ELL/my-first-git-hub-repository
 * [new branch]      temp -> origin/temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -r
  origin/HEAD -> origin/main
  origin/feature-1
  origin/main
  origin/temp

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git checkout temp
Switched to a new branch 'temp'
branch 'temp' set up to track 'origin/temp'.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
  feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt
 * temp     185aed4 [origin/temp] Create hello-github.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
Push URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
HEAD branch: main
Remote branches:
  feature-1 tracked
  main      tracked
  temp      tracked
Local branches configured for 'git pull':
  feature-1 merges with remote feature-1
  main      merges with remote main
  temp      merges with remote temp
Local refs configured for 'git push':
  feature-1 pushes to feature-1 (up to date)
  main      pushes to main      (up to date)
  temp      pushes to temp      (up to date)

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$
```

github.com/EL-ELL/my-first-git-hub-repository/branches/all

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

Overview Yours Active Stale All branches Search branches...

All branches

main	Updated 18 hours ago by EL-ELL	Default	
feature-1	Updated 18 hours ago by EL-ELL	0 1	New pull request
temp	Updated 18 hours ago by EL-ELL	0 0	New pull request

Delete this branch

Can we delete temp branch in remote repository? – yes.

git remote show origin
command shows all
branches including temp
but now it is marked as
stale and advises us to
prune it.

git remote prune to
remove stale branches

After pruning the temp
branch is removed from
the remote list of
branches but is still
present in the local
repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  Push URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  HEAD branch: main
  Remote branches:
    feature-1          tracked
    main               tracked
    refs/remotes/origin/temp stale (use 'git remote prune' to remove)
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main      merges with remote main
    temp      merges with remote temp
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main      pushes to main      (up to date)

ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote prune origin
Pruning origin
URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
* [pruned] origin/temp

ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  Push URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  HEAD branch: main
  Remote branches:
    feature-1 tracked
    main      tracked
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main      merges with remote main
    temp      merges with remote temp
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main      pushes to main      (up to date)

ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ |
```

Git branch -vv shows that temp is still tracked.

Lets move the HEAD to the main branch in the local repository.

Now we can delete temp branch locally.

git branch -vv now shows that temp branch is not being tracked because it has been deleted locally.

More detail of tracked branches is shown in **git remote show origin** command output.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
  feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt
  * temp     185aed4 [origin/temp: gone] Create hello-github.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

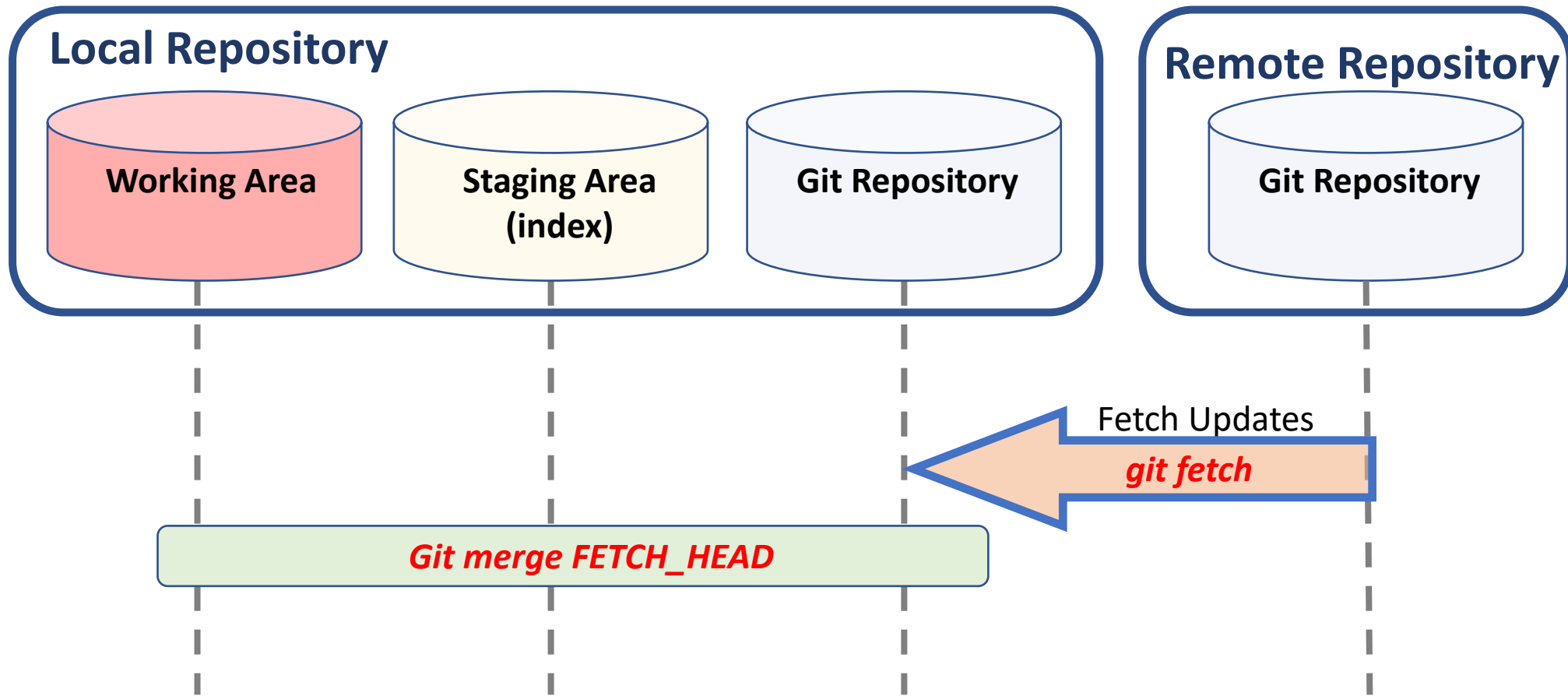
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -d temp
Deleted branch temp (was 185aed4).

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -vv
  feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  * main     185aed4 [origin/main] Create hello-github.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  Push URL: https://github.com/EL-ELL/my-first-git-hub-repository.git
  HEAD branch: main
  Remote branches:
    feature-1 tracked
    main      tracked
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main      merges with remote main
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main      pushes to main      (up to date)

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

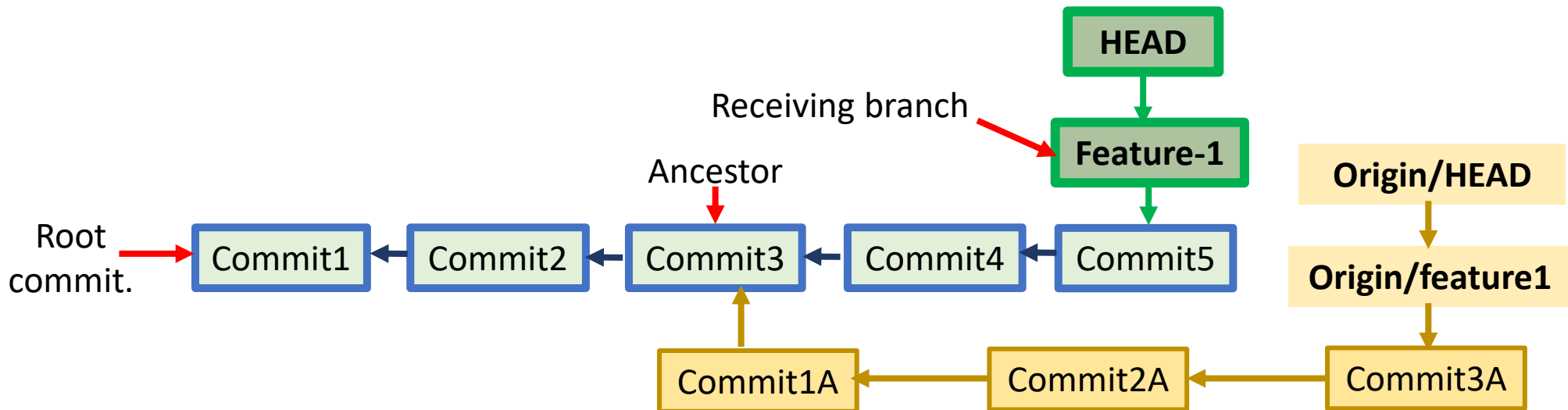
Git pull – merges remote branch into current local branch



Git pull is a two step process of git fetch and git merge and in order for it to work we need at least one tracking branch. First git will fetch all updates in remote repository and write them into local repository. Then git merge is performed locally using fast forward method where possible or if not then a three-way merge. Git pull only updates the currently checked out branch.

Git pull

1. Checkout local branch and make sure it is tracking branch and has corresponding remote branch. Use **git branch -vv**.
2. Enter **git pull**
3. Git will fetch all changes from remote repository – **git fetch** is executed in the background.
4. After fetching git updates **FETCH_HEAD** file that contains SHA1 hashes of last commits in remote repository for all tracking branches.
5. Git merges remote branch into current local branch **git merge FETCH_HEAD**.



git branch -a Shows all branches in local and remote repositories. →

git branch -vv Shows that both branches in the local repository are tracked in the remote repository. →

git fetch -v performs a git fetch with verbose output. We see that → both branches are up to date.

git pull -v is a verbose output. Note how the output is the same as a → git fetch. This is because a git pull performs a git fetch then a merge.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -a
* feature-1
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/feature-1
  remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt
```

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

$ git branch -vv
* feature-1 7422106 [origin/feature-1] file list.txt was created in the feature-1 branch
  main      185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git fetch -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-ELL/my-first-git-hub-repository
 = [up to date]      feature-1  -> origin/feature-1
 = [up to date]      main       -> origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git pull
Already up to date.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git pull -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-ELL/my-first-git-hub-repository
 = [up to date]      feature-1  -> origin/feature-1
 = [up to date]      main       -> origin/main
Already up to date.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ |
```


git pull -v performed on the main branch also gives us the fetch + merge output results.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git pull -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-ELL/my-first-git-hub-repository
 = [up to date]      main      -> origin/main
 = [up to date]      feature-1 -> origin/feature-1
Already up to date.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```

The contents of the hidden git directory now has a **FETCH_HEAD** file.

The **FETCH_HEAD** file contains a SHA1 hash for each branch. It lists the URL of the remote repository for that branch. Note how the HEAD branch is listed first in this table.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository/.git

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository/.git (GIT_D
IR!)
$ ls -la
total 19
drwxr-xr-x 1 ellio 197609  0 Mar  3 02:24 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 02:23 ../
-rw-r--r-- 1 ellio 197609 244 Mar  3 02:24 FETCH_HEAD
-rw-r--r-- 1 ellio 197609  21 Mar  3 02:23 HEAD
-rw-r--r-- 1 ellio 197609  41 Mar  3 02:24 ORIG_HEAD
-rw-r--r-- 1 ellio 197609 384 Mar  2 19:58 config
-rw-r--r-- 1 ellio 197609  73 Mar  2 02:01 description
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:01 hooks/
-rw-r--r-- 1 ellio 197609 217 Mar  3 02:23 index
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:01 info/
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:02 logs/
drwxr-xr-x 1 ellio 197609  0 Mar  3 02:24 objects/
-rw-r--r-- 1 ellio 197609 183 Mar  2 19:58 packed-refs
drwxr-xr-x 1 ellio 197609  0 Mar  2 02:02 refs/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository/.git (GIT_D
IR!)
$ cat FETCH_HEAD
185aed41542260681f23894a94531e49bb4acbd5      branch 'main' of https://github.com/EL-ELL/my-f
git-hub-repository
74221068ba4ffa278e8b1ea052ff2b47e88d8508      not-for-merge  branch 'feature-1' of https://github.co
ELL/my-first-git-hub-repository
```

Now we checkout the feature-1 branch and perform a fetch.

Notice now that in the FETCH_HEAD file the feature-1 branch is now at the top of the table.

The not for merge tag is set on non HEAD branches so that the merge is performed only on the first branch in the GIT_HEAD file without this tag.

Note if we check the file we see that the committer is GitHub because it was performed by the remote repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository/.git (GIT_DIR
!)
$ cd ..

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git checkout feature-1
Switched to branch 'feature-1'
Your branch is up to date with 'origin/feature-1'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git fetch

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ cat .git/FETCH_HEAD
74221068ba4ffa278e8b1ea052ff2b47e88d8508          branch 'feature-1' of https://github.com/EL-ELL
/my-first-git-hub-repository
185aed41542260681f23894a94531e49bb4acbd5          not-for-merge branch 'main' of https://github.com/EL-
ELL/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -p 74221068ba4ffa
tree 3a263ea2889051eaa137b26aaf887c1b4684b56d
parent 185aed41542260681f23894a94531e49bb4acbd5
author EL-ELL <100733779+EL-ELL@users.noreply.github.com> 1646182366 +0100
committer GitHub <noreply@github.com> 1646182366 +0100
pgpsig -----BEGIN PGP SIGNATURE-----

wsBcBAABCAAQBQJiHr/eCRBK7hj40v3rIwAAuUgIAErhpSBp78FgpSdpBzR/c06W
21lhziOLyqGK8i0GZP4wVvKdPlI+7SZ5P6BbV+0ca899Mctb+0x1/wd52u9r4cWG
ecV+s8GMx1KVuaBsw2y9tUg8N0wFv0qMDXrncf1bW8tEgv6yZKCN4Kgk0bd9L9d
1gy8hgVHZq5AnboaiNnIrYvEv0LUhkRRns0h6Iu6DxH88UtSrBvr6Y9gusPb5r0
KpSgxjU5HpFnSZ1e5BH2ydIB9/opgaktmNlTHNPy5fz0XBqCB/8qtCe/DcYwCXgL
2VGh0a86/F7IvqFPO9BwzFU+xDB1BA4aNTH5Muu0GmZGb0hIV+iYcWSUQ5ir9DY=
=Nthc
-----END PGP SIGNATURE-----

file list.txt was created in the feature-1 branch
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

Note in the GitHub remote repository in the list of commits that the last commit with comment “file-list.txt was created in the feature-1 branch” with a SHA1 of 7422106.

This is the same SHA1 as the feature-1 branch in the FETCH_HEAD file.

Git fetch simply compares SHA1 hashes in local FETCH_HEAD file to last commit if that branch in the remote repository to determine if the repository is up to date.

The screenshot shows the GitHub interface for the repository `EL-ELL / my-first-git-hub-repository`. The `feature-1` branch is selected. The commit history is displayed, showing four commits. The top commit, which is the most recent, has the message `file list.txt was created in the feature-1 branch` and a SHA1 hash of `7422106`. A red arrow points to this SHA1 hash. Below it are three more commits: `Create hello-github.txt` (SHA1: `185aed4`), `Update README.md` (SHA1: `bddce3f`), and `Initial commit` (SHA1: `d67d9ce`). Each commit entry includes a 'Verified' status, a copy icon, the SHA1 hash, and a code icon.

Commit Message	Author	SHA1 Hash	Verified
file list.txt was created in the feature-1 branch	EL-ELL committed yesterday	7422106	Yes
Create hello-github.txt	EL-ELL committed yesterday	185aed4	Yes
Update README.md	EL-ELL committed yesterday	bddce3f	Yes
Initial commit	EL-ELL committed yesterday	d67d9ce	Yes

Lets create another file in the remote repository under the feature-1 branch and commit it.

The screenshot shows the GitHub web interface for a repository named "my-first-git-hub-repository". The user is on the "feature-1" branch, creating a new file named "another-file.txt". The file path is displayed as "my-first-git-hub-repository / feature / another-file.txt" in the breadcrumb, with "feature-1" highlighted in a red box. A red arrow points to the "feature" part of the path, and a red text overlay states "The file is created in the feature folder". Below the file editor, the "Commit new file" section is visible, with the "Commit directly to the feature-1 branch" option selected. The "Commit new file" button is highlighted with a red box. The footer of the page shows the GitHub logo, copyright information, and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

github.com/EL-ELL/my-first-git-hub-repository/new/feature-1

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

my-first-git-hub-repository / feature / another-file.txt in feature-1 Cancel changes

<> Edit new file Preview The file is created in the feature folder Spaces 2 No wrap

1 Just another file created in the feature-1 branch.

Commit new file

Create another-file.txt

Add an optional extended description...

☒ Commit directly to the feature-1 branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

github.com/EL-ELL/my-first-git-hub-repository/commits/feature-1

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

feature-1

Commits on Mar 3, 2022

Create another-file.txt
EL-ELL committed 1 minute ago

Verified 3220afa <>

View commit details

Commits on Mar 2, 2022

file list.txt was created in the feature-1 branch
EL-ELL committed yesterday

Verified 7422106 <>

Create hello-github.txt
EL-ELL committed yesterday

Verified 185aed4 <>

Update README.md
EL-ELL committed yesterday

Verified bddce3f <> 1

Initial commit

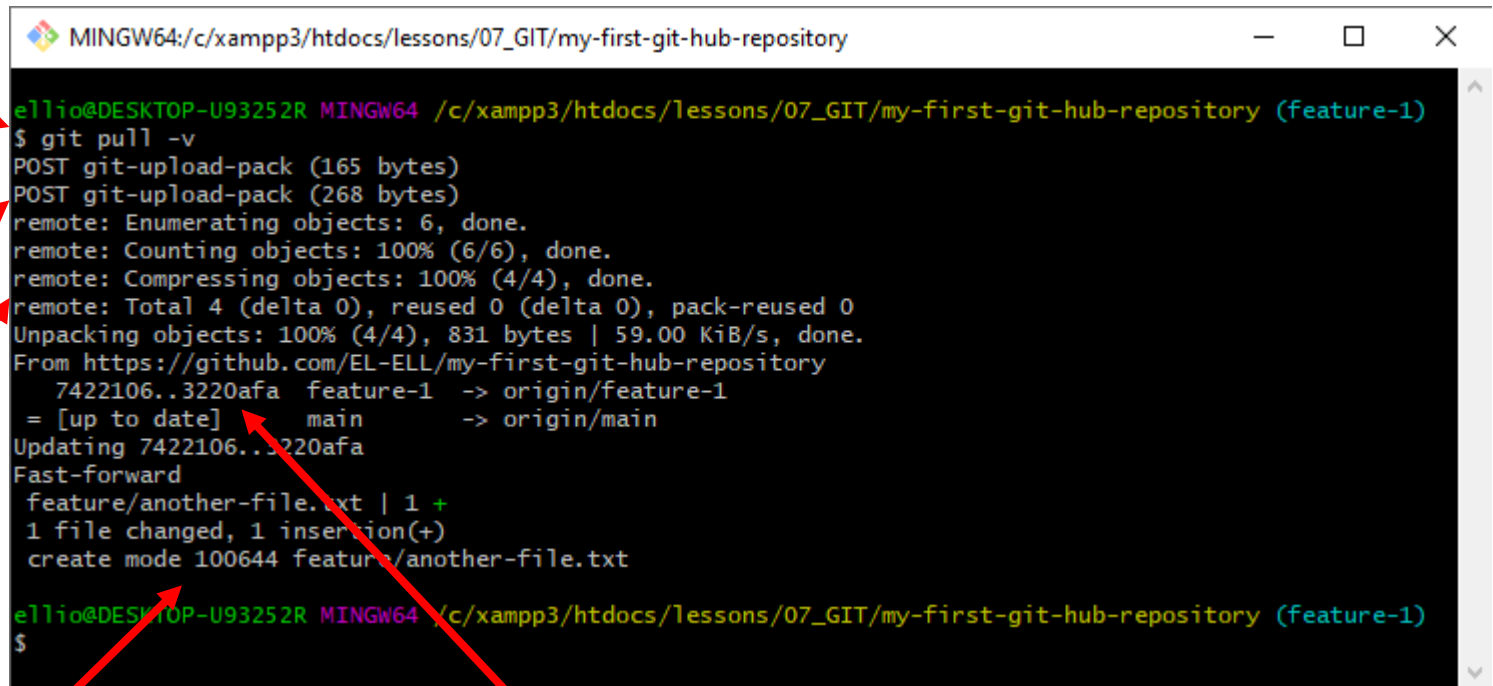
https://github.com/EL-ELL/my-first-git-hub-repository/commit/3220afa57193fe24e4319de63d0bd87899fd5f94

Reset 100 % 03:03

In the commits section of feature-1 branch we see the last commit with SHA1 hash of 3220afa

Now we perform a pull to feature-1 branch

Note how first it uploads in compressed format the feature-1 local repository changes so it can compare it with those in the remote repository. In this example it detects 4 files but that none of these files are missing from remote repository (delta 0)



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git pull -v
POST git-upload-pack (165 bytes)
POST git-upload-pack (268 bytes)
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 831 bytes | 59.00 KiB/s, done.
From https://github.com/EL-ELL/my-first-git-hub-repository
  7422106..3220afa  feature-1  -> origin/feature-1
   = [up to date]    main      -> origin/main
Updating 7422106..3220afa
Fast-forward
 feature/another-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature/another-file.txt
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

Red arrows point from the text on the left to the following lines in the terminal output: "POST git-upload-pack", "remote: Enumerating objects", "remote: Compressing objects", "7422106..3220afa feature-1 -> origin/feature-1", and the final prompt "\$".

It now performs the fetch noting that main branch was up to date but that feature-1 branch there were changes. Note how it was pointing to commit 7422106 but now points to **3220afa**. This is the last commit we made in the remote repository feature-1 branch.

After this step it then updates in the local repository the files up to the last commit in the remote repository and gives us an output of 1 file changed with one insertion (1 line added). It creates the missing file of feature/aonther-file.txt.

In this case there were no new files in the local feature-branch before changes were made in the feature-1 branch of the remote repository so it was able to perform a fast forward merge.

We can list the files in the local repository and see another-file.txt.

If we check the local repository staging area we also see another-file.txt.

Git log shows the latest commit SHA1 with HEAD->feature-1 and origin/feature-1.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la feature
total 6
drwxr-xr-x 1 ellio 197609  0 Mar  3 03:05 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 02:32 ../
-rw-r--r-- 1 ellio 197609 52 Mar  3 03:05 another-file.txt
-rw-r--r-- 1 ellio 197609 26 Mar  3 02:32 list.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git ls-files -s
100644 8f5a8bb3bb57c9896572dc4e45a4a2abcb4f0631 0      README.md
100644 b2fc5ba208692420e52476e6954216e3cf7b9f31 0      feature/another-file.txt
100644 e55dd9aec3e159e7621f957a90e882cccfc15370 0      feature/list.txt
100644 13677d3767eb5370882941a929b12015da599535 0      hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit 3220afa57193fe24e4319de63d0bd87899fd5f94 (HEAD -> feature-1, origin/feature-1)
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
Date: Thu Mar 3 03:01:54 2022 +0100

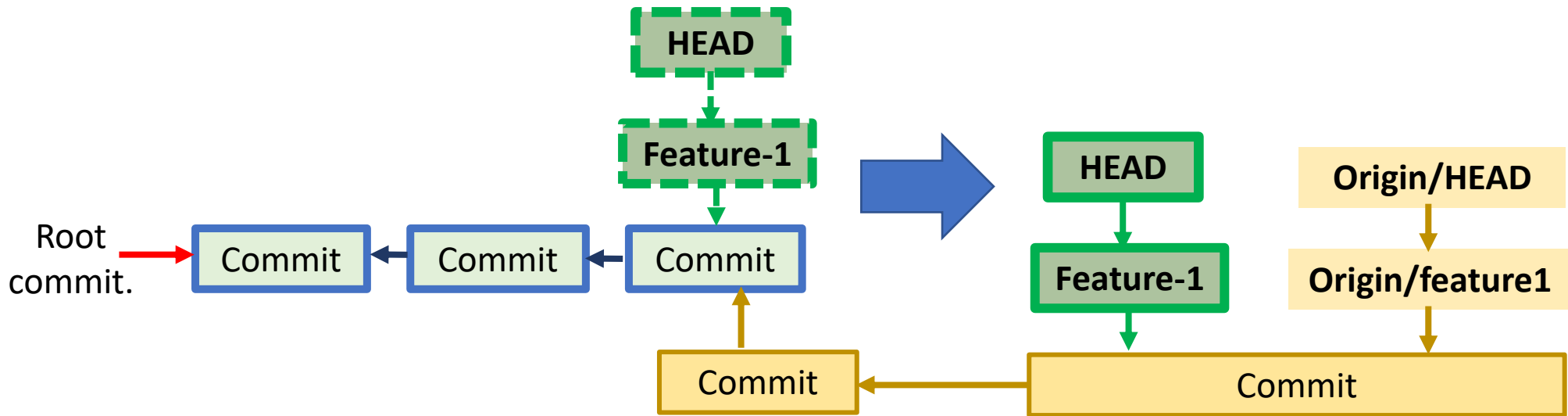
    Create another-file.txt

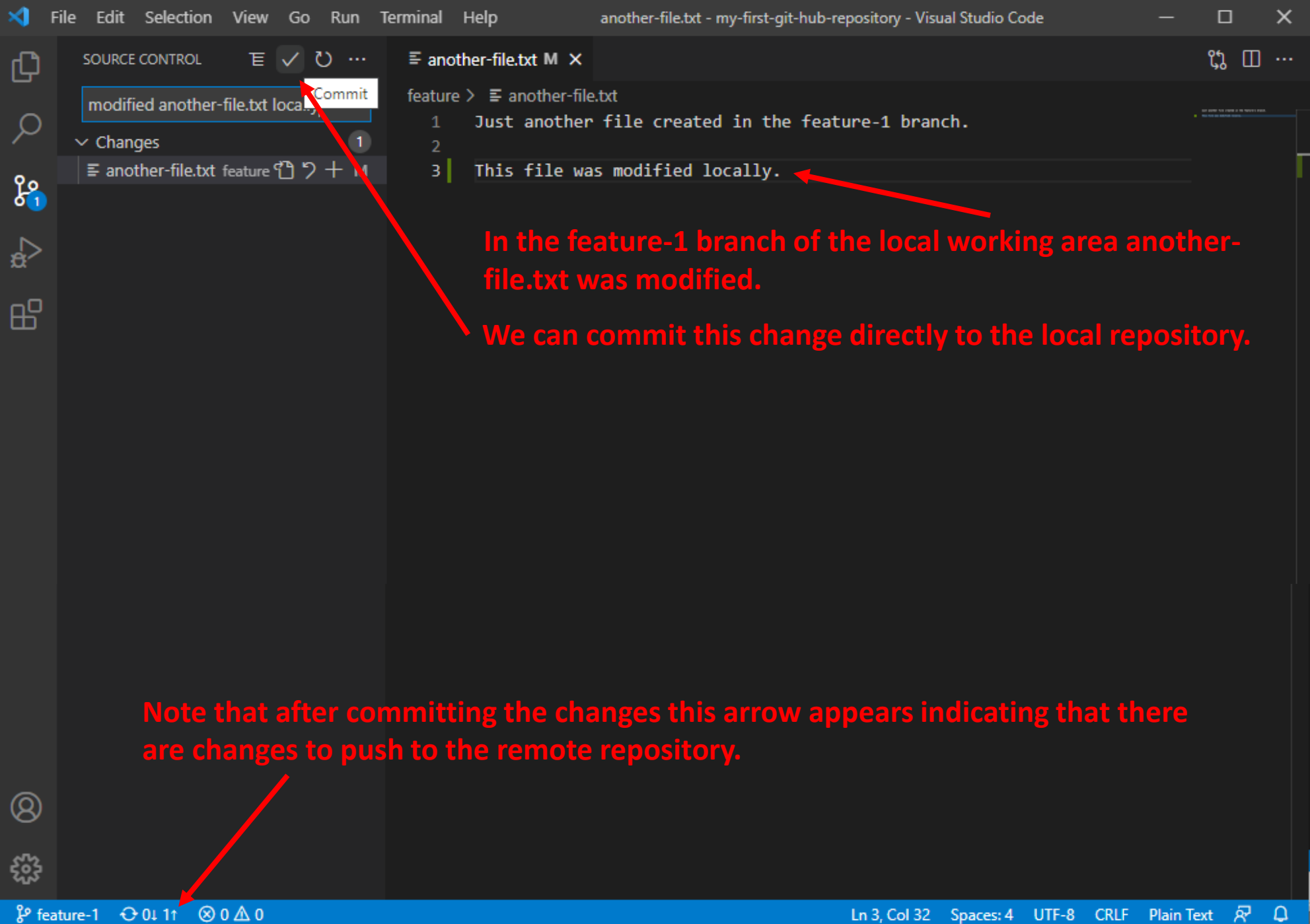
commit 74221068ba4ffa278e8b1ea052ff2b47e88d8508
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
Date: Wed Mar 2 01:52:46 2022 +0100

    file list.txt was created in the feature-1 branch

commit 185aed41542260681f23894a94531e49bb4acbd5 (origin/main, origin/HEAD, main)
```

After git pull operation git has seen that there are no more commits in feature 1 branch of local repository so it can perform a fast forward merge. In fast forward merge git simply moves pointer of local feature-2 branch to remote repository feature-1 branch.





Lets create another file in the remote repository in folder called feature of the feature-1 branch and commit it.

The screenshot shows the GitHub web interface for a repository named 'my-first-git-hub-repository' by user 'EL-ELL'. The page is for creating a new file. The breadcrumb path is 'my-first-git-hub-repository / feature / one-more-file.txt in feature-1'. A red box highlights the 'feature-1' branch name, and a red arrow points to it with the text 'The file is created in the feature folder'. Below the file path, there is a text area with the content '1 This file was created in the remote repository. |'. The 'Commit new file' section is visible, with a red box around the 'Commit new file' button. The commit message is 'Create one-more-file.txt'. The commit options are set to 'Commit directly to the feature-1 branch.'.

github.com/EL-ELL/my-first-git-hub-repository/new/feature-1

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

my-first-git-hub-repository / feature / one-more-file.txt in feature-1 Cancel changes

Edit new file Preview The file is created in the feature folder

1 This file was created in the remote repository. |

Commit new file

Create one-more-file.txt

Add an optional extended description...

☒ Commit directly to the feature-1 branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

github.com/EL-ELL/my-first-git-hub-repository/commit/fa84e35e35de2004217beb911f3d3313f2752234

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

Create one-more-file.txt feature-1

EL-ELL committed 1 minute ago Verified 1 parent 3220afa commit fa84e35e35de2004217beb911f3d3313f2752234

Showing 1 changed file with 1 addition and 0 deletions. Split Unified

Here we see the SHA1 of this latest commit in the remote repository. Of course there is no such file in the local repository.

We can see in the local repository the ultimate commit for the last file added to feature-1 branch. Note how it only points to local repository.

Notice the previous commit points to origin/feature-1 which is the remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit d0de7adfbf3c62c607321f9fce3ec523c79da30f (HEAD -> feature-1)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 04:03:37 2022 +0100

    modified another-file.txt locally

commit 3220afa57193fe24e4319de63d0bd87899fd5f94 (origin/feature-1)
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
Date: Thu Mar 3 03:01:54 2022 +0100

    Create another-file.txt

commit 74221068ba4ffa278e8b1ea052ff2b47e88d8508
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
```

Now we can perform the fetch and see that the feature-1 branch SHA1 has changed from 3220afa to fa84e35. This corresponds to the penultimate commit in the feature-1 branch and the ultimate commit in the remote repository.

In the local repository we can see another-file.txt has been added. But what about the staging area?

We can see four new files.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git fetch -v
POST git-upload-pack (165 bytes)
POST git-upload-pack (368 bytes)
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 862 bytes | 57.00 KiB/s, done.
From https://github.com/EL-ELL/my-first-git-hub-repository
   3220afa..fa84e35  feature-1  -> origin/feature-1
   = [up to date]      main      -> origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git ls-files -s
100644 8f5a8bb3bb57c9896572dc4e45a4a2abcb4f0631 0      README.md
100644 8e717ccc0db1b1ab06b6e17e49e441c14817080a 0      feature/another-file.txt
100644 e55dd9aec3e159e7621f957a90e882cccf15370 0      feature/list.txt
100644 13677d3767eb5370882941a929b12015da599535 0      hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ |
```

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la .git/objects
total 12
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:22 ./
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:03 ../
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:22 2a/
drwxr-xr-x 1 ellio 197609 0 Mar  3 03:05 32/
drwxr-xr-x 1 ellio 197609 0 Mar  3 03:05 4c/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:22 66/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:03 8e/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:22 b0/
drwxr-xr-x 1 ellio 197609 0 Mar  3 03:05 b2/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:03 b8/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:03 d0/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:03 dd/
drwxr-xr-x 1 ellio 197609 0 Mar  3 03:05 f1/
drwxr-xr-x 1 ellio 197609 0 Mar  3 04:22 fa/
drwxr-xr-x 1 ellio 197609 0 Mar  2 02:01 info/
drwxr-xr-x 1 ellio 197609 0 Mar  2 02:02 pack/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

Last commit from the remote repository. It is not merged into local branch but exists in the staging area

If we look in the contents of the b0 directory to find the hash we can check that file and see it is a tree. This tree points to the files fetched from the remote repository.

The other newly created file in the staging area is a blob with contents of the new file create in the remote repository.

In directory 66 we find another tree. This tree pertains to the new file created in the local repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la .git/objects/b0
total 5
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ../
-r--r--r-- 1 ellio 197609 128 Mar  3 04:22 b87de27f77e4677b63ca658c6876af5372611a

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -p b0b87de27
100644 blob 8f5a8bb3bb57c9896572dc4e45a4a2abcb4f0631    README.md
040000 tree 6665fb8e859b1bf350782f9394275e926f9a498c    feature
100644 blob 13677d3767eb5370882941a929b12015da599535    hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -t b0b87de27
tree

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la .git/objects/2a
total 5
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ../
-r--r--r-- 1 ellio 197609  61 Mar  3 04:22 3b958a8974c449cad6654dbcfa7050a042c98

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -t 2a3b958a8
blob

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -p 2a3b958a8
This file was created in the remote repository.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la .git/objects/66
total 5
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:22 ../
-r--r--r-- 1 ellio 197609 124 Mar  3 04:22 65fb8e859b1bf350782f9394275e926f9a498c

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git cat-file -t 6665fb8
tree

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

In the FETCH_HEAD file we see that the feature-1 branch has the SHA1 of the last commit in the remote repository.

But the last SHA1 in the git log is the last one from the local directory because we have not merged.

git merge
FETCH_HEAD will merge feature-1 branch of the remote and local repositories.

Now the working directory contains the new file from the remote repository and the new file from the local directory.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ cat .git/FETCH_HEAD
fa84e35e35de2004217beb911f3d3313f2752234      branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository
185aed41542260681f23894a94531e49bb4acbd5      not-for-merge branch 'main' of https://github.com/EL-ELL/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit d0de7adfbf3c62c607321f9fce3ec523c79da30f (HEAD -> feature-1)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 04:03:37 2022 +0100

    modified another-file.txt locally

commit 3220afa57193fe24e4319de63d0bd87899fd5f94
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
Date: Thu Mar 3 03:01:54 2022 +0100

    Create another-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git merge FETCH_HEAD
Merge made by the 'ort' strategy.
 feature/one-more-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature/one-more-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ ls -la feature
total 7
drwxr-xr-x 1 ellio 197609  0 Mar  3 04:58 ./
drwxr-xr-x 1 ellio 197609  0 Mar  3 02:32 ../
-rw-r--r-- 1 ellio 197609 85 Mar  3 04:00 another-file.txt
-rw-r--r-- 1 ellio 197609 26 Mar  3 02:32 list.txt
-rw-r--r-- 1 ellio 197609 49 Mar  3 04:58 one-more-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

In the log we see a new commit for the merge pointing to the feature-1 branch-> HEAD.

Notice how it has two parents of d0de7ad and fa84e35.

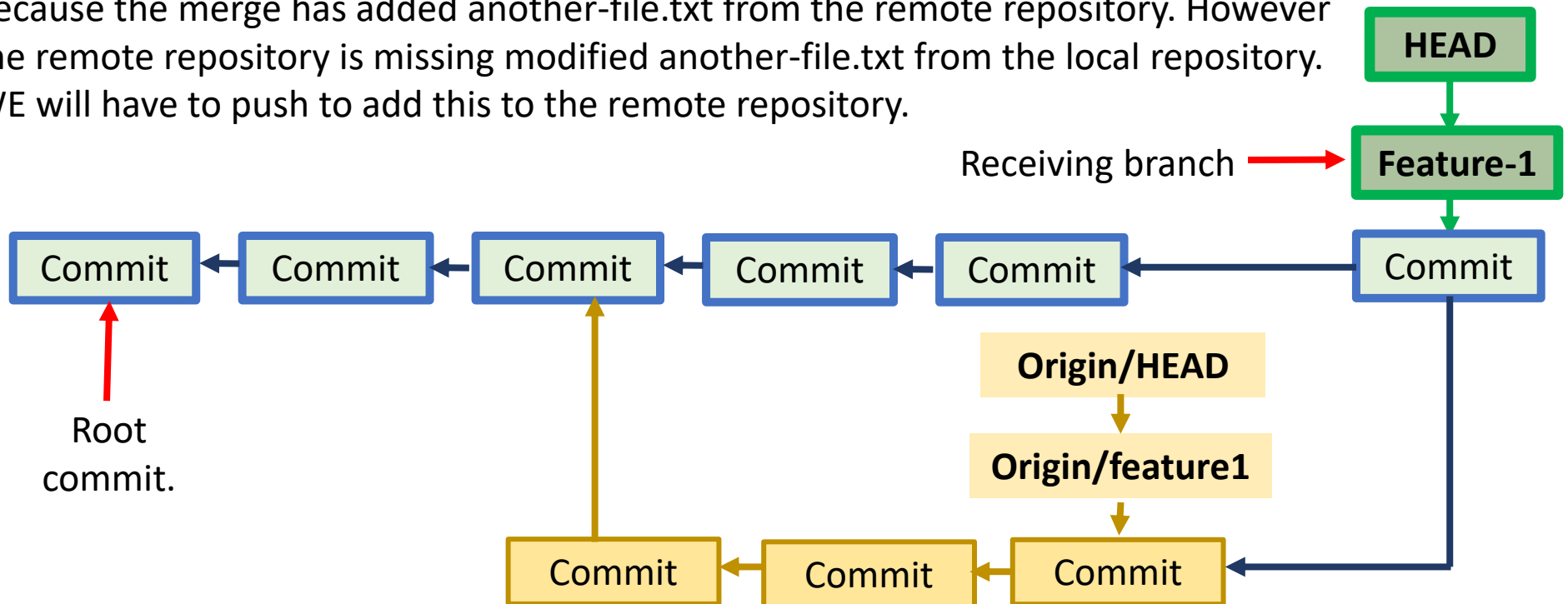
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit 721135b181653488c969e463251fbd7f8b0a2c74 (HEAD -> feature-1)
Merge: d0de7ad fa84e35
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 04:58:28 2022 +0100

    Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1

commit fa84e35e35de2004217beb911f3d3313f2752234 (origin/feature-1)
Author: EL-ELL <100733779+EL-ELL@users.noreply.github.com>
Date: Thu Mar 3 04:14:18 2022 +0100

    Create one-more-file.txt
```

The next commit down from the origin/feature-1 has a different SHA1 hash. This is because the merge has added another-file.txt from the remote repository. However the remote repository is missing modified another-file.txt from the local repository. WE will have to push to add this to the remote repository.



github.com/EL-ELL/my-first-git-hub-repository/edit/feature-1/README.md

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

my-first-git-hub-repository / README.md in feature-1 Cancel changes

<> Edit file Preview Spaces 2 Soft wrap

```
1 # my-first-git-hub-repository
2 Here is my first GitHub repository. Let's get started with GitHub.
3
4 README.md file was just edited manually. One more Edit at GitHub
```

Attach files by dragging & dropping, selecting or pasting them.

Commit changes

Update README.md

Add an optional extended description...

☒ Commit directly to the feature-1 branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

One more edit at GitHub.

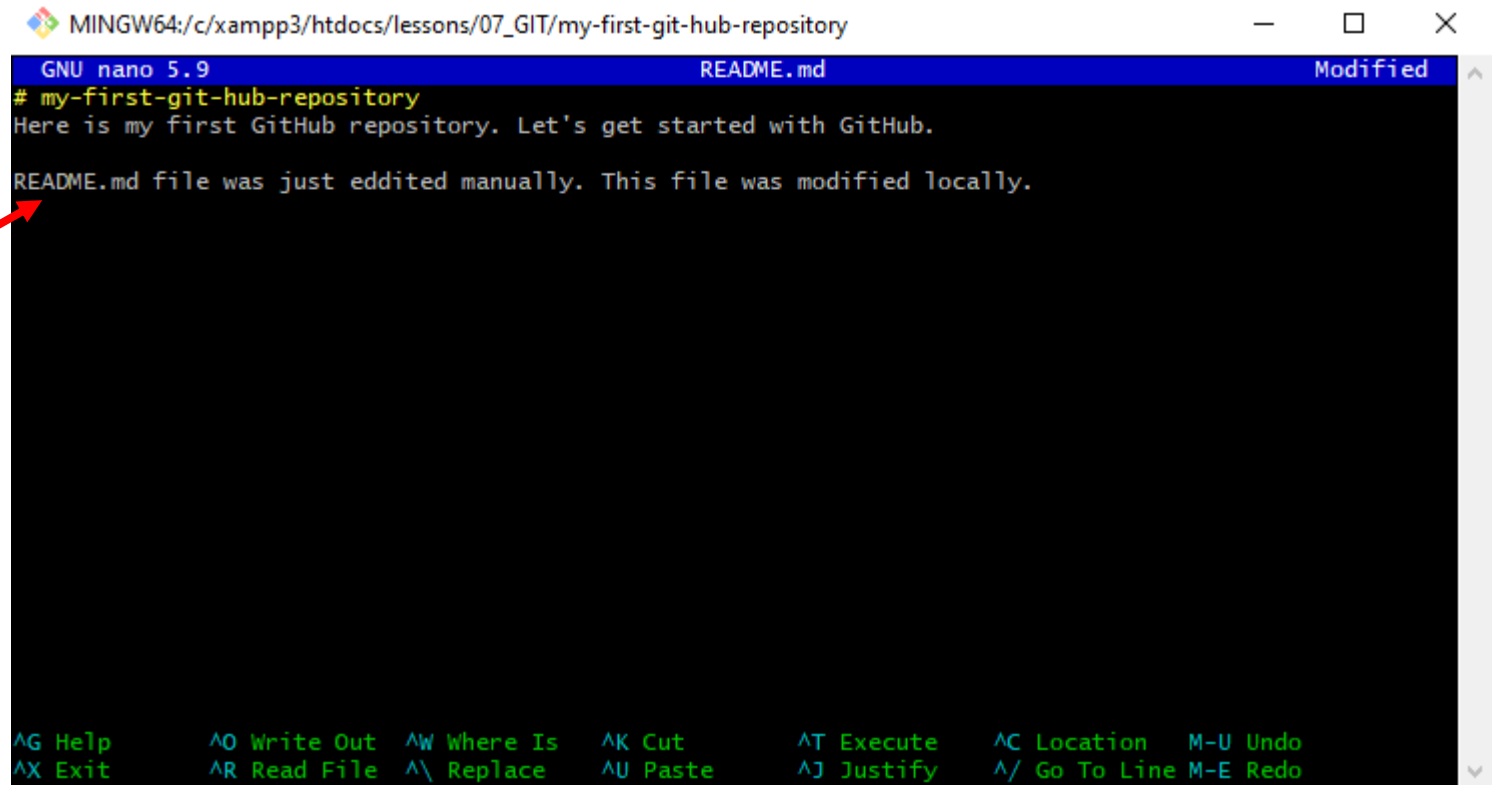
README.md file was modified in the feature-1 branch in remote repository and changes committed.

Activate Windows
Go to Settings to activate Windows.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

With a nano
README.md
command from the
feature-1 branch we
can modify this file
differently to that of
the remote repository.

CONTROL + X and Yes
to save.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
GNU nano 5.9 README.md Modified
# my-first-git-hub-repository
Here is my first GitHub repository. Let's get started with GitHub.

README.md file was just eddited manually. This file was modified locally.

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Git status shows the README.md file pending to be staged and committed.

We can do this with a combined add and commit command **git commit -a -m <comment>**

Git log now shows this last commit to the feature-1 branch of local repository.

We have modified the same file with different text in both the remote and local repository. What will happen if we try and do a pull from remote repository? – a conflict.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git status
On branch feature-1
Your branch is ahead of 'origin/feature-1' by 2 commits.
(use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

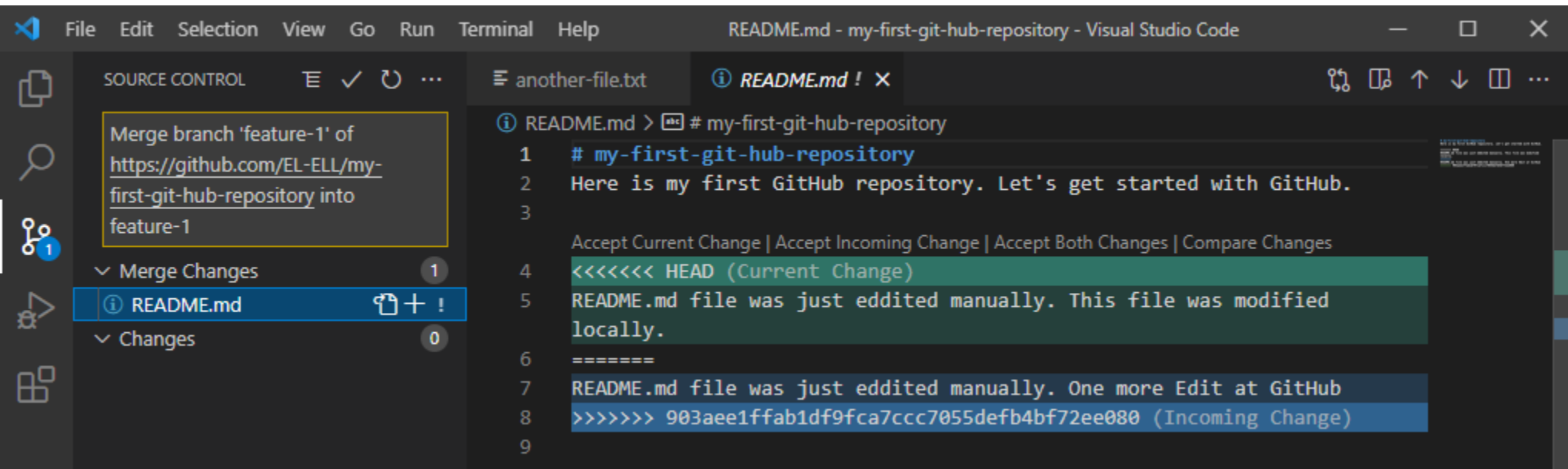
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git commit -a -m "Modified README.md file locally"
[feature-1 7b334c9] Modified README.md file locally
1 file changed, 1 insertion(+), 1 deletion(-)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit 7b334c94d05f6ea132a4538e995461f5eda1118a (HEAD -> feature-1)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 05:42:05 2022 +0100

    Modified README.md file locally
```

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git pull -v
POST git-upload-pack (165 bytes)
POST git-upload-pack (518 bytes)
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 746 bytes | 74.00 KiB/s, done.
From https://github.com/EL-ELL/my-first-git-hub-repository
   fa84e35..903aee1 feature-1 -> origin/feature-1
   = [up to date]      main      -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```



Local branch is receiving branch and remote branch is incoming branch.

In the staging area of the local repository we have 3 versions of the README.md file. 1 is common file for both branches before attempted merge. 2 is changed file in receiving branch. 3 is changed file of incoming branch.

```

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

e11lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1|MERGING)
$ git ls-files -s
100644 8f5a8bb3bb57c9896572dc4e45a4a2abcb4f0631 1      README.md
100644 1845abd202fb86fd6e07e0d2b456323c9fb95762 2      README.md
100644 b588bf2ce2c80fad579f965c719685a50e6464ab 3      README.md
100644 8e717ccc0db1b1ab06b6e17e49e441c14817080a 0      feature/another-file.txt
100644 e55dd9aec3e159e7621f957a90e882cccfc15370 0      feature/list.txt
100644 2a3b958a8974c449cad6654dbcf7050a042c98 0      feature/one-more-file.txt
100644 13677d3767eb5370882941a929b12015da599535 0      hello-github.txt

e11lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1|MERGING)
$

```

Git status shows the modified file pending to be staged and committed.

Note that there are still 3 versions of the file until the conflict resolution is committed.

After staging and committing the file the git log shows a new commit with two parents, one for the remote repository and one for the local repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1|MERGING)
$ git status
On branch feature-1
Your branch and 'origin/feature-1' have diverged,
and have 3 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.
(fix conflicts and run "git commit")
(use "git merge --abort" to abort the merge)

Unmerged paths:
(use "git add <file>..." to mark resolution)
    both modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1|MERGING)
$ git ls-files -s
100644 8f5a8bb3bb57c9896572dc4e45a4a2abcb4f0631 1      README.md
100644 1845abd202fb86fd6e07e0d2b456323c9fb95762 2      README.md
100644 b588bf2ce2c80fad579f965c719685a50e6464ab 3      README.md
100644 8e717ccc0db1b1ab06b6e17e49e441c14817080a 0      feature/another-file.txt
100644 e55dd9aec3e159e7621f957a90e882cccfc15370 0      feature/list.txt
100644 2a3b958a8974c449cad6654dbcfa7050a042c98 0      feature/one-more-file.txt
100644 13677d3767eb5370882941a929b12015da599535 0      hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1|MERGING)
$ git commit -a
[feature-1 9f1d8db] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1

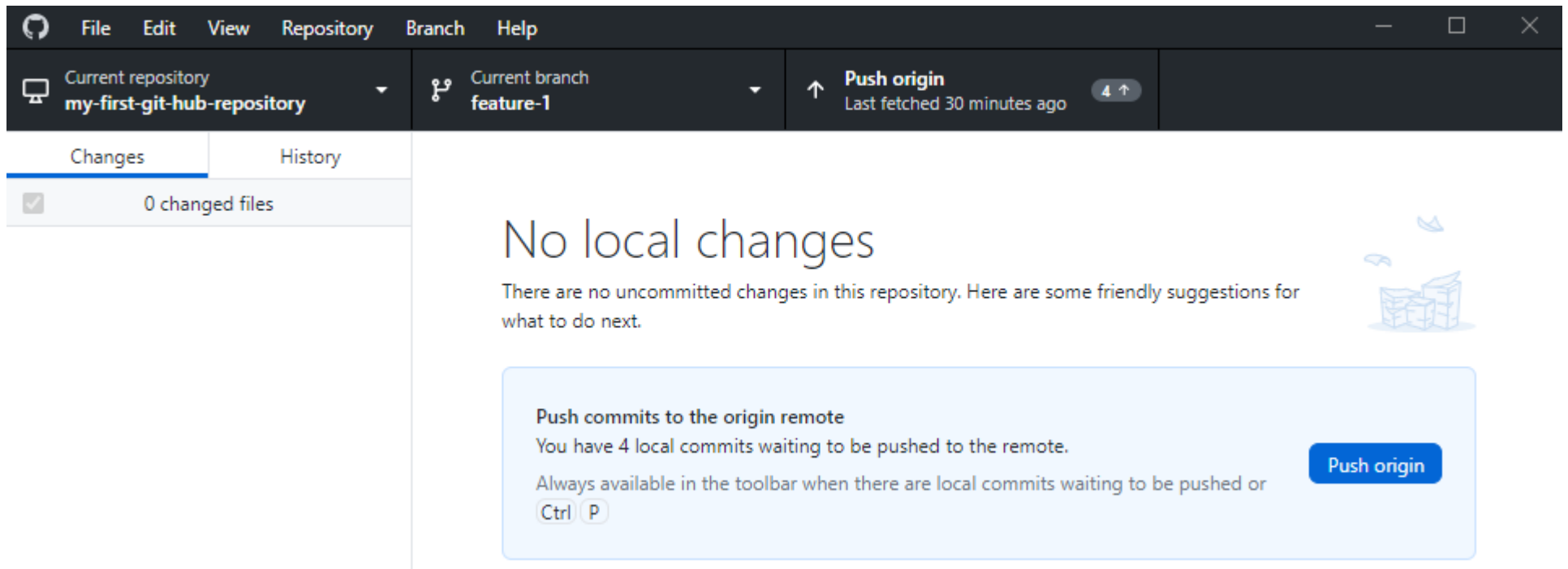
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit 9f1d8dbf66307291133142b56a6589532855141b (HEAD -> feature-1)
Merge: 7b334c9 903aee1
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 06:05:44 2022 +0100

    Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1

commit 7b334c94d05f6ea132a4538e995461f5eda1118a
```

Now we have performed git pull several times so the local repository is up to date with the remote repository.

However there are changes in the local repository which are absent in the remote repository. We can solve this with git push. Note that we need to have write access to the remote repository. GitHub desktop is showing 4 local commits waiting to be pushed to the remote repository and visual studio code shows pending pushes in the bottom bar.



git push -v will push changes from the local repository to the remote repository.

Note how pointer has updated from SHA1 of 903aee1 to 9f1d8db

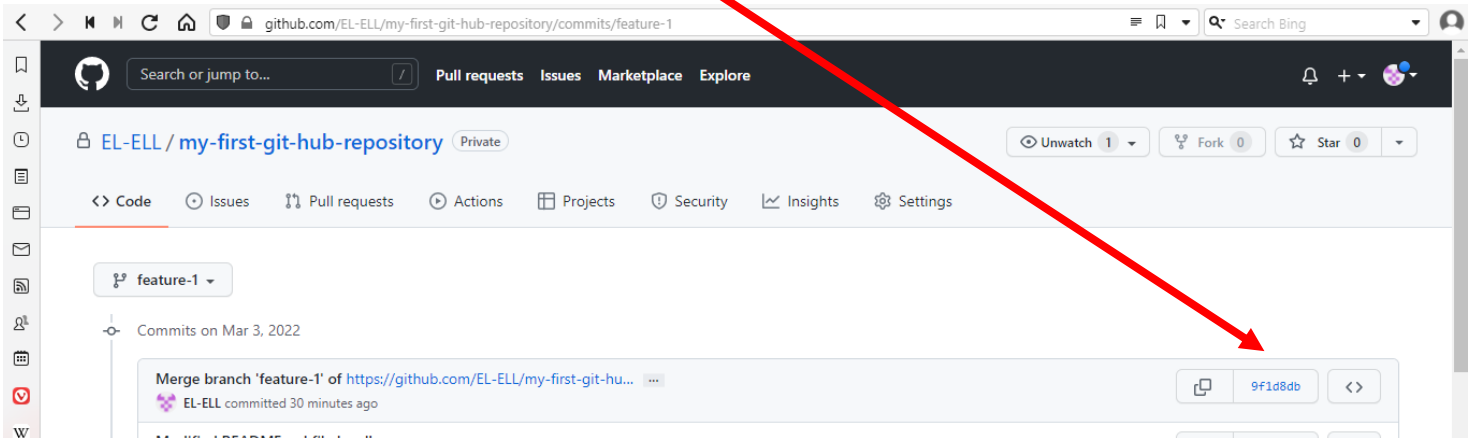
Also the refs/remotes/origin/feature-1 file has been updated with local tracking reference.

Git log shows this SHA1 as having two parents, one in the remote and one in the local.

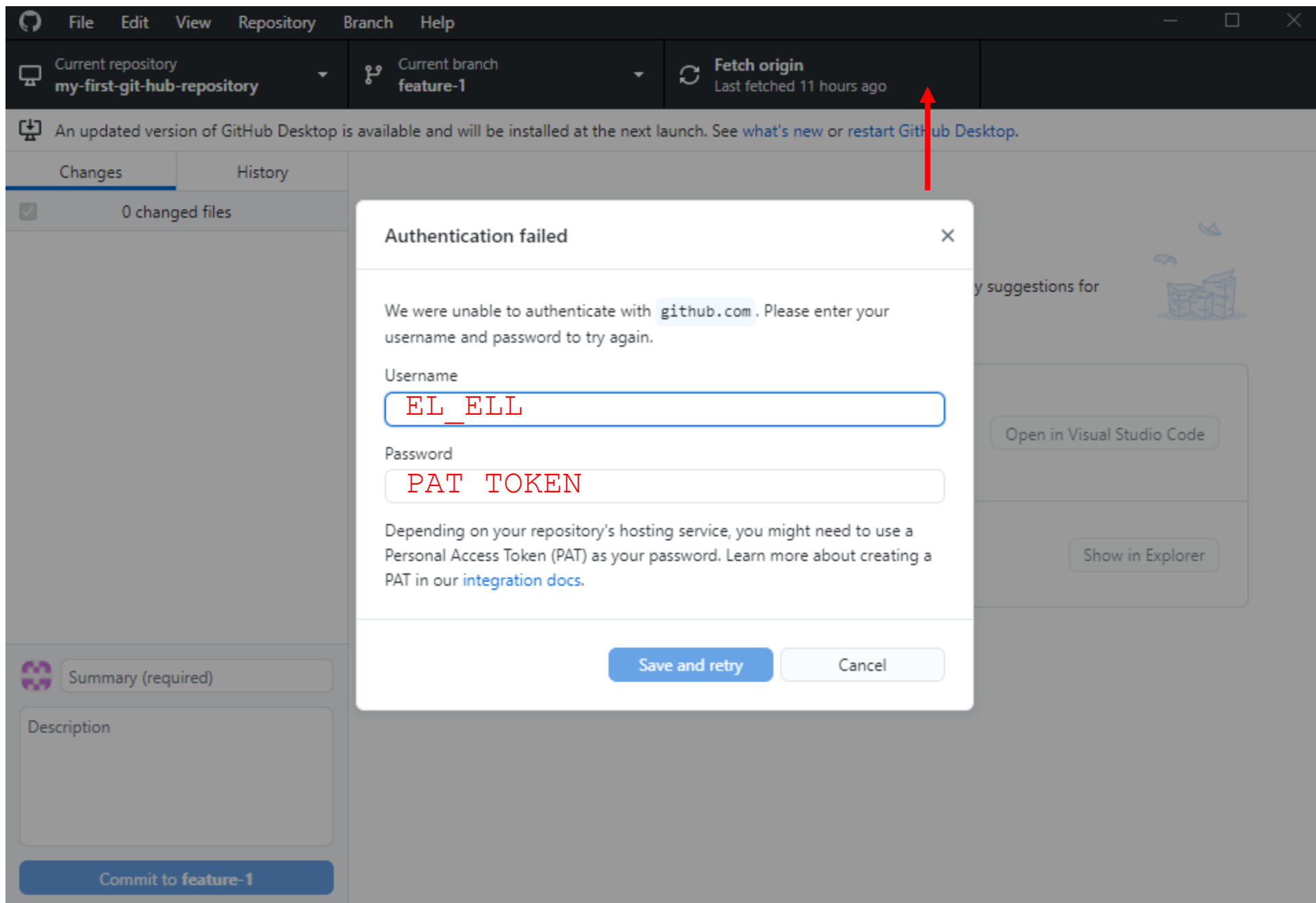
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git push -v
Pushing to https://github.com/EL-ELL/my-first-git-hub-repository.git
Enumerating objects: 25, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 1.77 KiB | 1.77 MiB/s, done.
Total 13 (delta 2), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (1999 bytes)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/EL-ELL/my-first-git-hub-repository.git
 903aee1..9f1d8db feature-1 -> feature-1
updating local tracking ref 'refs/remotes/origin/feature-1'

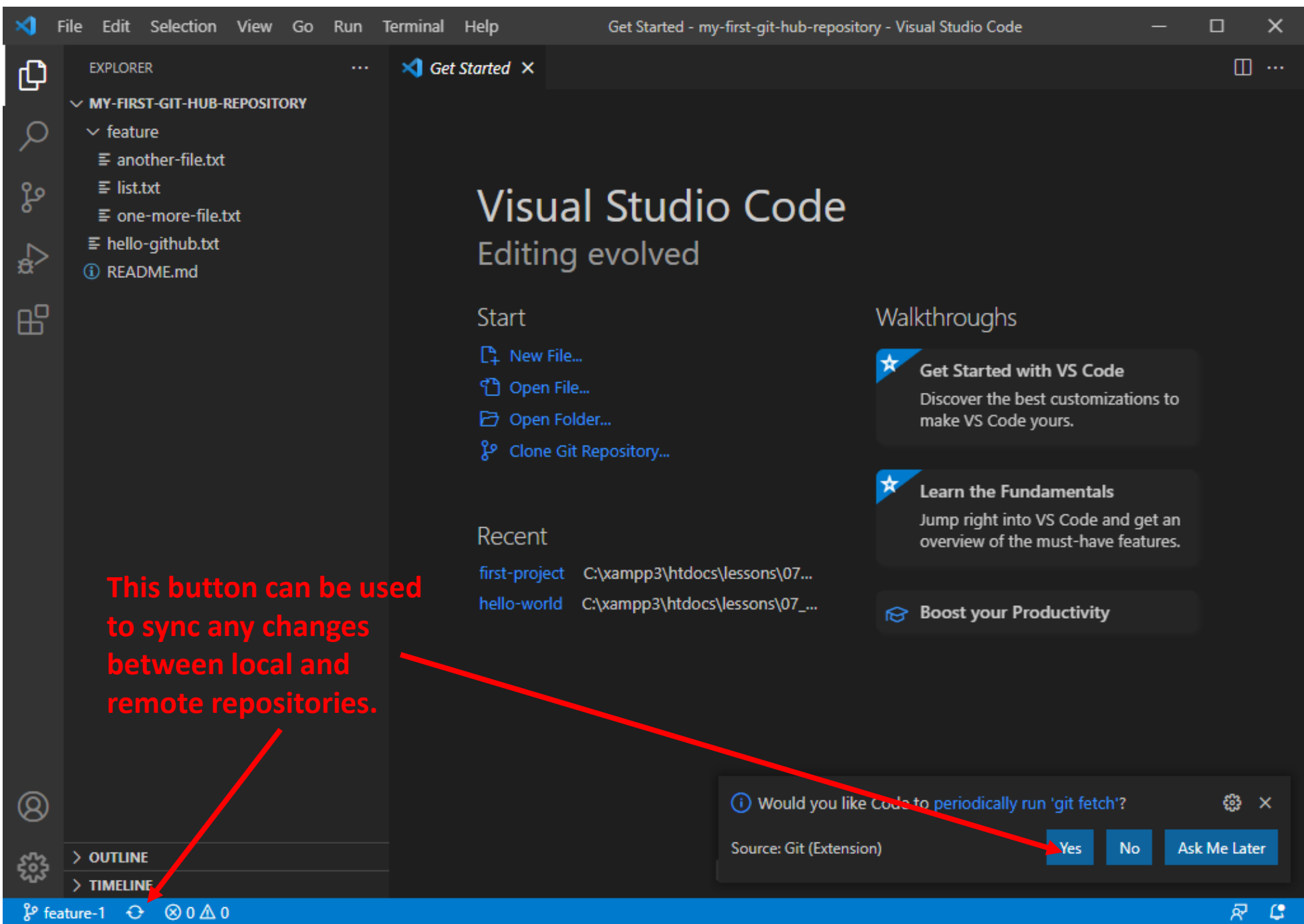
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git log
commit 9f1d8dbf66307291133142b55a6589532855141b (HEAD -> feature-1, origin/feature-1)
Merge: 7b334c9 903aee1
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 06:05:44 2022 +0100

    Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
```



GitHub allows multiple users to update to remote repository so that multiple users can contribute to a project. The first time a user tries to push to a remote repository GitHub will ask for





git pull -v and **git push -v** show that remote and local repositories are synchronised.

If we look at the contents of the tracking file **refs/remotes/origin/feature-1** we see the same SHA1.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git pull -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-ELL/my-first-git-hub-repository
= [up to date]      feature-1  -> origin/feature-1
= [up to date]      main       -> origin/main
Already up to date.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git push -v
Pushing to https://github.com/EL-ELL/my-first-git-hub-repository.git
To https://github.com/EL-ELL/my-first-git-hub-repository.git
= [up to date]      feature-1 -> feature-1
updating local tracking ref 'refs/remotes/origin/feature-1'
Everything up-to-date

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ cat .git/refs/remotes/origin/feature-1
9f1d8dbf66307291133142b56a6589532855141b

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$
```

git branch shows the local branches of main and feature-1.

git branch -a shows the local branches of main and feature-1 plus the remote branches of main and feature-1

git branch -vv shows that both branches in the local repository are tracked in the remote repository.

git checkout -b feature-2 command is used to create new branch in the local repository.

git branch -r shows that the locally created new branch is not automatically created in the remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch
* feature-1
  main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -a
* feature-1
  main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git branch -vv
* feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
  main      185aed4 [origin/main] Create hello-github.txt

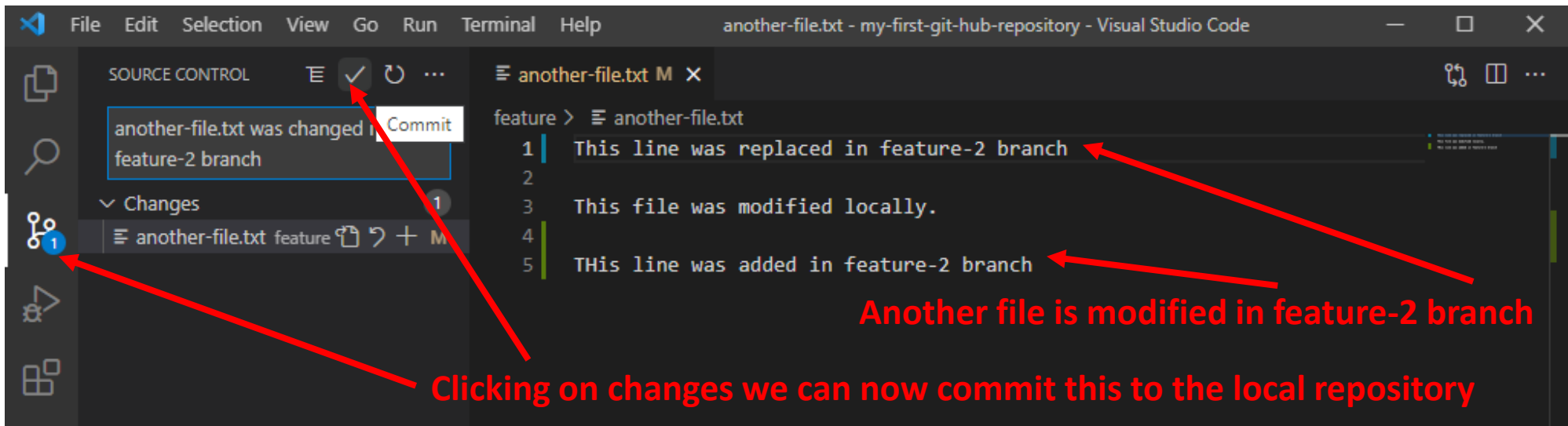
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-1)
$ git checkout -b feature-2
Switched to a new branch 'feature-2'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch
feature-1
* feature-2
  main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch -r
origin/HEAD -> origin/main
origin/feature-1
origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ |
```

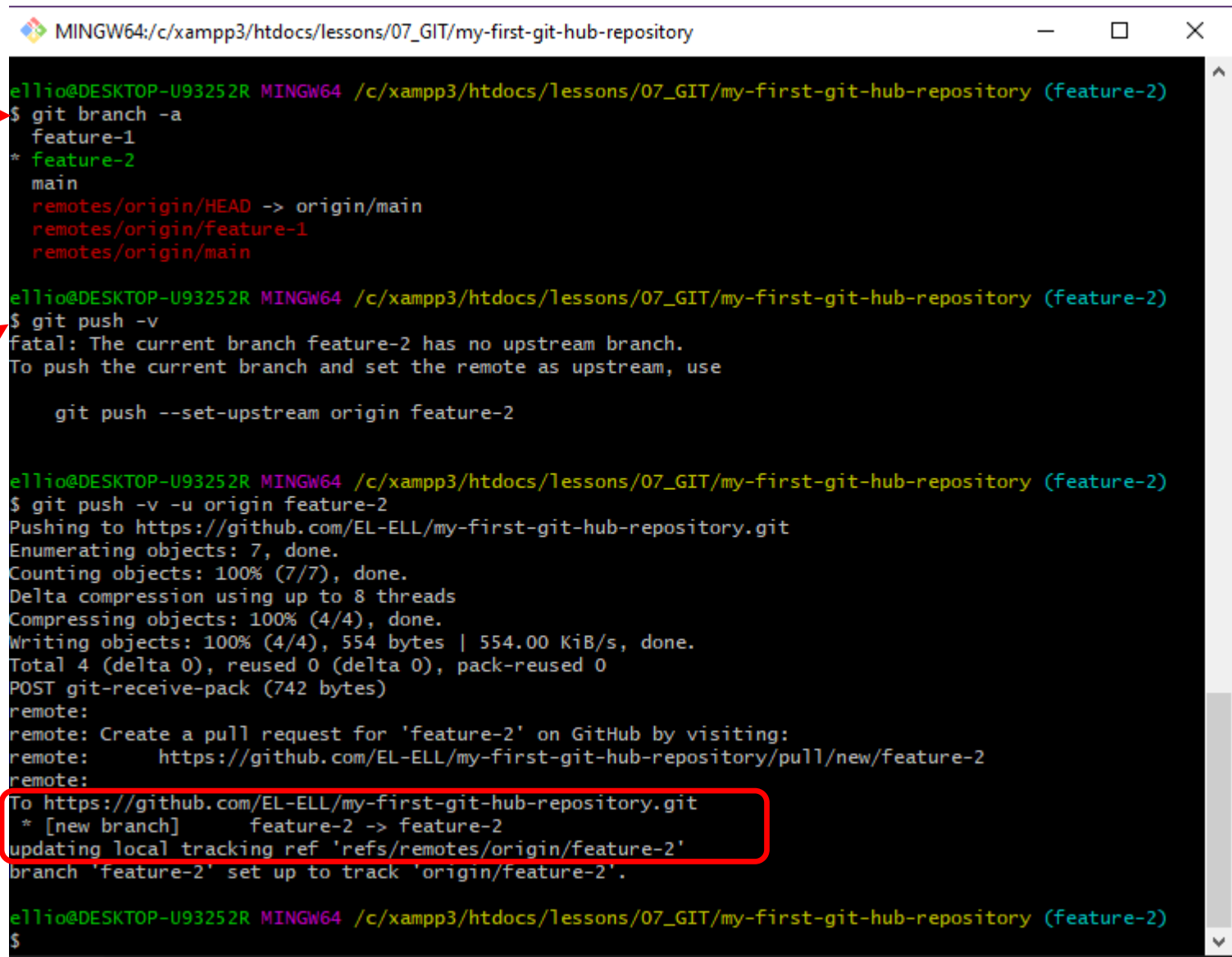
We have to manually create the new branch in the remote repository.



git branch -a shows the local branches of main, feature-1 and feature-2. However in the local repository there is only main and feature-1 branches.

If we try a normal git push then it fails because feature-2 branch has no upstream branch in the remote repository.

Git push -v -u origin feature-2 command can be used to specify the upstream path of feature-2 which will create it in the remote repository.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch -a
feature-1
* feature-2
main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git push -v
fatal: The current branch feature-2 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature-2

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git push -v -u origin feature-2
Pushing to https://github.com/EL-ELL/my-first-git-hub-repository.git
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 554 bytes | 554.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (742 bytes)
remote:
remote: Create a pull request for 'feature-2' on GitHub by visiting:
remote:   https://github.com/EL-ELL/my-first-git-hub-repository/pull/new/feature-2
remote:
To https://github.com/EL-ELL/my-first-git-hub-repository.git
 * [new branch]      feature-2 -> feature-2
 updating local tracking ref 'refs/remotes/origin/feature-2'
branch 'feature-2' set up to track 'origin/feature-2'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$
```

We can see in the command output that new branch of feature-2 has been pushed to remote repository and that local tracking refs/remotes/origin/feature-2 has been updated.

git branch -vv now shows that feature-2 branch is tracked with its corresponding branch in the remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch -vv
feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
* feature-2 9b83790 [origin/feature-2] another-file.txt was changed in feature-2 branch
main      185aed4 [origin/main] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$
```

And committed

Another file is modified in feature-2 branch

Now when we push changes we can just use **git push** command as all branches are tracked.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git push -v
Pushing to https://github.com/EL-ELL/my-first-git-hub-repository.git
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 547 bytes | 547.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (735 bytes)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/EL-ELL/my-first-git-hub-repository.git
  9b83790..51f6b86 feature-2 -> feature-2
updating local tracking ref 'refs/remotes/origin/feature-2'
```

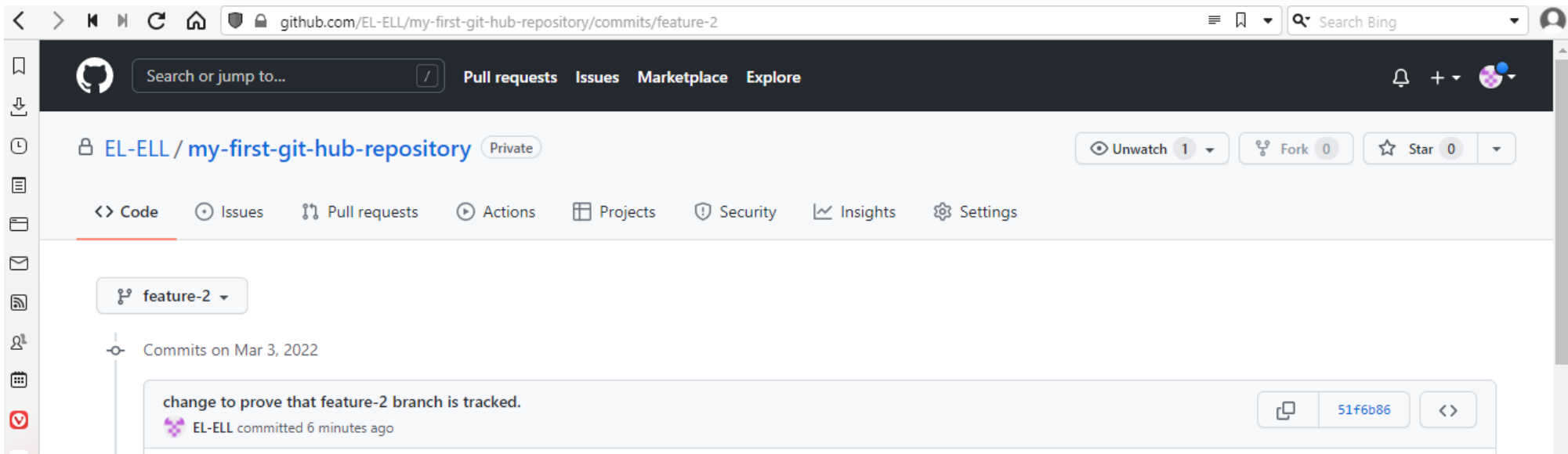
git log shows the latest commit pointing to head -> feature-2 in both local and remote repositories.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
$ git log
commit 51f6b860f99dde8102dc2c753a563e012d400d4f (HEAD -> feature-2, origin/feature-2)
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 18:23:43 2022 +0100

    change to prove that feature-2 branch is tracked.

commit 9b83790f46535c833ad318c46d547ac36d70ab7d
Author: Elliott Farmer <BarcelonaTravelHacks@gmail.com>
Date: Thu Mar 3 18:04:54 2022 +0100

    another-file.txt was changed in feature-2 branch
```

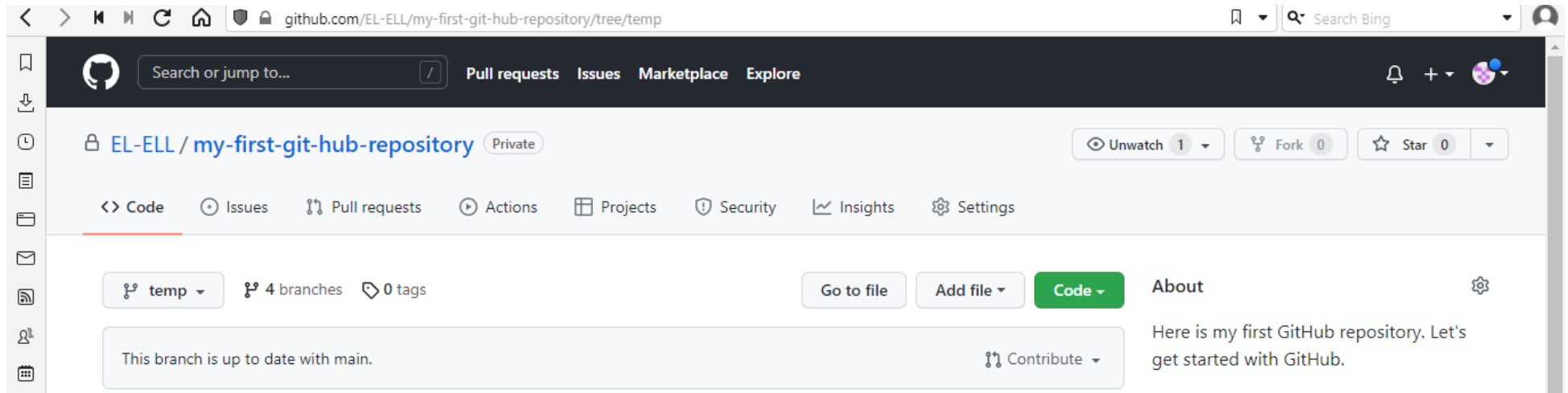


Once again the command to push a new branch to upstream remote repository is:

upstream
↓
Git push -v -u origin <remote branch name>
↑
Remote repository name

Update tracking statuses

1. Create new branch **temp** in the remote repository
2. Checkout this new branch locally using **git checkout temp**. Local tracking branch will be created.
3. Delete **temp** branch in the remote repository.
4. Use **git fetch**. After fetching we will observe tracking statuses **git branch -vv** and see that local temp branch still tracks remote **temp** branch.
5. Update tracking statuses using **git update origin --prune**.
5. Observe tracking again statuses **git branch -vv**



New **temp** branch is created in remote repository.

git branch -a shows the local branches of main, feature-1 and feature-2 but not the **temp** branch created in remote repository.

git fetch will add the branch to the remote repository.

git fetch only writes changes to remote repository so the temp branch will still not show up in the branch list.

git checkout <branch name> command has to be used add it to the staging area and working area of the local repository.

git branch -a now shows the new temp branch

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch -a
feature-1
* feature-2
main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git fetch
From https://github.com/EL-ELL/my-first-git-hub-repository
* [new branch]      temp      -> origin/temp

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git branch -a
feature-1
* feature-2
main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
remotes/origin/temp

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git checkout temp
Switched to a new branch 'temp'
branch 'temp' set up to track 'origin/temp'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -a
feature-1
feature-2
main
* temp
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
remotes/origin/temp

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$
```


github.com/EL-ELL/my-first-git-hub-repository/branches

Search or jump to... Pull requests Issues Marketplace Explore

EL-ELL / my-first-git-hub-repository Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

Overview Yours Active Stale All branches Search branches...

Default branch

main Updated 2 days ago by EL-ELL Default

Your branches

temp Updated 2 days ago by EL-ELL 0 | 0 New pull request Delete this branch

feature-2 Updated 1 hour ago by Elliott Farmer 0 | 10 New pull request

feature-1 Updated 13 hours ago by Elliott Farmer

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

The trash can icon can be used to delete temp branch

Delete this branch

git branch -a still shows the temp branch in the local repository even after deletion in the remote repository.

git branch -vv still shows the temp branch is tracked from the local repository.

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -a
feature-1
feature-2
main
* temp
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
remotes/origin/temp

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
main 185aed4 [origin/main] Create hello-github.txt
* temp 185aed4 [origin/temp] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$
```

git fetch -v updates from the remote repository to the local repository. Note how all the branches are updated minus the deleted temp branch.

git branch -vv shows that temp is still being tracked.

git remote update origin command does not remove temp from the local repository.

git remote update origin command needs to include **--prune** to remove temp.

git branch -vv now shows that temp branch has been flagged as “gone” from local repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git fetch -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-ELL/my-first-git-hub-repository
= [up to date]      feature-1  -> origin/feature-1
= [up to date]      feature-2  -> origin/feature-2
= [up to date]      main       -> origin/main

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
main      185aed4 [origin/main] Create hello-github.txt
* temp     185aed4 [origin/temp] Create hello-github.txt

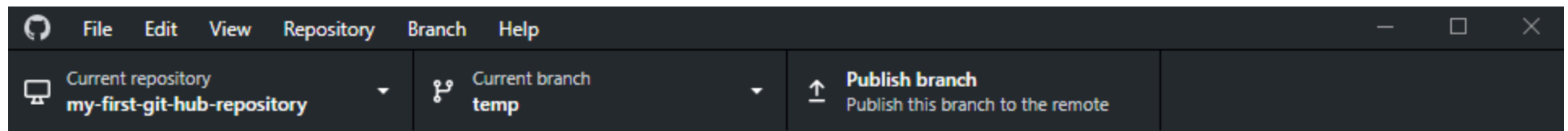
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote update origin
Fetching origin

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
main      185aed4 [origin/main] Create hello-github.txt
* temp     185aed4 [origin/temp] Create hello-github.txt

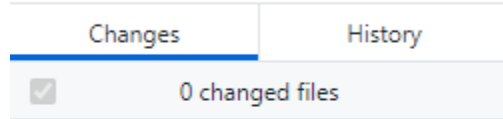
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git remote update origin --prune
Fetching origin
From https://github.com/EL-ELL/my-first-git-hub-repository
- [deleted]      (none)      -> origin/temp

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
main      185aed4 [origin/main] Create hello-github.txt
* temp     185aed4 [origin/temp: gone] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$
```



An updated version of GitHub Desktop is available and will be installed at the next launch. See [what's new](#) or [restart GitHub Desktop](#).



Note that in GitHub desktop it now gives us the option to publish temp branch to the remote repository. This is because temp branch is not deleted in the local repository, only the tracking status has been updated to remove it (prune) from tracking.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Publish your branch

The current branch (temp) hasn't been published to the remote yet. By publishing it you can share it, and collaborate with others.

Always available in the toolbar or **Ctrl P**

Publish branch

Now we can jump into the main branch and delete temp branch.

Now it is completely removed from local repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -d temp
Deleted branch temp (was 185aed4).

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -vv
  feature-1 9f1d8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
  feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
* main      185aed4 [origin/main] Create hello-github.txt
```

Remove remote branch from local terminal

1. Create new branch **temp** in the remote repository
2. Push this branch to remote repository using **git push -u origin temp**
3. Delete remote branch temp using **git push origin -d temp**

We can create new branch temp in local repository with **git checkout -b temp**

git branch -a shows temp branch in local repository but not remote repository.

git push -u origin temp will push this new branch into the remote repository

And in **git branch -vv** we see that temp is tracked in the remote repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git checkout -b temp
Switched to a new branch 'temp'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch
feature-1
feature-2
main
* temp

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -a
feature-1
feature-2
main
* temp
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git push -u origin temp
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'temp' on GitHub by visiting:
remote:   https://github.com/EL-ELL/my-first-git-hub-repository/pull/new/temp
remote:
To https://github.com/EL-ELL/my-first-git-hub-repository.git
 * [new branch]      temp -> temp
branch 'temp' set up to track 'origin/temp'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -vv
feature-1 9fd8db [origin/feature-1] Merge branch 'feature-1' of https://github.com/EL-ELL/my-first-git-hub-repository into feature-1
feature-2 51f6b86 [origin/feature-2] change to prove that feature-2 branch is tracked.
main      185aed4 [origin/main] Create hello-github.txt
* temp     185aed4 [origin/temp] Create hello-github.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ |
```

Temp branch is now in the remote repository.

The screenshot shows the GitHub interface for a repository named 'my-first-git-hub-repository' by user 'EL-ELL'. The 'branches' tab is selected, displaying a table of branches. The 'temp' branch is highlighted with a red arrow. The table also shows 'feature-2' and 'feature-1' branches with their respective commit differences from the 'main' branch.

Branch	Updated	By	Ahead of main	Behind main	Actions
main	Updated 2 days ago	EL-ELL	0	0	[Default]
temp	Updated 2 days ago	EL-ELL	0	0	[New pull request] [Edit] [Delete]
feature-2	Updated 2 hours ago	Elliott Farmer	0	10	[New pull request] [Edit] [Delete]
feature-1	Updated 14 hours ago	Elliott Farmer	0	8	[New pull request] [Edit] [Delete]

Note how GitHub shows the differences between the branches. For example feature-2 is 10 commits ahead of the main branch, feature-1 is 8 commits ahead of the main branch.

We can delete the temp branch in the remote repository with a push command.

The temp branch is now only present in the local repository.

Jump to main branch

And temp can be deleted from local repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git push origin -d temp
To https://github.com/EL-ELL/my-first-git-hub-repository.git
- [deleted]          temp

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git branch -a
feature-1
feature-2
main
* temp
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (temp)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -d temp
Deleted branch temp (was 185aed4).

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git branch -a
feature-1
feature-2
* main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$
```


git show-refs

command will show all refs files for local and remote (origin) repositories.

ls .git/refs/heads

Will show refs files for local repository only.

ls

.git/refs/remotes/origin

Will show refs files for remote repository only.

Or we can compare for each branch with a **git show-ref >branch name>** command. We can quickly see that the SHA1 hashes are the same for local and remote so that branch is synchronised.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git show-ref
9f1d8dbf66307291133142b56a6589532855141b refs/heads/feature-1
51f6b860f99dde8102dc2c753a563e012d400d4f refs/heads/feature-2
185aed41542260681f23894a94531e49bb4acbd5 refs/heads/main
185aed41542260681f23894a94531e49bb4acbd5 refs/remotes/origin/HEAD
9f1d8dbf66307291133142b56a6589532855141b refs/remotes/origin/feature-1
51f6b860f99dde8102dc2c753a563e012d400d4f refs/remotes/origin/feature-2
185aed41542260681f23894a94531e49bb4acbd5 refs/remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ ls .git/refs/heads
feature-1 feature-2 main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ ls .git/refs/remotes/origin
HEAD feature-1 feature-2

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git show-ref main
185aed41542260681f23894a94531e49bb4acbd5 refs/heads/main
185aed41542260681f23894a94531e49bb4acbd5 refs/remotes/origin/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git show-ref feature-1
9f1d8dbf66307291133142b56a6589532855141b refs/heads/feature-1
9f1d8dbf66307291133142b56a6589532855141b refs/remotes/origin/feature-1

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git show-ref feature-2
51f6b860f99dde8102dc2c753a563e012d400d4f refs/heads/feature-2
51f6b860f99dde8102dc2c753a563e012d400d4f refs/remotes/origin/feature-2

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ |
```


Jump to feature-2 branch



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (main)
$ git checkout feature-2
Switched to branch 'feature-2'
Your branch is up to date with 'origin/feature-2'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$
```

Now we edit another-file.txt and save it then commit it to the local repository.

Before the changes these files have the same SHA1



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git show-ref feature-2
51f6b860f99dde8102dc2c753a563e012d400d4f refs/heads/feature-2
51f6b860f99dde8102dc2c753a563e012d400d4f refs/remotes/origin/feature-2

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$ git show-ref feature-2
06fbf853904af43adb6d57719e2abbc5f88236b refs/heads/feature-2
51f6b860f99dde8102dc2c753a563e012d400d4f refs/remotes/origin/feature-2

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/my-first-git-hub-repository (feature-2)
$
```

After the commit the SHA1s are different proving that local and remote are not synchronised in feature-2 branch.



Summary of Shell Commands

git remote

Check name of remote repository

git remote -v

Show URL of remote repository

git branch -a

Shows local and remote repository branches

git branch -r

Shows remote repository branches

git branch -vv

Shows what branches are tracked between local and remote repositories

git remote show origin

Show detail of tracked branches

git fetch or git fetch -v

Is used to fetch branches from remote repository to local repository. -v gives verbose output.

git checkout <branch name>

Will checkout a branch from local repository to working area or if the branch is not in local repository but in remote repository it will checkout from remote repository to local working area.

Git remote prune origin

Will prune stale branches from the remote repository.

git remote update origin --prune

Prune tracking of branches in local repository.

git pull or git pull -v

Performs a pull -v options gives a verbose output.

git push or git push -v

Performs a push -v options gives a verbose output.

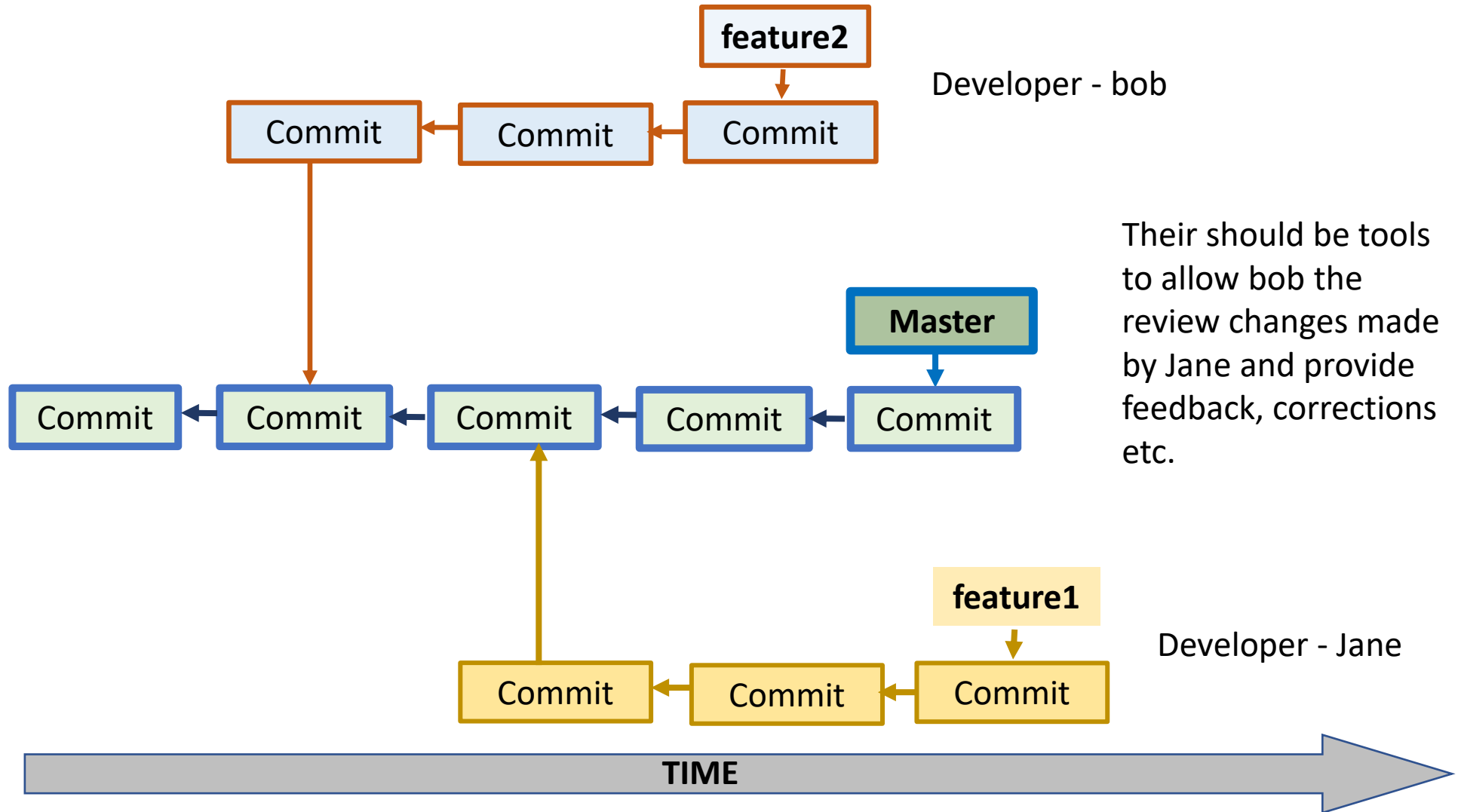
git push -v -u origin feature-2

Push update to remote repository creating new upstream branch.

GIT PULL REQUESTS

Collaboration Between Contributors

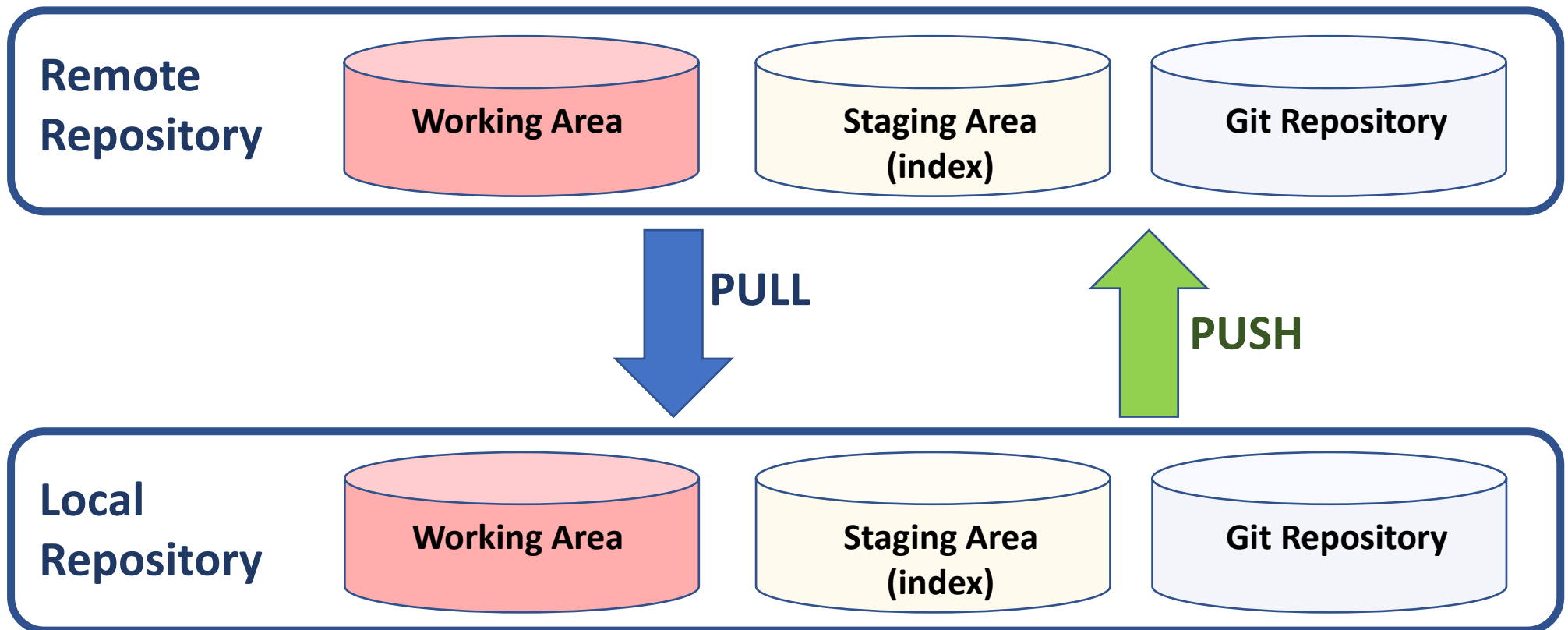
Multiple collaborators can work on the project simultaneously in different branches, and once complete with their part of the project can merge their branch into the master branch.



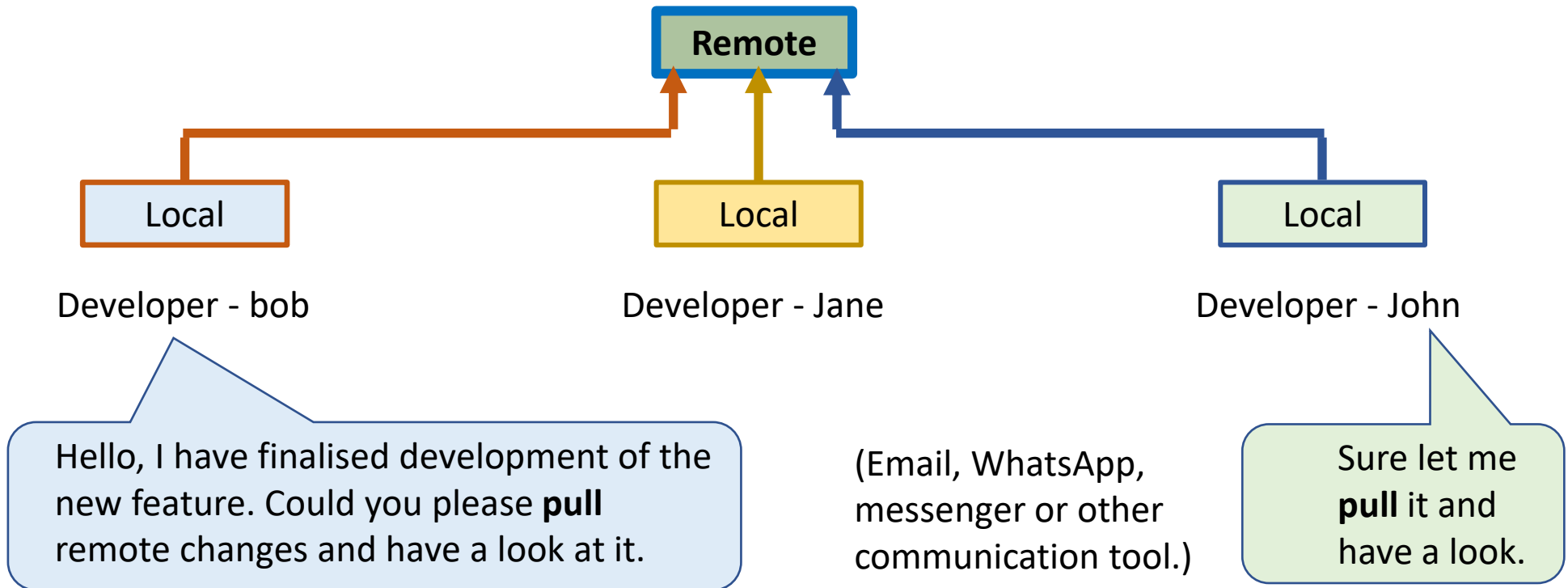
Pull Request

Pull request is a proposal of potential changes. Changes made by a collaborator may or may not be applied to the main or release branch.

Previously we have discovered how git pull and git push will move files from and to the remote repository to the working directory of the local repository.



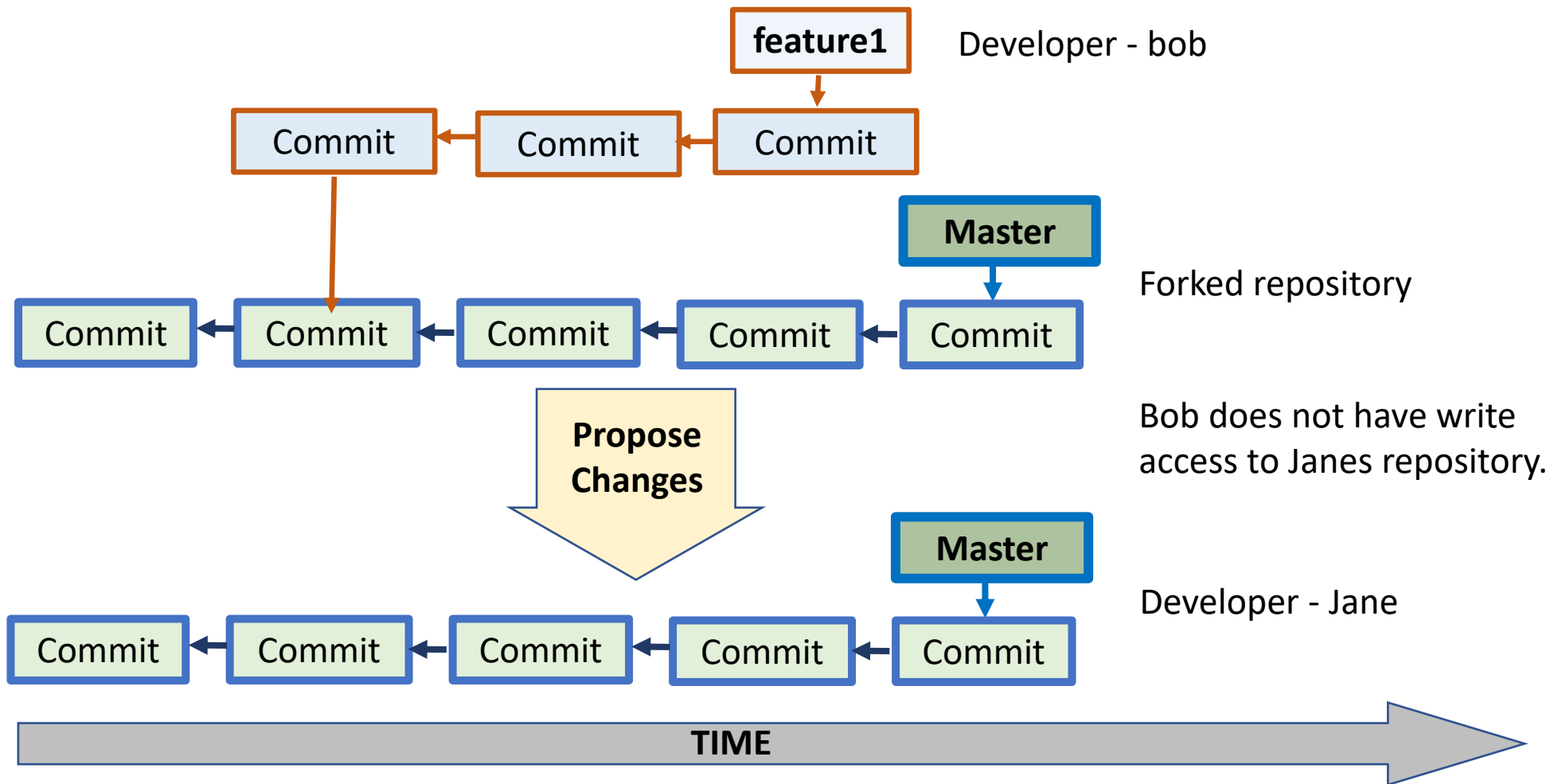
Why “Pull” Request?



A better name for pull request would be merge request because when a developer is working on a new feature in a separate branch he may not be able to merge these changes into the remote repository because in GitHub you can set a block on merges until changes of a developer have been approved by one or more other developers. There are multiple rights options that can be set to control merging of branches.

Collaboration Between Contributors

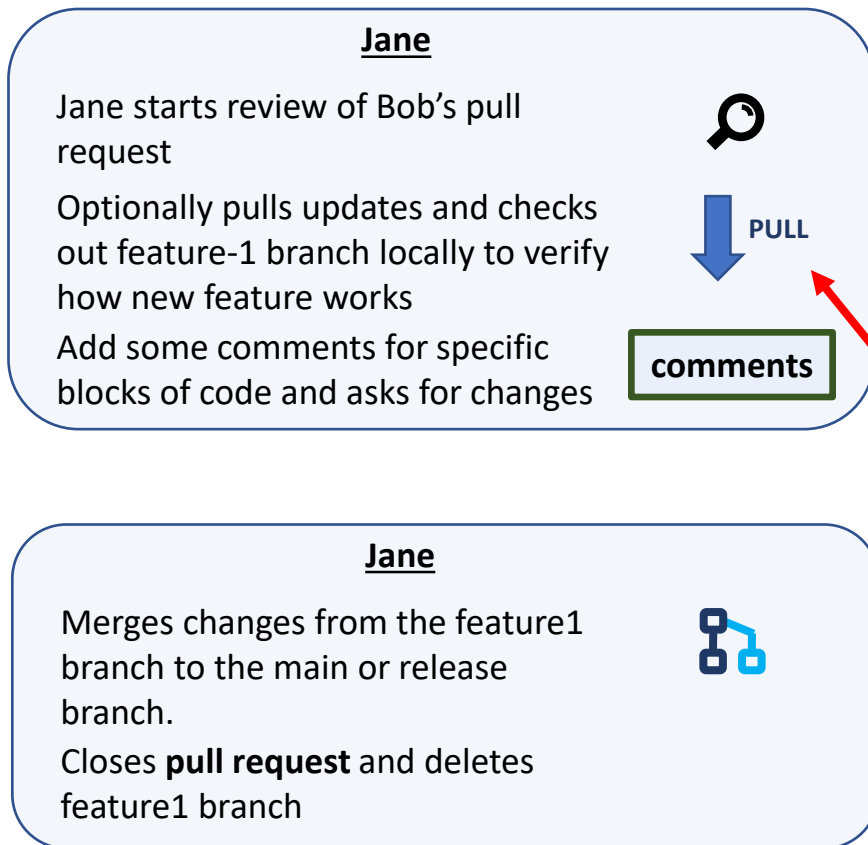
Jane is owner of public repository which many developers are contributing to. Any collaborators that want to contribute to this repository create a clone of Jane's repository and make their own changes. The collaborators do not have rights to write to Jan's repository, however Jane has the ability to merge changes from other collaborators. I.e. pull them into her repository.



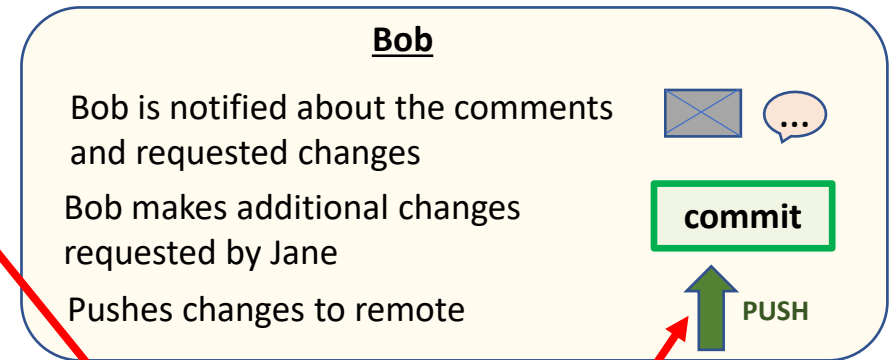
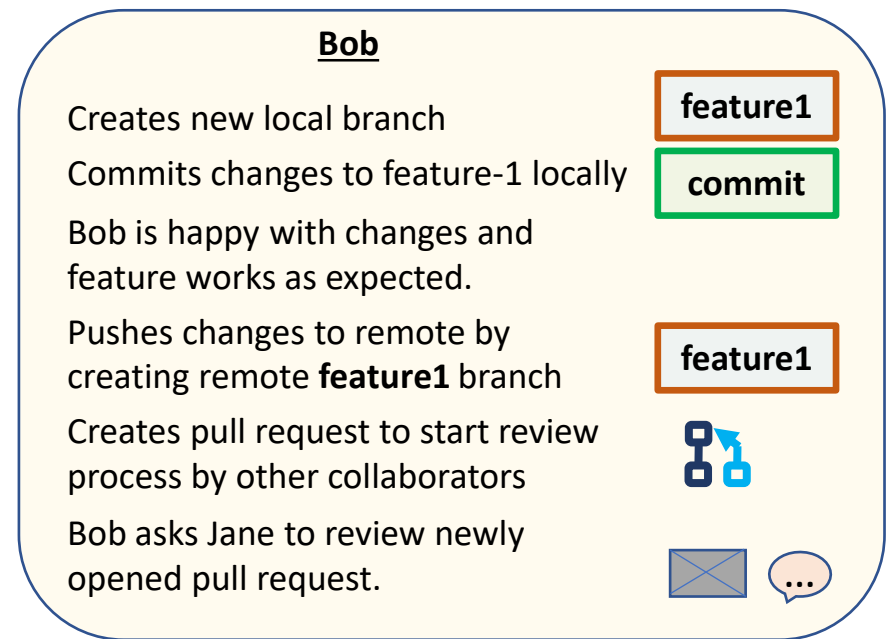
Pull Request Step by Step

In large projects several people may have merge permissions.

Often main branch is “production” branch and “release” is staging” version of the project.



TIME



Note Existing pull request is updated when new changes are pushed to it.

NEW FEATURE IMPLEMENTED

The screenshot shows the GitHub web interface for a repository named 'EL-EL-BCN/pull-requests'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows an initial commit by 'a1e5c5f' 1 minute ago, containing 'LICENSE' and 'README.md' files. The README file content is visible, showing the title 'pull-requests' and the text 'We will create different pull requests'. The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. A terminal window is overlaid on the bottom right, showing the command to clone the repository and the resulting directory listing.

github.com/EL-EL-BCN/pull-requests

Search or jump to...

Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests (Public)

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

About

We will create different pull requests

Readme 0 stars 1 watching 0 forks

Releases

EL-EL-BCN Initial commit a1e5c5f 1 minute ago 1 commit

File	Commit	Time
LICENSE	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

README.md

pull-requests

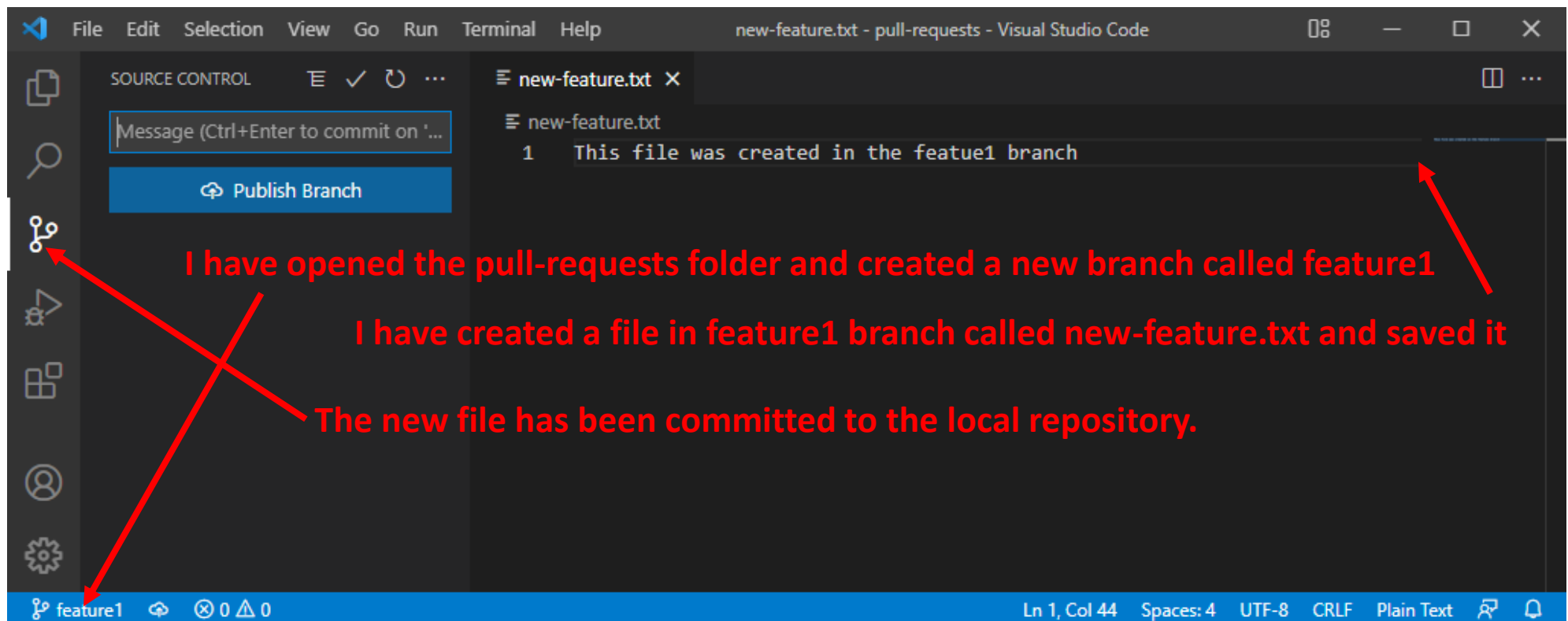
We will create different pull requests

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/pull-requests.git
Cloning into 'pull-requests'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ ls -la
total 32
drwxr-xr-x 1 ellio 197609 0 Mar  5 20:33 ./
drwxr-xr-x 1 ellio 197609 0 Feb 24 16:03 ../
drwxr-xr-x 1 ellio 197609 0 Mar  1 21:14 first-project/
drwxr-xr-x 1 ellio 197609 0 Feb 27 22:18 hello-world/
drwxr-xr-x 1 ellio 197609 0 Mar  3 20:35 my-first-git-hub-repository/
drwxr-xr-x 1 ellio 197609 0 Mar  5 20:33 pull-requests/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
```

A new repository called pull-requests is created in GitHub.com and cloned to the local machine.



The commit shows in the git log.

To amend author of commits first change git global settings using:

git config user.name "username"

git config user-email "email"

Then amend the commit

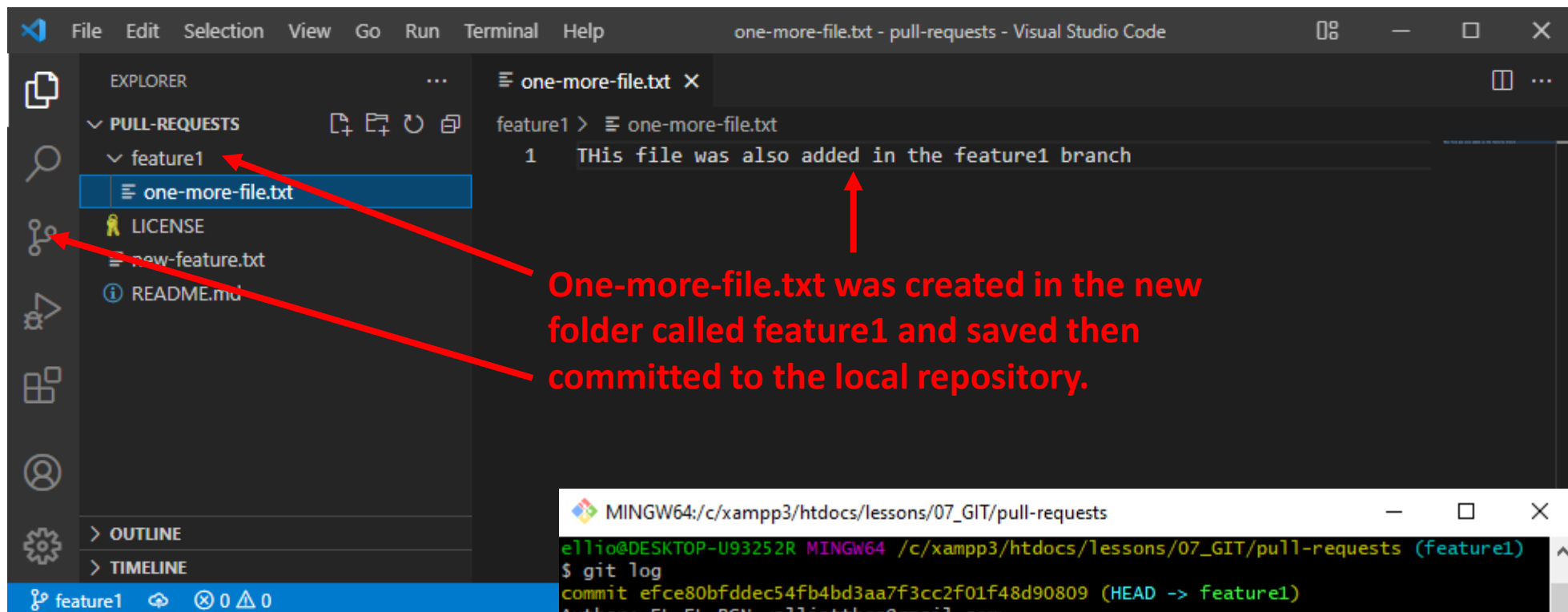
git commit --amend --author "username <email>"

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/pull-requests
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/pull-requests (feature1)
$ git log
commit 467e85d4986a83e69dbf325004c22c822542e23f (HEAD -> feature1)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Sat Mar 5 20:58:31 2022 +0100

    First file in the Feature1 branch

commit a1e5c5f6adf44f24489f4e70f8f46dfc86175dce (origin/main, origin/HEAD, main)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 5 20:32:35 2022 +0100

    Initial commit
```



Another file is committed to the local repository.

Then another change is made and committed.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/pull-requests
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/pull-requests (feature1)
$ git log
commit efce80bfddec54fb4bd3aa7f3cc2f01f48d90809 (HEAD -> feature1)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Sat Mar 5 21:21:45 2022 +0100

    Another change

commit 5e819f8c18d122b00778c1f7f3c92bbc16d7efa3
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Sat Mar 5 21:14:03 2022 +0100

    Another change in the feature1 branch

commit 467e85d4986a83e69dbf325004c22c822542e23f
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Sat Mar 5 20:58:31 2022 +0100

    First file in the Feature1 branch

commit a1e5c5f6adf44f24489f4e70f8f46dfc86175dce (origin/main, origin/HEAD, main)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 5 20:32:35 2022 +0100

    Initial commit
```

github.com/EL-EL-BCN/pull-requests/tree/feature1

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Clicking here we can create a pull request

feature1 had recent pushes 8 minutes ago

feature1 2 branches 0 tags

Go to file Add file Code

This branch is 3 commits ahead, 1 commit behind main.

Contribute

About

We will create different pull requests

Readme

0 stars

1 watching

0 forks

github.com/EL-EL-BCN/pull-requests/compare/feature1

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: main ← compare: feature1 ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

3 commits 2 files changed 1 contributor

Commits on Mar 5, 2022

First file in the Feature1 branch

EL-EL-BCN committed 1 hour ago 467e85d <>

Another change in the feature1 branch

EL-EL-BCN committed 1 hour ago 5e819f8 <>

Another change

EL-EL-BCN committed 1 hour ago efce80b <>

Showing 2 changed files with 4 additions and 0 deletions. Split Unified

3 feature1/one-more-file.txt

... @@ -0,0 +1,3 @@

1 + This file was also added in the feature1 branch
2 +
3 + One more change in feature1 branch

1 new-feature.txt

... @@ -0,0 +1 @@

1 + This file was created in the feature1 branch

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: feature1 ✓ Able to merge. These branches can be automatically merged.

[Feature1] first feature was implemented

Write Preview

Following changes have been made:

- new-feature-file.txt was added
- feature1 folder was created

Attach files by dragging & dropping, selecting or pasting them.

Create pull request → Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Use [Closing keywords](#) in the description to [Go to settings to activate Windows.](#)

Activate Windows
Go to settings to activate Windows.

22:30

It is advisable here to put the name of the branch in square brackets. If we view a list of pull requests they will stand out or if we use <https://atlassian.com/software.jira> then the pull requests will be correctly imported in to the Jira software

Add a detailed description of the changes made here

github.com/EL-EL-BCN/pull-requests/pull/1

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

[Feature1] first feature was implemented #1

Open EL-EL-BCN wants to merge 3 commits into main from feature1

Conversation 0 Commits 3 Checks 0 Files changed 2 +4 -0

EL-EL-BCN commented 1 minute ago

Following changes have been made:

- new-feature-file.txt was added
- feature1 folder was created

EL-EL-BCN added 3 commits 2 hours ago

- First file in the Feature1 branch
- Another change in the feature1 branch
- Another change

467e85d
5e819f8
efce80b

Reviews

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Activate Windows
Go to Settings to activate Windows.

Reset 100% 22:38

This page lets us review the pull request and shows the pull request status as open

Commits history of this pull request

Any checks that can be performed and some tools

Files that have been changed and what lines of the files

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

Watch 1 Fork 0 Star 0

Code Issues Pull requests 1 Actions Projects Wiki Security Insights

main 2 branches 0 tags

Go to file Add file Code

About

We will create different pull requests

Readme

0 stars

1 watching

0 forks

Releases

No releases published

EL-EL-BCN Delete LICENSE 7a72d27 1 hour ago 2 commits

README.md Initial commit 2 hours ago

README.md

pull-requests

We will create different pull requests

github.com/EL-EL-BCN/pull-requests/pull/1/files

[feature1] first feature was implemented #1

Open EL-EL-BCN wants to merge 3 commits into main from feature1

Conversation 0 Commits 3 Checks 0 Files changed 2 +4 -0

Changes from all commits File filter Conversations Jump to 0 / 2 files viewed Review changes

3 feature1/one-more-file.txt

... @@ -0,0 +1,3 @@

1 + This file was also added in the feature1 branch

2 +

3 + One more change in feature1 branch

1 new-feature.txt

... @@ -0,0 +1 @@

1 + This file was created in the feaue1 branch



Logging into GitHub via a different account I can view pull requests for the public repository called pull-requests

Under the files changed tab

I can hover over a particular line of code and click on the blue cross to add a comment

[Feature1] first feature was implemented #1


<> Code ▾

 Open EL-EL-BCN wants to merge 3 commits into `main` from `feature1` 



 Conversation 0  Commits 3  Checks 0  Files changed 2

+4 -0 

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ 

0 / 2 files viewed 



3  feature1/one-more-file.txt 

☐ Viewed ...

... @@ -0,0 +1,3 @@

**I can add a single comment or start a review.
A single comment is just that, a standalone
comment at a particular spot.**

**Start a review is a means to review the pull
request with multiple comments.**

**For each option the contributor will receive a
notification, i.e. one notification for multiple
comments or one notification for one
comment. In this case we will add a single
line of comment.**

```
1 + This file was also added in the feature1 branch
2 +
3 + One more change in feature1 branch
```

Write

Preview

 H B I  <>    
  

It is really god that you have added this line of code

Attach files by dragging & dropping, selecting or pasting them. 

Cancel

Add single comment



github.com/EL-EL-BCN/pull-requests/pull/1

Open [Feature1] first feature was implemented #1
EL-EL-BCN wants to merge 3 commits into `main` from `feature1`

Another change

EL-ELL reviewed 2 minutes ago

feature1/one-more-file.txt

```
... @@ -0,0 +1,3 @@
1 + This file was also added in the feature1 branch
2 +
3 + One more change in feature1 branch
```

EL-ELL 2 minutes ago
It is really god that you have added this line of code

Reply...

Resolve conversation

efce80b

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications
Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

Here we see the comment under conversation tab

github.com/EL-EL-BCN/pull-requests/pull/1/files

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

[Feature1] first feature was implemented #1

Open EL-EL-BCN wants to merge 3 commits into `main` from `feature1`

Conversation 1 Commits 3 Checks 0 Files changed 2

Changes from all commits File filter Conversations Jump to

0 / 2 files viewed

Review changes

Under the files changed tab I can review changes.

github.com/EL-EL-BCN/pull-requests/pull/1/files

Search Bing

Conversation 1Commits 3Checks 0Files changed 2

+4 -0

Changes from all commitsFile filterConversationsJump to0 / 2 files viewedReview changes

3 feature1/one-more-file.txt

... @@ -0,0 +1,3 @@

I can comment or Approve changes or request changes

1 new-feature.txt

... @@ -0,0 +1 @@

Finish your review

WritePreviewHBI≡<>🔗⋮⋮☑@🗨️↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Comment

Submit general feedback without explicit approval.

☐ Approve

Submit feedback and approve merging these changes.

☐ Request changes

Submit feedback that must be addressed before merging.

Submit review

1 + This file was created in the featue1 branch

Activate Windows
Go to Settings to activate Windows.

Reset100 %23:09

github.com/EL-EL-BCN/pull-requests/pull/1

Merged [Feature1] first feature was implemented #1
EL-EL-BCN merged 3 commits into main from feature1 now

EL-EL-BCN reviewed 40 seconds ago

EL-EL-BCN left a comment

Good job EL-EL-BCN

EL-EL-BCN merged commit 1ca6096 into main now

Pull request successfully merged and closed
You're all set—the feature1 branch can be safely deleted.

Delete branch

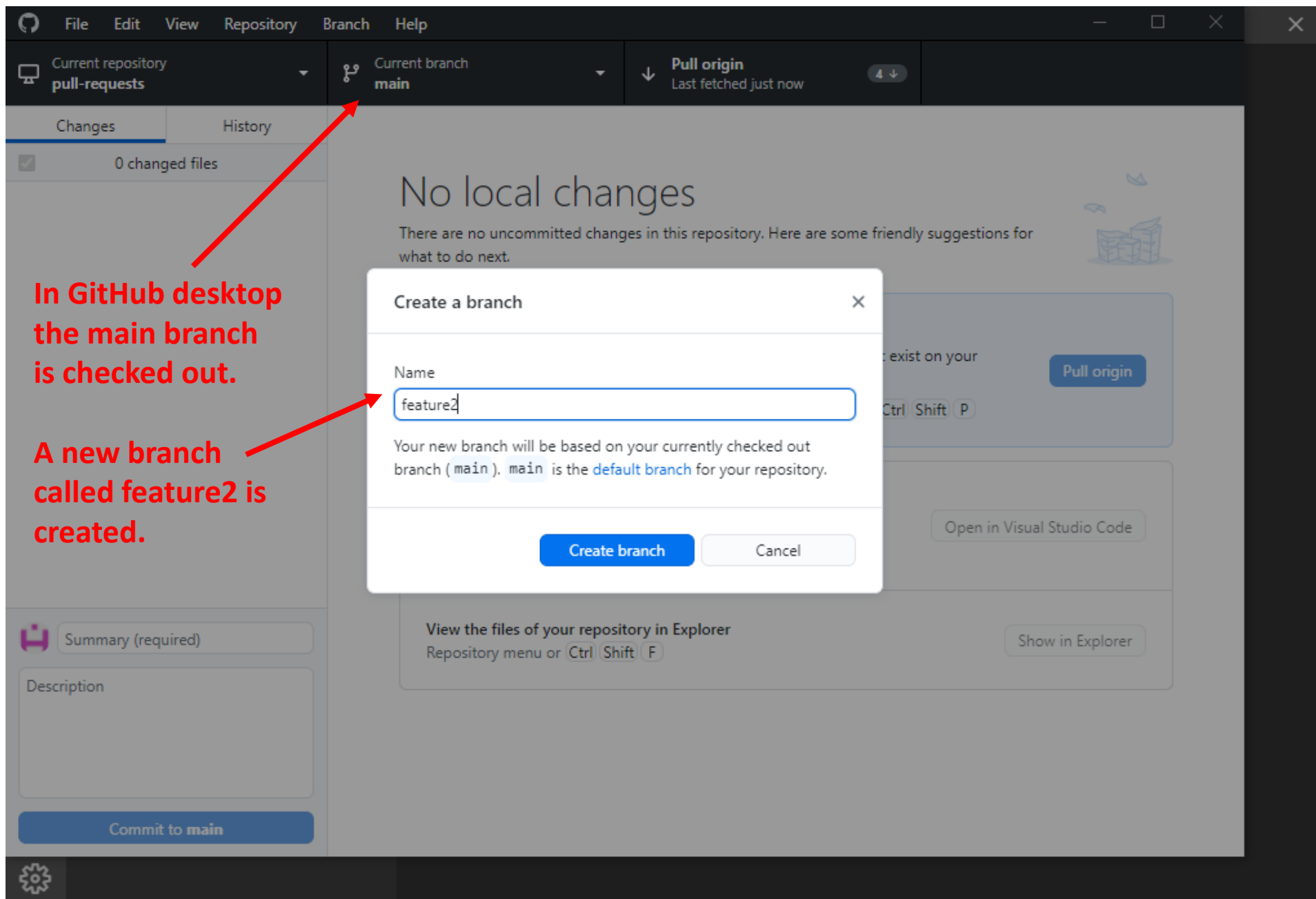
Write Preview

Leave a comment

Activate Windows
Go to Settings to activate Windows.

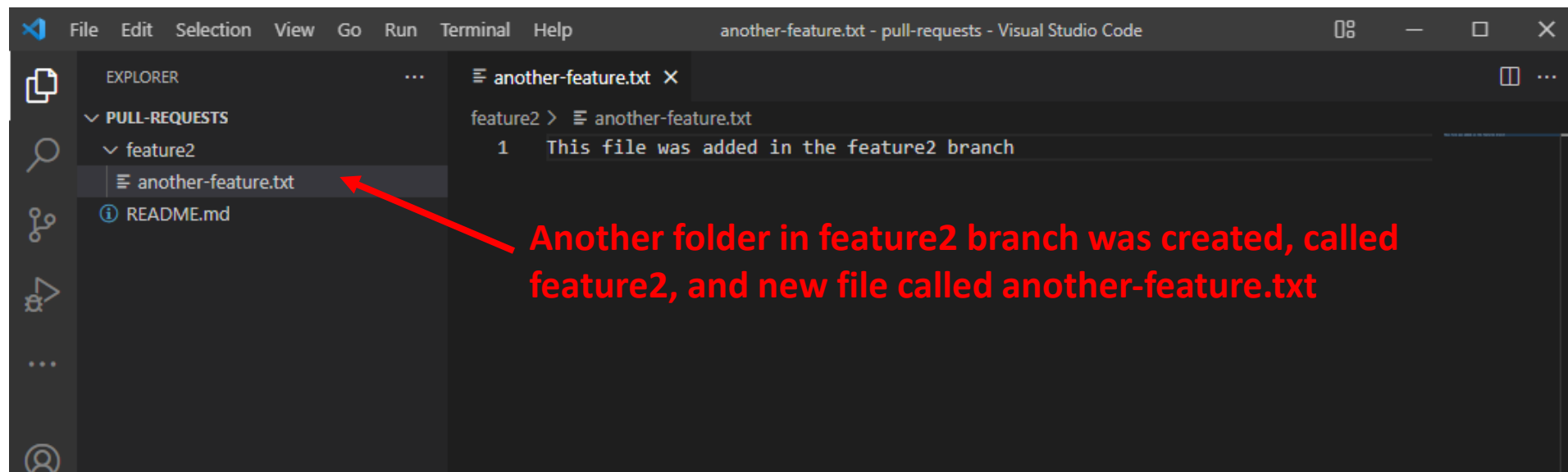
Reset 100% 23:12

I can merge changes. It gives me the option of deleting branch after the merge.

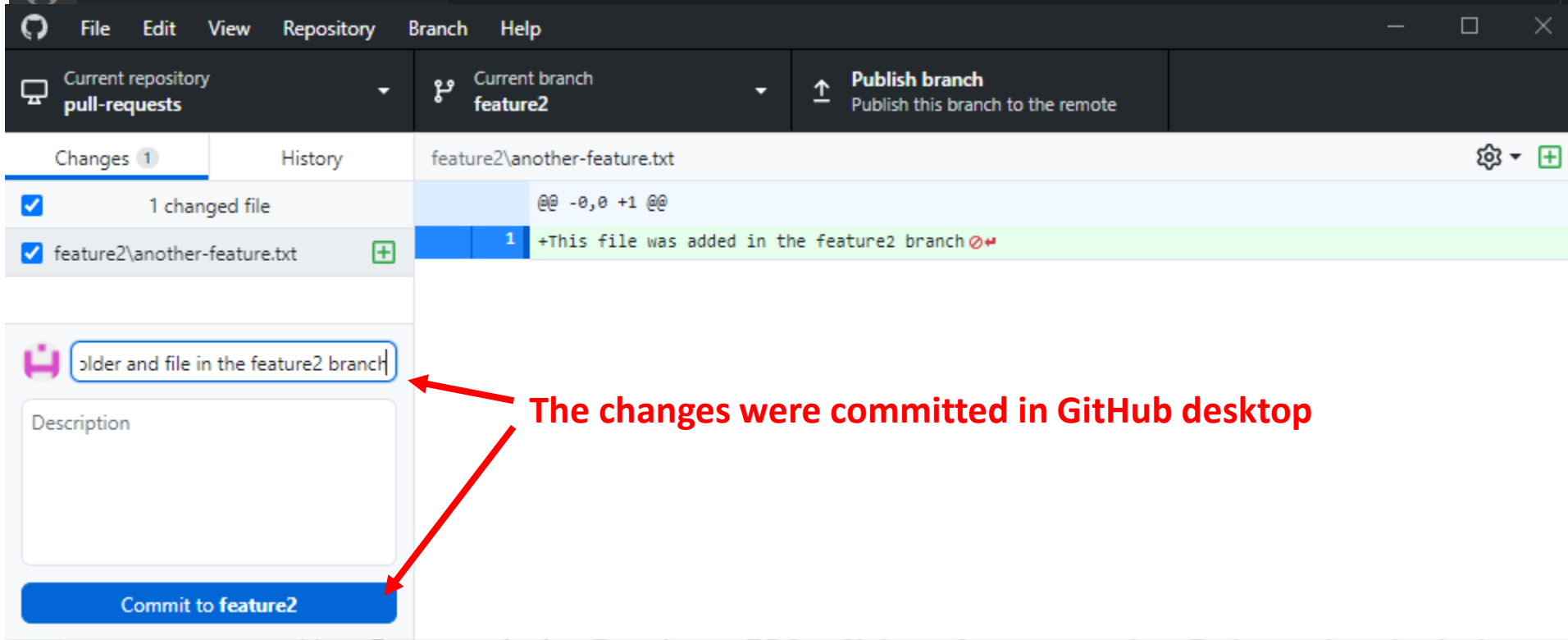


In GitHub desktop
the main branch
is checked out.

A new branch
called feature2 is
created.



Another folder in feature2 branch was created, called feature2, and new file called another-feature.txt



The changes were committed in GitHub desktop

File Edit View Repository Branch Help

Current repository pull-requests

Current branch feature2

Publish branch
Publish this branch to the remote

Changes History

Select branch to compare...

another change in feature2 bra...
EL-EL-BCN • just now

Added folder and file in the fea...
EL-EL-BCN • 1m

Delete LICENSE
EL-EL-BCN, GitHub • 2h

Initial commit
EL-EL-BCN, GitHub • 2h

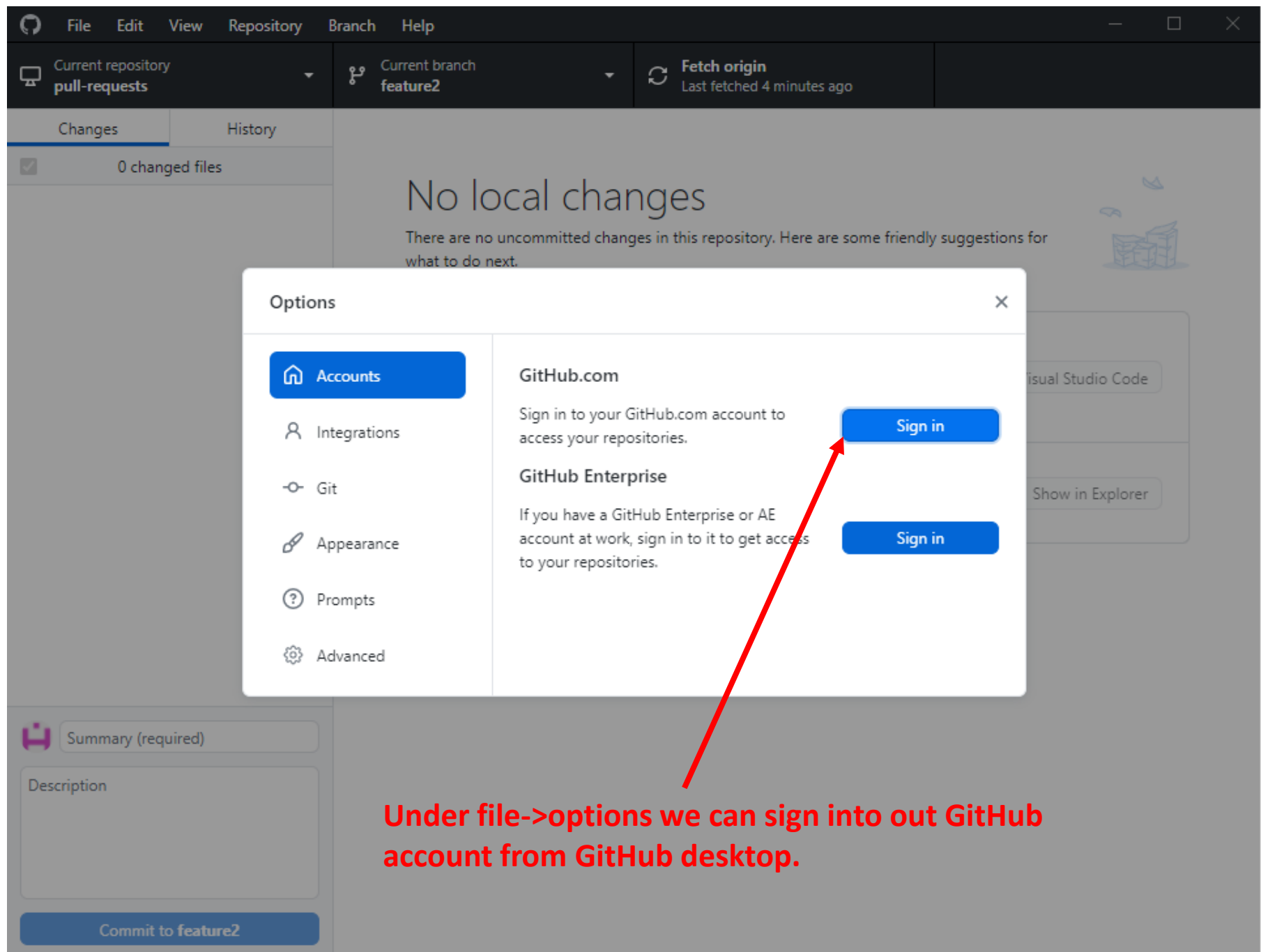
another change in feature2 branch

EL-EL-BCN 1d11f4a ± 1 changed file +3 -1

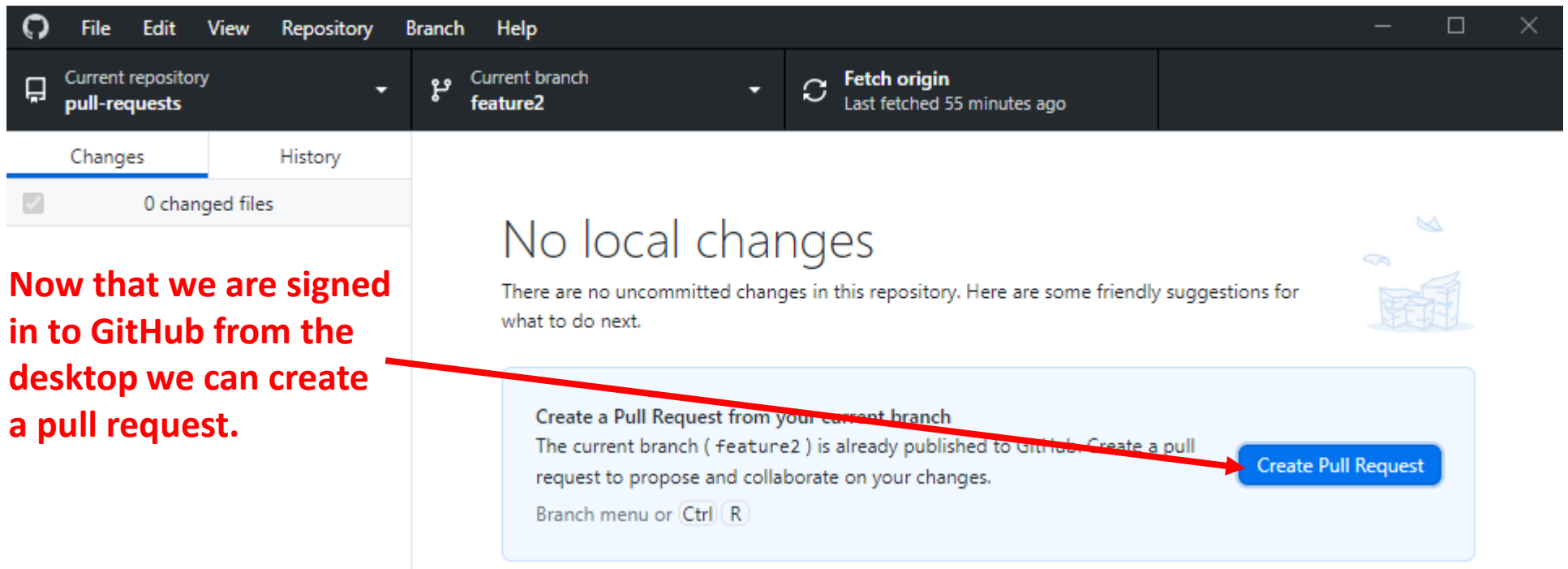
feature2\another-feature.txt

@@ -1 +1,3 @@
1 -This file was added in the feature2 branch
1 +This file was added in the feature2 branch
2 +
3 +another line

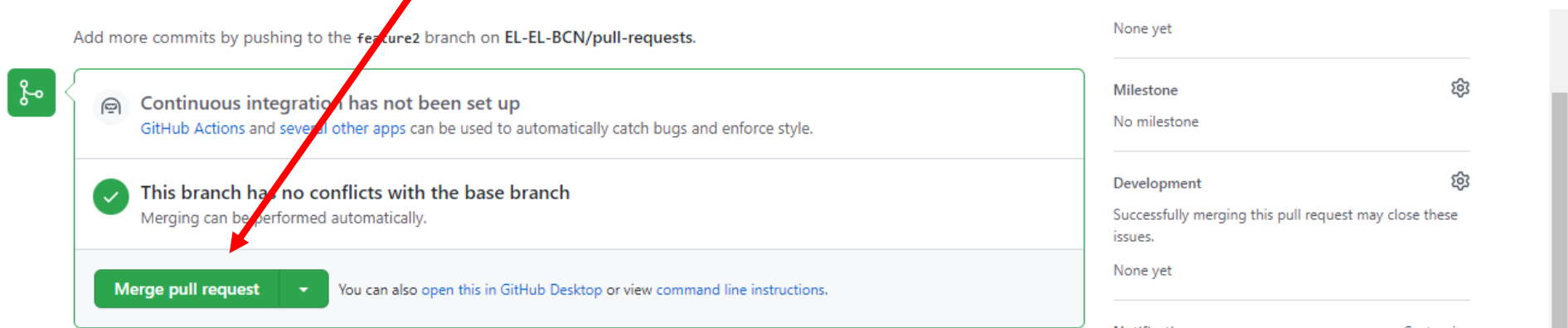
Another change was made and committed and pushed via GitHub desktop.

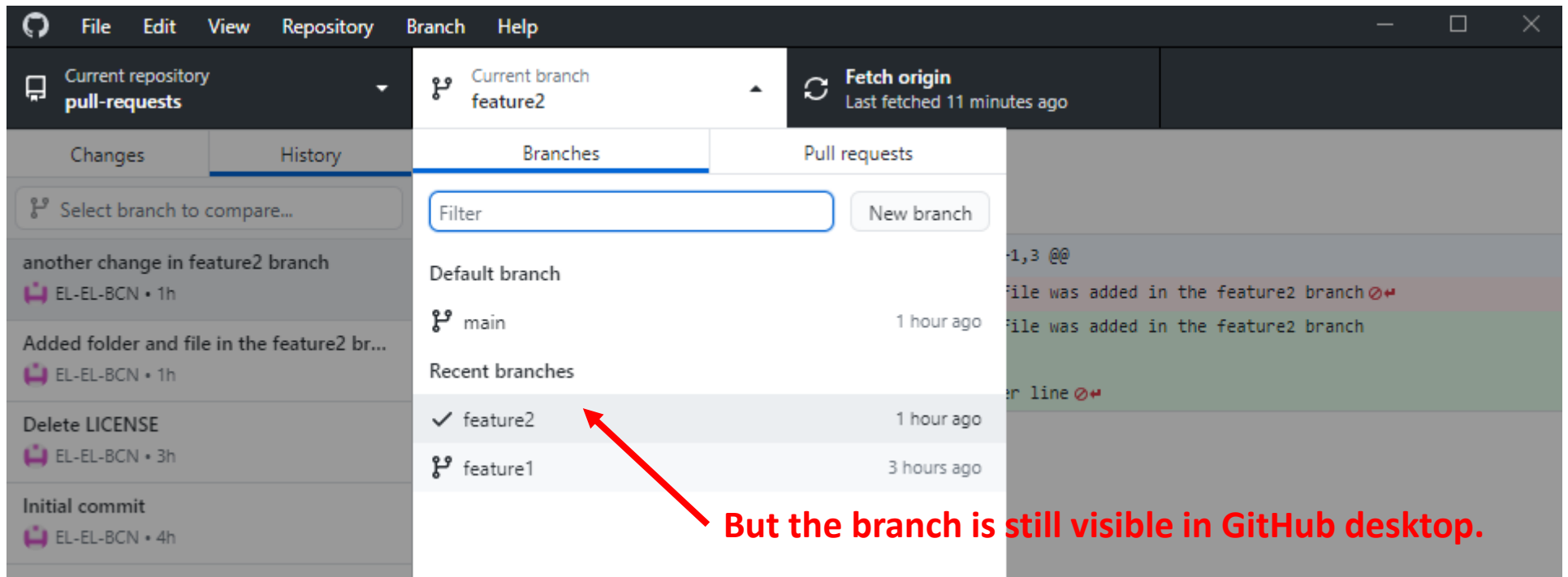


Under file->options we can sign into our GitHub account from GitHub desktop.

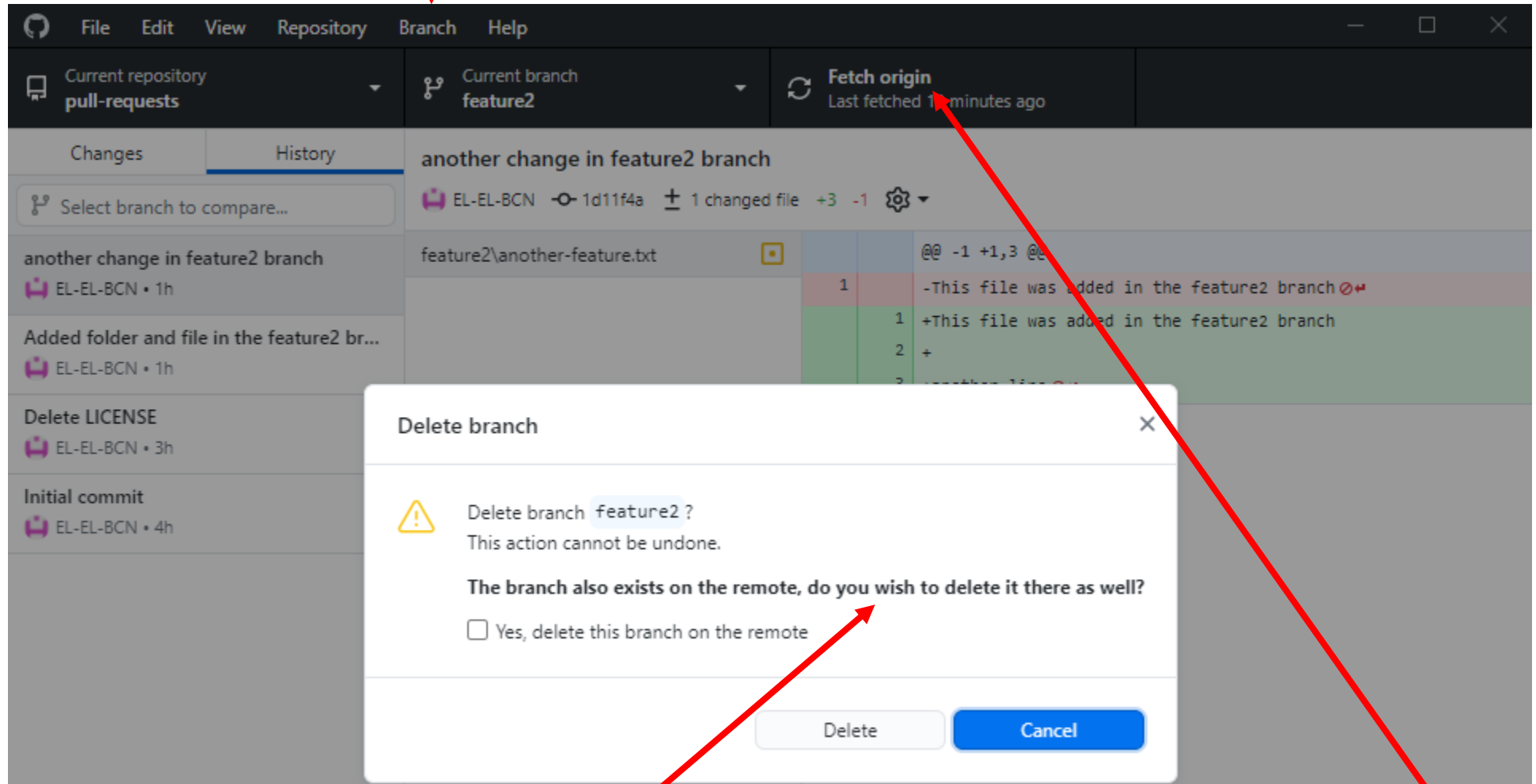


Note that in github.com pull requests can be merged without approval by anyone else. This is the default behaviour.



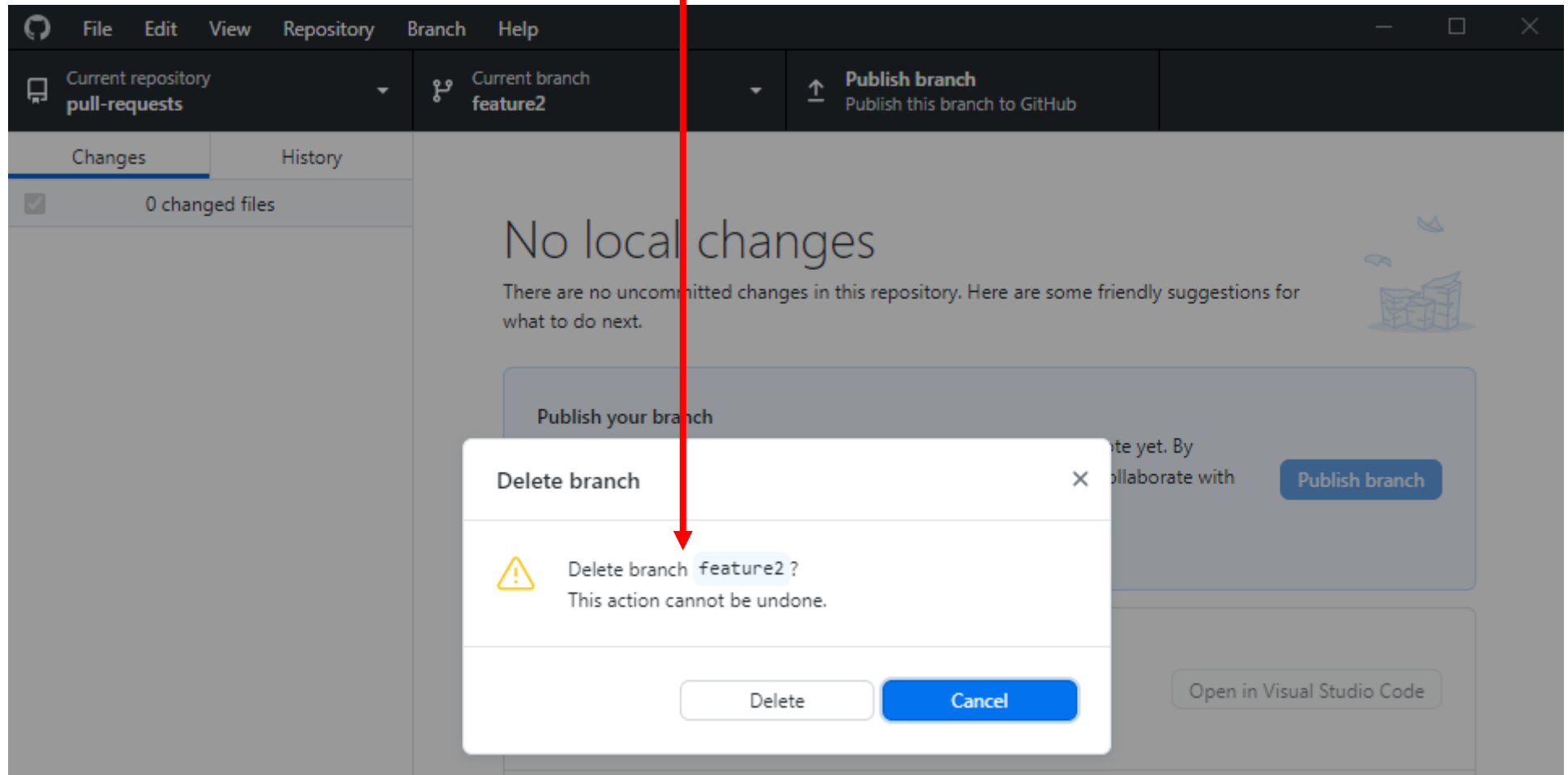


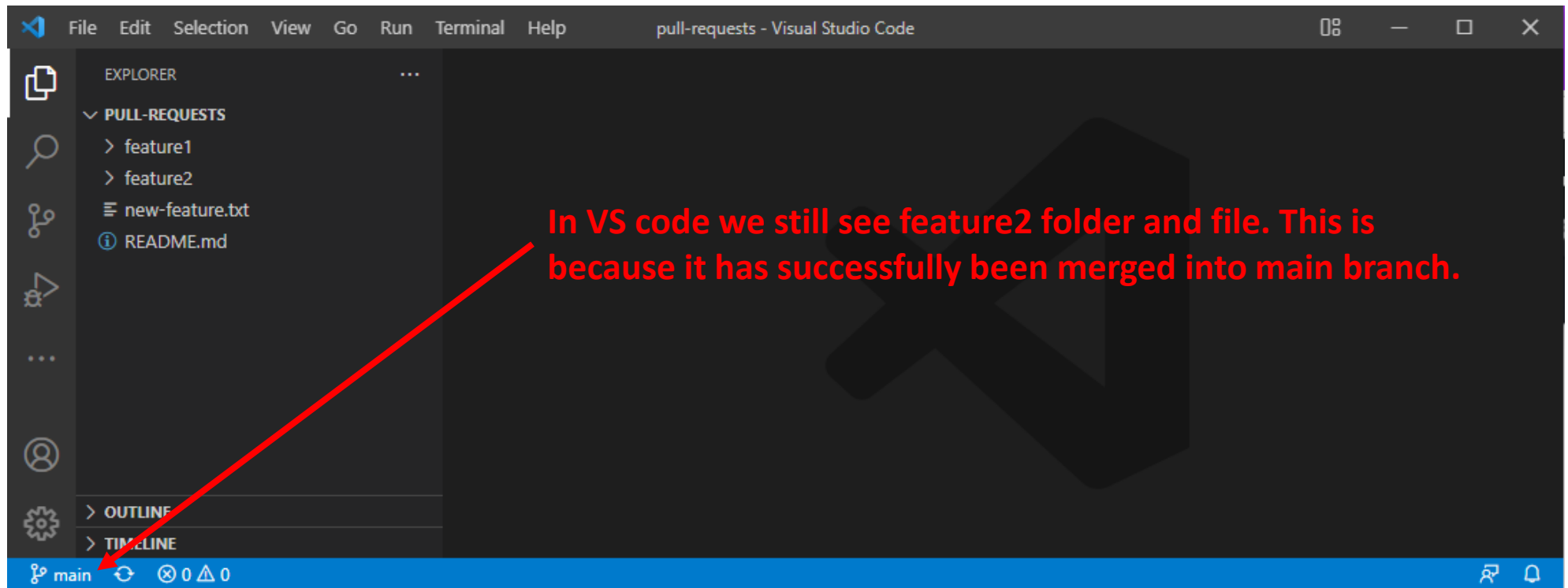
We can manually delete the feature2 branch locally by using the menu option branch->delete.



Note how GitHub desktop still thinks that the feature2 branch is still in the remote repository and asks us if we would like to delete it. This is because the local repository has not been updated with a fetch origin.

Now feature2 branch can be safely deleted.





Under the settings tab we can add collaborators to this repository.

The screenshot shows the GitHub interface for the repository 'EL-EL-BCN / pull-requests'. The 'Settings' tab is selected in the top navigation bar. In the left sidebar, under the 'Access' section, the 'Collaborators' link is highlighted with a red arrow. The main content area is titled 'Who has access' and contains two panels: 'PUBLIC REPOSITORY' (indicating the repository is public and visible to anyone) and 'DIRECT ACCESS' (showing 0 collaborators). Below these is a 'Manage access' section with a message: 'You haven't invited any collaborators yet'.

When I add a collaborator they receive an email with a link in it to confirm their participation.

The screenshot displays the GitHub interface for the repository `EL-EL-BCN/pull-requests`. The left sidebar contains navigation links: General, Access, Collaborators, Moderation options, Code and automation, Branches, Actions, Webhooks, Environments, Pages, Security, Code security and analysis, Deploy keys, Secrets, Integrations, GitHub apps, and Email notifications. The main content area is titled "Who has access" and includes sections for "PUBLIC REPOSITORY" and "DIRECT ACCESS". Below these is the "Manage access" section, which features a search bar "Find a collaborator..." and a list of collaborators. One collaborator, `MS-JM-ACC`, is listed with the status "Awaiting MS-JM-ACC's response". A red arrow points from the "Add people" button in the "Manage access" section to a red-bordered box on the right. This box contains a preview of an email invitation from `@EL-EL-BCN` to collaborate on the `EL-EL-BCN/pull-requests` repository. The email text includes instructions on how to accept or decline the invitation and a link to the repository. A "View invitation" button is located at the bottom of the email preview.

https://github.com/EL-EL-BCN/pull-requests/settings/access?guidance_task=

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)


DIRECT ACCESS

1 has access to this repository. 0 collaborators. 1 invitation.



Manage access

[Add people](#)

☐ Select all Type ▾

<input type="checkbox"/>	 MS-JM-ACC Awaiting MS-JM-ACC's response
--------------------------	--

[< Previous](#)

 + 

@EL-EL-BCN has invited you to collaborate on the **EL-EL-BCN/pull-requests** repository

You can [accept](#) or [decline](#) this invitation. You can also head over to <https://github.com/EL-EL-BCN/pull-requests> to check out the repository or visit [@EL-EL-BCN](#) to learn a bit more about them.

This invitation will expire in 7 days.

[View invitation](#)

GitHub user MS-JM-ACC has created a new branch, feature3 branch and created a pull request.

The screenshot shows a GitHub pull request page for the repository `EL-EL-BCN`. The pull request is titled `[feature3] some new changes from MS-JM-ACC #4` and is currently open. It shows 1 commit merged from the `feature3` branch into the `main` branch. A comment from the user `MS-JM-ACC` states: "This feature was added by another collaborator". Below the comment, a file change is listed: "This file was added by MS-JM-ACC" with a verified commit hash of `4dd7df0`. A green status box at the bottom indicates: "This branch has no conflicts with the base branch. Merging can be performed automatically." On the right side, the user profile dropdown menu is open, showing options like "Set status", "Your profile", "Your repositories", "Your codespaces", "Your projects", "Your stars", "Your gists", "Upgrade", "Feature preview", "Help", "Settings", and "Sign out".

https://github.com/EL-EL-BCN/pull-requests/pull/4

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public Watch 1 Fork

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights

[feature3] some new changes from MS-JM-ACC #4

Open MS-JM-ACC wants to merge 1 commit into `main` from `feature3`

Conversation 0 Commits 1 Checks 0 Files changed 1

MS-JM-ACC commented 1 minute ago Collaborator

This feature was added by another collaborator

This file was added by MS-JM-ACC Verified 4dd7df0

Add more commits by pushing to the `feature3` branch on EL-EL-BCN/pull-requests.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Sign out

Activate Windows
Go to Settings to activate Windows.

The pull request shows up in EL-EL-BCNs account and can be actioned

The screenshot displays the GitHub interface for the repository **EL-EL-BCN / pull-requests**. The page is filtered to show pull requests with the search criteria `is:pr is:open`. A single pull request is listed:

- ☐ **[feature3] some new changes from MS-JM-ACC**
#4 opened 5 minutes ago by MS-JM-ACC

A red arrow points to this pull request. Another red arrow points to the user profile dropdown menu in the top right corner, which is open and shows the user is signed in as **EL-EL-BCN**. The dropdown menu includes options like **Set status**, **Your profile**, **Your repositories**, **Your codespaces**, **Your projects**, **Your stars**, **Your gists**, **Upgrade**, **Feature preview**, **Help**, **Settings**, and **Sign out**.

Under the files changed tab -> review changes, User EL-EL-BCN can review the changes, approve them and submit a comment.

The screenshot shows a GitHub pull request page for user EL-EL-BCN. The title is "[feature3] some new changes from MS-JM-ACC #4". The interface includes tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). A red arrow points to the "Files changed" tab. Below the tabs, a file "feature3/new-collaborator.txt" is shown with a diff view. Another red arrow points to the "Review changes" button in the top right. A modal window titled "Finish your review" is open, showing a text area with the comment "Great Job, MS-JM-ACC!". Below the text area are three radio button options: "Comment", "Approve" (which is selected), and "Request changes". A red arrow points to the "Approve" option. At the bottom of the modal is a "Submit review" button. The footer of the page shows the GitHub logo, copyright information, and a Windows watermark.

github.com/EL-EL-BCN/pull-requests/pull/4/files

[feature3] some new changes from MS-JM-ACC #4

Open MS-JM-ACC wants to merge 1 commit into main from feature3

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to

1 feature3/new-collaborator.txt

... @@ -0,0 +1 @@

0 / 1 files viewed Review changes

Finish your review

Write Preview H B I

Great Job, MS-JM-ACC!

Attach files by dragging & dropping, selecting or pasting them.

☐ Comment
Submit general feedback without explicit approval.

☒ Approve
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

Submit review

Activate Windows
Go to Settings to activate Windows.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs

Reset 100% 23:29

User EL-EL-BCN can merge pull request.

The screenshot shows a GitHub pull request interface. At the top, the URL is `github.com/EL-EL-BCN/pull-requests/pull/4`. The pull request title is "[feature3] some new changes from MS-JM-ACC #4" and the description is "MS-JM-ACC wants to merge 1 commit into `main` from `feature3`".

A comment from "EL-EL-BCN" is visible, stating "Great Job, MS-JM-ACC!".

Below the comment, a message says: "Add more commits by pushing to the `feature3` branch on `EL-EL-BCN/pull-requests`."

The pull request status is "Changes approved" with "1 approving review" and "1 approval". A green checkmark indicates "This branch has no conflicts with the base branch".

A red arrow points to the "Merge pull request" button, which is highlighted in green. Below the button, it says: "You can also [open this in GitHub Desktop](#) or view [command line instructions](#)."

On the right side, there are sections for "Labels", "Projects", "Milestone", "Development", and "Notifications". The "Development" section notes: "Successfully merging this pull request may close these issues." and "None yet".

At the bottom, there are 2 participants and a watermark for "Activate Windows".

User EL-EL-BCN can view all pull requests by clicking on the pull request tab then deleting the code in this box.

The screenshot shows the GitHub interface for the repository `EL-EL-BCN`. The `Pull requests` tab is selected in the top navigation bar. A red arrow points to this tab. Below the navigation bar, there is a search bar and a `New pull request` button. The main content area displays a list of pull requests. The first pull request, `[Feature1] first feature was implemented`, is highlighted. It was created by `EL-EL-BCN` and was merged yesterday. The other three pull requests are also listed, showing their titles, authors, and merge status.

Author	Label	Projects	Milestones	Reviews	Assignee	Sort
MS-JM-ACC						
MS-JM-ACC						
EL-EL-BCN						
EL-EL-BCN						

User EL-EL-BCN can click on feature3 pull request and delete branch because it has been successfully merged.

The screenshot shows a GitHub pull request page for the repository 'EL-EL-BCN/pull-requests/pull/4'. The pull request, titled '[feature3] some new changes from MS-JM-ACC #4', has been merged. A red arrow points to the 'Delete branch' button in the 'Pull request successfully merged and closed' notification box. The notification text states: 'You're all set—the feature3 branch can be safely deleted.' The page also shows a comment from MS-JM-ACC: 'This feature was added by another collaborator' and an approval from EL-EL-BCN: 'Great Job, MS-JM-ACC!'. The right sidebar contains settings for Reviewers, Assignees, Labels, Projects, Milestone, and Development. The bottom of the page shows a Windows taskbar with the system clock at 23:39.

github.com/EL-EL-BCN/pull-requests/pull/4

Merged [feature3] some new changes from MS-JM-ACC #4
EL-EL-BCN merged 1 commit into main from feature3 3 minutes ago

MS-JM-ACC commented 18 minutes ago
This feature was added by another collaborator

This file was added by MS-JM-ACC ... Verified 4dd7df0

EL-EL-BCN approved these changes 6 minutes ago
View changes

EL-EL-BCN left a comment
Great Job, MS-JM-ACC!

EL-EL-BCN merged commit 4f34567 into main 3 minutes ago
Revert

Pull request successfully merged and closed
You're all set—the feature3 branch can be safely deleted.

Delete branch

Reviewers
EL-EL-BCN ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

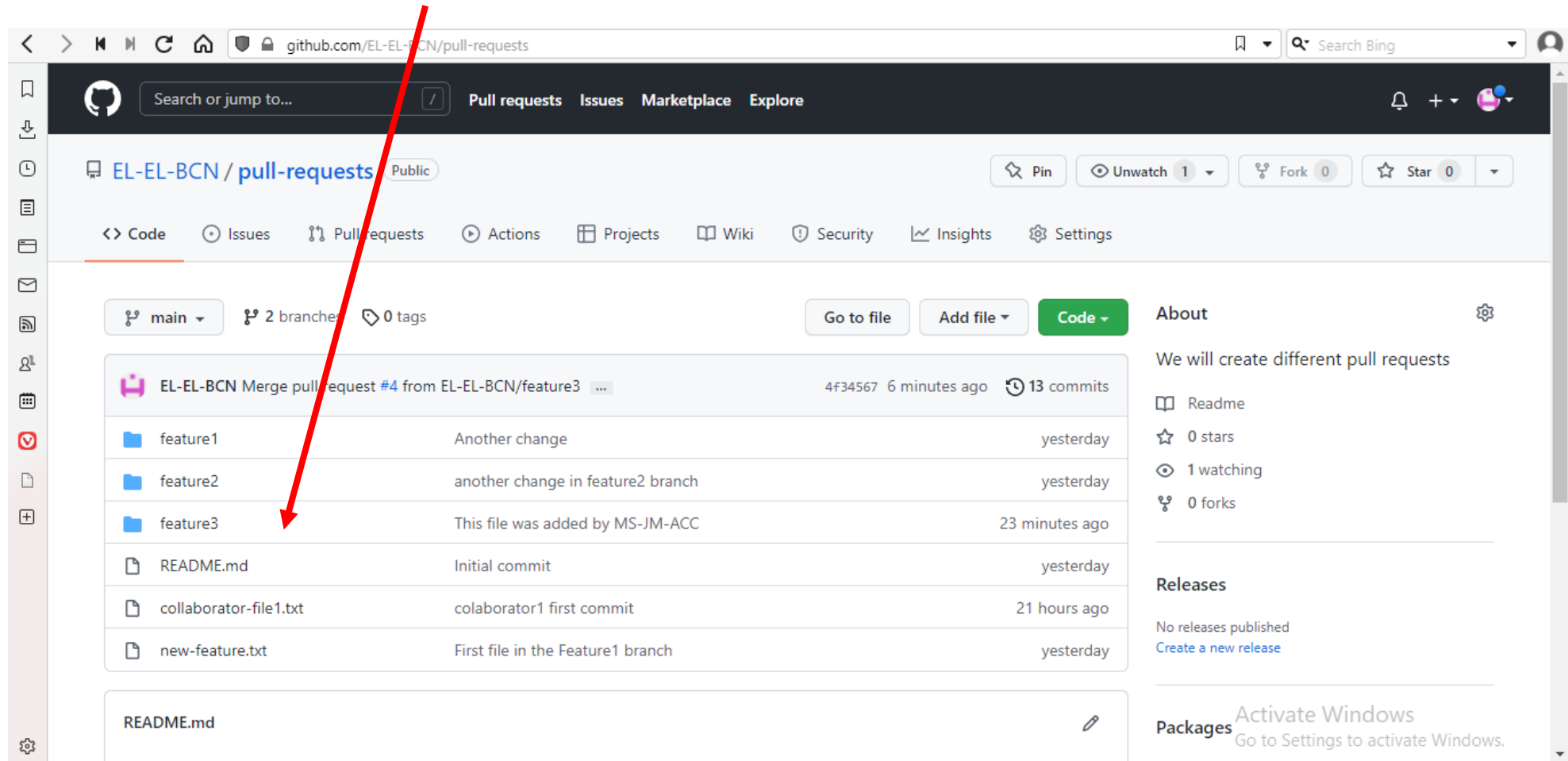
Development
Successfully merging this pull request may close these issues.
None yet

Activate Windows
Go to Settings to activate Windows.
Customize

Notifications

Reset 100 % 23:39

User EL-EL-BCN: feature3 folder created by MS-JM-ACC is now merged into main branch. The same changes are reflected in MS-LM-ACC's GitHub account.



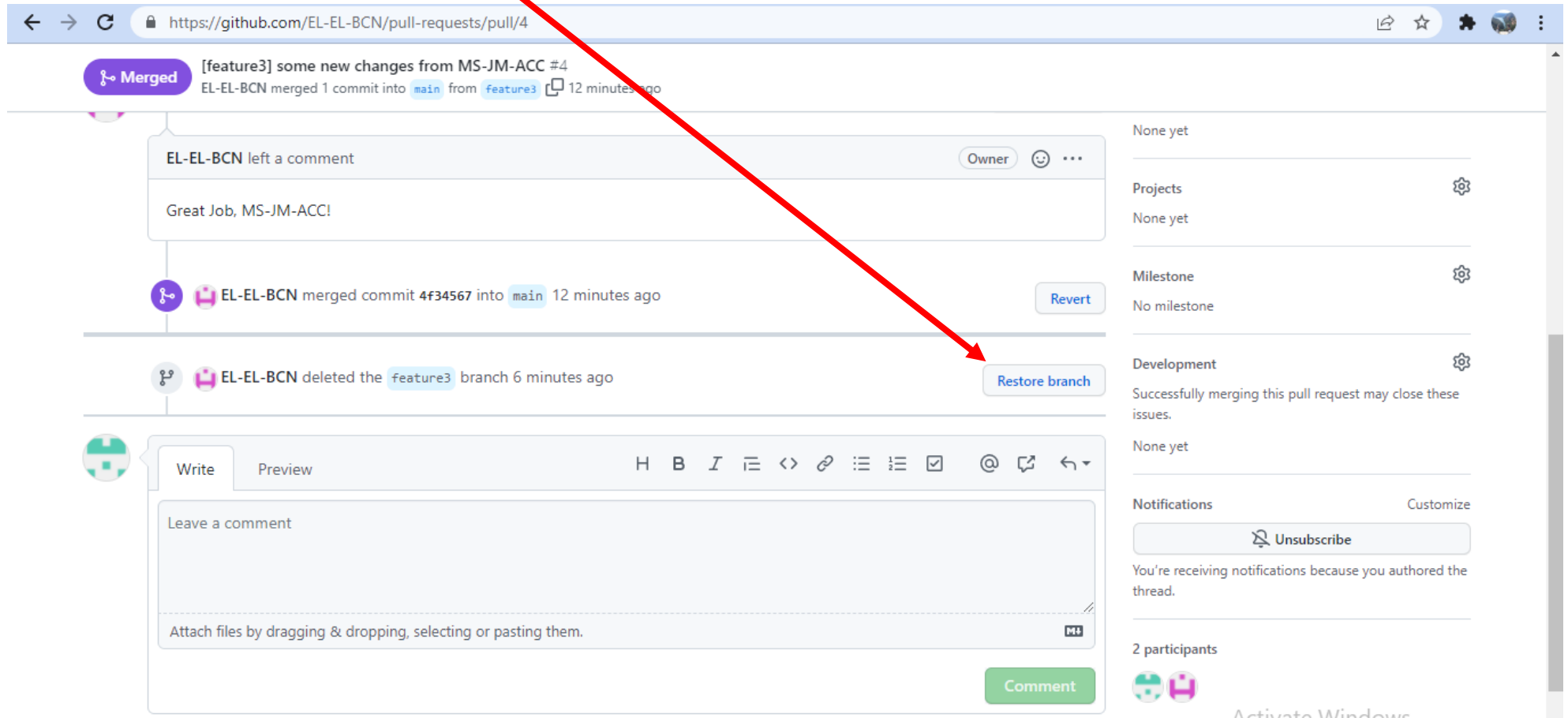
The screenshot shows a GitHub pull request page for the repository 'EL-EL-BCN'. The page title is 'pull-requests' and it is marked as 'Public'. The navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Pull requests' tab is active. Below the navigation bar, there are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files and folders that are part of the pull request. A red arrow points to the 'feature3' folder. The right sidebar contains information about the repository, including 'About', 'Releases', and 'Packages'.

File/Folder	Description	Commit Time
feature1	Another change	yesterday
feature2	another change in feature2 branch	yesterday
feature3	This file was added by MS-JM-ACC	23 minutes ago
README.md	Initial commit	yesterday
collaborator-file1.txt	colaborator1 first commit	21 hours ago
new-feature.txt	First file in the Feature1 branch	yesterday

Right sidebar content:

- About**
 - We will create different pull requests
 - Readme
 - 0 stars
 - 1 watching
 - 0 forks
- Releases**
 - No releases published
 - [Create a new release](#)
- Packages**
 - Activate Windows
 - Go to Settings to activate Windows.

In user MS-JM-ACC : feature3 branch can be restored by going to the pull requests and selecting all then choosing feature3.



The screenshot shows a GitHub pull request page for the repository EL-EL-BCN. The pull request, titled "[feature3] some new changes from MS-JM-ACC #4", has been merged. The merge history shows that EL-EL-BCN merged 1 commit into the main branch from the feature3 branch 12 minutes ago. Below the merge history, a message indicates that EL-EL-BCN deleted the feature3 branch 6 minutes ago. A red arrow points from the text above to the "Restore branch" button located next to the branch deletion message. The right sidebar contains settings for the pull request, including Projects, Milestone, Development, and Notifications. The bottom section shows a comment input area with a "Comment" button.

https://github.com/EL-EL-BCN/pull-requests/pull/4

Merged [feature3] some new changes from MS-JM-ACC #4
EL-EL-BCN merged 1 commit into main from feature3 12 minutes ago

EL-EL-BCN left a comment
Great Job, MS-JM-ACC!






EL-EL-BCN merged commit 4f34567 into main 12 minutes ago





EL-EL-BCN deleted the feature3 branch 6 minutes ago






Restore branch

Write Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment

None yet
Projects
None yet
Milestone
No milestone
Development
Successfully merging this pull request may close these issues.
None yet
Notifications
Unsubscribe
You're receiving notifications because you authored the thread.
2 participants




← → ↻ <https://github.com/EL-EL-BCN/pull-requests/edit/main/feature3/new-collaborator.txt>     

 Search or jump to... [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)   


 [EL-EL-BCN](#) / [pull-requests](#) Public  Watch **1**  Fork **0**  Star **0** 


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)


[pull-requests](#) / [feature3](#) / in [main](#) Cancel changes

[Edit file](#) [Preview changes](#) Spaces  2  Soft wrap 


```
1 This file was added by collaborator MS-JM-ACC
2
3 one more line added by MS-JM-ACC
```

 **Commit changes**




















☒  Commit directly to the [main](#) branch.

☐  Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

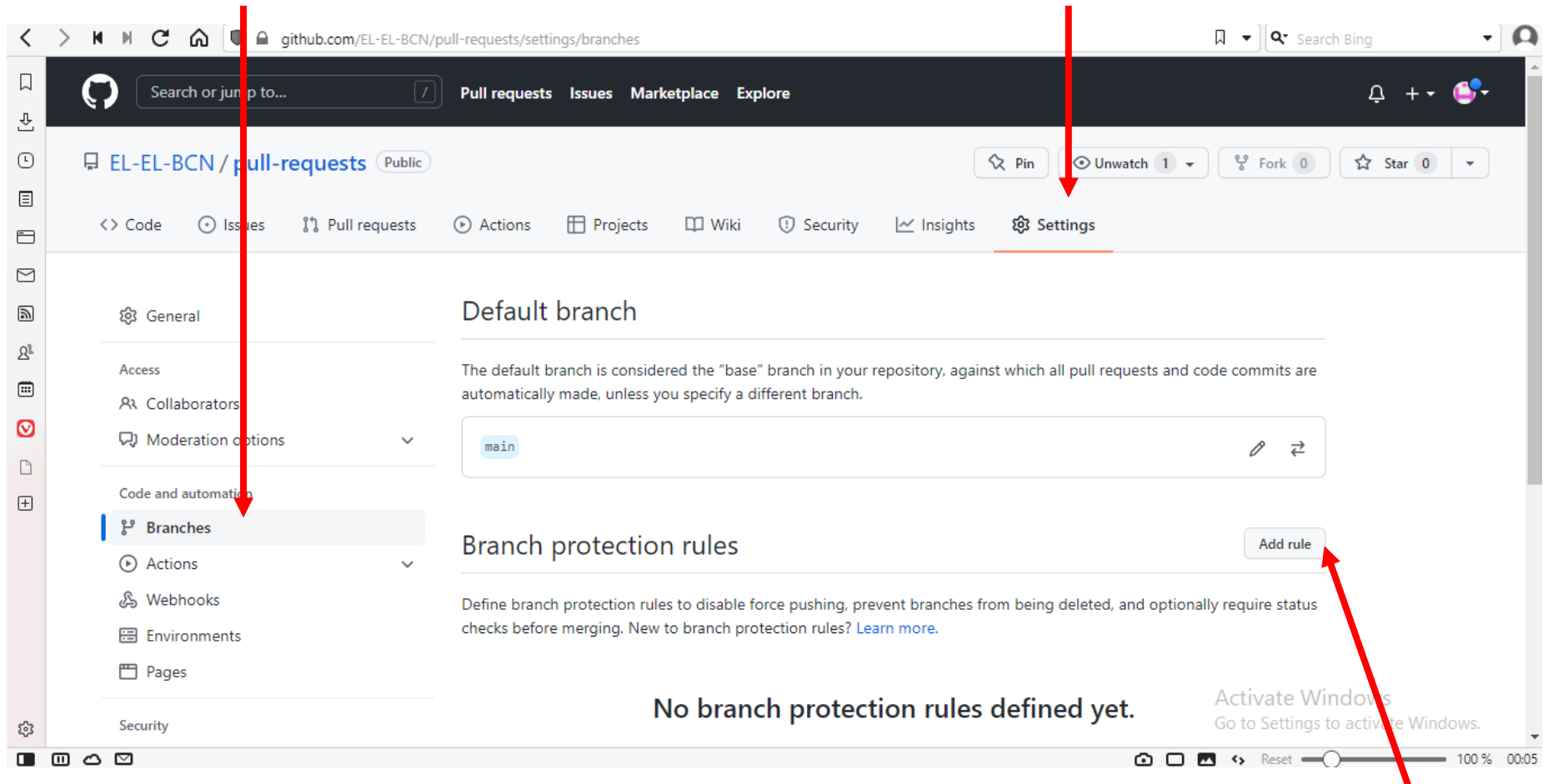
 © 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Activate Windows
Go to Settings to activate Windows.

               9°C    ENG
ES 23:54
06/03/2022 

User MS-JM-ACC: can now make changes to the files in feature3 and commit them. MS-JM-ACC can then create a new pull request from feature3 branch.

By default git allows merging of any new branches from a pull request into the main branch. We can add branch protection rules to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging.



Click on add rule

We need to name match the branch that we want to apply a protection rule to.

The screenshot shows the GitHub interface for creating a new branch protection rule. The left sidebar contains navigation links for General, Access, Collaborators, Moderation options, Code and automation, Branches, Actions, Webhooks, Environment, Pages, Security, Code security and analysis, Deploy keys, Secrets, and Integrations. The main content area is titled 'Branch protection rule' and includes a 'Branch name pattern' field with 'main' entered, a 'Protect matching branches' section with several checkboxes, and a 'Create' button at the bottom.

We can set requirement of a pull request before merging. This will require that all commits are made to a non-protected branch before merging.

We can also set require approvals. This means that x number of collaborators must approve the changes before merge is permitted. In this example we have only one collaborator so it is set to one.

Create rule

It will ask for password and to save changes.






If someone approves pull request but then a user tries to make changes to that pull request then these changes will be dismissed. This stops merging of changes into protected branch.

Create rule



It will ask for password and to save changes.






If someone approves pull request but then a user tries to make changes to that pull request then these changes will be dismissed. This stops merging of changes into protected branch.




Feature5 branch was created by MS-JM-ACC and a pull request sent. Note how it now will not allow merging because 1 review is required.

← → ↻ <https://github.com/EL-EL-BCN/pull-requests/pull/6>     




[feature5] protected-branch.txt #6

 Open MS-JM-ACC wants to merge 1 commit into `main` from `feature5` 


 Conversation 0  Commits 1  Checks 0  Files changed 1 +1 -0 



 MS-JM-ACC commented now Collaborator  

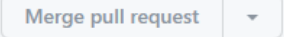

testing of protected branch feature5


  Create protected-branch.txt Verified  d172f62


Add more commits by pushing to the `feature5` branch on [EL-EL-BCN/pull-requests](#).





-  **Review required**
At least 1 approving review is required by reviewers with write access. [Learn more.](#)
-  **Merging is blocked**
Merging can be performed automatically with 1 approving review.


 Merge pull request  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers 
No reviews—at least 1 approving review is required.
Still in progress? Convert to draft

Assignees 
No one—assign yourself

Labels 
None yet

Projects 
None yet

Milestone 
No milestone

Activate Windows

Feature5 branch was created by MS-JM-ACC and a pull request sent. Note how also in EL-EL-BCNs account it will not allow merging because 1 approving review is required.

github.com/EL-EL-BCN/pull-requests/pull/6

[feature5] protected-branch.txt #6
MS-JM-ACC wants to merge 1 commit into main from feature5

Create protected-branch.txt

Verified d172f62

Add more commits by pushing to the feature5 branch on EL-EL-BCN/pull-requests.

Review required
At least 1 approving review is required by reviewers with write access. [Learn more.](#) [Add your review](#)

Merging is blocked
Merging can be performed automatically with 1 approving review.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications [Customize](#)
[Activate Windows](#) [Unsubscribe](#)
You're receiving notifications because you're watching this repository.

EL-EL-BCN can add a review

EL-EL-BCN can add a review

The screenshot shows a GitHub pull request for the file `[feature5] protected-branch.txt #6`. The pull request is from `feature5` to `main`, with 1 commit and 1 file changed. The interface includes a sidebar with navigation icons, a top bar with the repository name and search, and a main content area with a file diff view. A modal window titled "Finish your review" is open, showing a text area for a review comment, a list of review options (Comment, Approve, Request changes), and a "Submit review" button. Red arrows point from the text "EL-EL-BCN can add a review" to the "Review changes" button in the top right and the "Submit review" button in the modal.

github.com/EL-EL-BCN/pull-requests/pull/6/files

[feature5] protected-branch.txt #6

Open MS-JM-ACC wants to merge 1 commit into main from feature5

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to

1 feature5/protected-branch.txt

... @@ -0,0 +1 @@

Finish your review

Write Preview H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↩

I have approved the changes and feature5 can be merged into protected branch main

Attach files by dragging & dropping, selecting or pasting them.

☐ Comment
Submit general feedback without explicit approval.

☒ Approve
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

Submit review

Activate Windows
Go to Settings to activate Windows.

100% 00:37

With one review the feature5 branch can be merged into main

The screenshot shows a GitHub pull request interface. At the top, a green 'Open' button is next to the title '[feature5] protected-branch.txt #6'. Below the title, it says 'MS-JM-ACC wants to merge 1 commit into main from feature5'. A comment from 'EL-EL-BCN' is visible, stating 'I have approved the changes and feature5 can be merged into protected branch main'. Below the comment, a green box contains the status 'Changes approved' with '1 approving review by reviewers with write access'. A 'Merge pull request' button is at the bottom of this box. A red arrow points from the text 'With one review the feature5 branch can be merged into main' to this button. The right sidebar shows settings for labels, projects, milestones, and development. The bottom of the page shows a 'Write' and 'Preview' section for the pull request description.

github.com/EL-EL-BCN/pull-requests/pull/6

Open [feature5] protected-branch.txt #6
MS-JM-ACC wants to merge 1 commit into main from feature5

EL-EL-BCN left a comment
Owner

I have approved the changes and feature5 can be merged into protected branch main

Add more commits by pushing to the feature5 branch on EL-EL-BCN/pull-requests.

Changes approved
1 approving review by reviewers with write access. [Show all reviewers](#)

1 approval

Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications [Customize](#)
[Unsubscribe](#)
You're receiving notifications because you're watching this repository.

2 participants
[Activate Windows](#)
Go to Settings to activate Windows.

Write Preview

H B I [≡](#) [<>](#) [🔗](#) [⋮](#) [⋮](#) [☑](#) [@](#) [🗨](#) [↩](#)

Reset 100 % 00:38

Issues are like requests for bug fixes or small changes created in a particular repository. They are like comments from collaborators or if a public library anyone really.

The screenshot displays the GitHub interface for the repository `EL-EL-BCN / pull-requests`. The `Issues` tab is selected in the navigation bar. The search bar contains the query `is:issue is:open`. Below the search bar, it indicates `0 Open` and `0 Closed` issues. The main content area shows a message: `There aren't any open issues.` with a suggestion to search all of GitHub or try an advanced search. A green `New issue` button is visible on the right. Two red arrows originate from the text above: one points to the `Issues` tab, and the other points to the `New issue` button.

Issue title

Issue description

The screenshot shows the GitHub interface for creating a new issue. The browser address bar displays 'github.com/EL-EL-BCN/pull-requests/issues/new'. The repository name 'EL-EL-BCN / pull-requests' is shown as 'Public'. The 'Issues' tab is selected in the navigation bar. The issue title field contains the text 'Lets improve feature2 and add some additional files'. The issue description field contains the text 'Lets add one more file in the feature2 folder' and a code block with the text 'additional-file.txt'. The 'Submit new issue' button is located at the bottom right of the form. The right sidebar shows settings for Assignees, Labels, Projects, and Milestones, all currently set to 'None yet'. The bottom of the page shows a status bar with '100 %' and '00:56'.

github.com/EL-EL-BCN/pull-requests/issues/new

Search or jump to...

Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Lets improve feature2 and add some additional files

Write Preview

Lets add one more file in the feature2 folder
'additional-file.txt'

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Submit new issue

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Shows branches and pull requests linked to this issue.
Go to Settings to activate Windows.

100 % 00:56

Submit new issue

github.com/EL-EL-BCN/pull-requests/issues/7

EL-EL-BCN commented 2 minutes ago

Lets add one more file in the feature2 folder
'additional-file.txt'

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees

No one—assign yourself

Labels

Apply labels to this issue

Filter labels

- bug
Something isn't working
- documentation
Improvements or additions to documentation
- duplicate
This issue or pull request already exists
- enhancement
New feature or request
- good first issue
Good for newcomers
- help wanted
Extra attention is needed
- invalid
This doesn't seem right

The new issue can have a label
Such as Enhancement

Activate Windows
Go to Settings to activate Windows.

Reset 100 % 01:00

github.com/EL-EL-BCN/pull-requests/issues/7

Lets improve feature2 and add some additional files #7

Open EL-EL-BCN opened this issue 5 minutes ago · 0 comments

EL-EL-BCN commented 5 minutes ago

Lets add one more file in the feature2 folder
'additional-file.txt'

EL-EL-BCN added the **enhancement** label now

EL-EL-BCN assigned MS-JM-ACC now

EL-EL-BCN unassigned MS-JM-ACC now

The new issue can be assigned to a specific collaborator or collaborators

Assignees

Assign up to 10 people to this issue

Type or choose a user

Suggestions

EL-EL-BCN EL-EL-BCN

MS-JM-ACC MS-JM-ACC

Milestone

No milestone

Development

Create a branch for this issue or link a pull request.

Notifications

Customize

Unsubscribe

Go to Settings to activate Windows.

You're receiving notifications because you're watching

The new issue now shows up in the account of MS-JM-ACC in the repository.

EL-EL-BCN / pull-requests

Public

Watch

1

Fork

0

Star

0

<> Code

Issues

1

Pull requests

Actions

Projects

Wiki

Security

Insights

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters

is:issue is:open

Labels

9

Milestones

0

New issue

1 Open

0 Closed

Author

Label

Projects

Milestones

Assignee

Sort

Lets improve feature2 and add some additional files

enhancement

#7 opened 7 minutes ago by EL-EL-BCN

ProTip!

Notify someone on an issue with a mention, like: @MS-JM-ACC.

We can add new labels to the issue, i.e. work in progress.

github.com/EL-EL-BCN/pull-requests/issues/labels

Search Bing

Search or jump to...

Pull requestsIssuesMarketplaceExplore

EL-EL-BCN / pull-requestsPublic

PinUnwatch1Fork0Star0

<> CodeIssues1Pull requestsActionsProjectsWikiSecurityInsightsSettings

LabelsMilestones

Search all labels

New label

work in progress

Label name

work in progress

Description

Description (optional)

Color

#E8D52A

Cancel

Create label

9 labels

Sort

bug

Something isn't working

EditDelete

documentation

Improvements or additions to documentation

EditDelete

Activate Windows

Go to Settings to activate Windows.

Reset

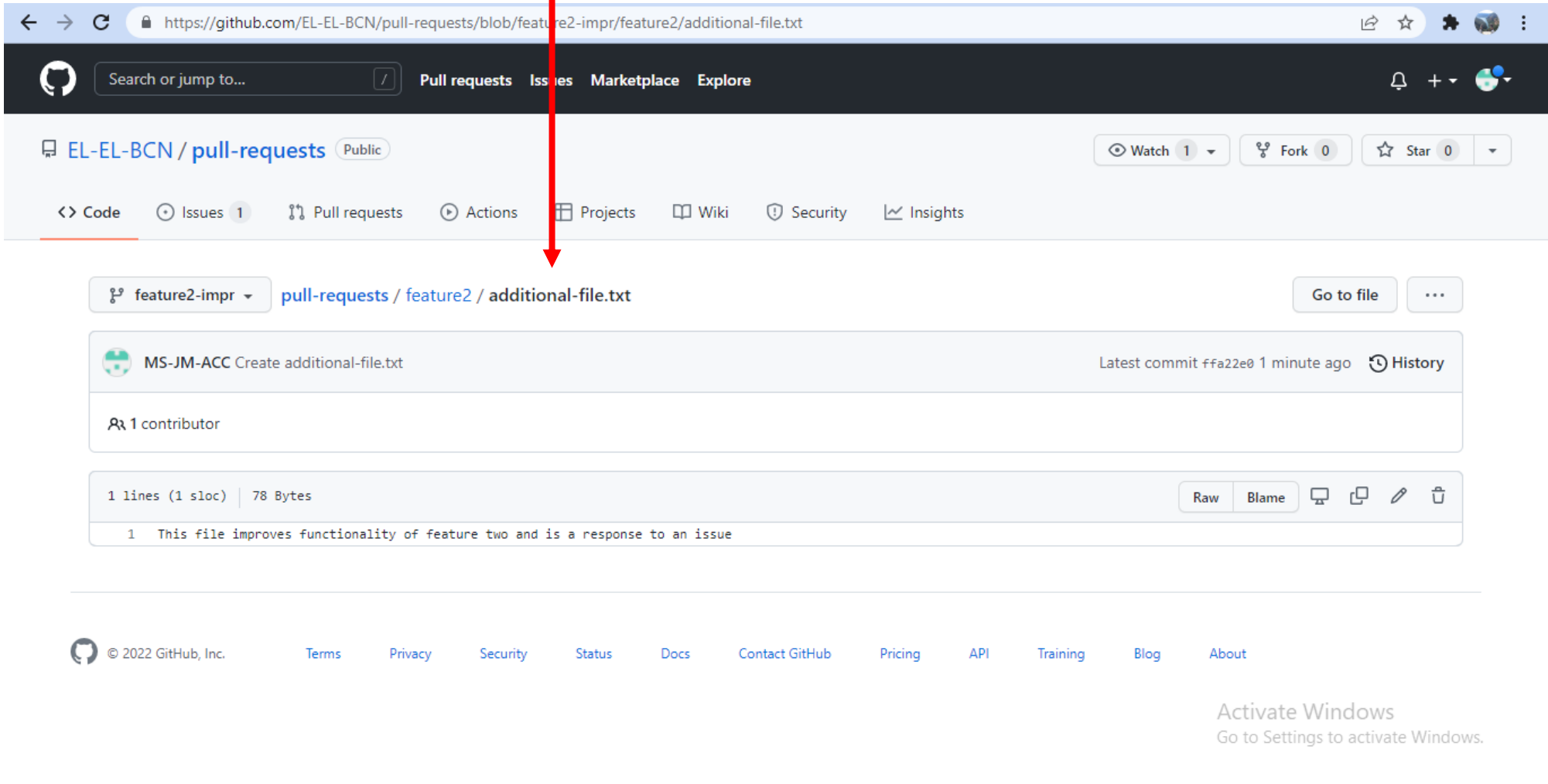
100 %

01:11

At the bottom we can create a new branch for this issue.



A new branch named feature2-impr is created and an additional-file.txt was added and committed.




A pull request was added by MS-JM-ACC which requires approval to merge. Note that a pull request can be assigned to a particular person.

← → ↺ 🔒 https://github.com/EL-EL-BCN/pull-requests/pull/8 ⌵ ☆ ⚙️ 👤 ⋮

Create additional-file.txt #8 Edit <> Code ▾

🔗 Open MS-JM-ACC wants to merge 1 commit into main from feature2-impr 📄

💬 Conversation 0 📄 Commits 1 🔍 Checks 0 📄 Files changed 1 +1 -0



MS-JM-ACC commented 2 minutes ago • edited ▾

Collaborator 😊 ⋮

Additional file for feature2 added by MS-JM-ACC

📄 Create additional-file.txt

Verified ffa22e0

🔖 MS-JM-ACC linked an issue that may be closed by this pull request 2 minutes ago

Lets improve feature2 and add some additional files #7 🔗 Open

🔗 Add more commits by pushing to the `feature2-impr` branch on `EL-EL-BCN/pull-requests`.

🔗

✖ Review required

At least 1 approving review is required by reviewers with write access. [Learn more.](#)

✖ Merging is blocked

Merging can be performed automatically with 1 approving review.

Merge pull request ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers


No reviews—at least 1 approving review is required.


Still in progress? [Convert to draft](#)

Assignees

Assign up to 10 people to this pull request

Suggestions

 MS-JM-ACC MS-JM-ACC

 EL-EL-BCN EL-EL-BCN

Milestone

No milestone

Development

Go to Settings to activate Windows.

Successfully merging this pull request may close these issues

github.com/EL-EL-BCN/pull-requests/pulls

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Filters is:pr is:closed Labels 10 Milestones 0 New pull request

Clear current search query, filters, and sorts

0 Open 7 Closed

Create additional-file.txt
#8 by MS-JM-ACC was merged 19 seconds ago • Approved

The pull request from MS-JM-ACC was approved and merged with the branch being deleted.

github.com/EL-EL-BCN/pull-requests/issues

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / pull-requests Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Filters is:issue is:closed Labels 10 Milestones 0 New issue

Clear current search query, filters, and sorts

0 Open 1 Closed

Lets improve feature2 and add some additional files enhancement work in progress
#7 by EL-EL-BCN was closed 2 minutes ago

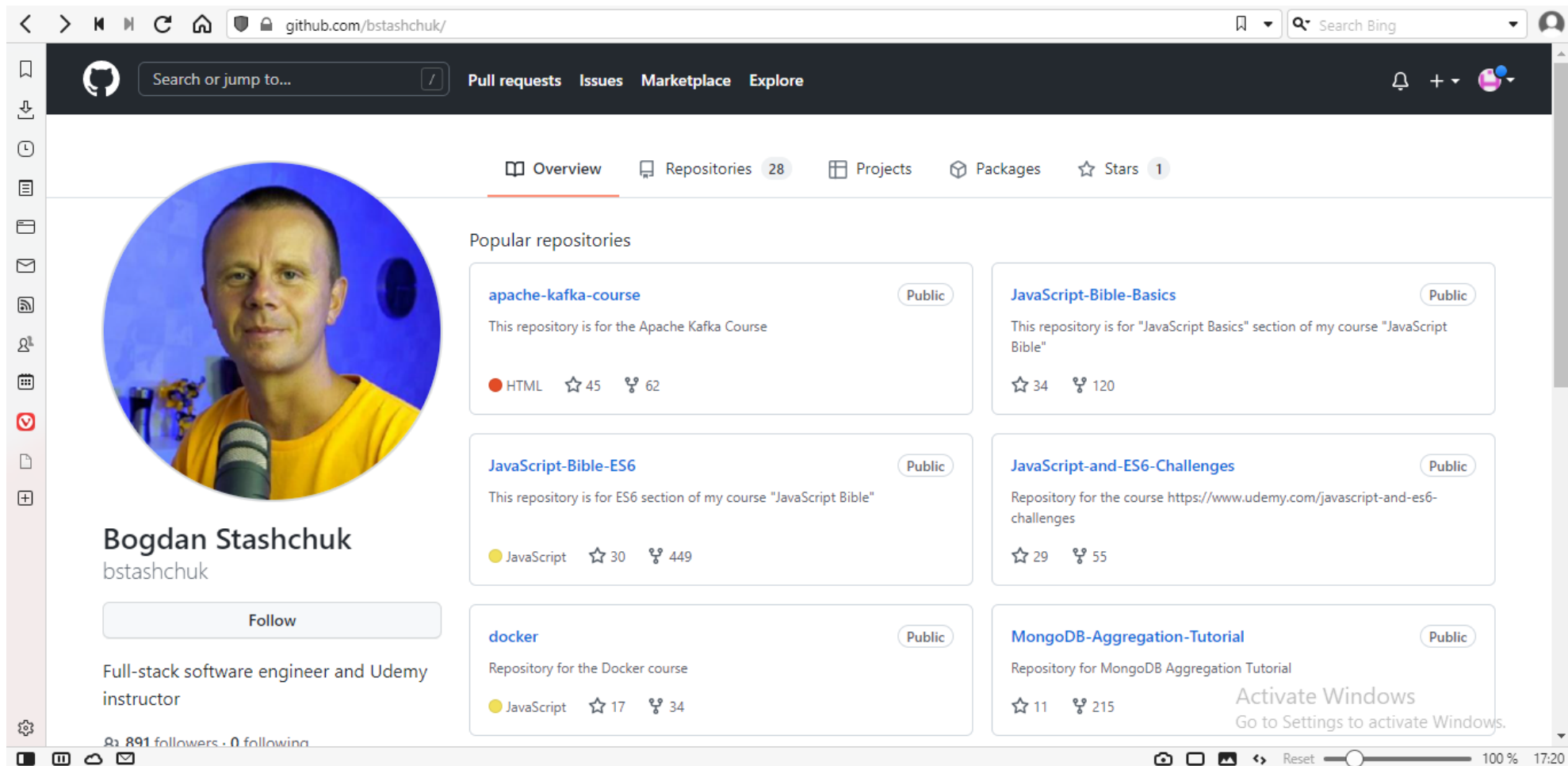
Because the branch was created from an issue when the branch is merged and deleted the issue gets automatically closed.

The screenshot shows a GitHub issue page for the repository `github.com/EL-EL-BCN/pull-requests/issues/7`. The issue is titled "Lets improve feature2 and add some additional files #7" and is marked as "Closed". It was opened 34 minutes ago and fixed by user #8. The issue history shows that user MS-JM-ACC linked a pull request to close the issue 12 minutes ago, and user EL-EL-BCN closed it in pull request #8 6 minutes ago. A comment box is visible with the text "Leave a comment" and a "Comment" button. A red arrow points from the bottom of the comment box to the "Lock conversation" button in the right sidebar. The sidebar also includes options like "Unsubscribe", "2 participants", "Pin issue", "Transfer issue", and "Delete issue".

Once issue is closed anyone can add a comment. By selecting lock conversation we prevent all but approved contributors from adding to the comments.

FORKS AND CONTRIBUTION TO THE PUBLIC REPOSITORIES

Go to <https://github.com/bstashchuk/> from my GitHub account and select any one of the repositories.



The screenshot displays the GitHub profile of Bogdan Stashchuk (bstashchuk). The profile includes a circular profile picture of a man in a yellow shirt, his name, and a bio identifying him as a full-stack software engineer and Udemmy instructor. A 'Follow' button is visible. Below the bio, it shows 891 followers and 0 following. The 'Repositories' tab is selected, showing 28 repositories. The 'Popular repositories' section lists six repositories, each with a description, language, star count, and fork count.

Repository Name	Description	Language	Stars	Forks	Visibility
apache-kafka-course	This repository is for the Apache Kafka Course	HTML	45	62	Public
JavaScript-Bible-Basics	This repository is for "JavaScript Basics" section of my course "JavaScript Bible"	JavaScript	34	120	Public
JavaScript-Bible-ES6	This repository is for ES6 section of my course "JavaScript Bible"	JavaScript	30	449	Public
JavaScript-and-ES6-Challenges	Repository for the course https://www.udemy.com/javascript-and-es6-challenges	JavaScript	29	55	Public
docker	Repository for the Docker course	JavaScript	17	34	Public
MongoDB-Aggregation-Tutorial	Repository for MongoDB Aggregation Tutorial	JavaScript	11	215	Public

Windows watermark: Activate Windows. Go to Settings to activate Windows.

We can click on fork button

The screenshot shows the GitHub interface for the repository 'bstashchuk / JavaScript-Bible-Basics'. The repository is public and has 8 watchers, 120 forks, and 34 stars. The 'Fork' button is highlighted with a red arrow. The repository description states: 'This repository is for "JavaScript Basics" section of my course "JavaScript Bible"'. The 'About' section also contains this text. The 'Releases' section shows 'No releases published'. The 'Packages' section has a message to 'Activate Windows' and 'Go to Settings to activate Windows.'.

github.com/bstashchuk/JavaScript-Bible-Basics

Search or jump to... Pull requests Issues Marketplace Explore

bstashchuk / JavaScript-Bible-Basics Public

Watch 8 Fork 120 Star 34

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights

master 32 branches 0 tags

Go to file Add file Code

About

This repository is for "JavaScript Basics" section of my course "JavaScript Bible"

Readme 34 stars 8 watching 120 forks

Releases

No releases published

Activate Windows
Packages Go to Settings to activate Windows.

Reset 100 % 17:21

Now user EL-EL-BCN has forked this repository from [bstashchuk](#)

The screenshot displays the GitHub web interface in a browser. The address bar shows 'github.com'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the left sidebar, under 'Recent Repositories', a list of repositories is shown, with a red arrow pointing to 'EL-EL-BCN/pull-requests'. The main content area features a green banner for 'Learn Git and GitHub without any code!' and a section titled 'All activity' showing recent pushes to the 'EL-EL-BCN/pull-requests' repository. On the right, a user profile dropdown menu is open, showing the user is signed in as 'EL-EL-BCN' and listing various account options. The Windows taskbar is visible at the bottom.

github.com

Search or jump to...

Pull requests Issues Marketplace Explore

Recent Repositories

New

Find a repository...

EL-EL-BCN/pull-requests

mikegithuber/pull-requests

EL-EL-BCN/my-first-git-hub-repository

EL-EL-BCN/JavaScript-Bible-Basics

Recent activity

Great course so far

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide Start a project

All activity

MS-JM-ACC pushed to EL-EL-BCN/pull-requests 16 hours ago

1 commit to feature2-impr

ffa22e0 Create additional-file.txt

MS-JM-ACC pushed to EL-EL-BCN/pull-requests 2 days ago

1 commit to collaborator1-branch

cb8ac3f colaborator1 first commit

Signed in as EL-EL-BCN

Set status

Your profile

Your repositories

Your codespaces

Your projects

Your stars

Your gists

Upgrade

Feature preview

Help

Settings

Sign out

Activate Windows

Go to Settings to activate Windows.

Reset 100 % 17:24

From the bash terminal in the habitual folder or git examples.

Git clone command can be used to clone the remote repository to the local computer. Note how the user is EL-EL-BCN in the clone URL because it has been forked and not directly cloned from bstashchuk.

Now we see that repository folder

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT

e1lio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  hello-world/  pull-requests/
first-project/  my-first-git-hub-repository/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git
Cloning into 'JavaScript-Bible-Basics'...
remote: Enumerating objects: 204, done.
remote: Total 204 (delta 0), reused 0 (delta 0), pack-reused 204
Receiving objects: 100% (204/204), 45.08 KiB | 485.00 KiB/s, done.
Resolving deltas: 100% (89/89), done.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls -la
total 40
drwxr-xr-x 1 e1lio 197609 0 Mar  6 03:24 -my-first-git-hub-repository/
drwxr-xr-x 1 e1lio 197609 0 Mar  7 17:30 ./
drwxr-xr-x 1 e1lio 197609 0 Feb 24 16:03 ../
drwxr-xr-x 1 e1lio 197609 0 Mar  7 17:30 JavaScript-Bible-Basics/
drwxr-xr-x 1 e1lio 197609 0 Mar  1 21:14 first-project/
drwxr-xr-x 1 e1lio 197609 0 Feb 27 22:18 hello-world/
drwxr-xr-x 1 e1lio 197609 0 Mar  6 01:07 my-first-git-hub-repository/
drwxr-xr-x 1 e1lio 197609 0 Mar  6 21:47 pull-requests/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$
```

We can cd into the directory.

And see a .git directory.

Push and pull remote server URLs are under username EL-EL-BCN

This means that this repository does not have any connection with parent directory from which it was forked.

And in the .git directory

We see only one remote origin server which is under EL-EL-BCNs account.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics/.git
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd JavaScript-Bible-Basics/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la
total 9
drwxr-xr-x 1 ellio 197609  0 Mar  7 17:30 ./
drwxr-xr-x 1 ellio 197609  0 Mar  7 17:30 ../
drwxr-xr-x 1 ellio 197609  0 Mar  7 17:30 .git/
-rw-r--r-- 1 ellio 197609 111 Mar  7 17:30 README.md

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git remote -v
origin  https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git (fetch)
origin  https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git (push)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ cd .git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics/.git (GIT_DIRECTORY)
$ cat config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[remote "origin"]
    url = https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
    remote = origin
    merge = refs/heads/master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics/.git (GIT_DIRECTORY)
$
```

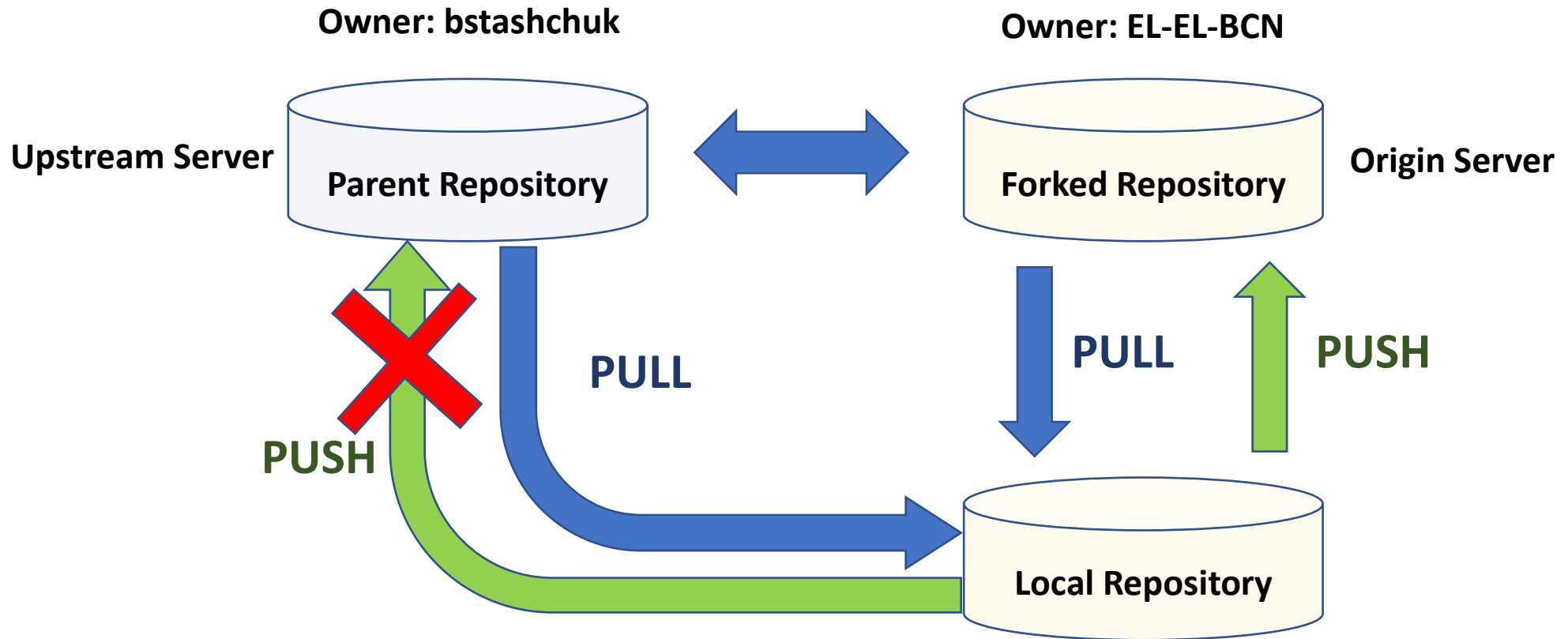
Note that this repository was forked from [bstashchuk/JavaScript-Bible-Basics](https://github.com/bstashchuk/JavaScript-Bible-Basics) And has 32 branches.

The screenshot shows the GitHub interface for the repository `EL-EL-BCN / JavaScript-Bible-Basics`. The repository is public and was forked from `bstashchuk/JavaScript-Bible-Basics`. The interface includes a top navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Pin, Watch (0), Fork (121), and Star (0). The main content area shows the `master` branch selected, with 32 branches and 0 tags. A message indicates that the branch is up to date with `bstashchuk:master`. Below this, there is a section for the `README.md` file, showing it was created by `bstashchuk` on 12 Nov 2018 with 4 commits. The README content is visible, starting with `JavaScript-Bible-Basics` and a description of the repository. On the right side, there is an 'About' section with details like 0 stars, 0 watching, and 121 forks. The bottom of the page shows a taskbar with various icons and a system clock.

Q1: How do we sync a forked repository with changes in the parent repository?

Q2: How to open a pull request in the parent repository from the forked repository?

In the local repository we can configure an upstream server that links back to the parent repository and we can then perform pulls but not pushes because EL-EL-BCN is not a collaborator with bstaschuk in this repository.



Conclusions: In order to keep your forked repository up to date then first it must be cloned to your computer then another remote server must be added and the convention is to name it upstream.

We can add a git remote repository for the upstream server using **git remote add upstream** command. Note how the URL is that of the parent repository.

Now we have two remote servers, Origin and upstream.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git remote add upstream https://github.com/bstashchuk/JavaScript-Bible-Basics.git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git remote
origin
upstream

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git remote -v
origin https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git (fetch)
origin https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git (push)
upstream https://github.com/bstashchuk/JavaScript-Bible-Basics.git (fetch)
upstream https://github.com/bstashchuk/JavaScript-Bible-Basics.git (push)

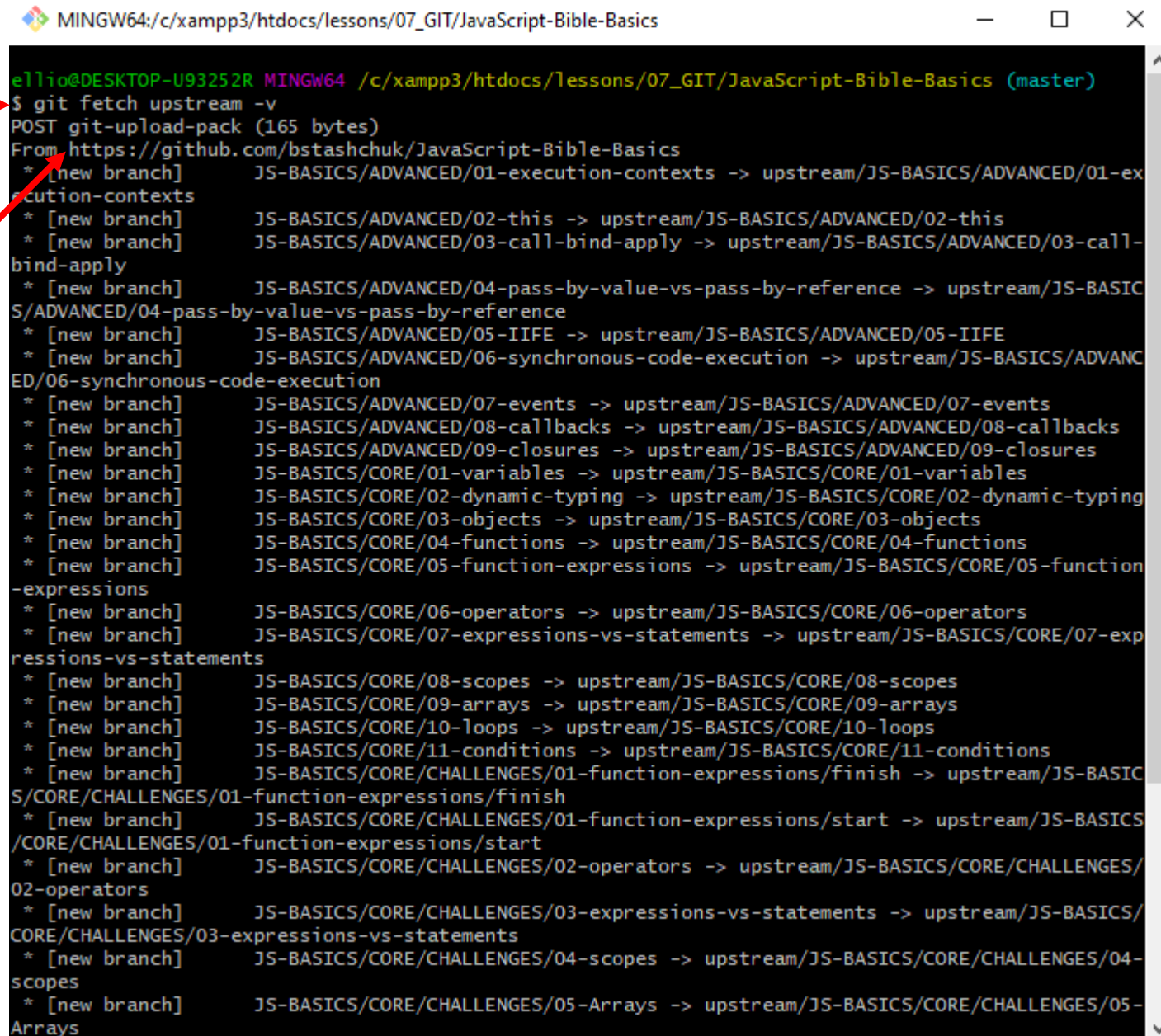
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ |
```

Note that a git branch -a command shows only branches from the origin server because we have not yet performed a fetch from the upstream server.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/JS-BASICS/ADVANCED/01-execution-contexts
remotes/origin/JS-BASICS/ADVANCED/02-this
remotes/origin/JS-BASICS/ADVANCED/03-call-bind-apply
remotes/origin/JS-BASICS/ADVANCED/04-pass-by-value-vs-pass-by-reference
remotes/origin/JS-BASICS/ADVANCED/05-IIFE
remotes/origin/JS-BASICS/ADVANCED/06-synchronous-code-execution
remotes/origin/JS-BASICS/ADVANCED/07-events
remotes/origin/JS-BASICS/ADVANCED/08-callbacks
remotes/origin/JS-BASICS/ADVANCED/09-closures
remotes/origin/JS-BASICS/CORE/01-variables
remotes/origin/JS-BASICS/CORE/02-dynamic-typing
remotes/origin/JS-BASICS/CORE/03-objects
remotes/origin/JS-BASICS/CORE/04-functions
remotes/origin/JS-BASICS/CORE/05-function-expressions
remotes/origin/JS-BASICS/CORE/06-operators
remotes/origin/JS-BASICS/CORE/07-expressions-vs-statements
remotes/origin/JS-BASICS/CORE/08-scopes
remotes/origin/JS-BASICS/CORE/09-arrays
remotes/origin/JS-BASICS/CORE/10-loops
remotes/origin/JS-BASICS/CORE/11-conditions
remotes/origin/JS-BASICS/CORE/CHALLENGES/01-function-expressions/finish
remotes/origin/JS-BASICS/CORE/CHALLENGES/01-function-expressions/start
remotes/origin/JS-BASICS/CORE/CHALLENGES/02-operators
remotes/origin/JS-BASICS/CORE/CHALLENGES/03-expressions-vs-statements
remotes/origin/JS-BASICS/CORE/CHALLENGES/04-scopes
remotes/origin/JS-BASICS/CORE/CHALLENGES/05-Arrays
remotes/origin/JS-BASICS/CORE/CHALLENGES/06-loops-and-conditions
remotes/origin/JS-BASICS/INTRO/01-embedded-script
remotes/origin/JS-BASICS/INTRO/02-external-script
remotes/origin/JS-BASICS/INTRO/CHALLENGE-1/external-scripts/finish
remotes/origin/JS-BASICS/INTRO/CHALLENGE-1/external-scripts/start
remotes/origin/master
(END)
```

Git fetch upstream -v
command will fetch the
branches from the
upstream or parent
server.

Note the from server.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git fetch upstream -v
POST git-upload-pack (165 bytes)
From https://github.com/bstashchuk/JavaScript-Bible-Basics
* [new branch]      JS-BASICS/ADVANCED/01-execution-contexts -> upstream/JS-BASICS/ADVANCED/01-ex
execution-contexts
* [new branch]      JS-BASICS/ADVANCED/02-this -> upstream/JS-BASICS/ADVANCED/02-this
* [new branch]      JS-BASICS/ADVANCED/03-call-bind-apply -> upstream/JS-BASICS/ADVANCED/03-call-
bind-apply
* [new branch]      JS-BASICS/ADVANCED/04-pass-by-value-vs-pass-by-reference -> upstream/JS-BASIC
S/ADVANCED/04-pass-by-value-vs-pass-by-reference
* [new branch]      JS-BASICS/ADVANCED/05-IIFE -> upstream/JS-BASICS/ADVANCED/05-IIFE
* [new branch]      JS-BASICS/ADVANCED/06-synchronous-code-execution -> upstream/JS-BASICS/ADVANC
ED/06-synchronous-code-execution
* [new branch]      JS-BASICS/ADVANCED/07-events -> upstream/JS-BASICS/ADVANCED/07-events
* [new branch]      JS-BASICS/ADVANCED/08-callbacks -> upstream/JS-BASICS/ADVANCED/08-callbacks
* [new branch]      JS-BASICS/ADVANCED/09-closures -> upstream/JS-BASICS/ADVANCED/09-closures
* [new branch]      JS-BASICS/CORE/01-variables -> upstream/JS-BASICS/CORE/01-variables
* [new branch]      JS-BASICS/CORE/02-dynamic-typing -> upstream/JS-BASICS/CORE/02-dynamic-typing
* [new branch]      JS-BASICS/CORE/03-objects -> upstream/JS-BASICS/CORE/03-objects
* [new branch]      JS-BASICS/CORE/04-functions -> upstream/JS-BASICS/CORE/04-functions
* [new branch]      JS-BASICS/CORE/05-function-expressions -> upstream/JS-BASICS/CORE/05-function
-expressions
* [new branch]      JS-BASICS/CORE/06-operators -> upstream/JS-BASICS/CORE/06-operators
* [new branch]      JS-BASICS/CORE/07-expressions-vs-statements -> upstream/JS-BASICS/CORE/07-exp
ressions-vs-statements
* [new branch]      JS-BASICS/CORE/08-scopes -> upstream/JS-BASICS/CORE/08-scopes
* [new branch]      JS-BASICS/CORE/09-arrays -> upstream/JS-BASICS/CORE/09-arrays
* [new branch]      JS-BASICS/CORE/10-loops -> upstream/JS-BASICS/CORE/10-loops
* [new branch]      JS-BASICS/CORE/11-conditions -> upstream/JS-BASICS/CORE/11-conditions
* [new branch]      JS-BASICS/CORE/CHALLENGES/01-function-expressions/finish -> upstream/JS-BASIC
S/CORE/CHALLENGES/01-function-expressions/finish
* [new branch]      JS-BASICS/CORE/CHALLENGES/01-function-expressions/start -> upstream/JS-BASICS
/CORE/CHALLENGES/01-function-expressions/start
* [new branch]      JS-BASICS/CORE/CHALLENGES/02-operators -> upstream/JS-BASICS/CORE/CHALLENGES/
02-operators
* [new branch]      JS-BASICS/CORE/CHALLENGES/03-expressions-vs-statements -> upstream/JS-BASICS/
CORE/CHALLENGES/03-expressions-vs-statements
* [new branch]      JS-BASICS/CORE/CHALLENGES/04-scopes -> upstream/JS-BASICS/CORE/CHALLENGES/04-
scopes
* [new branch]      JS-BASICS/CORE/CHALLENGES/05-Arrays -> upstream/JS-BASICS/CORE/CHALLENGES/05-
Arrays
```

Now we see branches
from the upstream server
in the list.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/JS-BASICS/ADVANCED/01-execution-contexts
remotes/origin/JS-BASICS/ADVANCED/02-this
remotes/origin/JS-BASICS/ADVANCED/03-call-bind-apply
remotes/origin/JS-BASICS/ADVANCED/04-pass-by-value-vs-pass-by-reference
remotes/origin/JS-BASICS/ADVANCED/05-IIFE
remotes/origin/JS-BASICS/ADVANCED/06-synchronous-code-execution
remotes/origin/JS-BASICS/ADVANCED/07-events
remotes/origin/JS-BASICS/ADVANCED/08-callbacks
remotes/origin/JS-BASICS/ADVANCED/09-closures
remotes/origin/JS-BASICS/CORE/01-variables
remotes/origin/JS-BASICS/CORE/02-dynamic-typing
remotes/origin/JS-BASICS/CORE/03-objects
remotes/origin/JS-BASICS/CORE/04-functions
remotes/origin/JS-BASICS/CORE/05-function-expressions
remotes/origin/JS-BASICS/CORE/06-operators
remotes/origin/JS-BASICS/CORE/07-expressions-vs-statements
remotes/origin/JS-BASICS/CORE/08-scopes
remotes/origin/JS-BASICS/CORE/09-arrays
remotes/origin/JS-BASICS/CORE/10-loops
remotes/origin/JS-BASICS/CORE/11-conditions
remotes/origin/JS-BASICS/CORE/CHALLENGES/01-function-expressions/finish
remotes/origin/JS-BASICS/CORE/CHALLENGES/01-function-expressions/start
remotes/origin/JS-BASICS/CORE/CHALLENGES/02-operators
remotes/origin/JS-BASICS/CORE/CHALLENGES/03-expressions-vs-statements
remotes/origin/JS-BASICS/CORE/CHALLENGES/04-scopes
remotes/origin/JS-BASICS/CORE/CHALLENGES/05-Arrays
remotes/origin/JS-BASICS/CORE/CHALLENGES/06-loops-and-conditions
remotes/origin/JS-BASICS/INTRO/01-embedded-script
remotes/origin/JS-BASICS/INTRO/02-external-script
remotes/origin/JS-BASICS/INTRO/CHALLENGE-1/external-scripts/finish
remotes/origin/JS-BASICS/INTRO/CHALLENGE-1/external-scripts/start
remotes/origin/master
remotes/upstream/JS-BASICS/ADVANCED/01-execution-contexts
remotes/upstream/JS-BASICS/ADVANCED/02-this
remotes/upstream/JS-BASICS/ADVANCED/03-call-bind-apply
remotes/upstream/JS-BASICS/ADVANCED/04-pass-by-value-vs-pass-by-reference
remotes/upstream/JS-BASICS/ADVANCED/05-IIFE
remotes/upstream/JS-BASICS/ADVANCED/06-synchronous-code-execution
remotes/upstream/JS-BASICS/ADVANCED/07-events
remotes/upstream/JS-BASICS/ADVANCED/08-callbacks
remotes/upstream/JS-BASICS/ADVANCED/09-closures
remotes/upstream/JS-BASICS/CORE/01-variables
remotes/upstream/JS-BASICS/CORE/02-dynamic-typing
```

If we check the **.git/refs/remotes** directory for the upstream server we can find more directories and keep going down till we find files.

A **cat** command will show a SHA1 for that commit of the file.

All remote files and folders are located in **.git/refs/remotes/upstream**

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la .git/refs/remotes
total 0
drwxr-xr-x 1 ellio 197609 0 Mar  7 18:16 ./
drwxr-xr-x 1 ellio 197609 0 Mar  7 17:30 ../
drwxr-xr-x 1 ellio 197609 0 Mar  7 17:30 origin/
drwxr-xr-x 1 ellio 197609 0 Mar  7 18:16 upstream/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la .git/refs/remotes/upstream
total 1
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ./
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ../
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 JS-BASICS/
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 master

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la .git/refs/remotes/upstream/JS-BASICS/
total 8
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ./
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ../
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ADVANCED/
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 CORE/
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 INTRO/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la .git/refs/remotes/upstream/JS-BASICS/ADVANCED/
total 13
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ./
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 ../
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 01-execution-contexts
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 02-this
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 03-call-bind-apply
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 04-pass-by-value-vs-pass-by-reference
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 05-IIFE
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 06-synchronous-code-execution
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 07-events
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 08-callbacks
-rw-r--r-- 1 ellio 197609 41 Mar  7 18:16 09-closures

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ cat .git/refs/remotes/upstream/JS-BASICS/ADVANCED/02-this
7f01fadcf5a877de9a0cabe9dcec089065a01bc0
```


Currently we are on the master branch and default pull location is origin server.

If we want to pull from upstream server then we have to specify **git pull upstream master**

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ ls -la
total 9
drwxr-xr-x 1 ellio 197609  0 Mar  7 17:30 ./
drwxr-xr-x 1 ellio 197609  0 Mar  7 17:30 ../
drwxr-xr-x 1 ellio 197609  0 Mar  7 18:16 .git/
-rw-r--r-- 1 ellio 197609 111 Mar  7 17:30 README.md

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git branch -vv
* master 9db9556 [origin/master] Create README.md

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git pull upstream master -v
POST git-upload-pack (327 bytes)
From https://github.com/bstashchuk/JavaScript-Bible-Basics
 * branch      master      -> FETCH_HEAD
 = [up to date] master      -> upstream/master
Already up to date.
```

To sync the origin repository we can push.

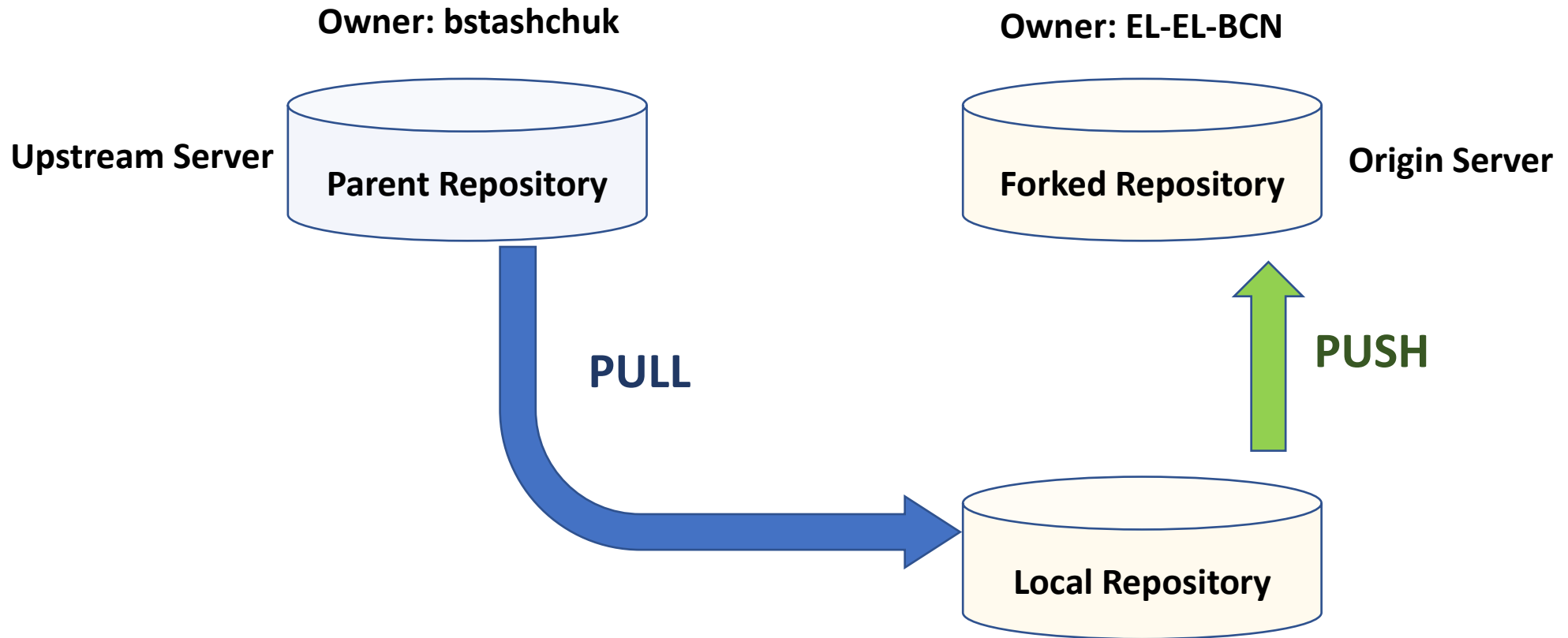
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git pull upstream master -v
POST git-upload-pack (327 bytes)
From https://github.com/bstashchuk/JavaScript-Bible-Basics
 * branch      master      -> FETCH_HEAD
 = [up to date] master      -> upstream/master
Already up to date.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$ git push -v
Pushing to https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git
To https://github.com/EL-EL-BCN/JavaScript-Bible-Basics.git
 = [up to date] master -> master
updating local tracking ref 'refs/remotes/origin/master'
Everything up-to-date

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/JavaScript-Bible-Basics (master)
$
```

To fully sync all repositories we first do a pull from the upstream or parent repository to our local repository then push any changes to our origin directory.



If there are any conflicts then git will ask us to manually correct them before performing any action.

How to open a pull request from the forked repository

Fork **Pull-requests** repository from EL-EL-BCN's account

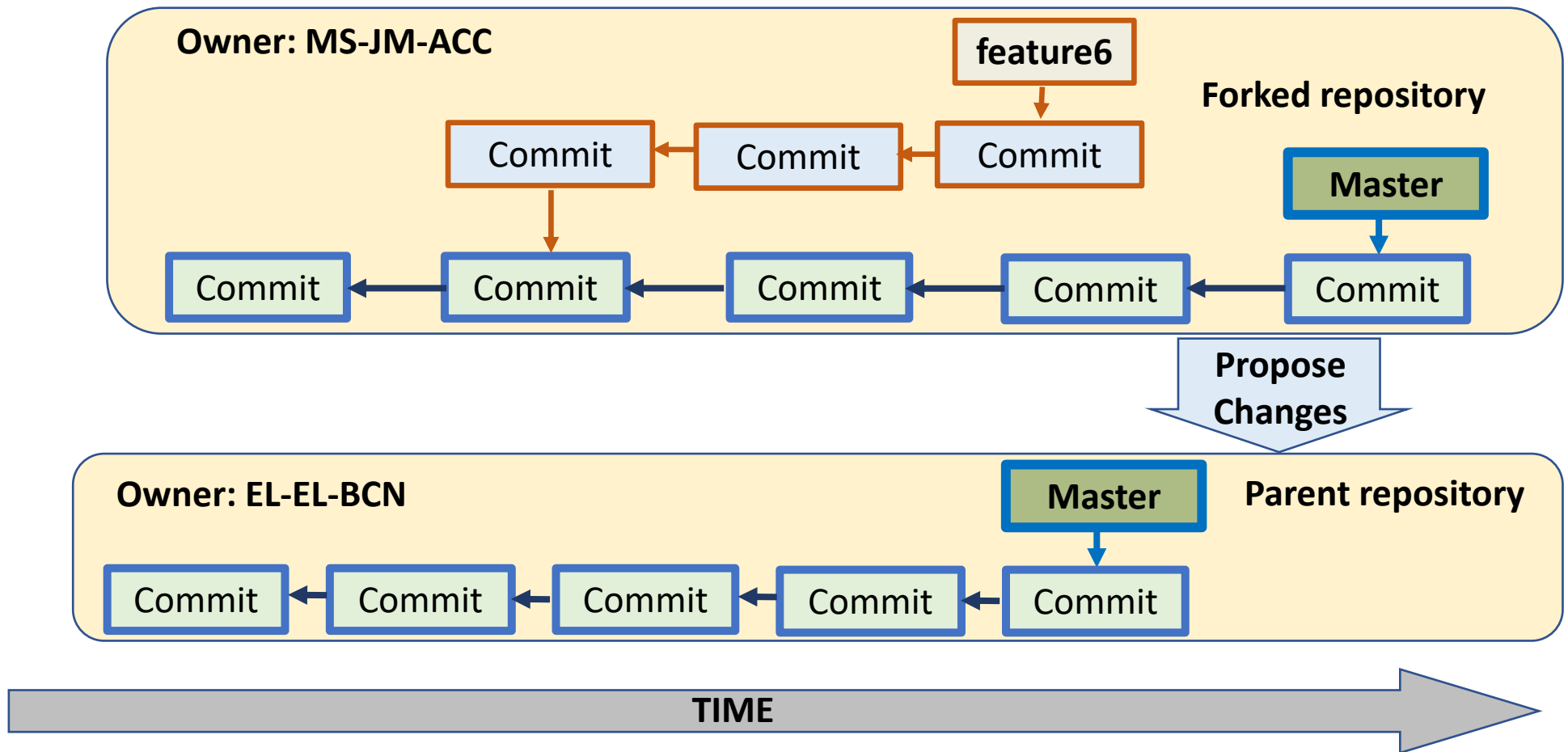
In the forked repository create a new feature branch

Make several commits in the new feature branch

Open a pull request from the forked repository to the parent repository

Request EL-EL-BCN review the changes and perform a merge

Owner of the forked repository is not able to directly push changes into parent repository but can do a pull request.



Fork **Pull-requests** repository from EL-EL-BCN's account

The screenshot shows the GitHub homepage in a web browser. The address bar displays `https://github.com`. The top navigation bar includes the GitHub logo, a search bar with the placeholder text "Search or jump to...", and links for "Pull requests", "Issues", "Marketplace", and "Explore". On the right side of the navigation bar, there are icons for notifications, a plus sign, and a user profile picture.

On the left sidebar, under "Recent Repositories", there is a "New" button and a search bar labeled "Find a repository...". Below this, two repositories are listed: "EL-EL-BCN/pull-requests" and "MS-JM-ACC/pull-requests". Under "Recent activity", a message states: "When you take actions across GitHub, we'll provide links to that activity here."

The main content area is titled "All activity" and features a large card with the text: "Discover interesting projects and people to populate your personal news feed." Below this, it says: "Your news feed helps you keep up with recent activity on repositories you [watch](#) or [star](#) and people you [follow](#)." There is an "Explore GitHub" button at the bottom of this card.

Below the card, a "ProTip!" message states: "The feed shows you events from people you [follow](#) and repositories you [watch](#) or [star](#)." There is also a link to "Subscribe to your news feed".

At the bottom of the page, the footer includes the GitHub logo, "© 2022 GitHub, Inc.", and links for "Blog", "About", "Shop", "API", "Training", "Status", "Terms", "Privacy", and "Docs".

On the right side, a user profile dropdown menu is open, showing the user is signed in as "MS-JM-ACC". The menu includes options: "Set status", "Your profile", "Your repositories", "Your codespaces", "Your projects", "Your stars", "Your gists", "Upgrade", "Feature preview" (with a blue dot), "Help", "Settings", and "Sign out".

In the desktop folder, the pull requests folder is deleted. I then clone the forked directory from user MS-JM-ACC.

I configure the upstream remote server to EL-EL-BCN/pull-request repository on github. And verify that I have an origin and an upstream server configured.

I then do a pull of the upstream main branch and verify that both upstream and origin branches are in the local repository.

```
MINGW64/c/Users/ellio/Desktop/pull-requests
ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ ls
'Exif Pilot.lnk'*      Sourcetree.lnk*      desktop.ini      'win10 to linux migration.ods'
'GitHub Desktop.lnk'* Vivaldi.lnk*         my-project/
JavaScript-Bible-Basics/ 'cita previa.jpg'    pull-requests/

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ rm -rf JavaScript-Bible-Basics/

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ ls
'Exif Pilot.lnk'*      Vivaldi.lnk*         my-project/
'GitHub Desktop.lnk'* 'cita previa.jpg'    pull-requests/
Sourcetree.lnk*        desktop.ini           'win10 to linux migration.ods'

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ rm -rf pull-requests/

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ ls
'Exif Pilot.lnk'*      Sourcetree.lnk*      'cita previa.jpg'    my-project/
'GitHub Desktop.lnk'* Vivaldi.lnk*         desktop.ini           'win10 to linux migration.ods'

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ git clone https://github.com/MS-JM-ACC/pull-requests.git
Cloning into 'pull-requests'...
remote: Enumerating objects: 61, done.
remote: Counting objects: 100% (61/61), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 61 (delta 12), reused 19 (delta 2), pack-reused 0
Receiving objects: 100% (61/61), 12.95 KiB | 4.32 MiB/s, done.
Resolving deltas: 100% (12/12), done.

ellio@DESKTOP-U93252R MINGW64 ~/Desktop
$ cd pull-requests/

ellio@DESKTOP-U93252R MINGW64 ~/Desktop/pull-requests (main)
$ git remote add upstream https://github.com/EL-EL-BCN/pull-requests.git

ellio@DESKTOP-U93252R MINGW64 ~/Desktop/pull-requests (main)
$ git remote -vv
origin  https://github.com/MS-JM-ACC/pull-requests.git (fetch)
origin  https://github.com/MS-JM-ACC/pull-requests.git (push)
upstream https://github.com/EL-EL-BCN/pull-requests.git (fetch)
upstream https://github.com/EL-EL-BCN/pull-requests.git (push)

ellio@DESKTOP-U93252R MINGW64 ~/Desktop/pull-requests (main)
$ git pull upstream -v
POST git-upload-pack (165 bytes)
From https://github.com/EL-EL-BCN/pull-requests
 * [new branch]      feature1  -> upstream/feature1
 * [new branch]      feature4  -> upstream/feature4
 * [new branch]      main      -> upstream/main
You asked to pull from the remote 'upstream', but did not specify
a branch. Because this is not the default configured remote
for your current branch, you must specify a branch on the command line.

ellio@DESKTOP-U93252R MINGW64 ~/Desktop/pull-requests (main)
$ git pull upstream main -v
POST git-upload-pack (315 bytes)
From https://github.com/EL-EL-BCN/pull-requests
 * branch            main      -> FETCH_HEAD
 * [up to date]      main      -> upstream/main
Already up to date.

ellio@DESKTOP-U93252R MINGW64 ~/Desktop/pull-requests (main)
$ git branch -a
* main
remotes/origin/HEAD -> origin/main
remotes/origin/feature1
remotes/origin/feature4
remotes/origin/main
remotes/upstream/feature1
remotes/upstream/feature4
remotes/upstream/main
```

Two additional files and commit were created in new branch called feature6 in the forked directory in user MS-JM-ACC. Both new files were committed.

The screenshot shows a GitHub pull request page for the user **MS-JM-ACC**. The page title is **pull-requests** (Public), forked from **EL-EL-BCN/pull-requests**. The navigation bar includes links for **Code**, **Pull requests**, **Actions**, **Projects**, **Wiki**, **Security**, **Insights**, and **Settings**. A dropdown menu is open on the right, showing the user's profile and options like **Set status**, **Your profile**, **Your repositories**, **Your codespaces**, **Your projects**, **Your stars**, **Your gists**, **Upgrade**, **Feature preview**, **Help**, **Settings**, and **Sign out**.

The main content area shows the **feature6** branch. Under the heading **Commits on Mar 7, 2022**, there is a list of commits:

- Create another-new-file.txt** (MS-JM-ACC committed 1 minute ago) - **Verified** (indicated by a red arrow)
- Create forked-branch-pull-request.txt** (MS-JM-ACC committed 2 minutes ago) - **Verified** (indicated by a red arrow)
- Merge pull request EL-EL-BCN#8 from EL-EL-BCN/feature2-impr** (EL-EL-BCN committed 19 hours ago) - **Verified**
- Create additional-file.txt** (MS-JM-ACC committed 10 hours ago) - **Verified**

Git now gives us the option to open pull request. Note how it now says base repository of EL-EL-BCN/pull-requests main branch and head repository of MS-JM-ACC/pull-requests feature6 branch.

The screenshot shows the GitHub 'Open a pull request' page. At the top, the URL is `https://github.com/EL-EL-BCN/pull-requests/compare/main...MS-JM-ACC:feature6?expand=1`. The page title is 'Open a pull request' with a subtitle 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this, the 'base repository' is set to 'EL-EL-BCN/pull-requests' and the 'base' branch is 'main'. The 'head repository' is 'MS-JM-ACC/pull-requests' and the 'compare' branch is 'feature6'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.'

The main form has a title field containing '[Feature6] new feature6 was implemented'. Below the title is a 'Write' tab and a 'Preview' tab. The 'Write' tab shows a description: 'I would like to add some additional features to the parent repository. I have created new folder feature6 with two files'. Below the description is a checkbox labeled 'Allow edits by maintainers' which is checked. A green 'Create pull request' button is on the right. At the bottom, a summary bar shows '2 commits', '2 files changed', and '1 contributor'.

Red annotations with arrows point to specific elements: one points to the 'base repository' dropdown, another to the 'head repository' dropdown, a third to the 'Allow edits by maintainers' checkbox, and a fourth to the '1 contributor' summary.

Here we add a title with the branch name in square brackets

Here we add a detailed description of the changes

Leave Allow edits by maintainers checked

Here it says we are creating pull request for 2 commits, 2 files changed and 1 contributor.

Now it tells us that pull request has been sent to EL-EL-BCN/pull-requests repository and they are pending review and approval.

Note that the pull request is for 2 commits into EL-EL-BCN-main from MS-JM-ACC-feature6

The screenshot shows a GitHub pull request page for the repository `EL-EL-BCN/pull-requests`, pull request #9. The title is "[Feature6] new feature6 was implemented #9". The description states "MS-JM-ACC wants to merge 2 commits into `EL-EL-BCN:main` from `MS-JM-ACC:feature6`".

The pull request is in a state where a review is required. The status bar at the bottom left shows a red 'X' icon and the text "Review required". Below this, it says "At least 1 approving review is required by reviewers with write access. [Learn more.](#)".

The commit history shows two commits added 18 minutes ago by MS-JM-ACC:

- Create `forked-branch-pull-request.txt` (Verified, 7c012c5)
- Create `another-new-file.txt` (Verified, a9d7011)

The right sidebar shows the pull request details, including Reviewers (No reviews), Assignees (No one assigned), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Successfully merging this pull request may close these issues).

A red arrow points from the text "Now it tells us that pull request has been sent to EL-EL-BCN/pull-requests repository and they are pending review and approval." to the "Review required" status. Another red arrow points from the text "Note that the pull request is for 2 commits into EL-EL-BCN-main from MS-JM-ACC-feature6" to the commit history.

GitHub sends an email to the repository owner about the pull request.

In the EL-EL-BCN GitHub account, the pull request tab shows the pending pull request and after clicking on it we can go to files changes tab to review the changes.

The screenshot shows the GitHub interface for a pull request. The top navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The pull request title is "[Feature6] new feature6 was implemented #9". Below the title, it says "MS-JM-ACC wants to merge 2 commits into EL-EL-BCN:main from MS-JM-ACC:feature6". The "Files changed" tab is selected, showing a list of changes. The first file is "feature6/another-new-file.txt" with 1 change. The second file is "feature6/forked-branch-pull-request.txt" with 3 changes. A "Review changes" button is visible on the right side of the page.

Lets assume we are happy with the changes we can review them and approve.

EL-EL-BCN can approve the changes from feature6 branch of MS-JM-ACC's forked repository.

The screenshot shows a GitHub pull request page for the repository `EL-EL-BCN`. The pull request is titled "MS-JM-ACC wants to merge 2 commits into `EL-EL-BCN:main` from `MS-JM-ACC:feature6`". The interface includes tabs for Conversation (0), Commits (2), Checks (0), and Files changed (2). A list of files changed is shown, including `feature6/another-new-file.txt` and `feature6/forked-branch-pull-request.txt`. A modal titled "Finish your review" is open, displaying a text area with the comment "changes from feature6 branch of forked repository approved". Below the text area are three radio button options: "Comment" (unselected), "Approve" (selected), and "Request changes" (unselected). A "Submit review" button is at the bottom of the modal. A red arrow points from the top of the page to the "Approve" option in the modal. The footer of the page includes the GitHub logo, copyright notice "© 2022 GitHub, Inc.", and links for Terms, Privacy, Security, Status, and Docs. An "Activate Windows" watermark is visible in the bottom right corner.

github.com/EL-EL-BCN/pull-requests/pull/9/files

Open MS-JM-ACC wants to merge 2 commits into `EL-EL-BCN:main` from `MS-JM-ACC:feature6`

Conversation 0 Commits 2 Checks 0 Files changed 2

Changes from all commits File filter Conversations Jump to

1 feature6/another-new-file.txt

... @@ -0,0 +1 @@

3 feature6/forked-branch-pull-request.txt

... @@ -0,0 +1,3 @@

0 / 2 files viewed Review changes

Finish your review

Write Preview H B I `< > Link List Unlink Checkmark @ Share Undo`

changes from feature6 branch of forked repository approved

Attach files by dragging & dropping, selecting or pasting them.

☐ Comment
Submit general feedback without explicit approval.

☒ Approve
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

Submit review

© 2022 GitHub, Inc. Terms Privacy Security Status Docs

Activate Windows
Go to Settings to activate Windows.

Reset 100% 20:59

The pull request can now be automatically merged because it has no conflicts.

github.com/EL-EL-BCN/pull-requests/pull/9

[Feature6] new feature6 was implemented #9
MS-JM-ACC wants to merge 2 commits into EL-EL-BCN:main from MS-JM-ACC:feature6

EL-EL-BCN left a comment
changes from feature6 branch of forked repository approved

Add more commits by pushing to the **feature6** branch on MS-JM-ACC/pull-requests.

Changes approved
1 approving review by reviewers with write access. [Learn more.](#)

1 approval

Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications
[Unsubscribe](#)
You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

Activate Windows
Go to Settings to activate Windows.

Note that there was no option to delete the branch after confirming the merge. This is because the branch was created in the personal forked repository of MS-JM-ACC. MS-JM-ACC does not have write access to EL-EL-BCN/pull-requests but can propose changes via a pull request.

Once the EL-EL-BCN has completed the merge then GitHub will send an email confirmation and MS-JS-ACC can delete the branch.

The screenshot shows the GitHub interface for the repository 'MS-JM-ACC' at the path 'pull-requests/branches'. The page has a navigation bar with links to Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there are tabs for 'Overview' (selected), 'Yours', 'Active', 'Stale', and 'All branches'. A search bar labeled 'Search branches...' is on the right. The main content area is divided into three sections: 'Default branch', 'Your branches', and 'Active branches'. The 'Default branch' section shows the 'main' branch, updated 20 hours ago by EL-EL-BCN, with a 'Default' label and an edit icon. The 'Your branches' section shows the 'feature6' branch, updated 40 minutes ago by MS-JM-ACC, with a 'New pull request' button, an edit icon, and a delete icon. The 'Active branches' section shows three branches: 'feature6' (updated 40 minutes ago by MS-JM-ACC), 'feature4' (updated 21 hours ago by EL-EL-BCN), and 'feature1' (updated 2 days ago by EL-EL-BCN). Each branch in the 'Active branches' section has a 'New pull request' button, an edit icon, and a delete icon. A red arrow points from the text above to the delete icon of the 'feature6' branch in the 'Your branches' section. A tooltip 'Delete this branch' is visible next to the delete icon.

Branch	Updated	By	Commits	Actions
main	Updated 20 hours ago	EL-EL-BCN	0 2	New pull request, Edit, Delete
feature6	Updated 40 minutes ago	MS-JM-ACC	0 2	New pull request, Edit, Delete
feature4	Updated 21 hours ago	EL-EL-BCN	6 0	New pull request, Edit, Delete
feature1	Updated 2 days ago	EL-EL-BCN	17 0	New pull request, Edit, Delete

© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Activate Windows
Go to Settings to activate Windows.

Note that in MS-JM-ACCs forked repository feature6 changes are not showing because this repository now needs to be sync'd with the parent.

The screenshot shows the GitHub interface for the repository **MS-JM-ACC / pull-requests**, which is a public fork of **EL-EL-BCN/pull-requests**. The page displays the commit history for the **main** branch, which is 3 commits behind the upstream **EL-EL-BCN:main**. The commit history table lists the following entries:

Commit	Message	Time
feature1	Another change	2 days ago
feature2	Create additional-file.txt	20 hours ago
feature3	Update new-collaborator.txt	21 hours ago
feature4	Update create-new-file.txt	21 hours ago
feature5	Create protected-branch.txt	21 hours ago
README.md	Initial commit	2 days ago
collaborator-file1.txt	collaborator1 first commit	2 days ago

A red arrow points to the **feature3** entry. The right sidebar shows repository statistics: 0 stars, 0 watching, and 1 fork. The 'About' section states: 'We will create different pull requests'. The 'Releases' section indicates 'No releases published' with a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' with a link to 'Publish your first package'.

Fetch upstream can be performed from the MS-JM-ACC/pull-requests repository.

The screenshot shows the GitHub repository page for `MS-JM-ACC / pull-requests`, which is a public repository forked from `EL-EL-BCN/pull-requests`. The page includes navigation tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A red arrow points to the `Fetch upstream` dropdown menu, which is open, showing the option to `Fetch and merge 3 upstream commits from EL-EL-BCN:main`. The dropdown also includes a `Compare` button and a `Fetch and merge` button. The main content area displays a list of files and their commit history, including `feature1` through `feature5`, `README.md`, and `collaborator-file1.txt`. The right sidebar contains information about the repository, including the number of stars (0), watching (0), and forks (1), as well as a section for releases and packages.

MS-JM-ACC / pull-requests Public

forked from EL-EL-BCN/pull-requests

Code Pull requests Actions Projects Wiki Security Insights Settings

main 3 branches 0 tags

This branch is 3 commits behind EL-EL-BCN:main.

Contribute Fetch upstream

Fetch and merge 3 upstream commits from EL-EL-BCN:main.
Keep your fork up-to-date with the upstream repository.
[Learn more](#)

Compare Fetch and merge

File	Commit	Time
feature1	Another change	
feature2	Create additional-file.txt	
feature3	Update new-collaborator.txt	
feature4	Update create-new-file.txt	21 hours ago
feature5	Create protected-branch.txt	21 hours ago
README.md	Initial commit	2 days ago
collaborator-file1.txt	colaborator1 first commit	2 days ago

About

We will create different pull requests

Readme

0 stars

0 watching

1 fork

Releases

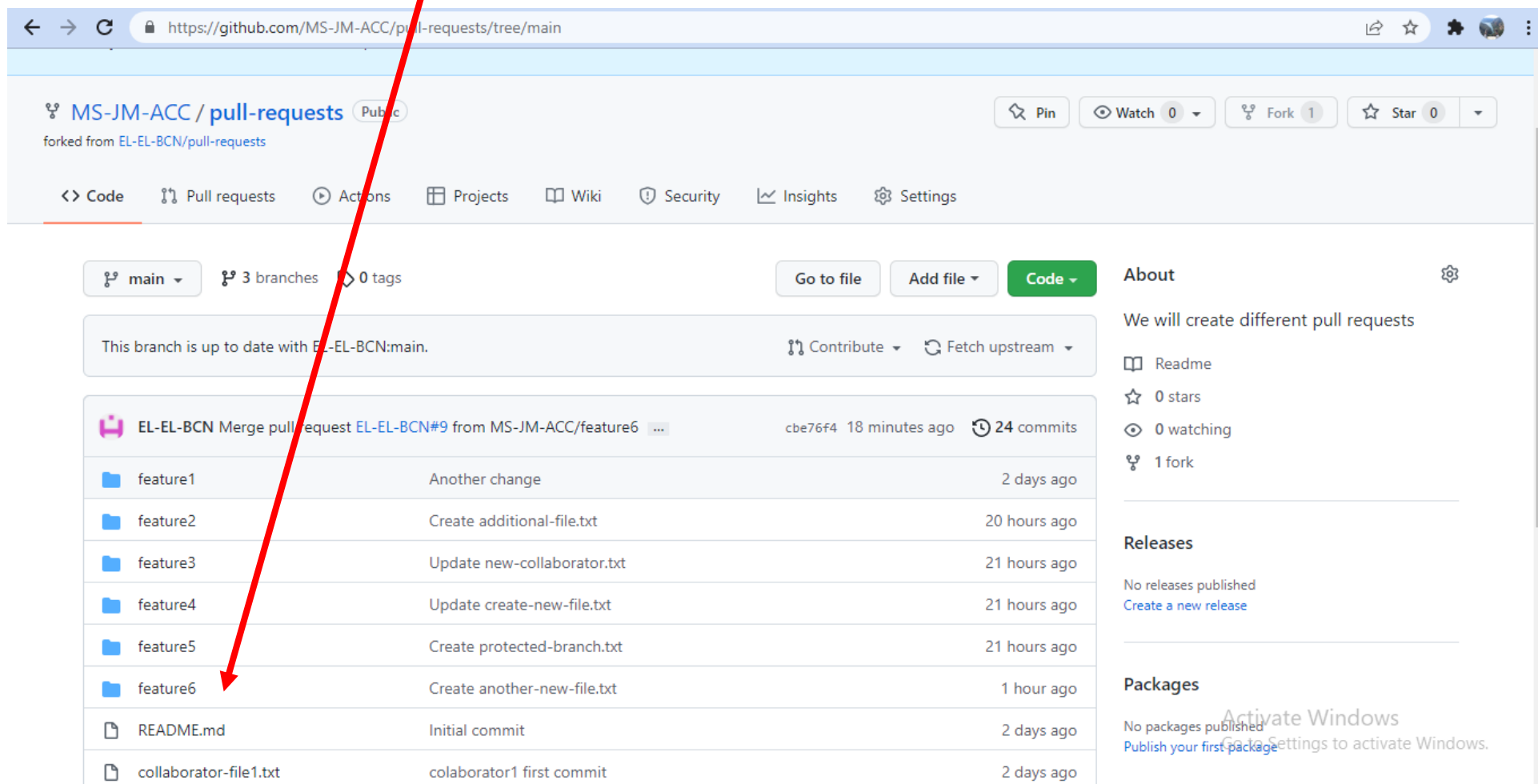
No releases published
[Create a new release](#)

Packages

Activate Windows
Go to Settings to activate Windows.

No packages published
[Publish your first package](#)

Now feature6 folder and files have been updated.



The screenshot shows a GitHub pull request page for the repository `MS-JM-ACC / pull-requests`, which is forked from `EL-EL-BCN / pull-requests`. The page displays a list of files and folders in the `main` branch. A red arrow points to the `feature6` folder, indicating it has been updated.

Repository: `MS-JM-ACC / pull-requests` (Public)
forked from `EL-EL-BCN / pull-requests`

Navigation: `<> Code`, `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, `Settings`

Branches: `main` (3 branches), `0 tags`

Buttons: `Go to file`, `Add file`, `Code`

Status: This branch is up to date with `EL-EL-BCN:main`.
Buttons: `Contribute`, `Fetch upstream`

Recent Activity: `EL-EL-BCN` Merge pull request `EL-EL-BCN#9` from `MS-JM-ACC/feature6` (cbe76f4, 18 minutes ago, 24 commits)

File/Folder	Description	Time
<code>feature1</code>	Another change	2 days ago
<code>feature2</code>	Create additional-file.txt	20 hours ago
<code>feature3</code>	Update new-collaborator.txt	21 hours ago
<code>feature4</code>	Update create-new-file.txt	21 hours ago
<code>feature5</code>	Create protected-branch.txt	21 hours ago
<code>feature6</code>	Create another-new-file.txt	1 hour ago
<code>README.md</code>	Initial commit	2 days ago
<code>collaborator-file1.txt</code>	collaborator1 first commit	2 days ago

Right sidebar:

- About**: We will create different pull requests. 0 stars, 0 watching, 1 fork.
- Releases**: No releases published. [Create a new release](#).
- Packages**: No packages published. [Publish your first package](#).

Watermark: Activate Windows. Go to Settings to activate Windows.

Summary of Shell Commands

git remote add upstream

Add an upstream (parent) repository for a forked repository

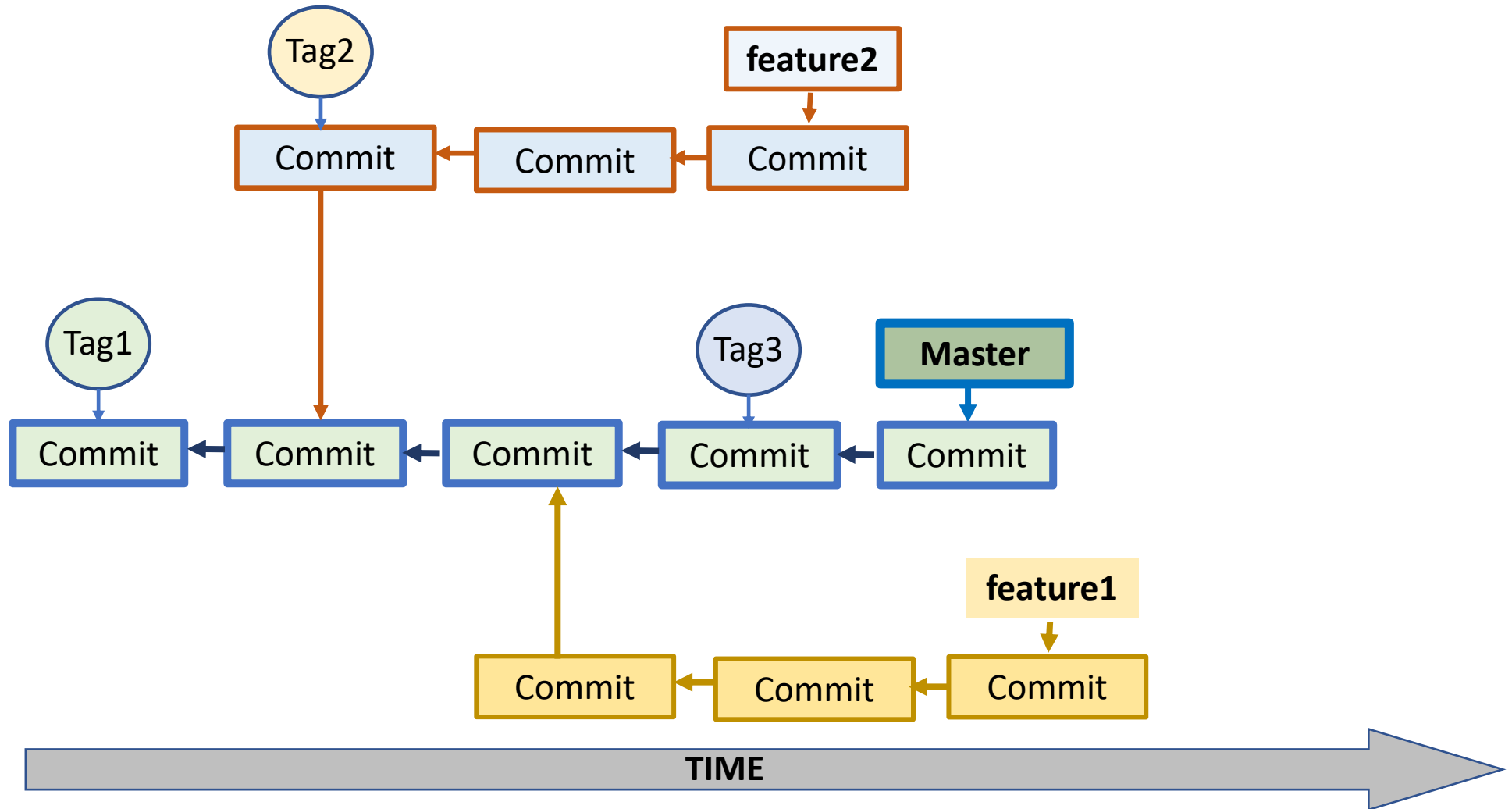
git pull upstream <branch name> -v

Pull changes from upstream remote repository

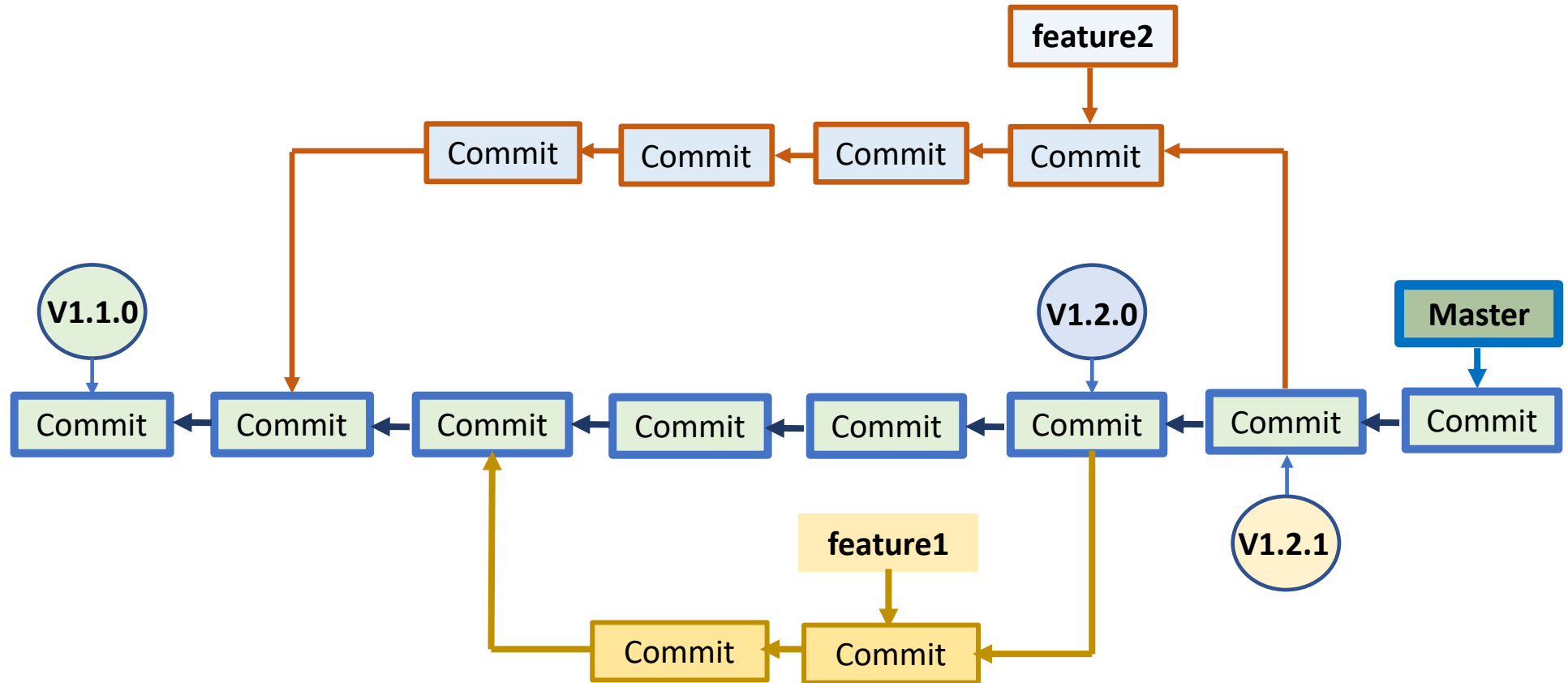
GIT TAGS

Git Tags

Branch pointer is dynamic which moves whenever a new commit is added. Tag is a static text pointer to specific commit. We can at any time checkout a specific tag. Tags are used mostly for release versions of a project.



Project release versioning



CI/CD – Continuous Integration / Continuous Development



- Primarily for testing
 - Usually internal use
 - Merging is performed frequently
 - Different feature branches are merged into release branch
 - Multiple people may have merge rights
- For stable production service
 - For customers
 - Merging happens each 2 weeks or 1 month
 - Usually only release branch is merged into master branch
 - Only few people have merge rights
- **Hotfixes may go straight into production**

Semantic Versioning

5.1.3

<https://semver.org>

Major

Minor

Patch

Major version is 2
Minor version is 9
Patch version is 11



GitHub Desktop

Version 2.9.11 (x64) ([release notes](#))

[Terms and Conditions](#)

[License and Open Source Notices](#)

You have the latest version (last checked just now)

[Check for Updates](#)

About

Windows specifications

Edition	Windows 10 Home
Version	21H1
Installed on	01/07/2021
OS build	19043.1526
Experience	Windows Feature Experience Pack 120.2212.4170.0

Major version is 120
Minor version is 2212
Patch version is 4170.0

Major version should be incremented when implementing major changes and the new release is no longer compatible with the previous version. In this case minor and patch are reset to zero.

Minor updates are new features that do not break any existing functionality of the software. Patch is reset to zero when minor version increments.

Patches are small changes and bug fixes incrementing by 1 each time a small fix is implemented.

As most software has external libraries and dependencies versioning needs to be considered when updating this software, i.e. a major upgrade might break the project completely.

5.1.3-4-1 would be an example of pre-release where 5 is major version, 1 is minor version and 3-4.1 is patch. When the software is released the version becomes 5.1.3 and is considered higher version than the pre-release version. Rc can be used to identify release candidate i.e. 3.4.1-RC-3

Git Tags

```
graph TD; A[Git Tags] --> B[Lightweight]; A --> C[Annotated];
```

The diagram shows 'Git Tags' at the top, with two arrows pointing down to 'Lightweight' and 'Annotated'.

git tag v1.0.0

Is stored in the **.git/refs/tags**

git tag -a v1.0.0 -m "new tag"

Is stored in the **.git/refs/tags**

Is also stored in the **.git/objects**

Stores tag message

Stores Tag author and date

It is highly advised to only use annotated tags because they store tag author and datetime
Tag names must be unique across entire repository

github.com/new

Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

New repository called tags is created in github.com

Owner * EL-EL-BCN / Repository name * tags ✓

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-tribble?](#)

Description (optional)
lets play with tags

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

We clone this new repository to our habitual local directory on the local computer.

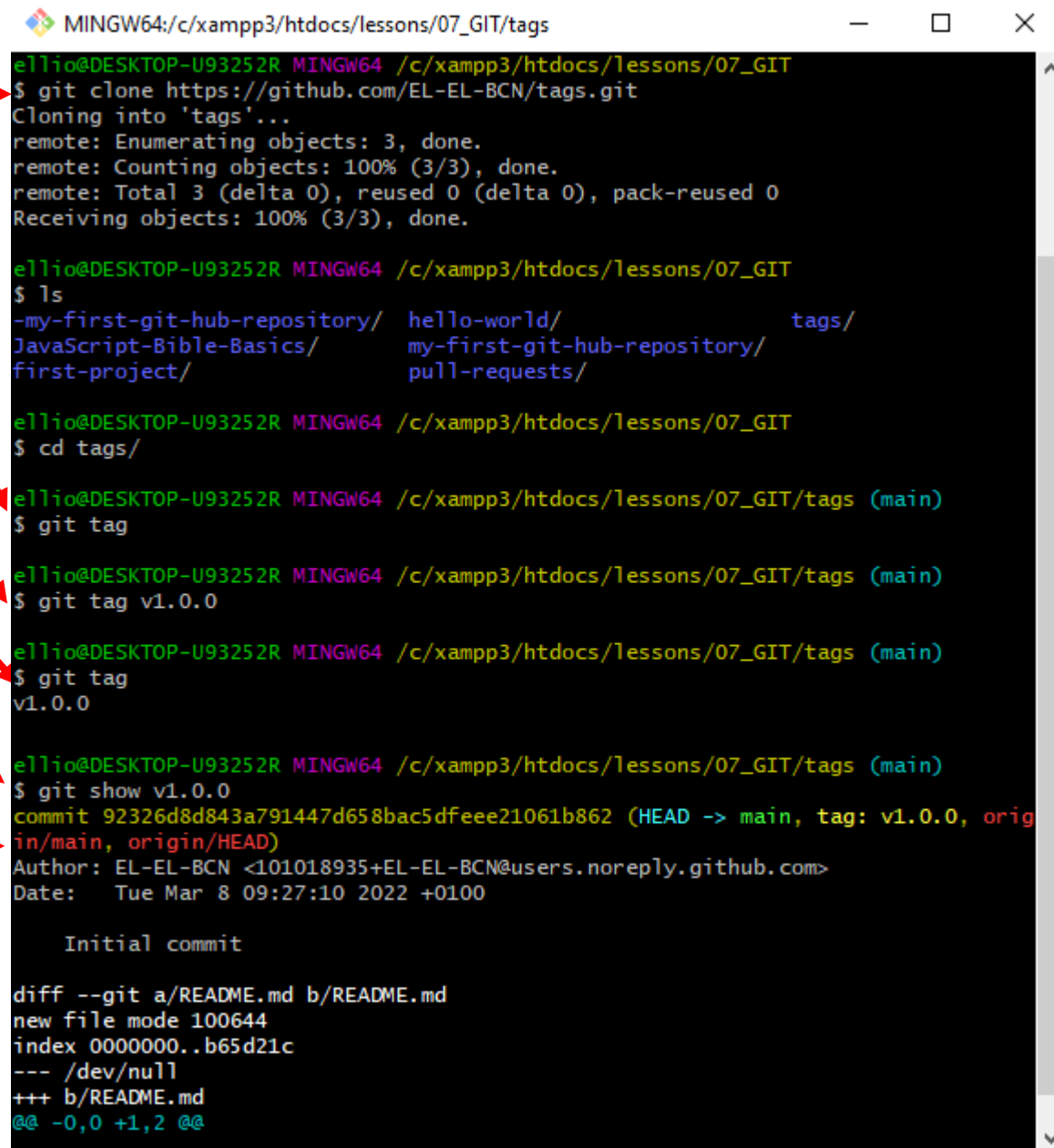
Once in the git folder we can use the command **git tag** to view any tags in the project.

git tag v1.0.0 command is used to add a lightweight tag to the project.

The v1.0.0 tag shows in **git tag** command.

git show v.1.0.0 command gives more detail about the tag.

We have the initial commit with head pointing to main branch, remote (origin) pointing to origin head. Notice how the tag is added into this commit.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/tags
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/tags.git
Cloning into 'tags'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  hello-world/  tags/
JavaScript-Bible-Basics/      my-first-git-hub-repository/
first-project/                pull-requests/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd tags/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag v1.0.0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag
v1.0.0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git show v1.0.0
commit 92326d8d843a791447d658bac5dfcee21061b862 (HEAD -> main, tag: v1.0.0, origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 09:27:10 2022 +0100

    Initial commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..b65d21c
--- /dev/null
+++ b/README.md
@@ -0,0 +1,2 @@
```

We create a new file in the local repository.

We stage the file

We commit the file

Note that It gives me an author identity unknown error because username and email is not set. I am setting this per repository rather than globally.

Once username and email are set the file can be committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ echo "Hello, Git tags" > new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls -la
total 10
drwxr-xr-x 1 ellio 197609  0 Mar  8 09:45 ./
drwxr-xr-x 1 ellio 197609  0 Mar  8 09:27 ../
drwxr-xr-x 1 ellio 197609  0 Mar  8 09:27 .git/
-rw-r--r-- 1 ellio 197609 29 Mar  8 09:27 README.md
-rw-r--r-- 1 ellio 197609 16 Mar  8 09:45 new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git add .
warning: LF will be replaced by CRLF in new-file.txt.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git commit -m "new file was added"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'ellio@DESKTOP-U93252R.(none)')

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git config user.name EL-EL-BCN

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git config user.email elliotbcbn@gmail.com

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git commit -m "new file was added"
[main b079a49] new file was added
1 file changed, 1 insertion(+)
create mode 100644 new-file.txt
```

In git log we see the two commits.
Notice how the tag is fixed to the initial
commit where as the head pointer
moves to the last commit. Tags are
static pointers

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/tags
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git log
commit b079a49c31d5966614cb5cb7e6e0a02671e420f7 (HEAD -> main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 09:48:43 2022 +0100

    new file was added

commit 92326d8d843a791447d658bac5dfeee21061b862 (tag: v1.0.0, origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 09:27:10 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$
```


The **.git/tag/refs** file just has one file with the name v1.0.0

The **.git/tag/refs/v1.0.0** file has the SHA1 of the commit that the tag is attached to.

Because the v.1.0.0 tag is a lightweight tag there are no additional files in the objects folder.

Lightweight tags do not contain additional information.

Git tags are unique in the repository and git will not let us create another tag with the same name.

Now lets push the local repository to the remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/tags
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls .git/refs/tags
v1.0.0
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ cat .git/refs/tags/v1.0.0
92326d8d843a791447d658bac5dfcee21061b862
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls -la .git/objects
total 12
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:48 ./
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:48 ../
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:46 0c/
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:48 b0/
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:47 ff/
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:27 info/
drwxr-xr-x 1 ellio 197609 0 Mar  8 09:27 pack/
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -v v1.0.0
error: v1.0.0: cannot verify a non-tag object of type commit.
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag v1.0.0
fatal: tag 'v1.0.0' already exists
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git push -v
Pushing to https://github.com/EL-EL-BCN/tags.git
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (486 bytes)
To https://github.com/EL-EL-BCN/tags.git
 92326d8..b079a49  main -> main
updating local tracking ref 'refs/remotes/origin/main'
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$
```

github.com/EL-EL-BCN/tags

Search or jump to...

Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

No tags seen in the repository after the push

main 1 branch 0 tags

Go to file Add file Code

About

lets play with tags

Readme

0 stars

1 watching

0 forks

EL-EL-BCN new file was added b079a49 32 minutes ago 2 commits

File	Commit	Time
README.md	Initial commit	1 hour ago
new-file.txt	new file was added	32 minutes ago

Git by default does not push tags created in local repository. This is to avoid conflicts that could arise when multiple collaborators working on the project name tags the same

Now locally we delete the repository and then clone it from remote server.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ rm -rf tags

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/ first-project/ my-first-git-hub-repository/
JavaScript-Bible-Basics/      hello-world/ pull-requests/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/tags.git
Cloning into 'tags'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$
```

The repository does not have any tags

We can create an annotated tag

Now we can see the tag

We can view verbose tag details if we specify the tag name

Here is the SHA1 of the commit object that the tag is attached to.

Here is author and date of annotated tag

Here is tag comment

It is also possible to add signature to an annotated tag

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -a v1.0.0 -m "initial tag"
Committer identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'ellio@DESKTOP-U93252R.(none)')

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git config user.name EL-EL-BCN

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git config user.email elliottnbcn@gmail.com

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -a v1.0.0 -m "initial tag"

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag
v1.0.0

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -v

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -v v1.0.0
object b079a49c31d5966614cb5cb7e6e0a02671e420f7
type commit
tag v1.0.0
tagger EL-EL-BCN <elliottnbcn@gmail.com> 1646731927 +0100
initial tag
error: no signature found

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ |
```

The **.git/tag/refs** file just has one file with the name v1.0.0

The **.git/tag/refs/v1.0.0** file has the SHA1 of the commit that the tag is attached to.

Annotated tags also are stored in the objects directory.

In this case we see the SHA1 hash of the tag object.

This object is a tag

With the contents of the tag object containing the SHA1 of the commit that the tag is attached to plus the author, date time and comment.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags
error: no signature found

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls .git/refs/tags
v1.0.0

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ cat .git/refs/tags/v1.0.0
b369c662d81ae17cb4858cc9a868cbd126c831d3

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls .git/objects
b3/  info/  pack/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls -la .git/objects/b3
total 1
drwxr-xr-x 1 ellio 197609  0 Mar  8 10:32 ./
drwxr-xr-x 1 ellio 197609  0 Mar  8 10:32 ../
-r--r--r-- 1 ellio 197609 136 Mar  8 10:32 69c662d81ae17cb4858cc9a868cbd126c831d3

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git cat-file -t b369c662d81ae17cb4858cc9a868cbd126c831d3
tag

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git cat-file -p b369c662d81ae17cb4858cc9a868cbd126c831d3
object b079a49c31d5966614cb5cb7e6e0a02671e420f7
type commit
tag v1.0.0
tagger EL-EL-BCN <elliottbcn@gmail.com> 1646731927 +0100

initial tag

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ |
```

Git log shows that the main and HEAD are the same in local and remote repositories meaning they are synchronised. In local repository we have tag v1.0.0 attached to new file commit.

Git push does not push tags by default

We have to use **git push --tags** command.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags
$ git log
commit b079a49c31d5966614cb5cb7e6e0a02671e420f7 (HEAD -> main, tag: v1.0.0, origin/main, origin/HEAD)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 09:48:43 2022 +0100

    new file was added

commit 92326d8d843a791447d658bac5dfeee21061b862
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 09:27:10 2022 +0100

    Initial commit

elliott@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git push -v
Pushing to https://github.com/EL-EL-BCN/tags.git
To https://github.com/EL-EL-BCN/tags.git
    = [up to date]      main -> main
    updating local tracking ref 'refs/remotes/origin/main'
    Everything up-to-date

elliott@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git push -v --tags
Pushing to https://github.com/EL-EL-BCN/tags.git
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 165 bytes | 165.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (349 bytes)
To https://github.com/EL-EL-BCN/tags.git
    * [new tag]          v1.0.0 -> v1.0.0

elliott@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$
```

github.com/EL-EL-BCN/tags

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 1 tag Go to file Add file Code About

EL-EL-BCN new file was added b079a49 1 hour ago 2 commits

README.md	Initial commit	2 hours ago
new-file.txt	new file was added	1 hour ago

lets play with tags

Readme 0 stars 1 watching 0 forks

github.com/EL-EL-BCN/tags/tags

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags

Tags

v1.0.0 ...

initial tag

28 minutes ago b079a49 zip tar.gz

Now we can add another file then stage and commit it.

We can add an annotated tag to this commit.

We can push to remote server the tag using **git push -v origin <tagname>**

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ ls
README.md  new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ echo "another file" > another-file.txt

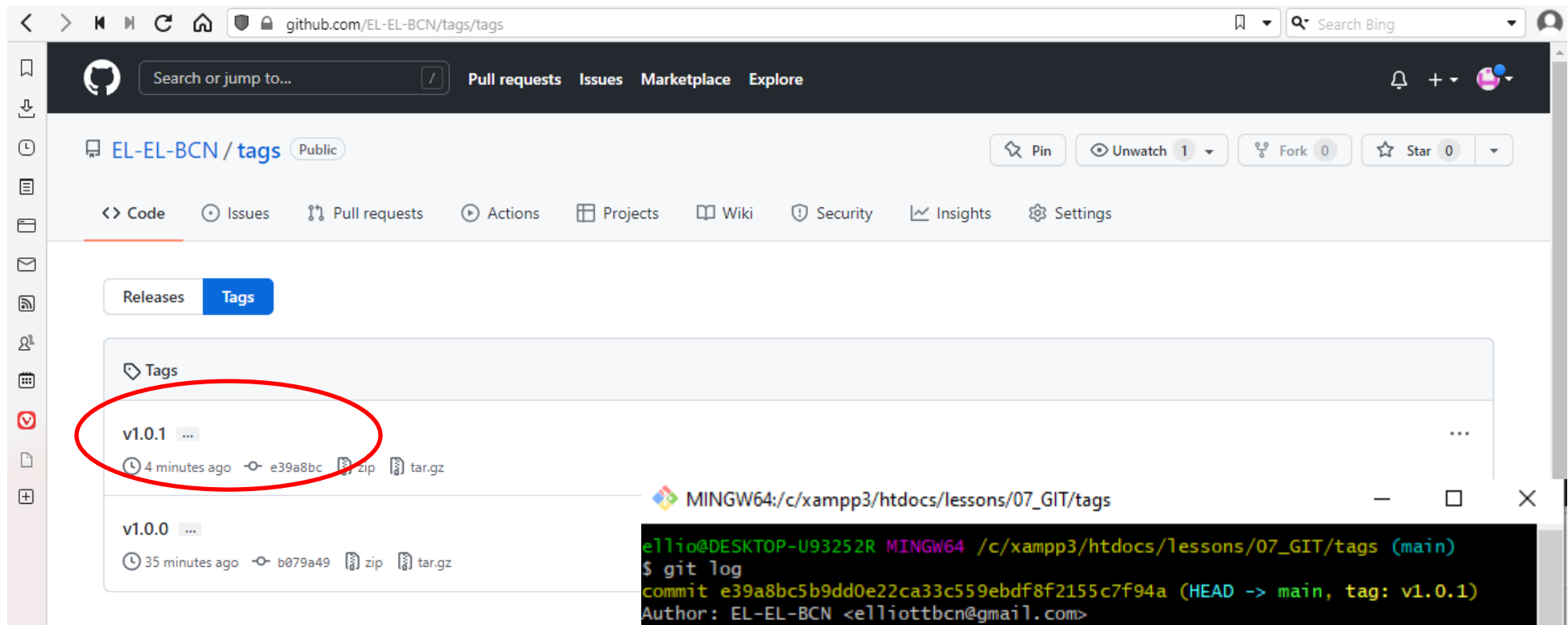
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git add .
warning: LF will be replaced by CRLF in another-file.txt.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git commit -m "Another file"
[main e39a8bc] Another file
1 file changed, 1 insertion(+)
create mode 100644 another-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag -a v1.0.1 -m "patch for v1.0.0"

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git push -v origin v1.0.1
Pushing to https://github.com/EL-EL-BCN/tags.git
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 464 bytes | 464.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (648 bytes)
To https://github.com/EL-EL-BCN/tags.git
 * [new tag]          v1.0.1 -> v1.0.1

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$
```

But the changes were not pushed to the remote repository, only the tag. The main and Head still point to the penultimate commit.

We need to do **git push -v** to push the commits.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/tags
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git log
commit e39a8bc5b9dd0e22ca33c559ebdf8f2155c7f94a (HEAD -> main, tag: v1.0.1)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 11:02:25 2022 +0100

    Another file

commit b079a49c31d5966614cb5cb7e6e0a02671e420f7 (tag: v1.0.0, origin/main, origin/HEAD)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 09:48:43 2022 +0100

    new file was added

commit 92326d8d843a791447d658bac5dfeee21061b862
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 09:27:10 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/tags (main)
$
```


github.com/EL-EL-BCN/tags/edit/main/another-file.txt

Search Bing

Search or jump to...

Pull requestsIssuesMarketplaceExplore

🔔

+

EL-EL-BCN / tags

Public

Pin

Unwatch 1

Fork 0

Star 0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

tags /

another-file.txt

in main

Cancel changes

<> Edit file

Preview changes

Spaces 2Soft wrap

1 another file

2

3 Some changes in this file

Commit changes

Update another-file.txt

Some changes to another file.txt

☒ Commit directly to the main branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

© 2022 GitHub, Inc.

TermsPrivacySecurityStatusDocsContact GitHubPricingAPITrainingBlog

Activate Windows
Go to Settings to activate Windows.

Another-file.txt was changed and committed.

github.com/EL-EL-BCN/tags/releases

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags Find a release

There aren't any releases here

You can create a release to package software, along with release notes and links to binary files, for other people to use. Learn more about releases in [our docs](#).

→ Create a new release

In git hub we can create a new release

Activate Windows
Go to Settings to activate Windows.

https://github.com/EL-EL-BCN/tags/releases/new 100 % 11:16

github.com/EL-EL-BCN/tags/releases/new

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags

v1.0.2 Target: main

Excellent! This tag will be created from the target when you publish this release.

patch to another-file.txt

Write Preview

H B I

+ Auto-generate release notes

a new line was added to another-file.txt

Attach files by dragging & dropping, selecting or pasting them.

No file chosen

↓ Attach binaries by dropping them here or selecting them.

☐ This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish release Save draft

Add the release with the version number

Release title

Release description

Tagging suggestions

It's common practice to prefix your version names with the letter v. Some good tag names might be v1.0.0 or v2.3.4.

If the tag isn't meant for production use, add a pre-release version after the version name. Some good pre-release versions might be v0.2.0-alpha or v5.9-beta.3.

Semantic versioning

If you're new to releasing software, we highly recommend reading about [semantic](#)

github.com/EL-EL-BCN/tags/tags

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / tags Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags

Tags

v1.0.2 ...

Update another-file.txt
Some changes to another file.txt

7 minutes ago c9715d0 zip tar.gz Notes

v1.0.1 ...

20 minutes ago e39a8bc zip tar.gz

v1.0.0 ...

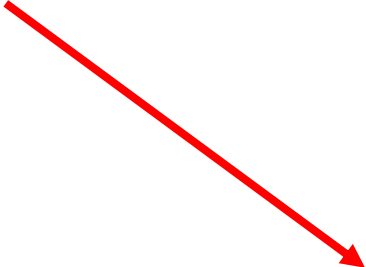
1 hour ago b079a49 zip tar.gz

When we create a release GitHub will automatically provide the option to download the project at this point in zip or tar format.

Activate Windows
Go to Settings to activate Windows.

Reset 100 % 11:23

We can update the local repository by pulling from the remote. Now we see three tags.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/tags
ello@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 693 bytes | 63.00 KiB/s, done.
From https://github.com/EL-EL-BCN/tags
   e39a8bc..c9715d0  main      -> origin/main
* [new tag]         v1.0.2     -> v1.0.2
Updating e39a8bc..c9715d0
Fast-forward
 another-file.txt | 2 ++
 1 file changed, 2 insertions(+)

ello@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ git tag
v1.0.0
v1.0.1
v1.0.2

ello@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/tags (main)
$ |
```

Summary of Shell Commands

git tag

Add lightweight tag

git tag -a v1.0.0 -m "new tag"

Add annotated tag

git show <tag name>

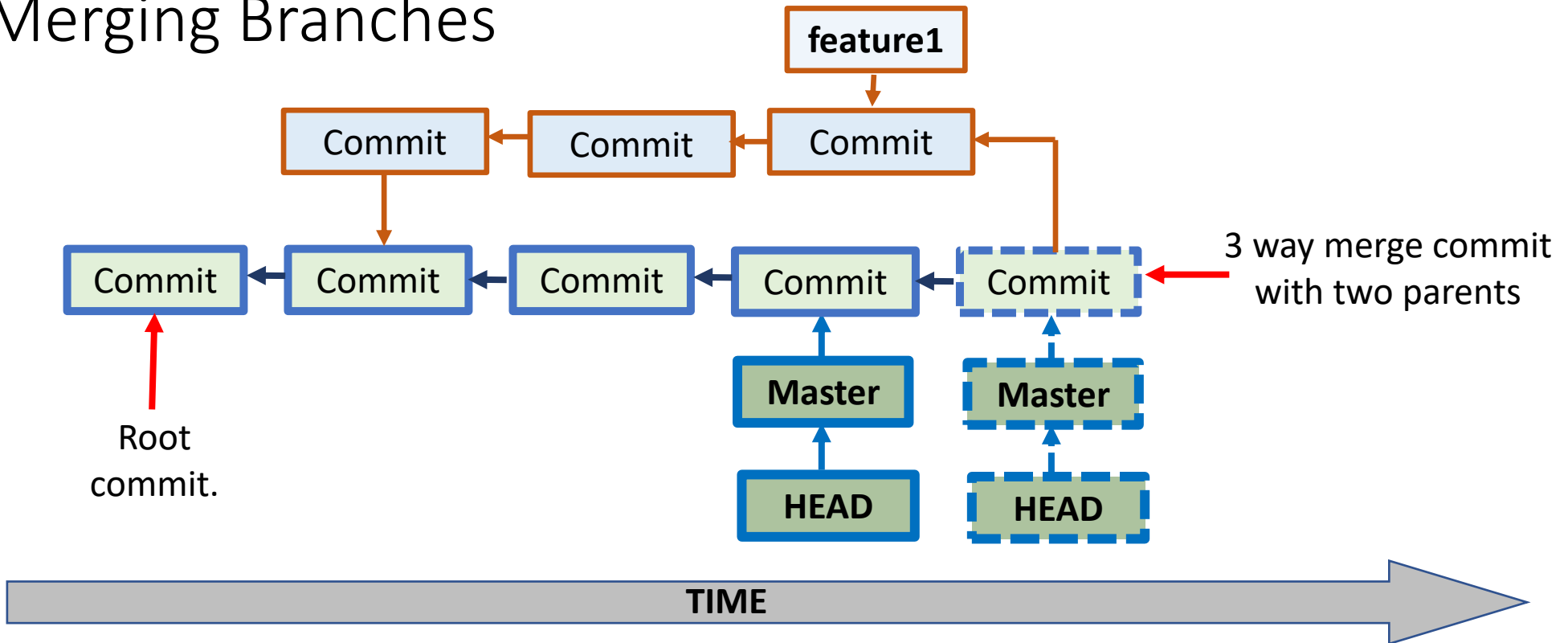
Show detailed information about tag

GIT REBASING

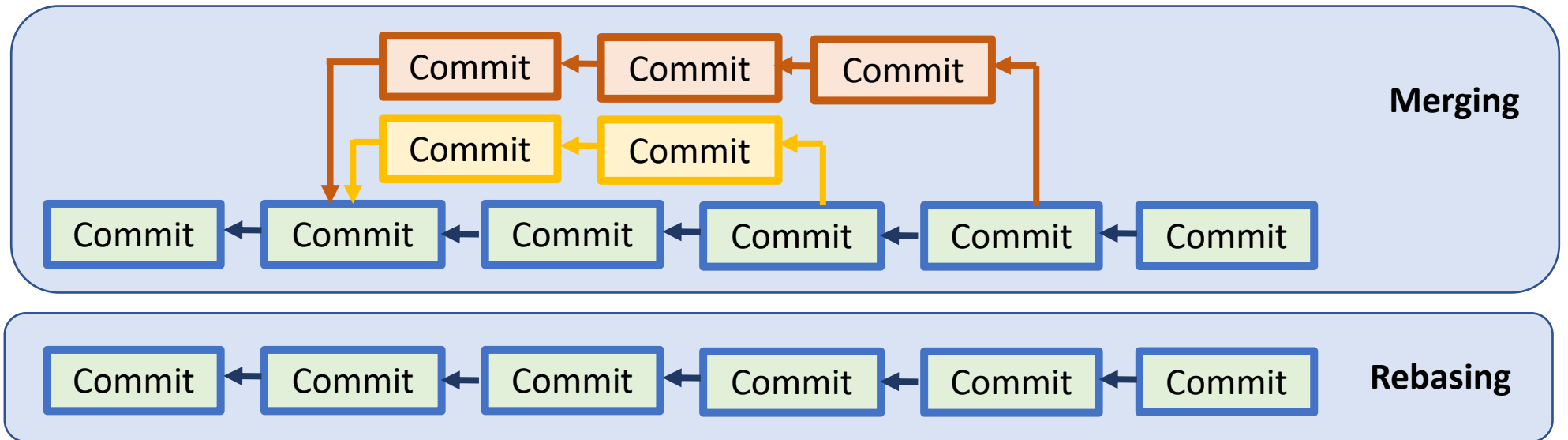
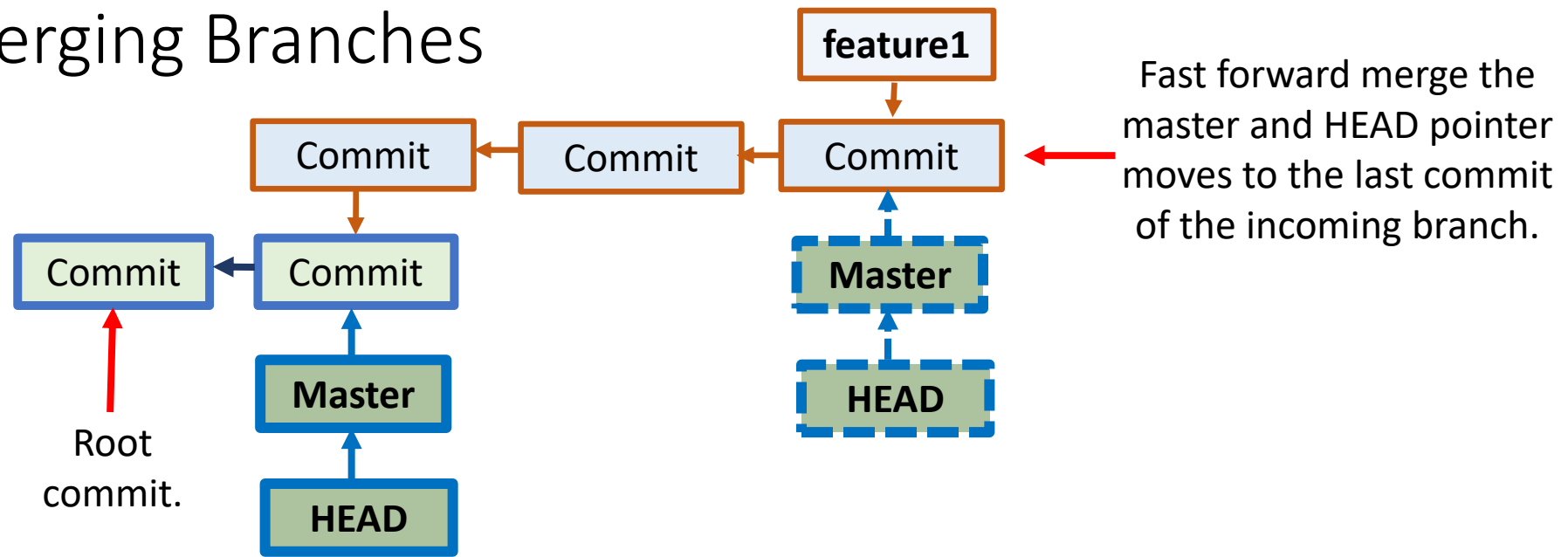
Rebasing Branches

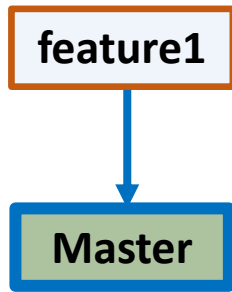
- Rewrites history
- History Becomes Linear. With Merges, commits have multiple parents, Rebasing removes multiple parents so that the history becomes linear.
- Does not keep entire History of all commits. Some commits are lost when rebasing is performed.

Merging Branches



Merging Branches





Merging of the feature1 branch into the master branch using rebasing is a two step process.

- | | |
|---|------------------------------|
| 1. Checkout feature branch | git checkout feature1 |
| 2. Rebase feature branch on top of the base branch | git rebase master |
| 3. Checkout base branch | git checkout master |
| 4. Merge feature branch into the base branch. Fast forward merge will be used. | git merge feature1 |

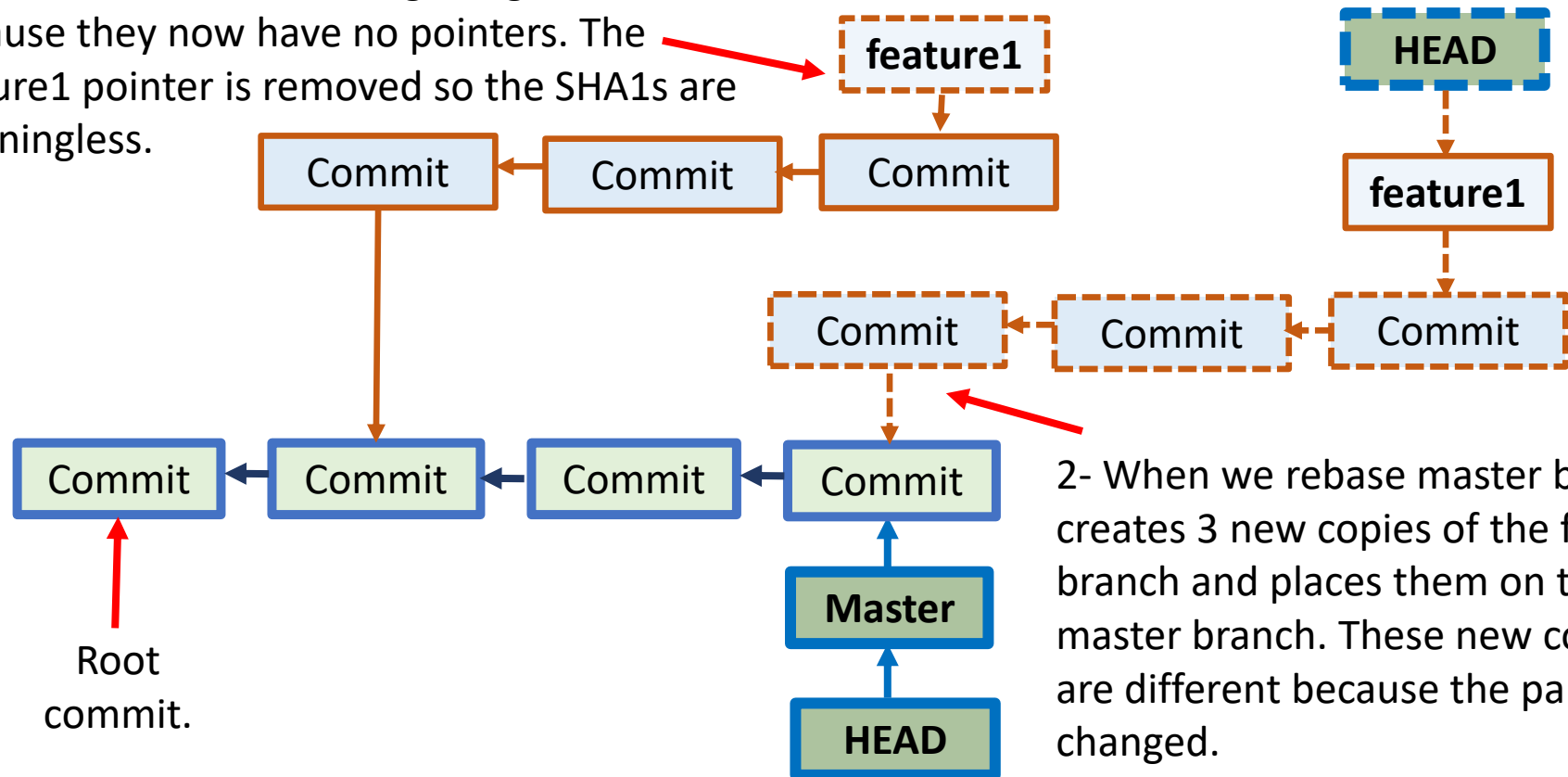
Rebasing Branches Step 1

git checkout feature1

git rebase master

1- Lets suppose that feature 1 has 3 commits stemming from the second commit of the master branch. Notice that there are 2 commits in the base master branch after the creation of the feature 1 branch.

3- The old commits will be garbage collected because they now have no pointers. The feature1 pointer is removed so the SHA1s are meaningless.



2- When we rebase master branch git creates 3 new copies of the feature 1 branch and places them on top of the master branch. These new commits are different because the parent has changed.

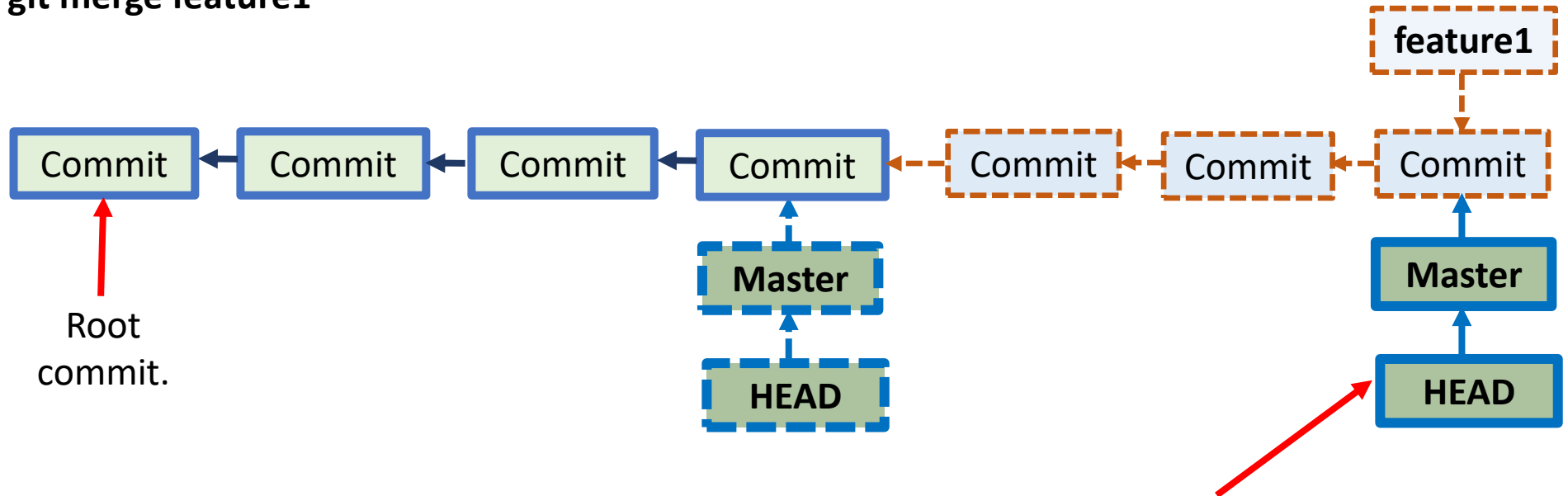
TIME

Rebasing Branches Step 2

git checkout master

git merge feature1

1- we checkout master branch so the pointer of master and HEAD are pointing to the last commit in the master branch.



2- Fast forward merge is used to move the Master and Head pointer to the last commit of the feature 1 branch and now feature 1 branch can be safely removed.

Now we see that rebasing a branch makes the commits linear and removes branch history. None of the commits have more than one parent.

TIME

In the habitual folder for git labs

We can clone the new repository created in github.com

In this new repository called rebasing we can create a simple file

And stage the new file

Add in repository credentials of username and email

Now we can commit the first file

And we see the commit in the log.

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing

```
e1lio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/rebasing.git
Cloning into 'rebasing'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd rebasing/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ echo "First file" > first-file.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git add first-file.txt
warning: LF will be replaced by CRLF in first-file.txt.
The file will have its original line endings in your working directory

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git config user.name EL-EL-BCN

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git config user.email elliotbcbn@gmail.com

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git commit -m "First file in master branch was added"
[main 42be4f0] First file in master branch was added
1 file changed, 1 insertion(+)
create mode 100644 first-file.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git log
commit 42be4f08eb8837dc6126d5ea953b74fef5778009 (HEAD -> main)
Author: EL-EL-BCN <elliotbcbn@gmail.com>
Date: Tue Mar 8 21:49:15 2022 +0100

    First file in master branch was added

commit b1a0e6586e5e0c6738089fa6927d1c2e2ec75deb (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 21:45:18 2022 +0100

    Initial commit

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$
```

Feature1 branch was created
and checked out

Third file was created in
feature 1 branch and staged
then committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git branch feature1
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git checkout feature1
Switched to branch 'feature1'
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ echo "Third file" > Third-file.txt
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git add .
warning: LF will be replaced by CRLF in Third-file.txt.
The file will have its original line endings in your working directory
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git commit -m "Third file in feature1 branch was added"
[feature1 19c433a] Third file in feature1 branch was added
1 file changed, 1 insertion(+)
create mode 100644 Third-file.txt
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$
```

Third file in feature1 branch was edited in nano editor and saved.

These changes were staged and committed.

Now in git log we have two commits in the feature1 branch and two commits in the main branch.

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ nano third-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git add .
warning: LF will be replaced by CRLF in Third-file.txt.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git commit -m "Third file was modified in feature1 branch"
[feature1 512430e] Third file was modified in feature1 branch
1 file changed, 2 insertions(+)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git log
commit 512430e6255b6f4965cd4869840fb6dcb61cf7b6 (HEAD -> feature1)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 22:04:53 2022 +0100

    Third file was modified in feature1 branch

commit 19c433a866d4b7f0f1dd66f529db44662b5268eb
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 21:59:24 2022 +0100

    Third file in feature1 branch was added

commit 5736979a14f1425df5e4d6d1b761ce580cf79cc2 (main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 21:57:46 2022 +0100

    Second file in master branch was added

commit 42be4f08eb8837dc6126d5ea953b74fef5778009
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 21:49:15 2022 +0100

    First file in master branch was added

commit b1a0e6586e5e0c6738089fa6927d1c2e2ec75deb (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Tue Mar 8 21:45:18 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$
```

The main branch is checked out and a fourth file created, staged and committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

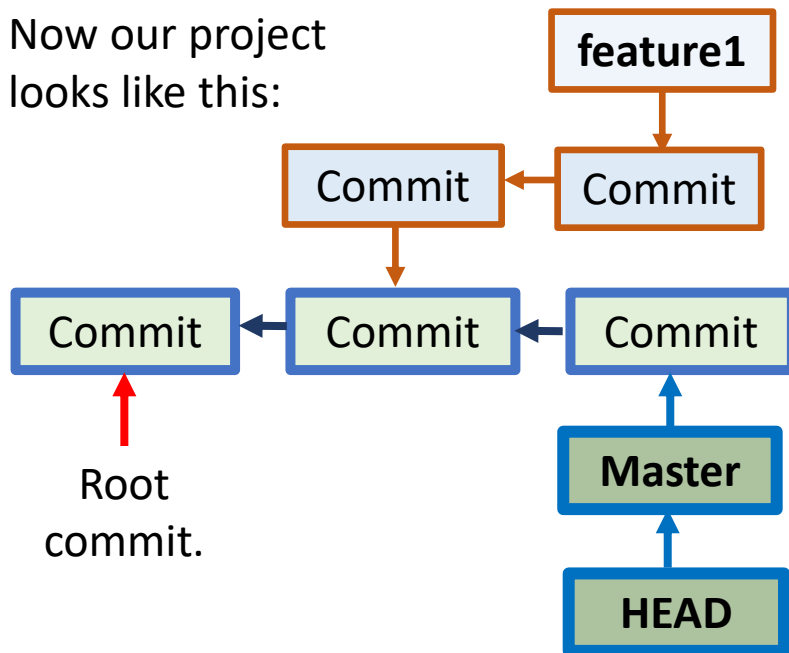
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ echo "Fourth file" > fourth-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git add .
warning: LF will be replaced by CRLF in fourth-file.txt.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git commit -m "Fourth file was added in master branch"
[main f59ddce] Fourth file was added in master branch
1 file changed, 1 insertion(+)
create mode 100644 fourth-file.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ |
```

Now our project looks like this:

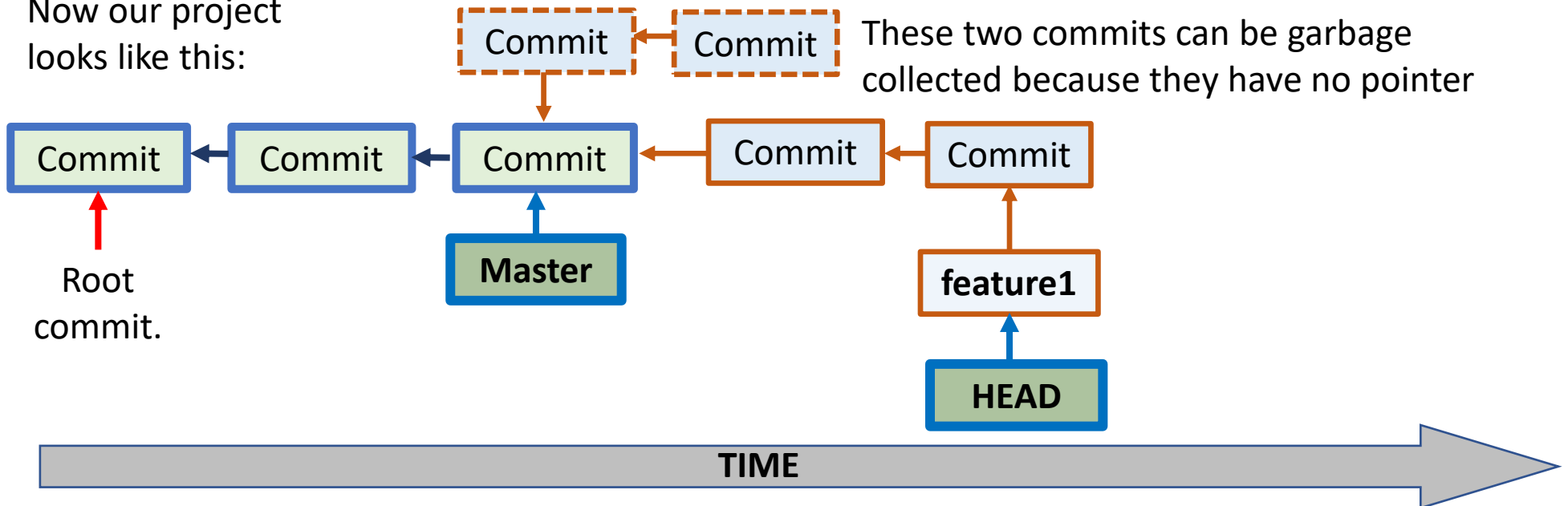


First we checkout feature1 branch

Then rebase main

Note that after the rebase the commits made in feature1 branch appear after the last commit in the main branch.

Now our project looks like this:



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git checkout feature1
Switched to branch 'feature1'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git rebase main
Successfully rebased and updated refs/heads/feature1.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git log
commit 0804d1a6d6bde61b0f06b76661bf3418bc593faa (HEAD -> feature1)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 22:04:53 2022 +0100

    Third file was modified in feature1 branch

commit e746cdeb2dc1acd03876aada7050999ec90e3057
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 21:59:24 2022 +0100

    Third file in feature1 branch was added

commit f59ddce09cb1cbc7d4e05ff4b45e64dc50194c3c (main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 21:59:24 2022 +0100

    Initial commit
```

Now we checkout main branch

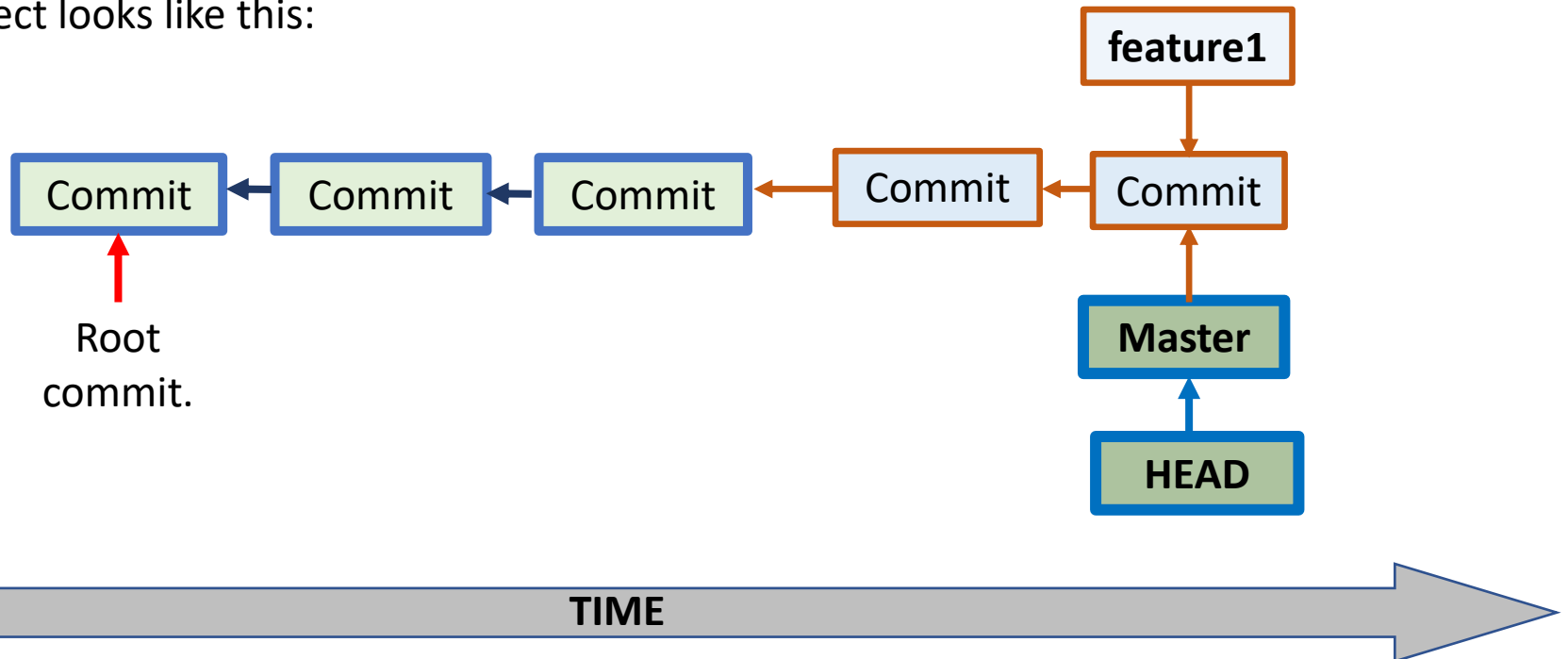
And perform a fast forward merge

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git merge feature1 -v
Updating f59ddce..0804d1a
Fast-forward
 Third-file.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 Third-file.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ |
```

Now our project looks like this:



Now in got log we see that the commits are linear and all under the main branch with the HEAD pointer.

Note that by using the **git log --graph** option the log is displayed with a red line showing time starting with most recent at the top.

Every commit has a single parent.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git log --graph
* commit 0804d1a6d6bde61b0f06b76661bf3418bc593faa (HEAD -> main, feature1)
  Author: EL-EL-BCN <elliottbcn@gmail.com>
  Date: Tue Mar 8 22:04:53 2022 +0100

    Third file was modified in feature1 branch

* commit e746cdeb2dc1acd03876aada7050999ec90e3057
  Author: EL-EL-BCN <elliottbcn@gmail.com>
  Date: Tue Mar 8 21:59:24 2022 +0100

    Third file in feature1 branch was added

* commit f59ddce09cb1cbc7d4e05ff4b45e64dc50194c3c
  Author: EL-EL-BCN <elliottbcn@gmail.com>
  Date: Tue Mar 8 22:10:19 2022 +0100

    Fourth file was added in master branch

* commit 5736979a14f1425df5e4d6d1b761ce580cf79cc2
  Author: EL-EL-BCN <elliottbcn@gmail.com>
  Date: Tue Mar 8 21:57:46 2022 +0100

    Second file in master branch was added

* commit 42be4f08eb8837dc6126d5ea953b74fef5778009
  Author: EL-EL-BCN <elliottbcn@gmail.com>
  Date: Tue Mar 8 21:49:15 2022 +0100

    First file in master branch was added

* commit b1a0e6586e5e0c6738089fa6927d1c2e2ec75deb (origin/main, origin/HEAD)
  Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
  Date: Tue Mar 8 21:45:18 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$
```

rebaseing

Commit Pull Push Fetch Branch Merge Stash Discard Tag

Git-flow Remote Terminal Explorer Settings

WORKSPACE

File Status

History

Search

BRANCHES

feature1

Current Branch ☐ Show Remote Branches Date Order

Graph

	Description	Date	Author	Commit
○	Third file was modified in feature1 branch	8 Mar 2022 22:38	EL-EL-BCN <elliott	0804d1a
●	Third file in feature1 branch was added	8 Mar 2022 22:38	EL-EL-BCN <elliott	e746cde
●	Fourth file was added in master branch	8 Mar 2022 22:10	EL-EL-BCN <elliott	f59ddce
●	Second file in master branch was added	8 Mar 2022 21:57	EL-EL-BCN <elliott	5736979
●	First file in master branch was added	8 Mar 2022 21:49	EL-EL-BCN <elliott	42be4f0
●	Initial commit	8 Mar 2022 21:45	EL-EL-BCN <10101	b1a0e65

origin/main origin/HEAD

rebaseing

Commit Pull Push Fetch Branch Merge Stash Discard Tag

Git-flow Remote Terminal Explorer Settings

WORKSPACE

File Status

History

Search

BRANCHES

feature1

main 5t

Tags

Remotes

Stashes

All Branches ☐ Show Remote Branches Date Order

Graph

	Description	Date	Author	Commit
○	Third file was modified in feature1 branch	8 Mar 2022 22:38	EL-EL-BCN <elliott	0804d1a
●	Third file in feature1 branch was added	8 Mar 2022 22:38	EL-EL-BCN <elliott	e746cde
●	Fourth file was added in master branch	8 Mar 2022 22:10	EL-EL-BCN <elliott	f59ddce
●	Second file in master branch was added	8 Mar 2022 21:57	EL-EL-BCN <elliott	5736979
●	First file in master branch was added	8 Mar 2022 21:49	EL-EL-BCN <elliott	42be4f0
●	Initial commit	8 Mar 2022 21:45	EL-EL-BCN <10101	b1a0e65

origin/main origin/HEAD

Both the main and feature 1 branch now show commits in linear order and it is impossible to tell which commit was made in which branch originally.

Now from the main branch the feature1 branch can be deleted and the local repository pushed to the remote repository.

Git log shows that the main branch and head point to the same commit in the local repository and remote repository meaning that they are synchronised.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git branch -d feature1
Deleted branch feature1 (was 0804d1a).

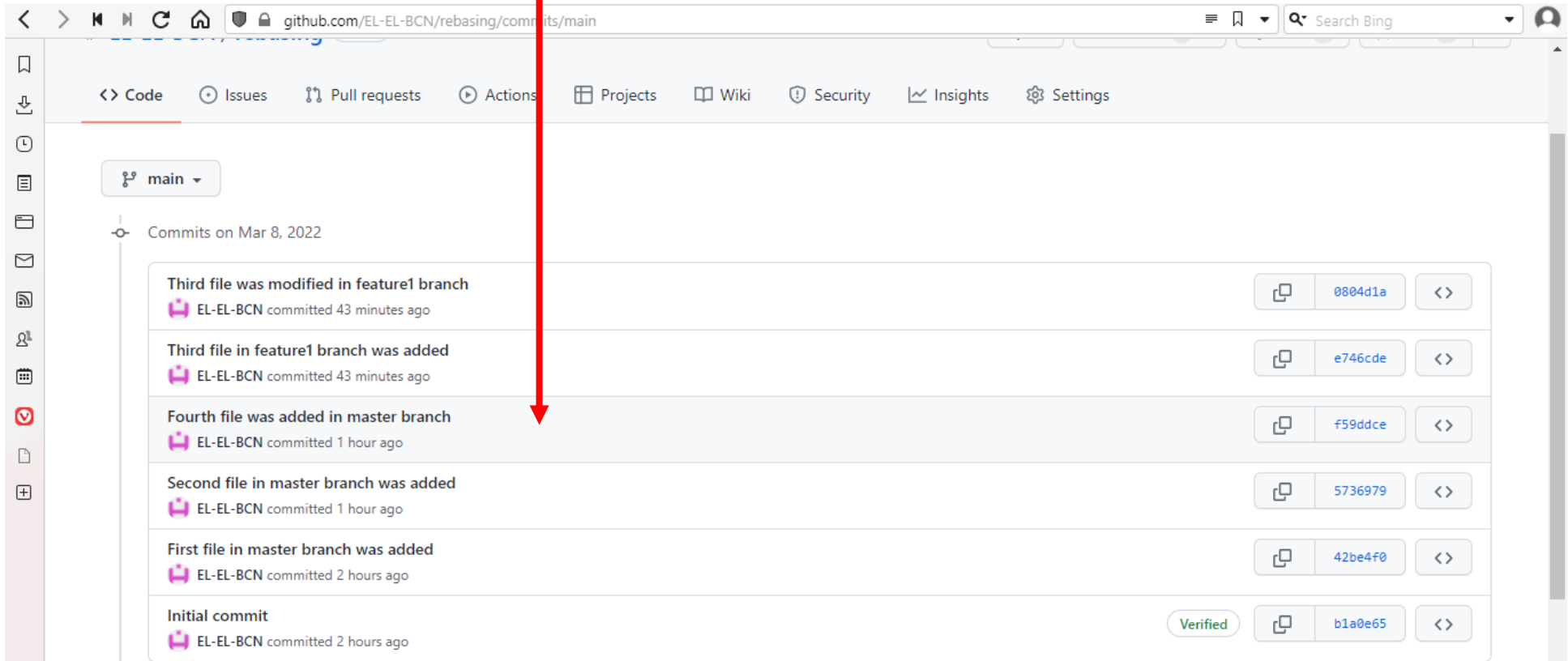
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git push -v
Pushing to https://github.com/EL-EL-BCN/rebasing.git
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 1.30 KiB | 1.30 MiB/s, done.
Total 15 (delta 4), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (1511 bytes)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/EL-EL-BCN/rebasing.git
    b1a0e65..0804d1a  main -> main
updating local tracking ref 'refs/remotes/origin/main'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing (main)
$ git log
commit 0804d1a6d6bde61b0f06b76661bf3418bc593faa (HEAD -> main, origin/main, origin/HEAD)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Tue Mar 8 22:04:53 2022 +0100

    Third file was modified in feature1 branch

commit e746cdeb2dc1acd03876aada7050999ec90e3057
```

Note how GitHub also reflects the out of sequence files after the rebasing.



The screenshot shows the GitHub interface for the repository `github.com/EL-EL-BCN/rebasing/commits/main`. The page displays the commit history for the `main` branch. A red arrow points to the commit titled "Fourth file was added in master branch", which is highlighted in light blue. The commit history shows the following sequence of commits:

- Initial commit (Verified) by EL-EL-BCN, committed 2 hours ago. Commit hash: `b1a0e65`.
- First file in master branch was added by EL-EL-BCN, committed 2 hours ago. Commit hash: `42be4f0`.
- Second file in master branch was added by EL-EL-BCN, committed 1 hour ago. Commit hash: `5736979`.
- Fourth file was added in master branch (highlighted) by EL-EL-BCN, committed 1 hour ago. Commit hash: `f59ddce`.
- Third file in feature1 branch was added by EL-EL-BCN, committed 43 minutes ago. Commit hash: `e746cde`.
- Third file was modified in feature1 branch by EL-EL-BCN, committed 43 minutes ago. Commit hash: `0804d1a`.

Rebasing is a destructive process as branch history is deleted and unrecoverable.

Rebasing creates new commits with current timestamps when rebasing is performed so order of changes is also lost as the rebasing reorders the commits in a linear fashion.

Merging is a much better solution to rebasing.

Summary of Shell Commands

Git rebase <branch>

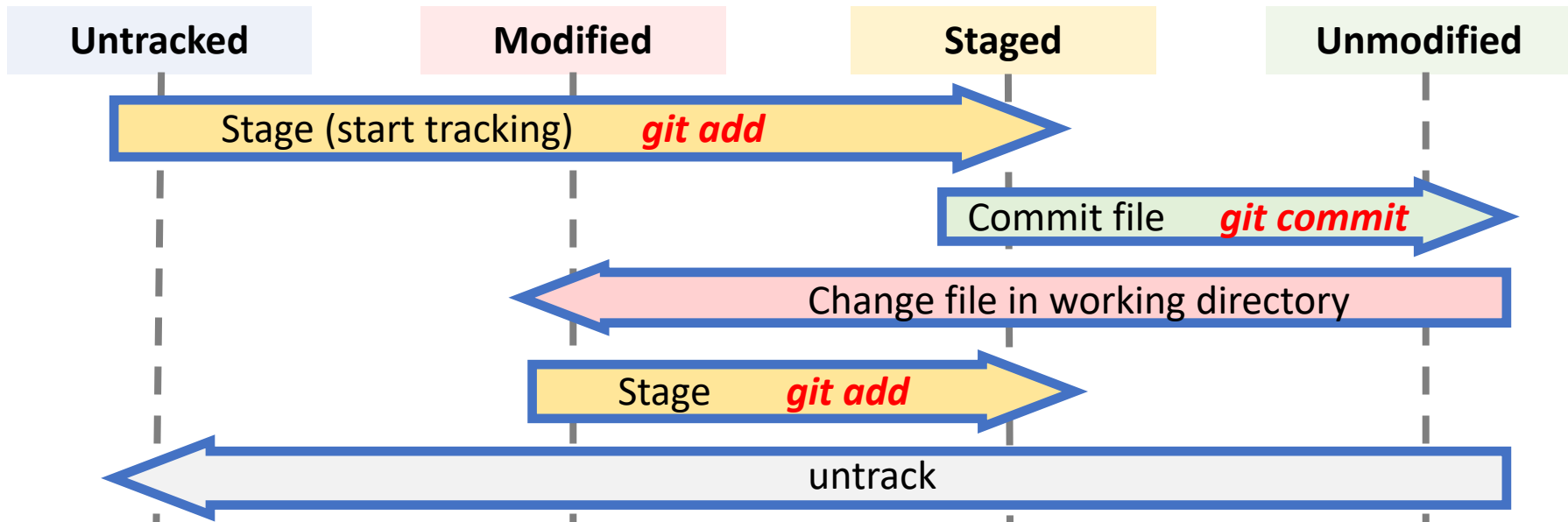
Rebase branch onto another branch

IGNORING FILES IN GIT

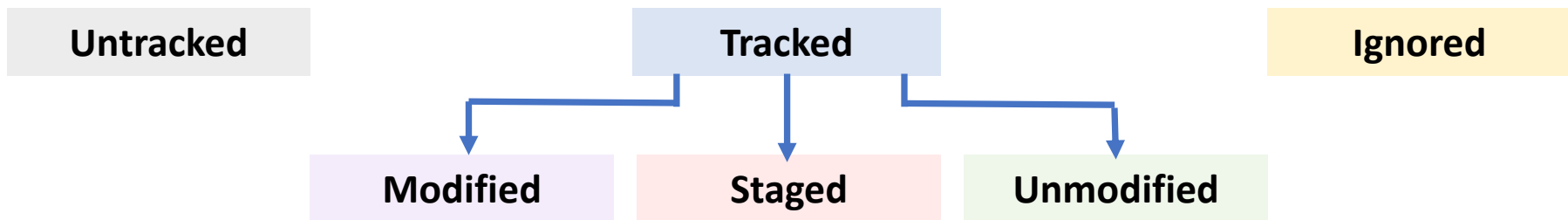
Git Ignore

- Git Ignore tells git which files and folders to ignore
- Change in Ignored files and folders are ignored by git
- Rules are defined in a separate file **.gitignore**
- **.gitignore** file itself must be submitted
- By default there are not **.gitignore** files or folders in git, they have to be manually added to the **.gitignore** file

The below diagram shows git file statuses



By default any file created in git has a status of Untracked. The file can then progress to a tracked sub status of modified, staged and unmodified. Unless it has file status of Ignored.



In the habitual directory a new repository is created called git ignore.

A new file called **.gitignore** is created in the root of the directory.

The status of this file is unstaged/Uncommitted.

the **.gitignore** file must be staged and committed.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gitignore
ellio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  hello-world/          rebasing/
JavaScript-Bible-Basics/      my-first-git-hub-repository/  tags/
first-project/                pull-requests/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ mkdir gitignore

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd gitignore/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/gitignore/.git/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ touch .gitignore

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ ls -la
total 8
drwxr-xr-x 1 ellio 197609 0 Mar  9 19:58 ./
drwxr-xr-x 1 ellio 197609 0 Mar  9 19:57 ../
drwxr-xr-x 1 ellio 197609 0 Mar  9 19:58 .git/
-rw-r--r-- 1 ellio 197609 0 Mar  9 19:58 .gitignore

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git status
On branch master

No commits yet

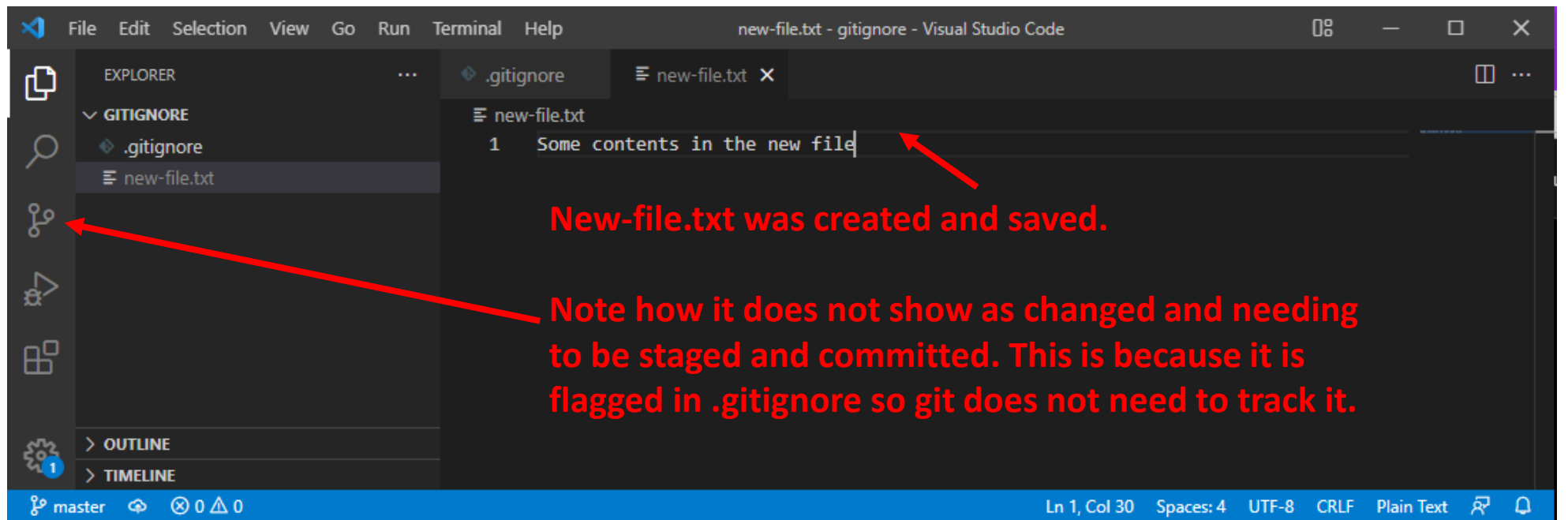
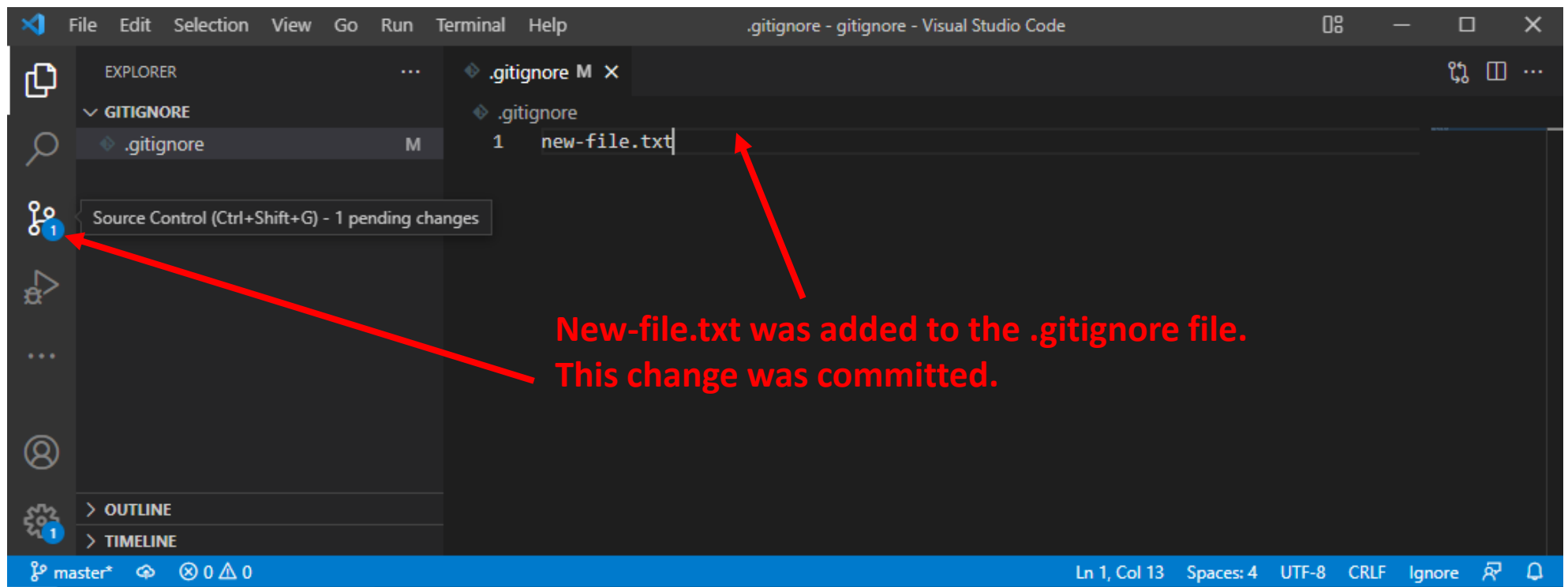
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .gitignore

nothing added to commit but untracked files present (use "git add" to track)

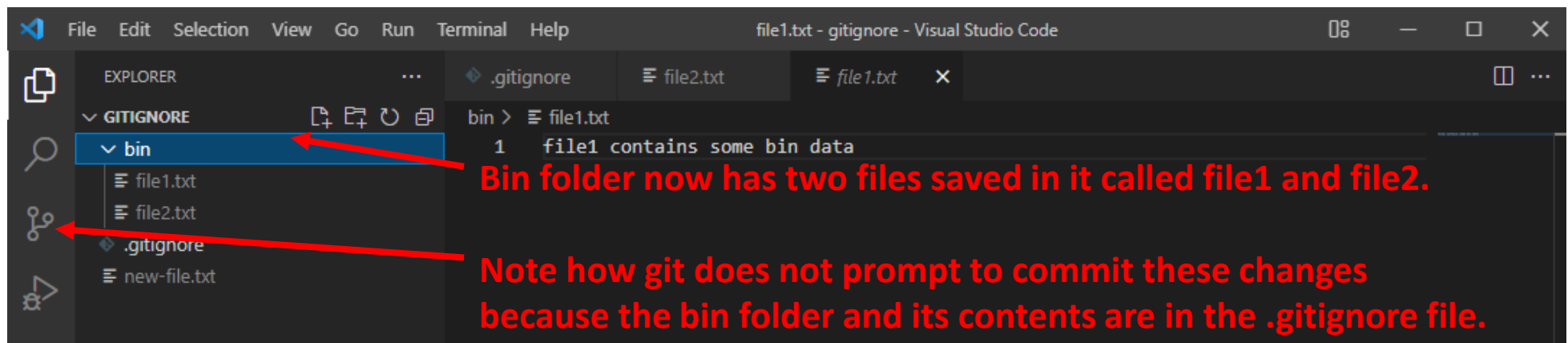
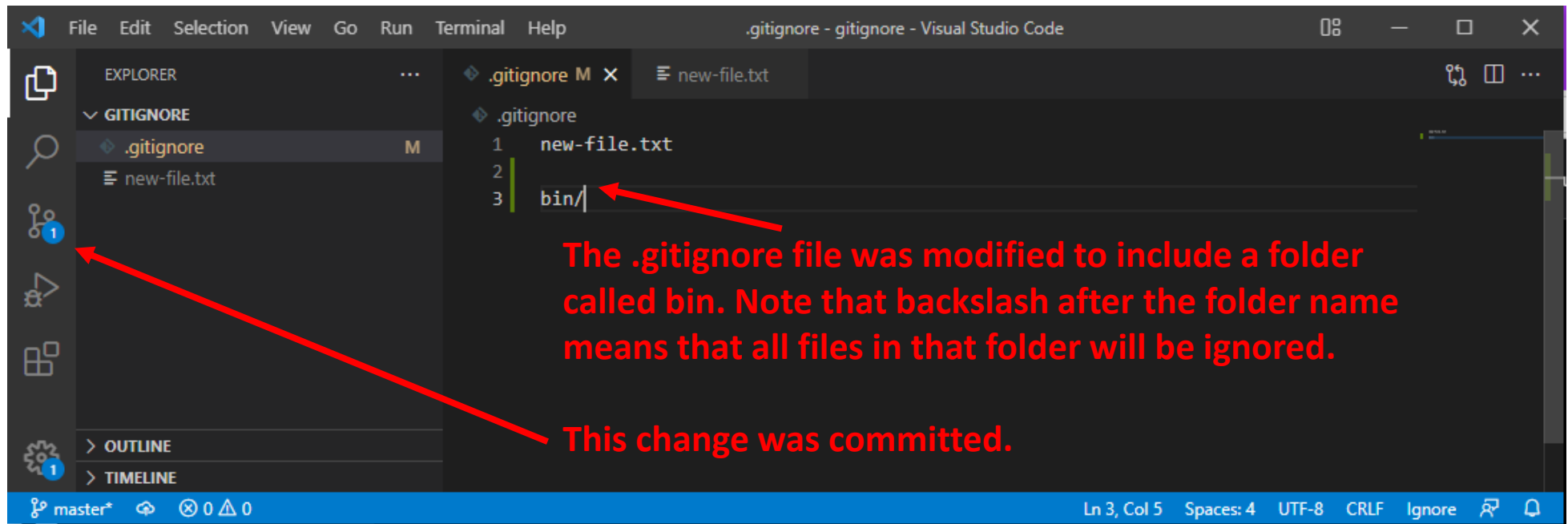
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git add .gitignore

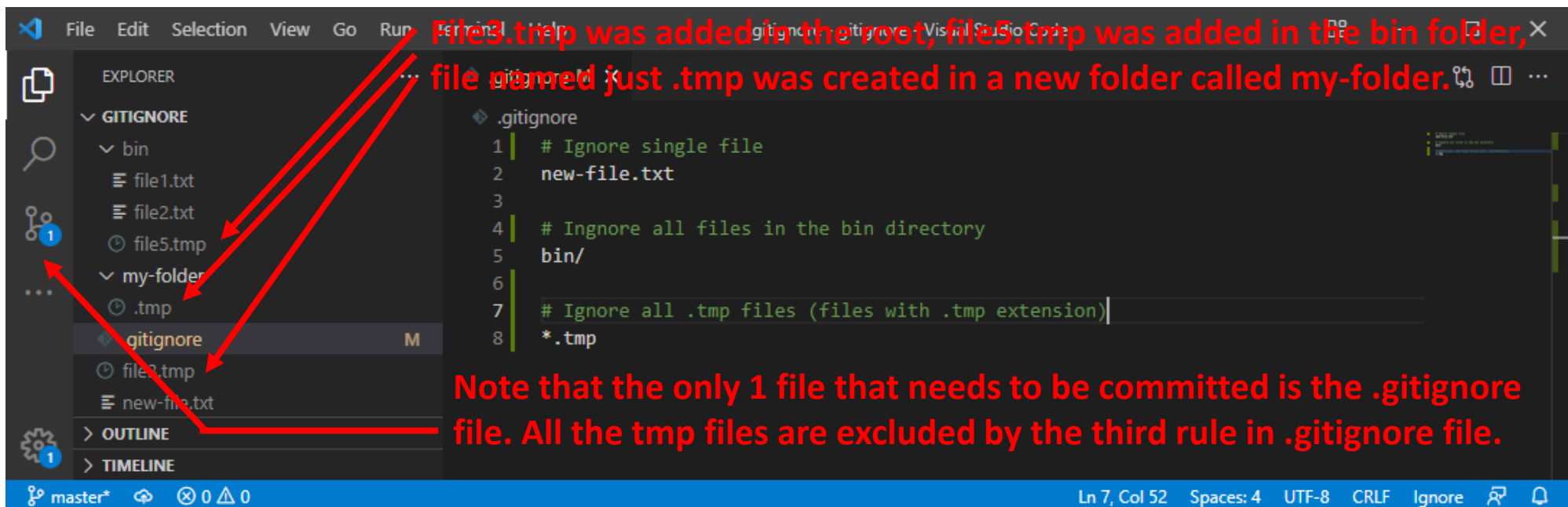
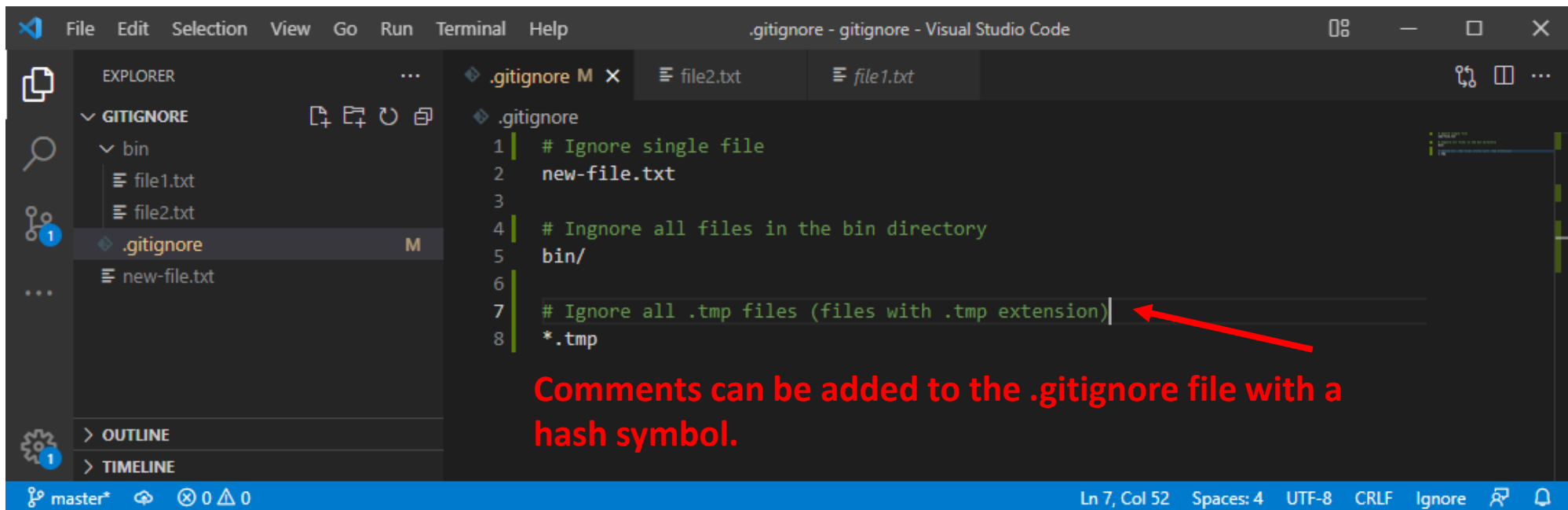
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git commit -m "Create gitignore file"
[master (root-commit) 661f9d9] Create gitignore file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 .gitignore

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$
```



This means that when the repository is pushed to remote repository git will ignore the file and it will not show up in the remote repository. If someone clones or forks this remote repository then the files in .gitignore will not be present.





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

EL-EL-BCN / gitignore ✓

Lets create a remote public repository in github.com

Great repository names are short and memorable. Need inspiration? How about [glowing-waffle?](#)

Description (optional)

lets play with git ignore rules

- ☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Leaving README.txt unchecked.

Create repository

The remote repository is configured in the local repository and local repository is pushed to remote.

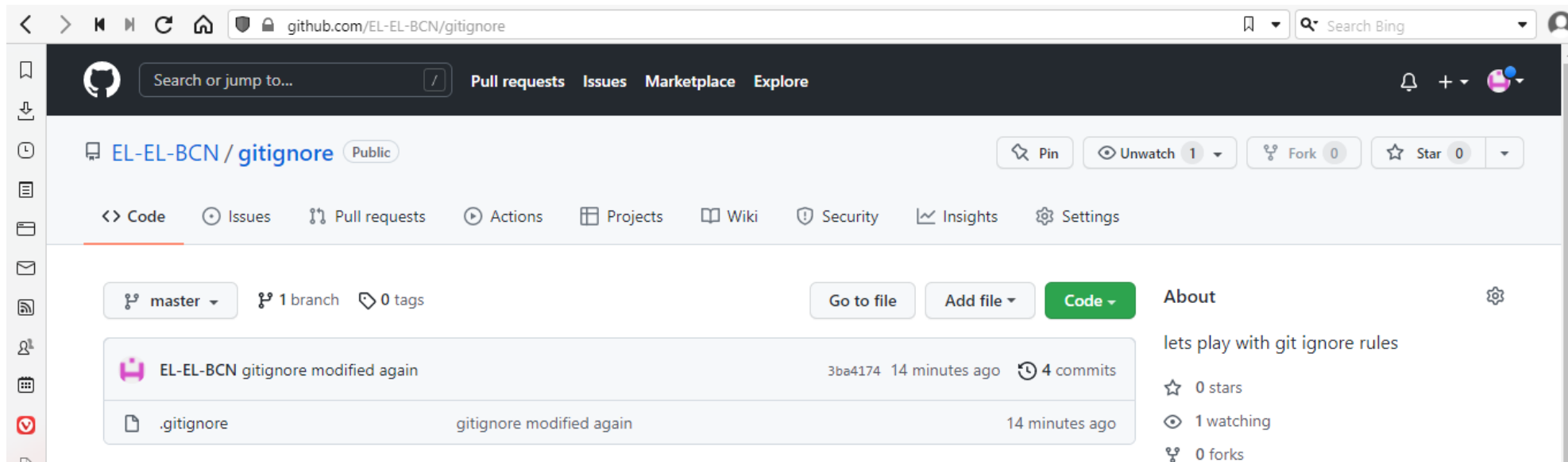
Below in the remote repository we can see that only one file was pushed because the rest are excluded in the .gitignore file

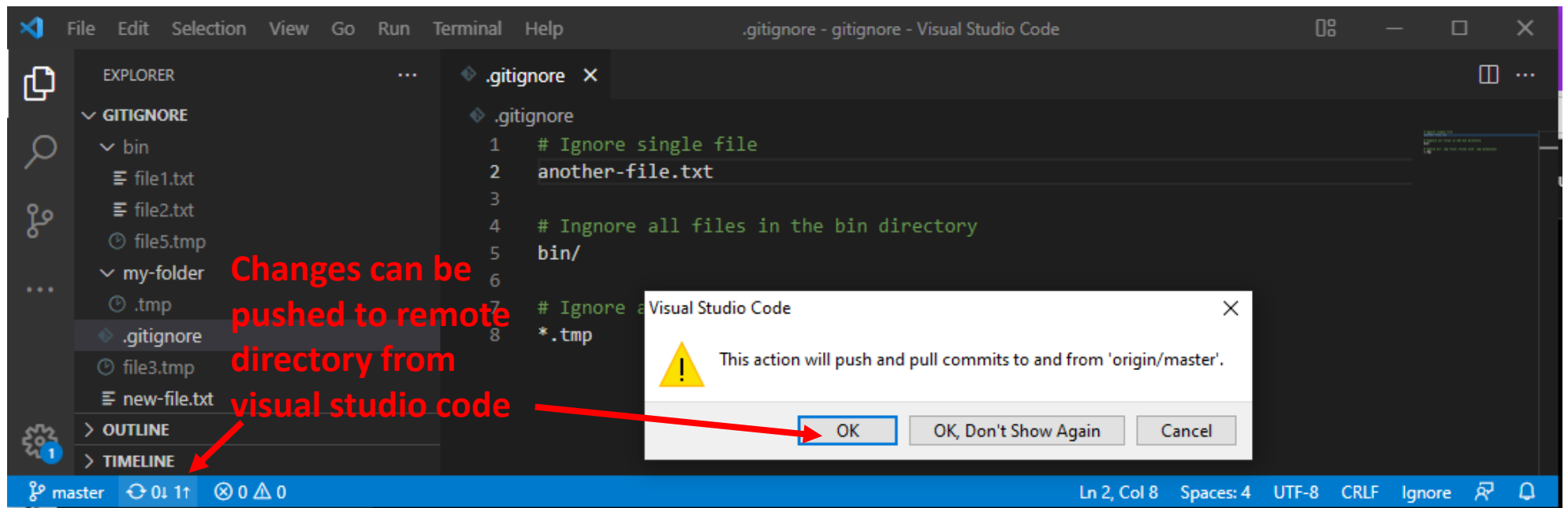
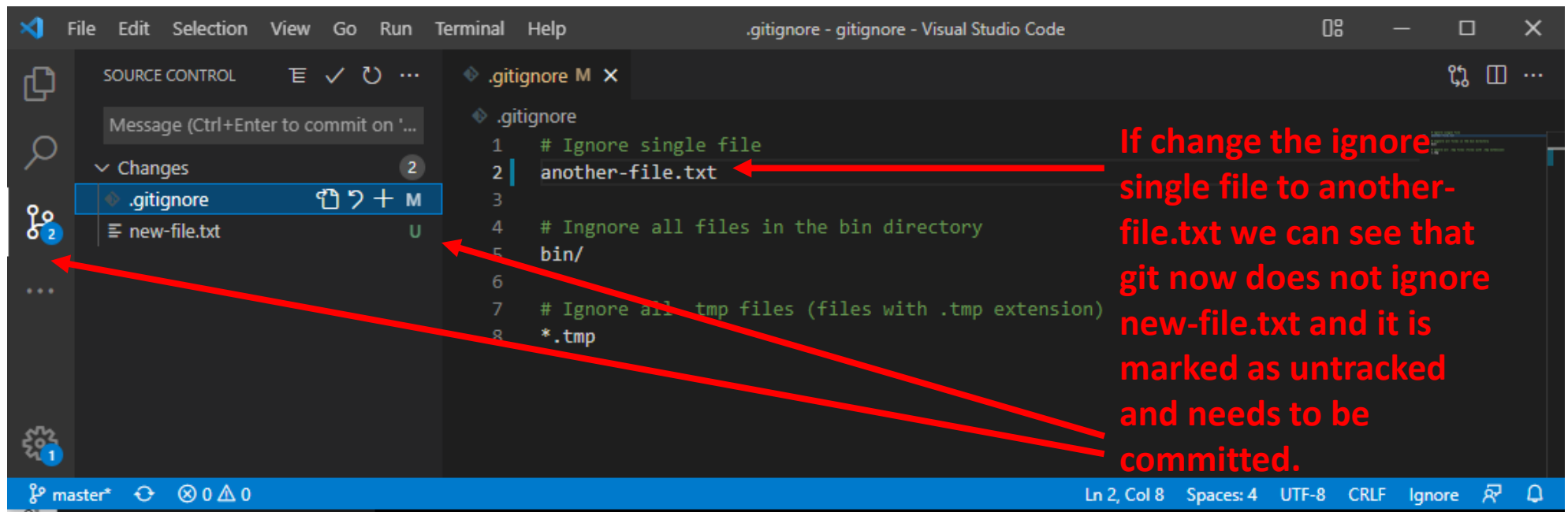
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gitignore
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git remote add origin https://github.com/EL-EL-BCN/gitignore.git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git push -u origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (12/12), 1018 bytes | 1018.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EL-EL-BCN/gitignore.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git remote -vv
origin https://github.com/EL-EL-BCN/gitignore.git (fetch)
origin https://github.com/EL-EL-BCN/gitignore.git (push)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ |
```





github.com/EL-EL-BCN/gitignore

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / gitignore Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

About
lets play with git ignore rules
0 stars
1 watching
0 forks

EL-EL-BCN create new file a230438 2 minutes ago 5 commits

.gitignore	create new file	2 minutes ago
new-file.txt	create new file	2 minutes ago

Now that new-file.txt is removed from the .gitignore file it is tracked and so is committed and pushed to the remote repository.

File Edit Selection View Go Run Terminal Help new-file.txt - gitignore - Visual Studio Code

EXPLORER

- GITIGNORE
 - bin
 - file1.txt
 - file2.txt
 - files.tmp
 - my-folder
 - .tmp
 - .gitignore
 - file3.tmp
 - new-file.txt M
- OUTLINE
- TIMELINE

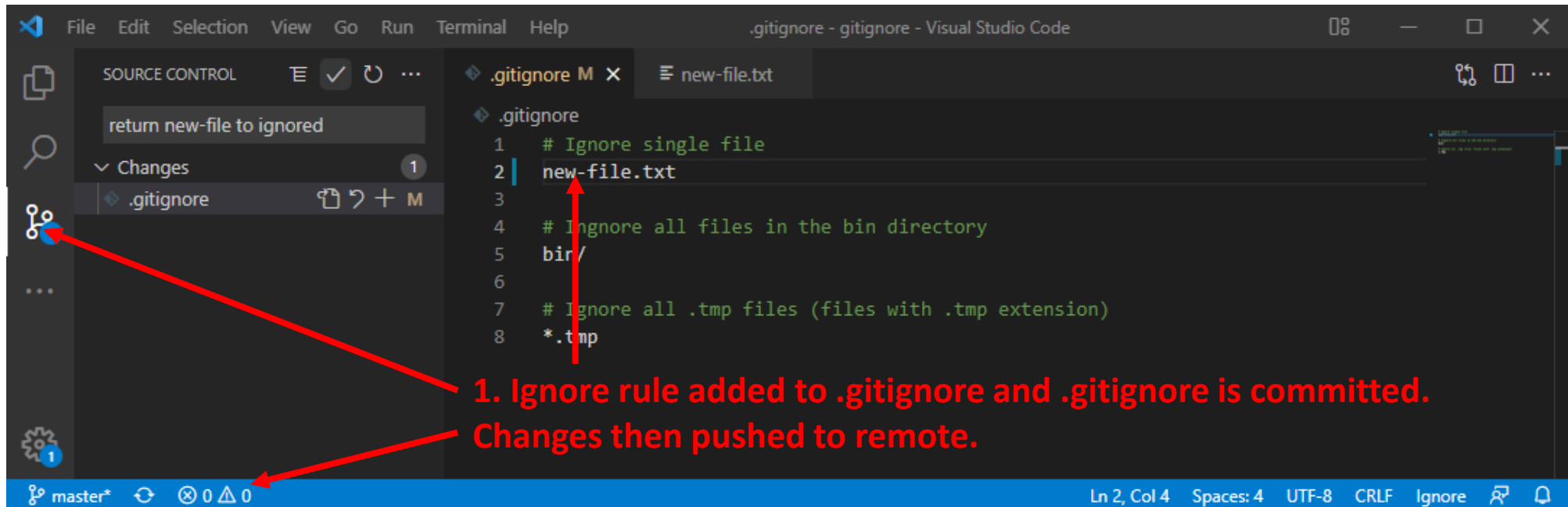
new-file.txt

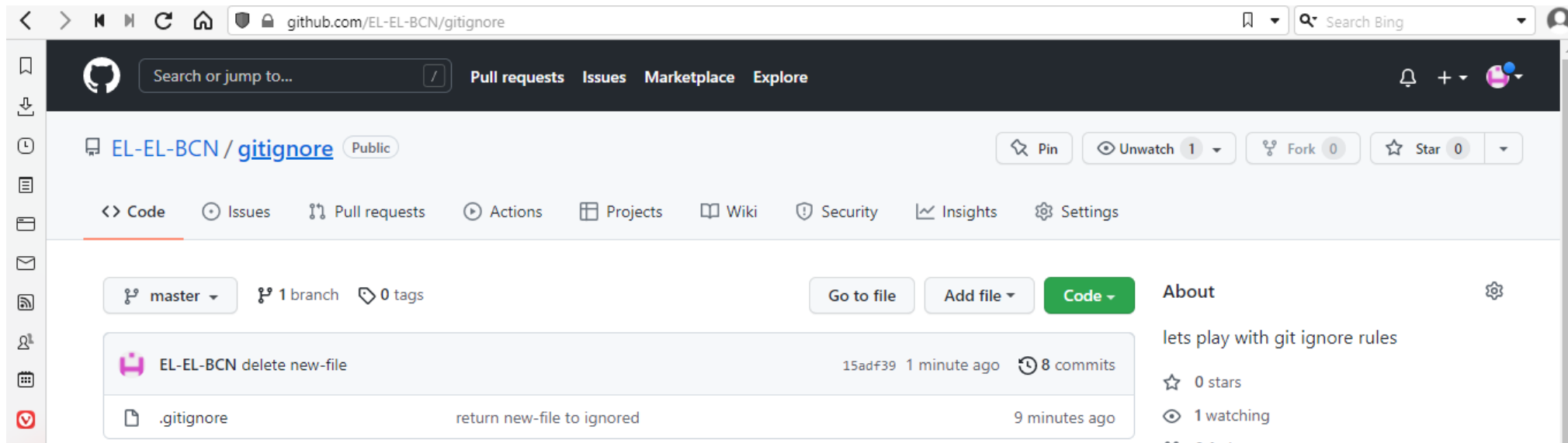
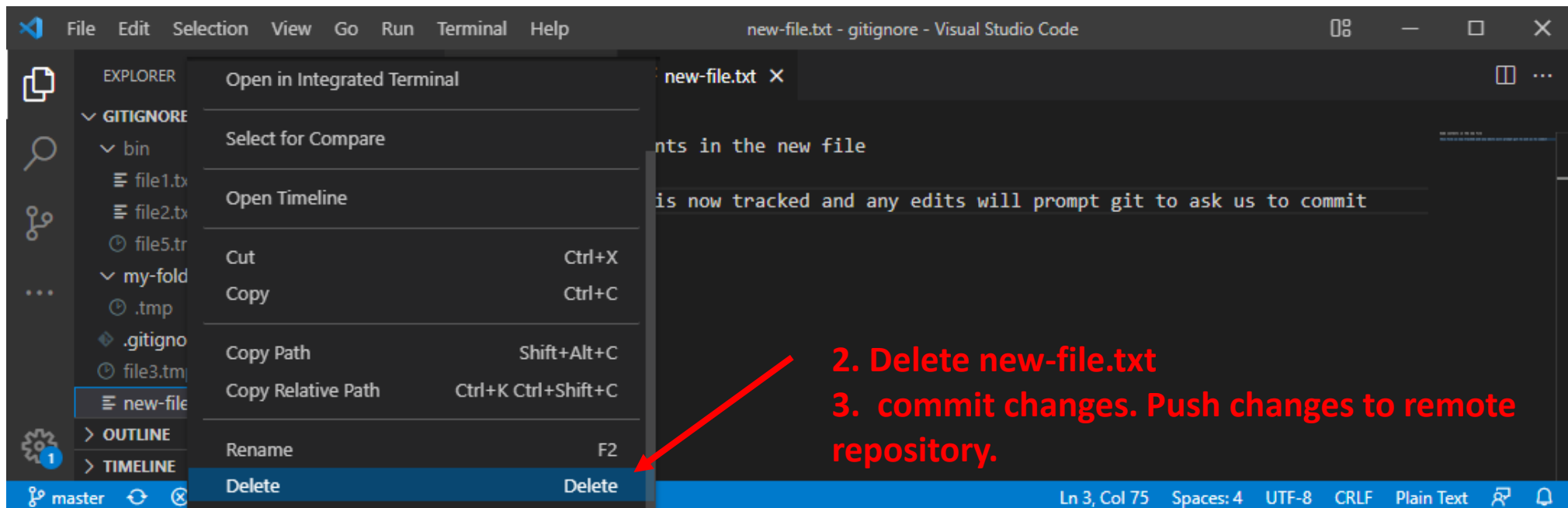
```
1 Some contents in the new file
2
3 this file is now tracked and any edits will prompt git to ask us to commit
```

Any subsequent edits to new-file.txt will prompt git to ask us to commit.

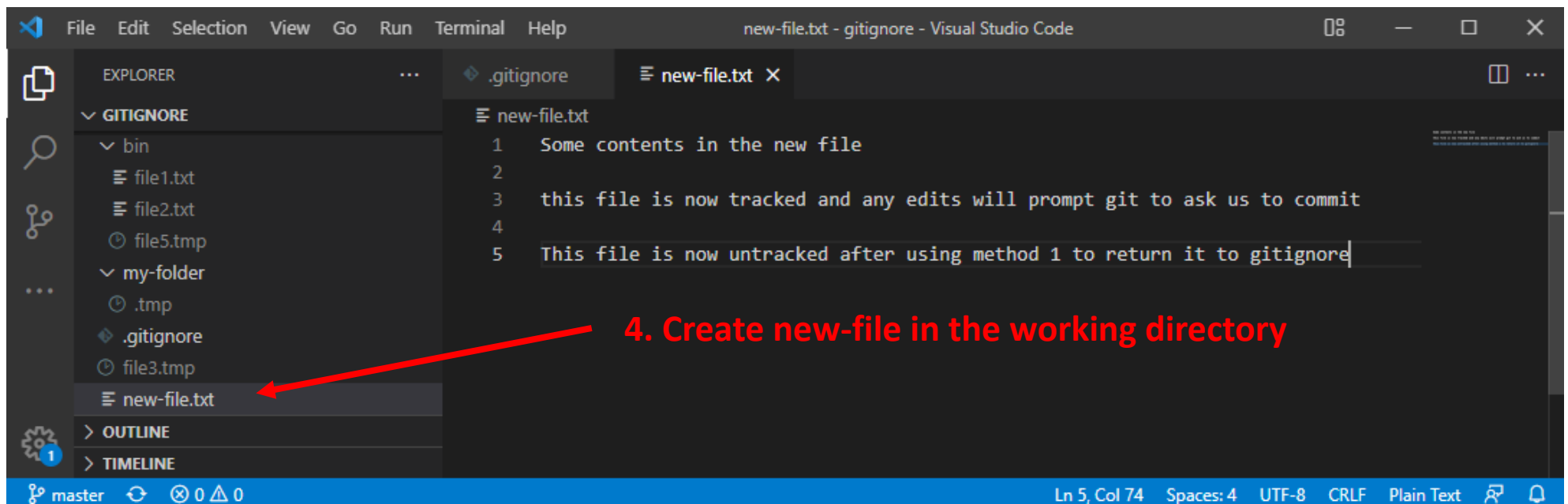
Ignore previously committed file – method 1

1. Add ignore rule to **.gitignore**
2. Delete file in working directory
3. Commit changes
4. Add again previously deleted file

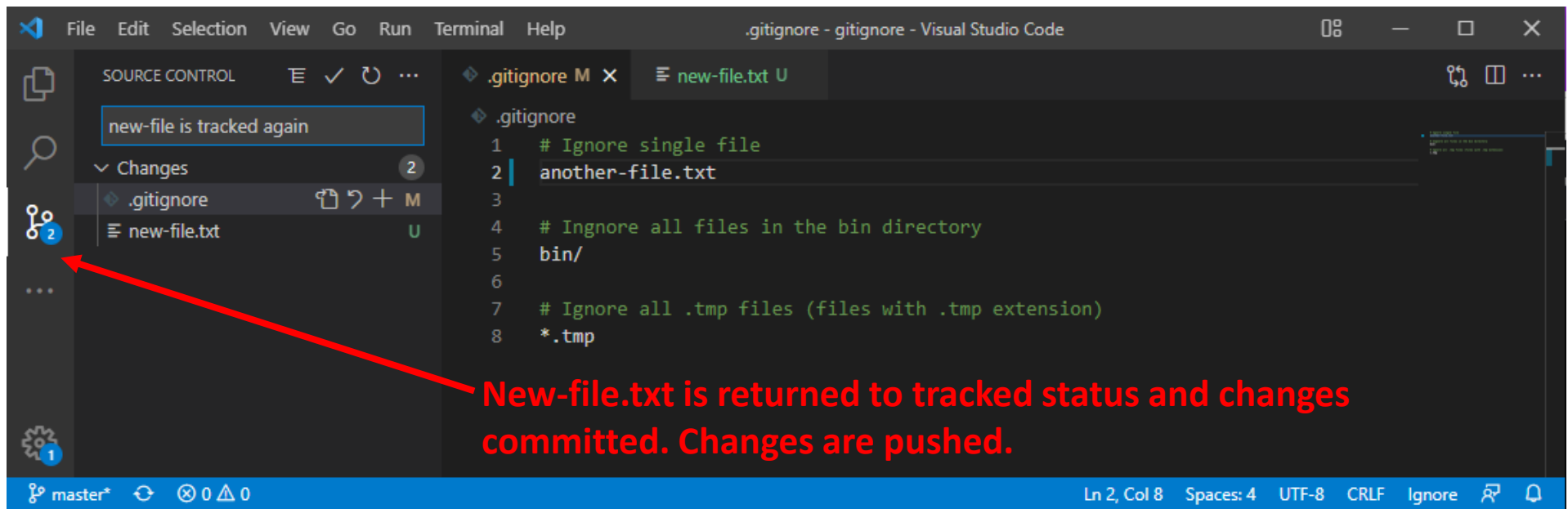




New-file.txt has now gone from remote repository.

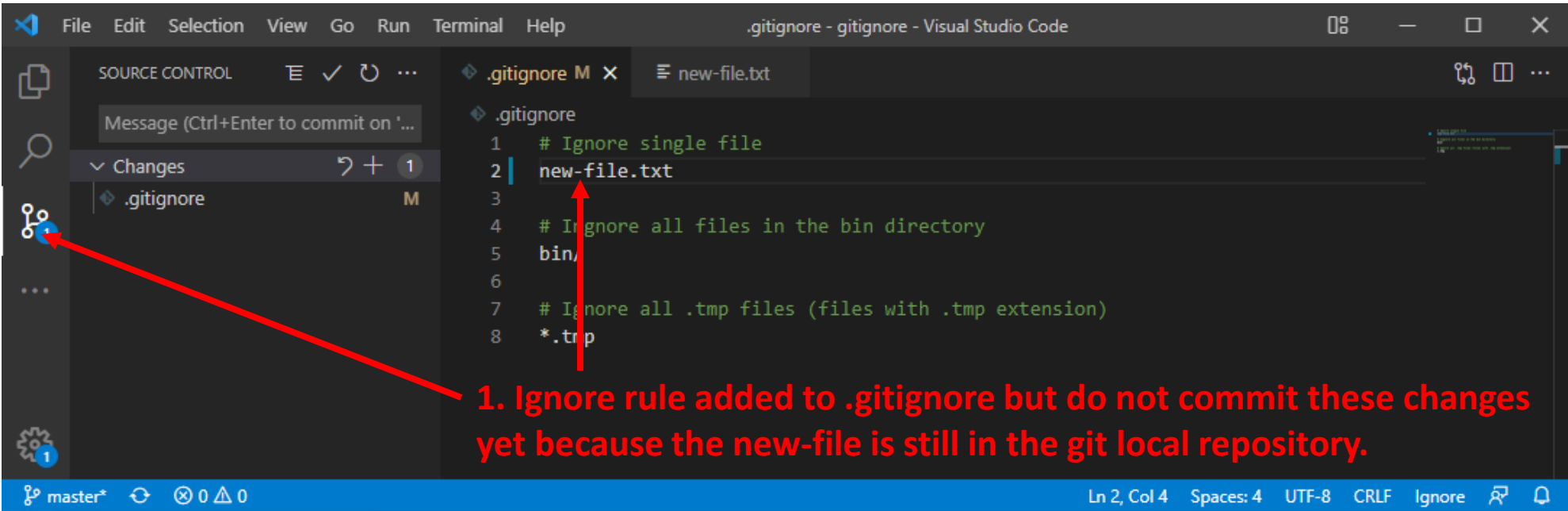


But if you want to keep the file in the working directory it is not convenient to delete it first then commit and recreate the file again in the working directory.



Ignore previously committed file – method 2

1. Add ignore rule to **.gitignore**
2. Delete file only from repository, keeping it in working directory using command **git rm --cached <filename>**



The screenshot shows the Visual Studio Code interface with the `.gitignore` file open. The file contains the following content:

```
1 # Ignore single file
2 new-file.txt
3
4 # Ignore all files in the bin directory
5 bin/
6
7 # Ignore all .tmp files (files with .tmp extension)
8 *.tmp
```

A red arrow points from the Source Control panel on the left to the new rule `new-file.txt` in the `.gitignore` file. The Source Control panel shows a message: "Message (Ctrl+Enter to commit on '...'" and a list of changes with `.gitignore` marked as modified (M).

1. Ignore rule added to `.gitignore` but do not commit these changes yet because the new-file is still in the git local repository.

The new-file.txt can be deleted from the repository with **git rm --cached <filename>** command

Note how new-file.txt is still in the working directory.

git status shows the new-file.txt as deleted and the .gitignore as modified.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gitignore
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git rm --cached new-file.txt
rm 'new-file.txt'

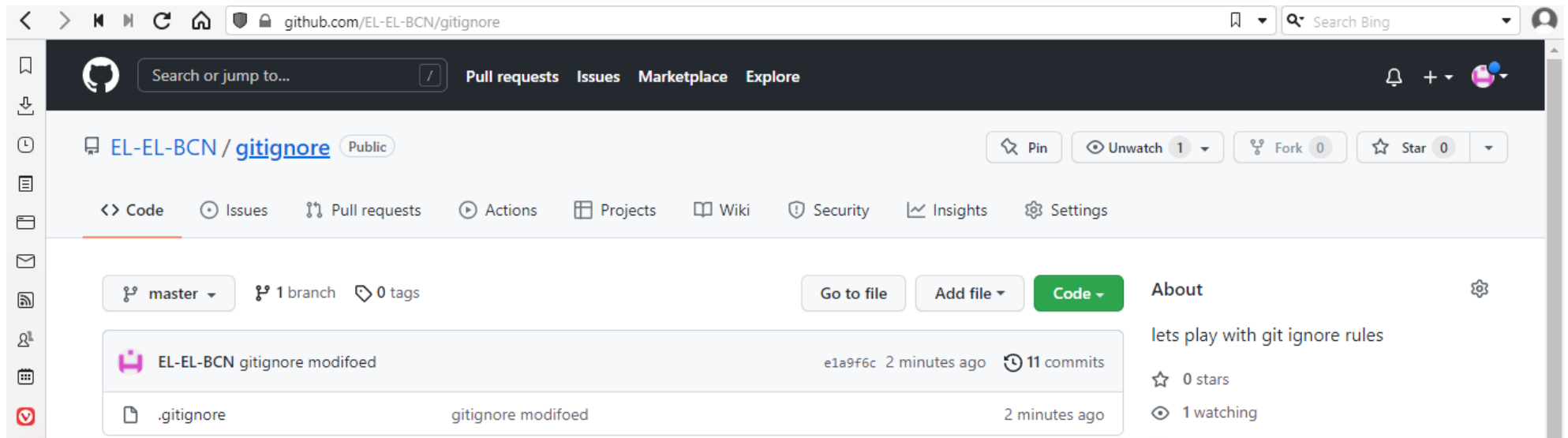
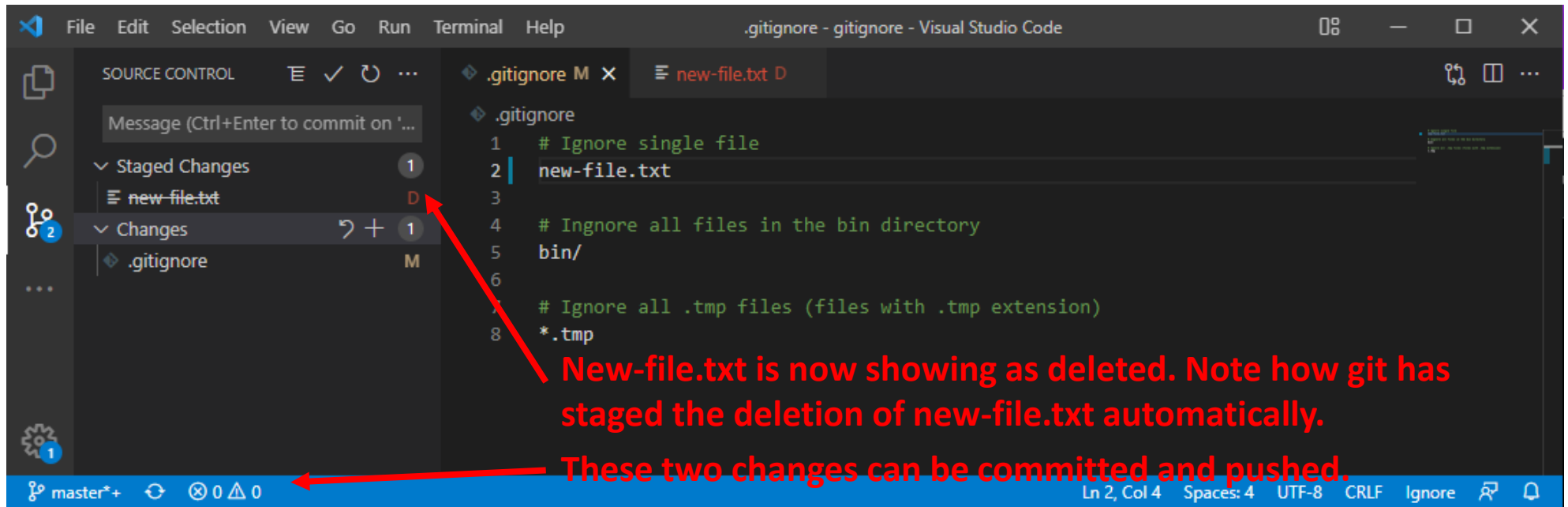
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ ls -la
total 14
drwxr-xr-x 1 ellio 197609  0 Mar  9 21:38 ./
drwxr-xr-x 1 ellio 197609  0 Mar  9 19:57 ../
drwxr-xr-x 1 ellio 197609  0 Mar  9 21:50 .git/
-rw-r--r-- 1 ellio 197609 146 Mar  9 21:48 .gitignore
drwxr-xr-x 1 ellio 197609  0 Mar  9 20:45 bin/
-rw-r--r-- 1 ellio 197609  0 Mar  9 20:44 file3.tmp
drwxr-xr-x 1 ellio 197609  0 Mar  9 20:46 my-folder/
-rw-r--r-- 1 ellio 197609 184 Mar  9 21:39 new-file.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   new-file.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$
```



New-file.txt has now gone from remote repository but is still present in local working directory.

Best Practices for git ignore

- Build folders like bin/

Build folders are usually created in production repositories and will not appear in the final release so these files can be ignored.

- Dependency folders like node_modules/

Dependency folders like node modules or JQuery etc are ignored because these files are usually large and anyone can install these dependency files themselves after cloning or forking the repository.

- Compiled and log files like *.pyc , *.log

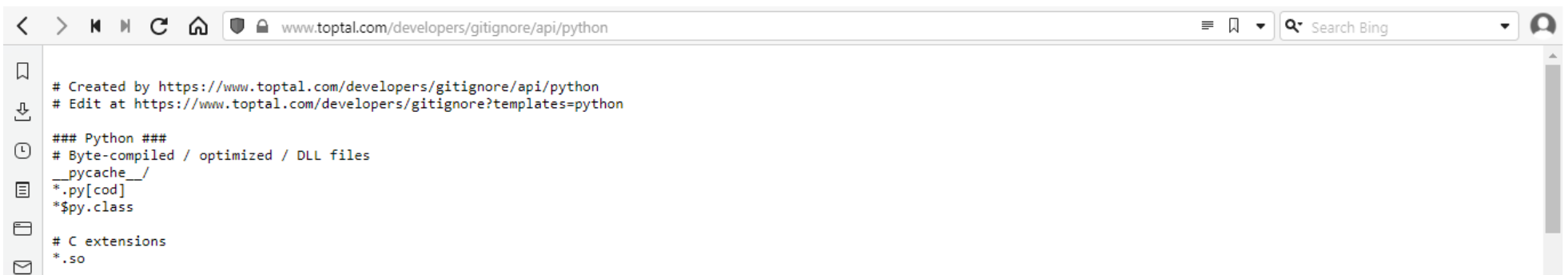
Log files should be ignored as they will be full of junk information that was created during the development stages.

- Hidden OS files like thumbs.db or DS_Store

Do not include such system files that the OS will sometimes create automatically.

Gitignore.io <https://www.toptal.com/developers/gitignore>

Gitignore.io is a website where we can download default templates for a .gitignore file depending on the programming language OS etc. for example we can add python and click create and it will generate a default gitignore template that can be copied into the .gitignore file.



GitHub also has a list of git ignore templates in the repository <https://github.com/github/gitignore>

Search or jump to... Pull requests Issues Marketplace Explore

github / gitignore Public

Watch 3.2k Fork 73k Star 130k

Code Pull requests 219 Actions Security Insights

main 6 branches 0 tags

Go to file Add file Code

About

A collection of useful .gitignore templates

gitignore git

Readme CC0-1.0 License Code of conduct 130k stars 3.2k watching 73k forks

Releases Activate Windows

No releases published

bdougie	Merge pull request #3943 from spdegabrielle/Racket1	ce5da10 22 days ago	3,520 commits
.github	Adds a relationship question	3 months ago	
Global	Update Xcode.gitignore	3 months ago	
community	Merge pull request #3956 from snxx-lppxx/add-toit	27 days ago	
AL.gitignore	Add .gitignore for Microsoft Business Central	4 months ago	
Actionscript.gitignore	Fix comments on same line causing ignore to break	5 years ago	
Ada.gitignore	ensure single trailing newline	8 years ago	
Agda.gitignore	Add MAlonzo directory. (#2978)	3 years ago	
Android.gitignore	Android Studio	last month	

Summary of Shell Commands

touch .gitignore

Create git ignore hidden file in the root directory

git add .

Stage all files

git commit -m "gitignore file created"

Commit git ignore file

git rm --cached <filename>

Remove a file from the repository but not the working area

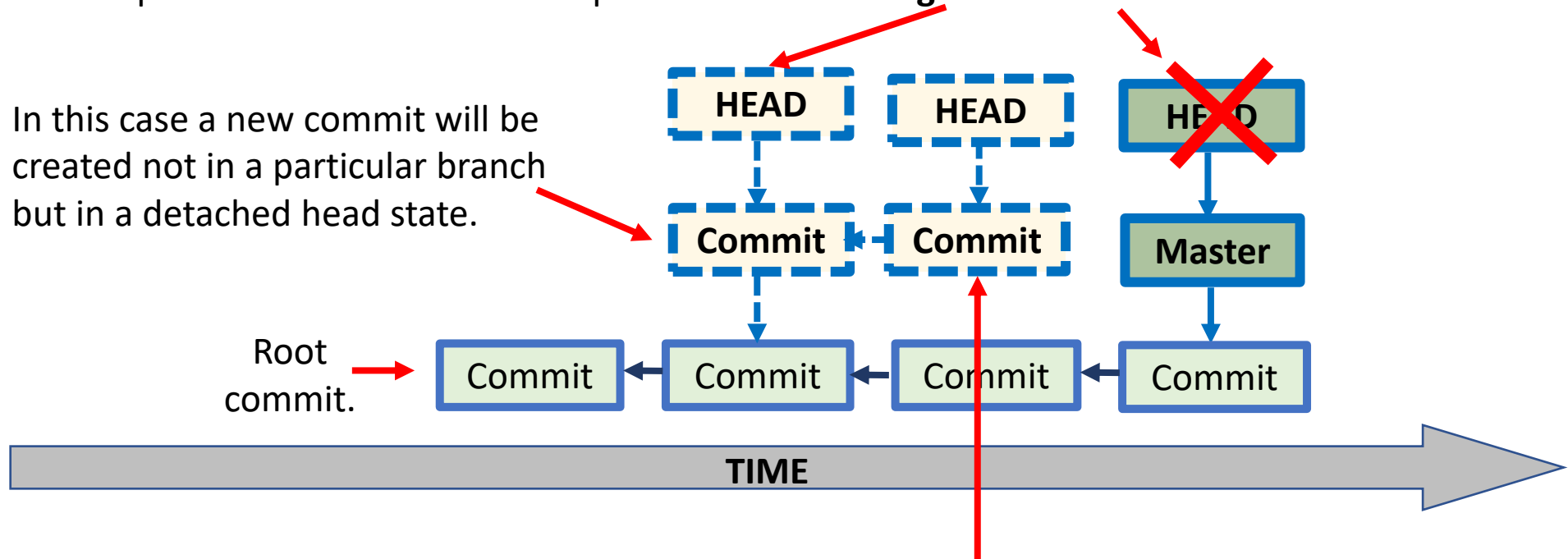
DETACHED HEAD

Detached Head

Normally, HEAD points to a specific branch so any new commit made in for example master branch, the head pointer automatically moves with the branch pointer.

But it is possible to move head to a specific commit with **git checkout <SHA1>** command.

In this case a new commit will be created not in a particular branch but in a detached head state.



Any commits made in detached head state will chain up behind the first detached head commit and the head pointer will move to the latest detached head commit.

If you checkout the master branch again then the head will move back to the master branch and any commits in the detached head state will be garbage collected because they do not have any head or branch pointer. However when head points to detached head commit, if you are happy with these experimental changes then a new branch can be created.

A new repository was created in GitHub and cloned to the habitual directory.

Empty files of file1.txt, file2.txt and file3.txt were created, staged and committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/detached-head
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gitignore (master)
$ cd ..

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  gitignore/          pull-requests/
JavaScript-Bible-Basics/      hello-world/         rebasing/
first-project/                my-first-git-hub-repository/  tags/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/detached-head.git
Cloning into 'detached-head'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), 636 bytes | 636.00 KiB/s, done.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd detached-head/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git config user.name EL-EL-BCN

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git config user.email elliotbnc@gmail.com

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ touch file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git add file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git commit -m "File1 created"
[main b95c8d6] File1 created
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git add file2.txt
fatal: pathspec 'file2.txt' did not match any files

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ touch file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git add file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git commit -m "File2 created"
[main 60a13a3] File2 created
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ touch file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git add file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git commit -m "File3 created"
[main 291674f] File3 created
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.txt
```

In git log we can see the commits and that the head and main branch pointers are on the last commit.

`cat .git/head` shows that the head pointer is in the main branch also.

We can checkout any commit to enter detached head mode with the command `git checkout <SHA1>`. I have checked out the commit for file1.txt

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/detached-head
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git log
commit 291674f9bc040b5626d66f1b43c5e2ce9fba202e (HEAD -> main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:54:09 2022 +0100

    File3 created

commit 60a13a3249623dfec3948ff5a2bf709d3497ee3a
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:53:53 2022 +0100

    File2 created

commit b95c8d6e6566177e3303ff3261a9953b229495a0
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:53:01 2022 +0100

    File1 created

commit 2d9bad0d4d6cd80398effa6837c80ab18e514f59 (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Wed Mar 9 22:49:43 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ cat .git/head
ref: refs/heads/main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git checkout b95c8d6e6
Note: switching to 'b95c8d6e6'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

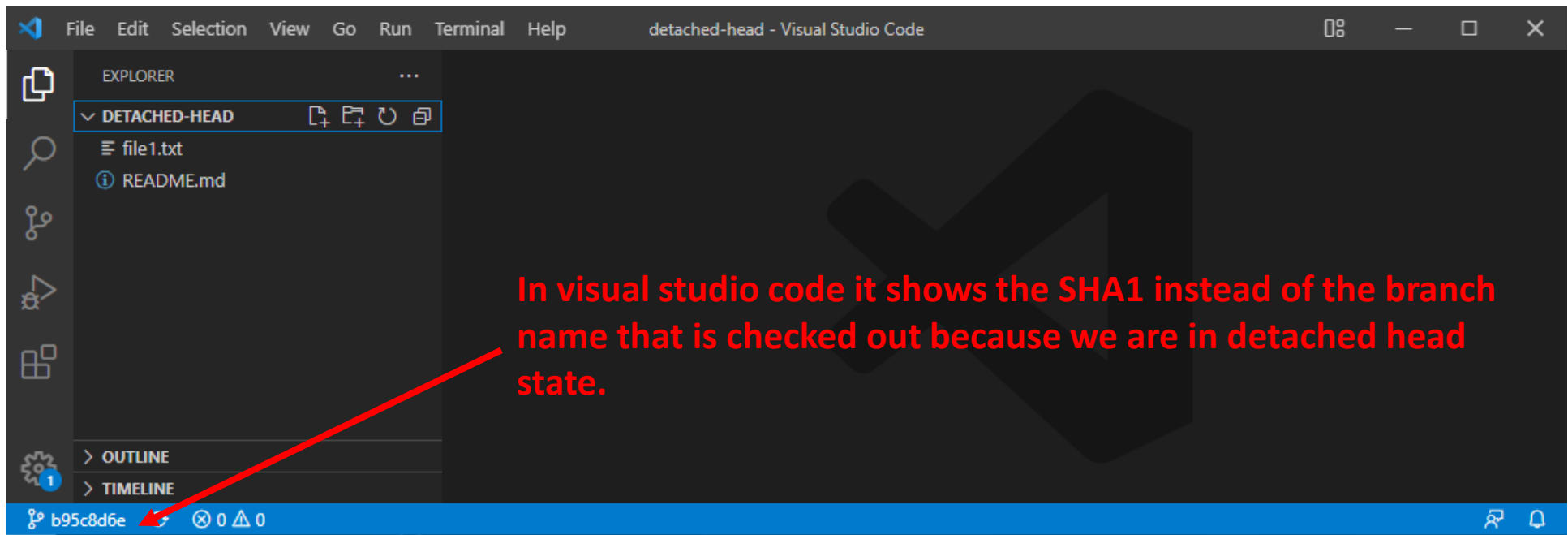
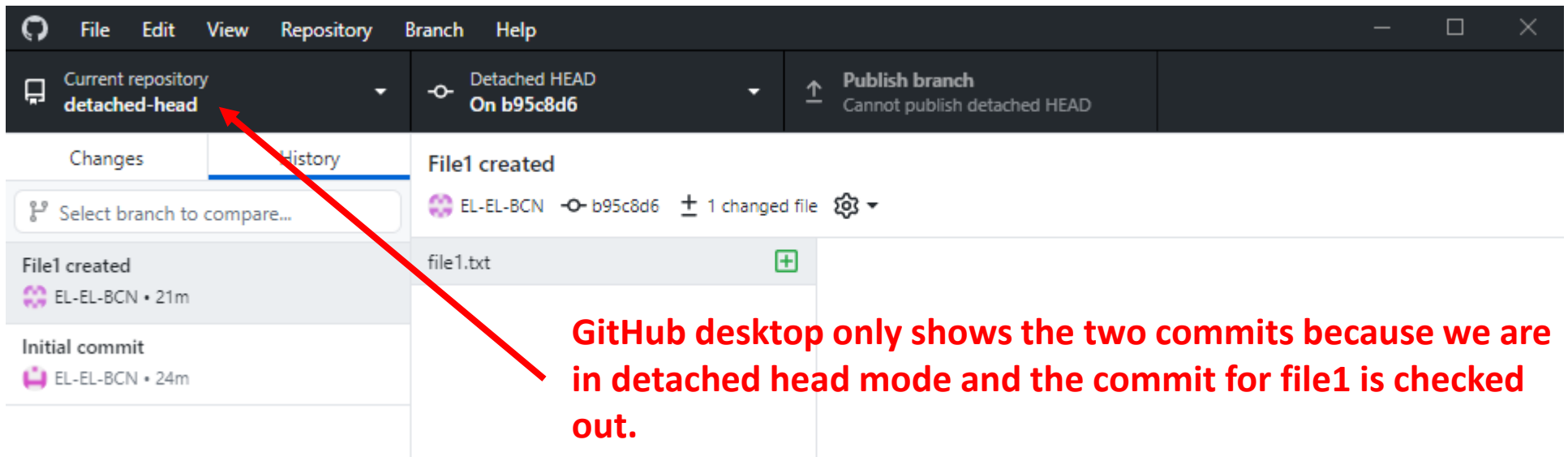
    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at b95c8d6 File1 created
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((b95c8d6...))
$
```



Create, stage and commit another file

In git log we can see that head points to a SHA1 but with out a branch name.
I.e. detached head state.

Create, stage and commit another file

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/detached-head
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((b95c8d6...))
$ touch file4.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((b95c8d6...))
$ git add file4.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((b95c8d6...))
$ git commit -m "file4 created in detached head"
[detached HEAD 4dff257] file4 created in detached head
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((4dff257...))
$ git log
commit 4dff257b5464657bbfbd947ea451381056f93686 (HEAD)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:18:48 2022 +0100

    file4 created in detached head

commit b95c8d6e6566177e3303ff3261a9953b229495a0
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:53:01 2022 +0100

    File1 created

commit 2d9bad0d4d6cd80398effa6837c80ab18e514f59 (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Wed Mar 9 22:49:43 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((4dff257...))
$ touch file5.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((4dff257...))
$ git add file5.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((4dff257...))
$ git commit -m "one more change in deatched head state"
[detached HEAD e571732] one more change in deatched head state
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file5.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((e571732...))
$
```

File Edit View Repository Branch Help

Current repository **detached-head**

Detached HEAD
On e571732

Publish branch
Cannot publish detached HEAD

Changes History

Select branch to compare...

one more change in detached head state
EL-EL-BCN • 4m

file4 created in detached head
EL-EL-BCN • 5m

File1 created
EL-EL-BCN • 31m

Initial commit
EL-EL-BCN • 34m

File1 created
EL-EL-BCN b95c8d6 ± 1 changed file

file1.txt

We can see all the commits made in detached head state.
Note that moving back to main branch now would result in these commits being garbage collected.

If I checkout the main branch now I get a message telling me that there are two commits that will be lost because they were created in detached head state.

I can revert the loss of these commits by checking out one of the SHA1's in the message.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/detached-head
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((e571732...))
$ git checkout main
Warning: you are leaving 2 commits behind, not connected to
any of your branches:
e571732 one more change in deatched head state
4dff257 file4 created in detached head

If you want to keep them by creating a new branch, this may be a good time
to do so with:

git branch <new-branch-name> e571732

Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git checkout e571732
Note: switching to 'e571732'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at e571732 one more change in deatched head state

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((e571732...))
```

Two more files were created, staged and committed in a detached head state.

These commits in detached head state can be preserved if we like them by creating a new branch with **git checkout -b <branch name>**

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/detached-head
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((71d189b...))
$ git log
commit 71d189b8a721ba5dddb4fa4bf27985c8d3233905 (HEAD)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:35:14 2022 +0100

    file8 created in detached head state

commit 0cf1d0a0d45fe3f381fd0c5bb1b9530fcc175c07
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:34:44 2022 +0100

    file7 created in detached head state

commit e571732d90687449b0f838921e1aae3d5548fac7
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:19:47 2022 +0100

    one more change in detached head state

commit 4dff257b5464657bbfbd947ea451381056f93686
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:18:48 2022 +0100

    file4 created in detached head

commit b95c8d6e6566177e3303ff3261a9953b229495a0
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:53:01 2022 +0100

    File1 created

commit 2d9bad0d4d6cd80398effa6837c80ab18e514f59 (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Wed Mar 9 22:49:43 2022 +0100

    Initial commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head ((71d189b...))
$ git checkout -b experimental
Switched to a new branch 'experimental'

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (experimental)
$
```

We can see the head pointer points to the experimental branch.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/detached-head
$
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (experimental)
$ cat .git/HEAD
ref: refs/heads/experimental
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/detached-head (experimental)
$ git log
commit 71d189b8a721ba5dddb4fa4bf27985c8d3233905 (HEAD -> experimental)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:35:14 2022 +0100

    file8 created in detached head state

commit 0cf1d0a0d45fe3f381fd0c5bb1b9530fcc175c07
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:34:44 2022 +0100

    file7 created in detached head state

commit e571732d90687449b0f838921e1aae3d5548fac7
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:19:47 2022 +0100

    one more change in detached head state

commit 4dff257b5464657bbfbd947ea451381056f93686
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 23:18:48 2022 +0100

    file4 created in detached head

commit b95c8d6e6566177e3303ff3261a9953b229495a0
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Wed Mar 9 22:53:01 2022 +0100

    File1 created

commit 2d9bad0d4d6cd80398effa6837c80ab18e514f59 (origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Wed Mar 9 22:49:43 2022 +0100

    Initial commit
```

If we now go back into detached head state by checking out the last commit we get the detached head warning.

The head pointer file now points to a SHA1 rather than a branch name.

If I checkout experimental branch then I do not get any warnings of lost commits because the detached head was checked out on the last commit so nothing is lost.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/detached-head
Initial commit

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (experimental)
$ git checkout 71d189b8
Note: switching to '71d189b8'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 71d189b file8 created in detached head state

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head ((71d189b...))
$ cat .git/HEAD
71d189b8a721ba5dddb4fa4bf27985c8d3233905

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head ((71d189b...))
$ git branch
* (HEAD detached at 71d189b)
  experimental
  main

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head ((71d189b...))
$ git checkout experimental
Switched to branch 'experimental'

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (experimental)
$
```

From main
branch the
experimental
branch was
merged. Then
experimental
branch was
deleted.

```
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main|MERGING)
$ git merge -v experimental
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main|MERGING)
$ git commit -m "merge experimental"
[main 90b5608] merge experimental

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$ git branch -d experimental
Deleted branch experimental (was 71d189b).

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/detached-head (main)
$
```

The screenshot shows the GitHub Desktop interface. The top bar includes a search bar with 'rebaseing' and a tab for 'detached-head'. The left sidebar shows the 'History' tab selected. The main area displays a commit history table with columns: Graph, Description, Date, Author, and Commit. The current commit, 'merge experimental', is highlighted in blue. The graph on the left shows a vertical line of commits, with the current commit at the top. The 'main' branch is shown as a blue circle with a '9' next to it, indicating 9 commits. The 'experimental' branch is shown as a red circle, indicating it has been deleted. The commit history table lists the following commits:

Graph	Description	Date	Author	Commit
○	merge experimental	9 Mar 2022 23:56	EL-EL-BCN <elliott	90b5608
●	file9 created im main branch	9 Mar 2022 23:52	EL-EL-BCN <elliott	7d42189
●	file8 created in detached head state	9 Mar 2022 23:35	EL-EL-BCN <elliott	71d189b
●	file7 created in detached head state	9 Mar 2022 23:34	EL-EL-BCN <elliott	0cf1d0a
●	one more change in deatched head state	9 Mar 2022 23:19	EL-EL-BCN <elliott	e571732
●	file4 created in detached head	9 Mar 2022 23:18	EL-EL-BCN <elliott	4dff257
●	File3 created	9 Mar 2022 22:54	EL-EL-BCN <elliott	291674f
●	File2 created	9 Mar 2022 22:53	EL-EL-BCN <elliott	60a13a3
●	File1 created	9 Mar 2022 22:53	EL-EL-BCN <elliott	b95c8d6
●	Initial commit	9 Mar 2022 22:49	EL-EL-BCN <10101	2d9bad0

Note that all changes were retained because a branch was created from detached head state and it was merged so these commits were not garbage collected.

ADVANCED GIT

Cloning a public repository

Lets start by cloning any public remote repository to our local computer.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue

e1lio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/vuejs/create-vue.git
Cloning into 'create-vue'...
remote: Enumerating objects: 3967, done.
remote: Counting objects: 100% (333/333), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 3967 (delta 288), reused 256 (delta 254), pack-reused 3634
Receiving objects: 100% (3967/3967), 766.33 KiB | 3.89 MiB/s, done.
Resolving deltas: 100% (2908/2908), done.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  first-project/          pull-requests/
JavaScript-Bible-Basics/      gitignore/              rebasing/
create-vue/                  hello-world/            tags/
detached-head/               my-first-git-hub-repository/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd create-vue/

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git config user.name EL-EL-BCN

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git config user.email elliottbcn@gmail.com

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ |
```

File Edit View Repository Actions Tools Help

create-vue

Commit Pull Push Fetch Branch Merge Stash Discard Tag

Git-flow Remote Terminal Explorer Settings

WORKSPACE

File Status

History

Search

BRANCHES

main

TAGS

REMOTES

STASHES

SUBMODULES

All Branches Show Remote Branches Date Order

Graph

main origin/main origin/HEAD

fix: only add vitest type

8 Mar 2022 14:13

Haoqun Jiang <ha

bc7bee8

origin/v2 chore: dependency maintenance

chore: update snapshot

2.0.4

chore: update snapshot

v3.1.7 3.1.7

feat: update vitest from 0.2 to 0.5

chore: update esbuild

fix: fix build script

chore: update zx

chore: dependency maintenance (update all none-major versions)

fix(cypress-ct): avoid cypress types polluting normal source files (#62)

feat: add vue-template-babel-compiler for optional chaining (#59)

feat: update dts (#60)

23 Feb 2022 8:47

22 Feb 2022 9:13

22 Feb 2022 9:12

22 Feb 2022 9:11

22 Feb 2022 9:11

21 Feb 2022 15:20

21 Feb 2022 14:57

21 Feb 2022 14:50

21 Feb 2022 14:50

21 Feb 2022 12:22

21 Feb 2022 7:58

18 Feb 2022 8:04

16 Feb 2022 11:39

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

Haoqun Jiang <ha

ZHAO Jinxiang <xi

ZHAO Jinxiang <xi

5915e98

3cbd6b7

7556e9d

8ad42c2

4ee6bd8

566f6a1

4e0e498

f693fc7

4a95e37

ec076b5

74cafa6

755b5cd

ed0232b

Sorted by file status

template/config/typescript/tsconfig.json

Deleted file

Reverse hunk

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Commit: bc7bee87de1769967343e35c49984f4bcb608726

[bc7bee8]

Parents: 8ad42c20ed

Author: Haoqun Jiang <haoqunjiang@gmail.com>

Date: 08 March 2022 21:13:34

Committer: Haoqun Jiang

fix: only add vitest type when it is chosen

template/tsconfig/base/tsconfig.vite-config.json

template/tsconfig/vitest/tsconfig.vite-config.json

template/config/typescript/tsconfig.json

We have git tags, branches, files, commits etc in this repository

File

Edit

View

Repository

Branch

Help

Current repository

create-vue

Current branch

main

Fetch origin

Never fetched

Changes

History

Select branch to compare...

fix: only add vitest type when it is chosen

Haoqun Jiang • 1d

chore: update snapshot

Haoqun Jiang • Feb 22, 2022

3.1.7

Haoqun Jiang • Feb 22, 2022

v3.1.7

feat: update vitest from 0.2 to 0.5

Haoqun Jiang • Feb 21, 2022

chore: update esbuild

Haoqun Jiang • Feb 21, 2022

fix: fix build script

Haoqun Jiang • Feb 21, 2022

chore: update zx

Haoqun Jiang • Feb 21, 2022

chore: dependency maintenance (upda...

Haoqun Jiang • Feb 21, 2022

fix(cypress-ct): avoid cypress types poll...

Haoqun Jiang • Feb 21, 2022

docs: note that the version number mu...

Haoqun Jiang • Feb 16, 2022

chore: update snapshot

fix: only add vitest type when it is chosen

Haoqun Jiang • bc7bee8 • 3 changed files • +9 -21

template\config\typescr...\tsconfig.json

template\tsc...\tsconfig.vite-config.json

template\tsc...\tsconfig.vite-config.json

@@ -1,20 +0,0 @@

1 -{

2 - "compilerOptions": {

3 - "target": "esnext",

4 - "useDefineForClassFields": true,

5 - "module": "esnext",

6 - "moduleResolution": "node",

7 - "isolatedModules": true,

8 - "strict": true,

9 - "jsx": "preserve",

10 - "sourceMap": true,

11 - "resolveJsonModule": true,

12 - "esModuleInterop": true,

13 - "paths": {

14 - "@/*": ["./*src/*"]

15 - },

16 - "lib": ["esnext", "dom", "dom.iterable", "scripthost"],

17 - "skipLibCheck": true

18 - },

19 - "include": ["vite.config.*", "env.d.ts", "src/**/*", "src/*

20 - }

We have git tags, branches, files, commits etc in this repository

Fix git lg command

If **git lg** command is not available, you should create an **alias** to make it work

1. `git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset - %C(yellow)%d%Creset %s %Cgreen(%cr)%C(bold blue)<%an>%Creset' --abbrev-commit"`

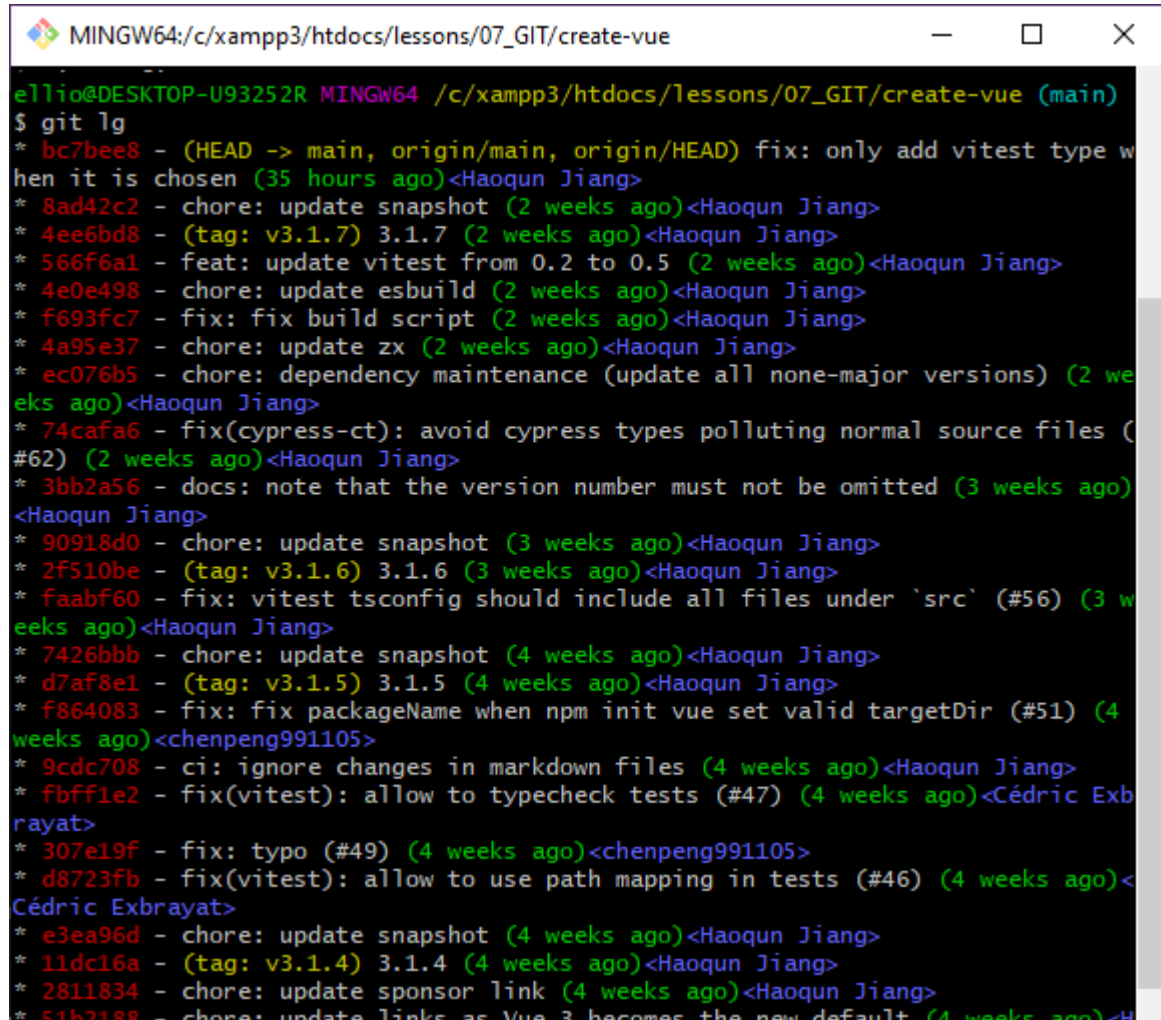
git lg command is a summary of git log where:

Red is the SHA1

Yellow is the head, branch and tag

Blue is the author and date time

White is the commit comment



```
MINGW64: /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* bc7bee8 - (HEAD -> main, origin/main, origin/HEAD) fix: only add vitest type w
hen it is chosen (35 hours ago)<Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago)<Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>
* 4a95e37 - chore: update zx (2 weeks ago)<Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (2 we
eks ago)<Haoqun Jiang>
* 74cafa6 - fix(cypress-ct): avoid cypress types polluting normal source files (
#62) (2 weeks ago)<Haoqun Jiang>
* 3bb2a56 - docs: note that the version number must not be omitted (3 weeks ago)
<Haoqun Jiang>
* 90918d0 - chore: update snapshot (3 weeks ago)<Haoqun Jiang>
* 2f510be - (tag: v3.1.6) 3.1.6 (3 weeks ago)<Haoqun Jiang>
* faabf60 - fix: vitest tsconfig should include all files under 'src' (#56) (3 w
eeks ago)<Haoqun Jiang>
* 7426bbb - chore: update snapshot (4 weeks ago)<Haoqun Jiang>
* d7af8e1 - (tag: v3.1.5) 3.1.5 (4 weeks ago)<Haoqun Jiang>
* f864083 - fix: fix packageName when npm init vue set valid targetDir (#51) (4
weeks ago)<chenpeng991105>
* 9cdc708 - ci: ignore changes in markdown files (4 weeks ago)<Haoqun Jiang>
* fbff1e2 - fix(vitest): allow to typecheck tests (#47) (4 weeks ago)<Cédric Exb
rayat>
* 307e19f - fix: typo (#49) (4 weeks ago)<chenpeng991105>
* d8723fb - fix(vitest): allow to use path mapping in tests (#46) (4 weeks ago)<
Cédric Exbrayat>
* e3ea96d - chore: update snapshot (4 weeks ago)<Haoqun Jiang>
* 11dc16a - (tag: v3.1.4) 3.1.4 (4 weeks ago)<Haoqun Jiang>
* 2811834 - chore: update sponsor link (4 weeks ago)<Haoqun Jiang>
* 51b2188 - chore: update links as Vue 3 becomes the new default (4 weeks ago)<H
```

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --oneline
bc7bee8 (HEAD -> main, origin/main, origin/HEAD) fix: only add vitest type when
it is chosen
8ad42c2 chore: update snapshot
4ee6bd8 (tag: v3.1.7) 3.1.7
566f6a1 feat: update vitest from 0.2 to 0.5
4e0e498 chore: update esbuild
f693fc7 fix: fix build script
4a95e37 chore: update zx
ec076b5 chore: dependency maintenance (update all none-major
74cafa6 fix(cypress-ct): avoid cypress types polluting norma
3bb2a56 docs: note that the version number must not be omitt
90918d0 chore: update snapshot
2f510be (tag: v3.1.6) 3.1.6
faabf60 fix: vitest tsconfig should include all files under
7426bbb chore: update snapshot
```

git log --oneline command will display the git log as collapsed view with one commit per line

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
$
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --graph
* commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/HEAD)
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Mar 8 21:13:34 2022 +0800
|
| fix: only add vitest type when it is chosen
|
* commit 8ad42c20ed7c98413e31f6e6dabc48dff06b068a
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Feb 22 16:11:18 2022 +0800
|
| chore: update snapshot
|
* commit 4ee6bd81759bef1ec055c6594247223c247a0bcb (tag: v3.1.7)
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Feb 22 16:11:03 2022 +0800
|
|
|
* commit ff3cbe1d14099269d001c34fcf01e91b89e8f528
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Mon Dec 27 10:59:04 2021 +0800
|
| fix: fixup the template code
|
* commit e068c4752dc33a7a62465de97af2a61528327f5e
| Merge: 3b69f07 0e6899a
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Mon Dec 27 10:56:05 2021 +0800
|
| Merge branch 'main' of https://github.com/vuejs/create-vue
|
* commit 0e6899accb5b6eb48701eb42b216241a70ecc49c
| Author: zhangenming <282126346@qq.com>
| Date: Mon Dec 27 10:50:11 2021 +0800
|
```

git log --graph command show the commits in branches with relationships to parents defined by a line (pipe symbol and stars!)

git log --graph --oneline command show the commits in branches with relationships to parents with one commit per line.

git log --stat command is a summary of git log with stats for each commit.

For the last commit we see that three files were changed then it gives a breakdown of those changes. i.e. the first file had 20 lines removed, the second file had 2 lines added and 2 lines removed, The third file had 8 lines added.

A total of three files were modified, 9 lines inserted and 21 removed.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --stat
commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/HEAD)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen

template/config/typescript/tsconfig.json | 20 -----
template/tsconfig/base/tsconfig.vite-config.json | 2 +-
template/tsconfig/vitest/tsconfig.vite-config.json | 8 ++++++
3 files changed, 9 insertions(+), 21 deletions(-)

commit 8ad42c20ed7c98413e31f6e6dabc48dff06b068a
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Feb 22 16:11:18 2022 +0800

    chore: update snapshot

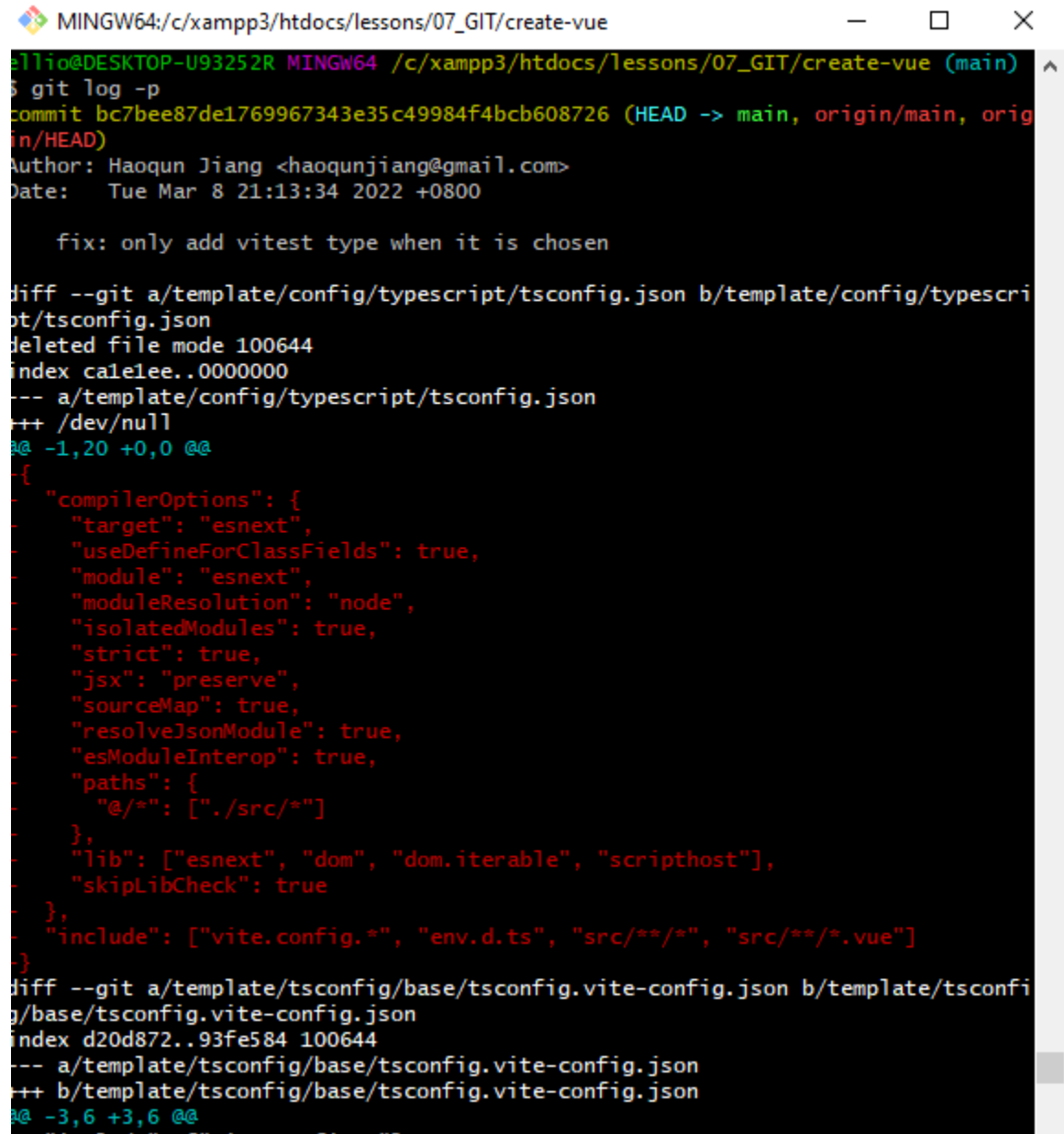
playground | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 4ee6bd81759bef1ec055c6594247223c247a0bcb (tag: v3.1.7)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
: ... skipping ...
commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/HEAD)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen
```


git log -p command shows exactly what was changed in each file during that commit.

This is not really practical to use in terminal and it is better to use graphical interface like GitHub desktop or SourceTree.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ello@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log -p
commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/HEAD)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen

diff --git a/template/config/typescript/tsconfig.json b/template/config/typescript/tsconfig.json
deleted file mode 100644
index cae1ee..0000000
--- a/template/config/typescript/tsconfig.json
+++ /dev/null
@@ -1,20 +0,0 @@
- {
-   "compilerOptions": {
-     "target": "esnext",
-     "useDefineForClassFields": true,
-     "module": "esnext",
-     "moduleResolution": "node",
-     "isolatedModules": true,
-     "strict": true,
-     "jsx": "preserve",
-     "sourceMap": true,
-     "resolveJsonModule": true,
-     "esModuleInterop": true,
-     "paths": {
-       "@/*": [".src/*"]
-     },
-     "lib": ["esnext", "dom", "dom.iterable", "scripthost"],
-     "skipLibCheck": true
-   },
-   "include": ["vite.config.*", "env.d.ts", "src/**/*", "src/**/*.vue"]
- }

diff --git a/template/tsconfig/base/tsconfig.vite-config.json b/template/tsconfig/base/tsconfig.vite-config.json
index d20d872..93fe584 100644
--- a/template/tsconfig/base/tsconfig.vite-config.json
+++ b/template/tsconfig/base/tsconfig.vite-config.json
@@ -3,6 +3,6 @@
- {
-   "compilerOptions": {
-     "target": "esnext",
-     "useDefineForClassFields": true,
-     "module": "esnext",
-     "moduleResolution": "node",
-     "isolatedModules": true,
-     "strict": true,
-     "jsx": "preserve",
-     "sourceMap": true,
-     "resolveJsonModule": true,
-     "esModuleInterop": true,
-     "paths": {
-       "@/*": [".src/*"]
-     },
-     "lib": ["esnext", "dom", "dom.iterable", "scripthost"],
-     "skipLibCheck": true
-   },
-   "include": ["vite.config.*", "env.d.ts", "src/**/*", "src/**/*.vue"]
- }
- }
```

git log -4 --graph command shows us the last four commits in graph format.

Equally **git log -4 --oneline** or **git log -4 --graph --oneline** command can be used.

The dash4 option where 4 is the number of commits from latest so dash3 or dash 2 etc works.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
chore: update snapshot

diff --git a/playground b/playground
index 062d1f3..284fcf5 160000
--- a/playground
+++ b/playground
@@ -1,1 @@
-Subproject commit 062d1f31bb7a4c94e9b456db1b0166298ae7b4ea

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log -4 --graph
* commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/HEAD)
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Mar 8 21:13:34 2022 +0800
|
| fix: only add vitest type when it is chosen
|
* commit 8ad42c20ed7c98413e31f6e6dabc48dff06b068a
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Feb 22 16:11:18 2022 +0800
|
| chore: update snapshot
|
* commit 4ee6bd81759bef1ec055c6594247223c247a0bcb (tag: v3.1.7)
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Tue Feb 22 16:11:03 2022 +0800
|
| 3.1.7
|
* commit 566f6a1fab87b85b7251860aeaa927db68526801
| Author: Haoqun Jiang <haoqunjiang@gmail.com>
| Date: Mon Feb 21 22:20:45 2022 +0800
|
| feat: update vitest from 0.2 to 0.5

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ |
```

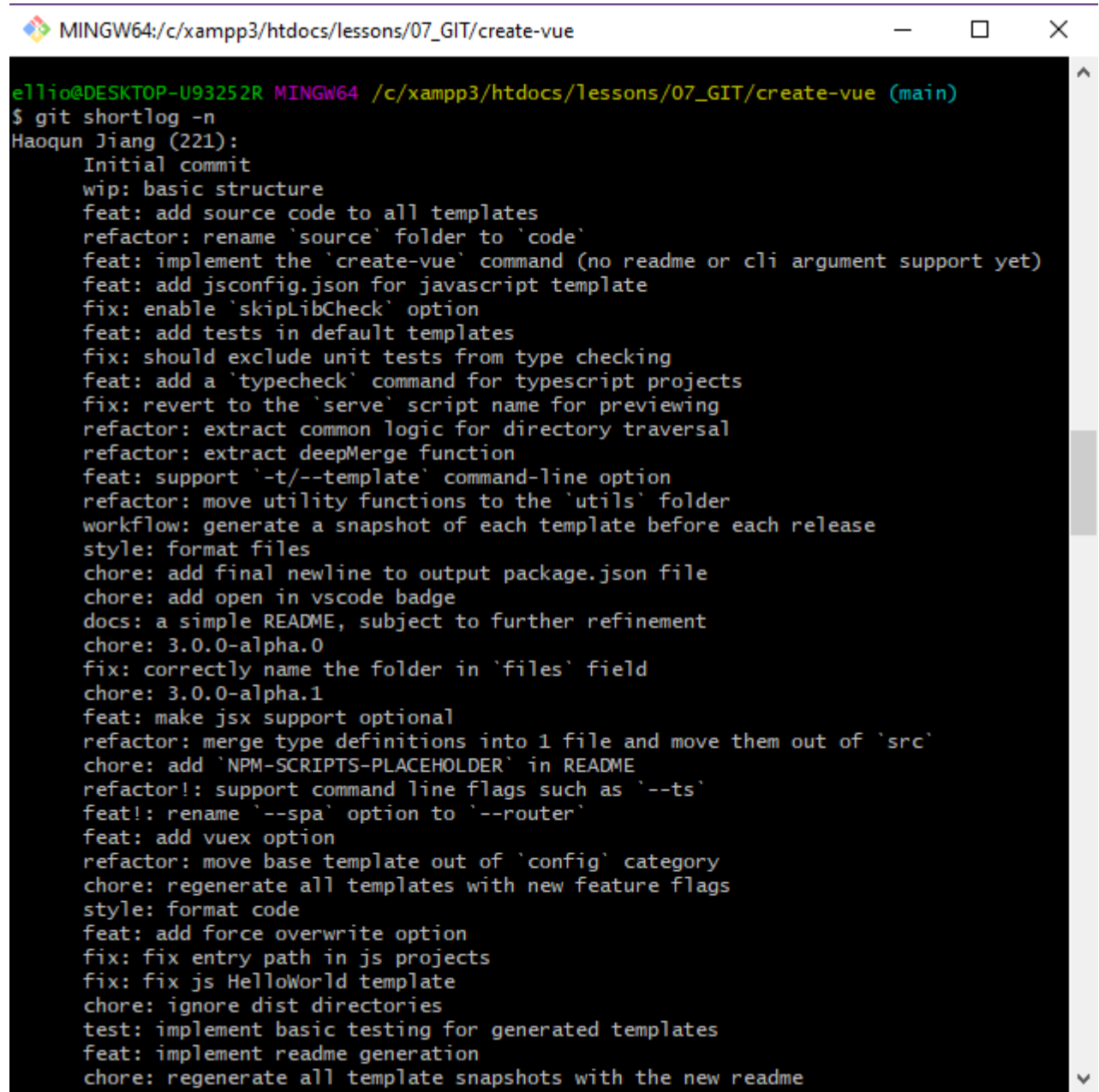

git shortlog command shows us commits sorted by author alphabetically.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git shortlog
Cédric Exbrayat (7):
    fix: use string as prop type in TS (#13)
    fix: eslint with prettier (#25)
    chore: remove console.log (#30)
    chore: remove outdated comment (#31)
    chore: add coverage to gitignore (#41)
    fix(vitest): allow to use path mapping in tests (#46)
    fix(vitest): allow to typecheck tests (#47)

Haoqun Jiang (221):
    Initial commit
    wip: basic structure
    feat: add source code to all templates
    refactor: rename 'source' folder to 'code'
    feat: implement the 'create-vue' command (no readme or cli argument support yet)
    feat: add jsconfig.json for javascript template
    fix: enable 'skipLibCheck' option
    feat: add tests in default templates
    fix: should exclude unit tests from type checking
    feat: add a 'typecheck' command for typescript projects
    fix: revert to the 'serve' script name for previewing
    refactor: extract common logic for directory traversal
    refactor: extract deepMerge function
    feat: support '-t/--template' command-line option
    refactor: move utility functions to the 'utils' folder
    workflow: generate a snapshot of each template before each release
    style: format files
    chore: add final newline to output package.json file
    chore: add open in vscode badge
```

git shortlog -n command shows us commits sorted by author in order of number of commits from highest to lowest.

In this repository we see that Haoquin Jiang has made 221 commits.

A terminal window with a dark background and light-colored text. The window title is 'MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue'. The prompt is 'e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)'. The command '\$ git shortlog -n' has been executed, resulting in a list of 221 commits by Haoqun Jiang, sorted by the number of commits in descending order. The first commit is the 'Initial commit'. The commits include various features, refactors, fixes, and chores related to the 'create-vue' project.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git shortlog -n
Haoqun Jiang (221):
  Initial commit
  wip: basic structure
  feat: add source code to all templates
  refactor: rename 'source' folder to 'code'
  feat: implement the 'create-vue' command (no readme or cli argument support yet)
  feat: add jsconfig.json for javascript template
  fix: enable 'skipLibCheck' option
  feat: add tests in default templates
  fix: should exclude unit tests from type checking
  feat: add a 'typecheck' command for typescript projects
  fix: revert to the 'serve' script name for previewing
  refactor: extract common logic for directory traversal
  refactor: extract deepMerge function
  feat: support '-t/--template' command-line option
  refactor: move utility functions to the 'utils' folder
  workflow: generate a snapshot of each template before each release
  style: format files
  chore: add final newline to output package.json file
  chore: add open in vscode badge
  docs: a simple README, subject to further refinement
  chore: 3.0.0-alpha.0
  fix: correctly name the folder in 'files' field
  chore: 3.0.0-alpha.1
  feat: make jsx support optional
  refactor: merge type definitions into 1 file and move them out of 'src'
  chore: add 'NPM-SCRIPTS-PLACEHOLDER' in README
  refactor!: support command line flags such as '--ts'
  feat!: rename '--spa' option to '--router'
  feat: add vuex option
  refactor: move base template out of 'config' category
  chore: regenerate all templates with new feature flags
  style: format code
  feat: add force overwrite option
  fix: fix entry path in js projects
  fix: fix js HelloWorld template
  chore: ignore dist directories
  test: implement basic testing for generated templates
  feat: implement readme generation
  chore: regenerate all template snapshots with the new readme
```

git shortlog -n -s command shows us commits and author sorted in order of number of commits from highest to lowest.

git shortlog -n -s -e command shows us commits, author and email sorted in order of number of commits from highest to lowest.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git shortlog -n -s
 221 Haoqun Jiang
   7 Cédric Exbrayat
   3 btea
   2 chenpeng991105
   2 zhangenming
   1 JD
   1 Sneaken
   1 ZHAO Jinxiang

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git shortlog -n -s -e
 221 Haoqun Jiang <haoqunjiang@gmail.com>
   7 Cédric Exbrayat <cexbrayat@users.noreply.github.com>
   3 btea <2356281422@qq.com>
   2 chenpeng991105 <chenpeng991105@foxmail.com>
   2 zhangenming <282126346@qq.com>
   1 JD <47495003+jd-0001@users.noreply.github.com>
   1 Sneaken <37062838+Sneaken@users.noreply.github.com>
   1 ZHAO Jinxiang <xiaoxiangmoe@gmail.com>

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$
```

git log --author <name> command shows us commits for a specific author.

git log --author <name> --oneline command shows us commits for a specific author with one commit per line

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
$ cd create-vue/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git shortlog -n -s -e
 221 Haoqun Jiang <haoqunjiang@gmail.com>
   7 Cédric Exbrayat <cexbrayat@users.noreply.github.com>
   3 btea <2356281422@qq.com>
   2 chenpeng991105 <chenpeng991105@foxmail.com>
   2 zhangenming <282126346@qq.com>
   1 JD <47495003+jd-0001@users.noreply.github.com>
   1 Sneaken <37062838+Sneaken@users.noreply.github.com>
   1 ZHAO Jinxiang <xiaoxiangmoe@gmail.com>

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --author "Cédric Exbrayat"
commit fbff1e2e3e095393d0efeecfc341b85bef9f57c9
Author: Cédric Exbrayat <cexbrayat@users.noreply.github.com>
Date: Fri Feb 11 08:49:18 2022 +0100

    fix(vitest): allow to typecheck tests (#47)

commit d8723fb59c34d3e5599442903170240417efb47c
Author: Cédric Exbrayat <cexbrayat@users.noreply.github.com>
Date: Thu Feb 10 08:00:11 2022 +0100

    fix(vitest): allow to use path mapping in tests (#46)

commit b5851bcba8c54beedd0f3a91f3f831e094ce10ef
Author: Cédric Exbrayat <cexbrayat@users.noreply.github.com>
Date: Sat Jan 29 03:55:30 2022 +0100

    chore: add coverage to gitignore (#41)

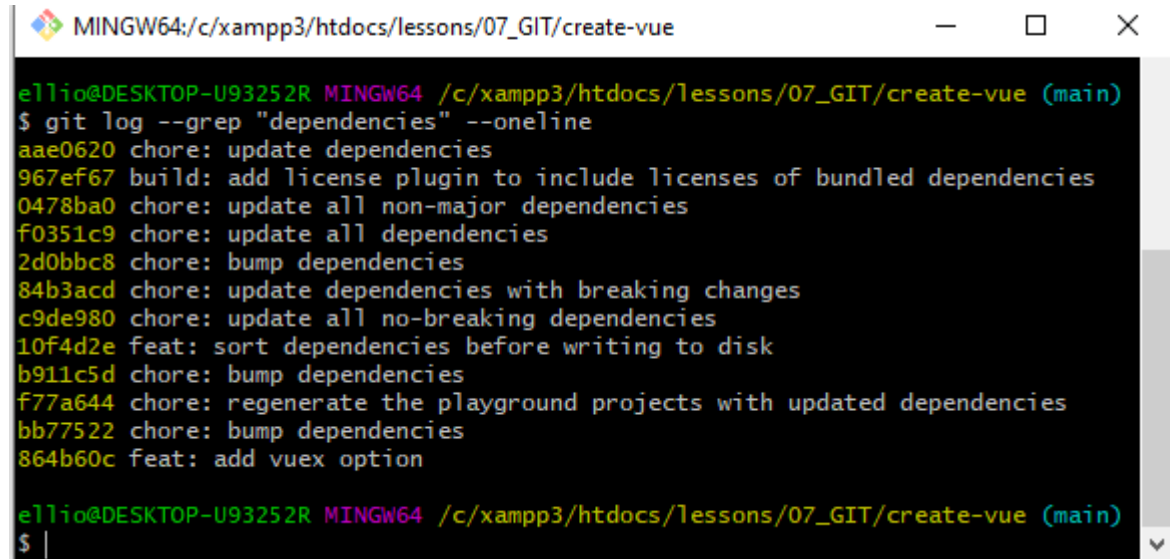
commit fe47e81d9373fbcbb890c7bb2bafb3d81b6f2ba0
Author: Cédric Exbrayat <cexbrayat@users.noreply.github.com>
Date: Wed Jan 19 09:23:11 2022 +0100

    chore: remove outdated comment (#31)
...skipping...
commit fbff1e2e3e095393d0efeecfc341b85bef9f57c9
Author: Cédric Exbrayat <cexbrayat@users.noreply.github.com>
```

git log --grep <word> command is like a wildcard search where logs containing a particular word or text or numbers can be filtered

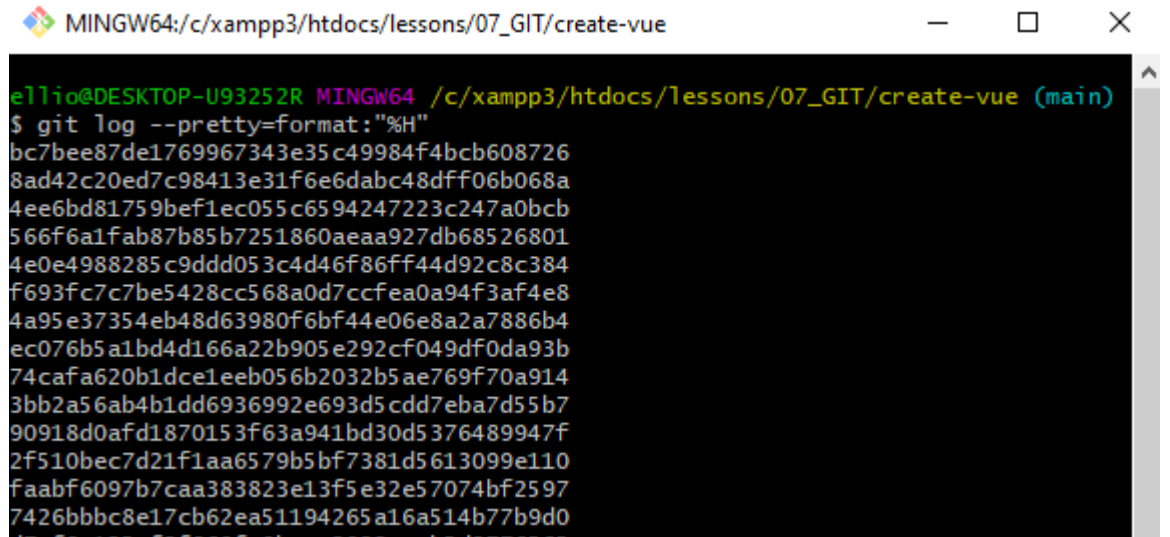
git log --grep <word> --oneline wildcard search results on one line

git log --pretty=format:"%H" command is a wildcard search to output only the SHA1 hash of all the commits.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --grep "dependencies" --oneline
aae0620 chore: update dependencies
967ef67 build: add license plugin to include licenses of bundled dependencies
0478ba0 chore: update all non-major dependencies
f0351c9 chore: update all dependencies
2d0bbc8 chore: bump dependencies
84b3acd chore: update dependencies with breaking changes
c9de980 chore: update all no-breaking dependencies
10f4d2e feat: sort dependencies before writing to disk
b911c5d chore: bump dependencies
f77a644 chore: regenerate the playground projects with updated dependencies
bb77522 chore: bump dependencies
864b60c feat: add vuex option

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ |
```



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --pretty=format:"%H"
bc7bee87de1769967343e35c49984f4bcb608726
8ad42c20ed7c98413e31f6e6dabc48dff06b068a
4ee6bd81759bef1ec055c6594247223c247a0bcb
566f6a1fab87b85b7251860aeaa927db68526801
4e0e4988285c9ddd053c4d46f86ff44d92c8c384
f693fc7c7be5428cc568a0d7ccfea0a94f3af4e8
4a95e37354eb48d63980f6bf44e06e8a2a7886b4
ec076b5a1bd4d166a22b905e292cf049df0da93b
74cafa620b1dce1eeb056b2032b5ae769f70a914
3bb2a56ab4b1dd6936992e693d5cdd7eba7d55b7
90918d0afd1870153f63a941bd30d5376489947f
2f510bec7d21f1aa6579b5bf7381d5613099e110
faabf6097b7caa383823e13f5e32e57074bf2597
7426bbbc8e17cb62ea51194265a16a514b77b9d0
```

git log --pretty=format:"%cn %H"
command is a wildcard search to
output only the author and SHA1
hash of all the commits.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --pretty=format:"%cn %H"
Haoqun Jiang bc7bee87de1769967343e35c49984f4bcb608726
Haoqun Jiang 8ad42c20ed7c98413e31f6e6dabc48dff06b068a
Haoqun Jiang 4ee6bd81759bef1ec055c6594247223c247a0bcb
Haoqun Jiang 566f6a1fab87b85b7251860aeaa927db68526801
Haoqun Jiang 4e0e4988285c9ddd053c4d46f86ff44d92c8c384
Haoqun Jiang f693fc7c7be5428cc568a0d7ccfea0a94f3af4e8
Haoqun Jiang 4a95e37354eb48d63980f6bf44e06e8a2a7886b4
Haoqun Jiang ec076b5a1bd4d166a22b905e292cf049df0da93b
GitHub 74cafa620b1dce1eeb056b2032b5ae769f70a914
Haoqun Jiang 3bb2a56ab4b1dd6936992e693d5cdd7eba7d55b7
Haoqun Jiang 90918d0afd1870153f63a941bd30d5376489947f
Haoqun Jiang 2f510be3d31f1aa6570b5bf7381d5613090e110
```

**git log --pretty=format:"author of
commit: %cn SHA1 hash: %h"**
lower h indicates shortened sha1
hash. We can also add text into the
format string.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --pretty=format:"Author of commit: %cn SHA1 hash: %h"
Author of commit: Haoqun Jiang SHA1 hash: bc7bee8
Author of commit: Haoqun Jiang SHA1 hash: 8ad42c2
Author of commit: Haoqun Jiang SHA1 hash: 4ee6bd8
Author of commit: Haoqun Jiang SHA1 hash: 566f6a1
Author of commit: Haoqun Jiang SHA1 hash: 4e0e498
Author of commit: Haoqun Jiang SHA1 hash: f693fc7
Author of commit: Haoqun Jiang SHA1 hash: 4a95e37
Author of commit: Haoqun Jiang SHA1 hash: ec076b5
Author of commit: GitHub SHA1 hash: 74cafa6
Author of commit: Haoqun Jiang SHA1 hash: 3bb2a56
Author of commit: Haoqun Jiang SHA1 hash: 90918d0
Author of commit: Haoqun Jiang SHA1 hash: 2f510be
Author of commit: GitHub SHA1 hash: faabf60
Author of commit: Haoqun Jiang SHA1 hash: 7426bbb
Author of commit: Haoqun Jiang SHA1 hash: d7af8e1
Author of commit: GitHub SHA1 hash: 6851893
```


git log --pretty=format:"author: %cn; SHA1 hash: %h; Date: %cd"
lower h indicates shortened sha1 hash. We can also add text into the format string.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --pretty=format:"Author: %cn; SHA1 hash: %h; Date: %cd"
Author: Haoqun Jiang; SHA1 hash: bc7bee8; Date: Tue Mar 8 21:13:34 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 8ad42c2; Date: Tue Feb 22 16:11:18 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 4ee6bd8; Date: Tue Feb 22 16:11:03 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 566f6a1; Date: Mon Feb 21 22:20:45 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 4e0e498; Date: Mon Feb 21 21:57:01 2022 +0800
Author: Haoqun Jiang; SHA1 hash: f693fc7; Date: Mon Feb 21 21:50:51 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 4a95e37; Date: Mon Feb 21 21:50:00 2022 +0800
Author: Haoqun Jiang; SHA1 hash: ec076b5; Date: Mon Feb 21 19:22:42 2022 +0800
Author: GitHub; SHA1 hash: 74cafa6; Date: Mon Feb 21 14:58:46 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 3bb2a56; Date: Wed Feb 16 15:57:41 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 90918d0; Date: Wed Feb 16 14:35:14 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 2f510be; Date: Wed Feb 16 14:34:50 2022 +0800
Author: GitHub; SHA1 hash: faabf60; Date: Wed Feb 16 14:34:28 2022 +0800
Author: Haoqun Jiang; SHA1 hash: 7426bbb; Date: Fri Feb 11 17:48:44 2022 +0800
Author: Haoqun Jiang; SHA1 hash: d7af8e1; Date: Fri Feb 11 17:48:26 2022 +0800
Author: GitHub; SHA1 hash: f864083; Date: Fri Feb 11 16:08:08 2022 +0800
```

git log --merges --oneline command shows only git logs for merges.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --merges --oneline
e068c47 Merge branch 'main' of https://github.com/vuejs/create-vue

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --merges
commit e068c4752dc33a7a62465de97af2a61528327f5e
Merge: 3b69f07 0e6899a
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Mon Dec 27 10:56:05 2021 +0800

    Merge branch 'main' of https://github.com/vuejs/create-vue

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ |
```

git log --no-merges --oneline

command shows all commits minus
merge commits

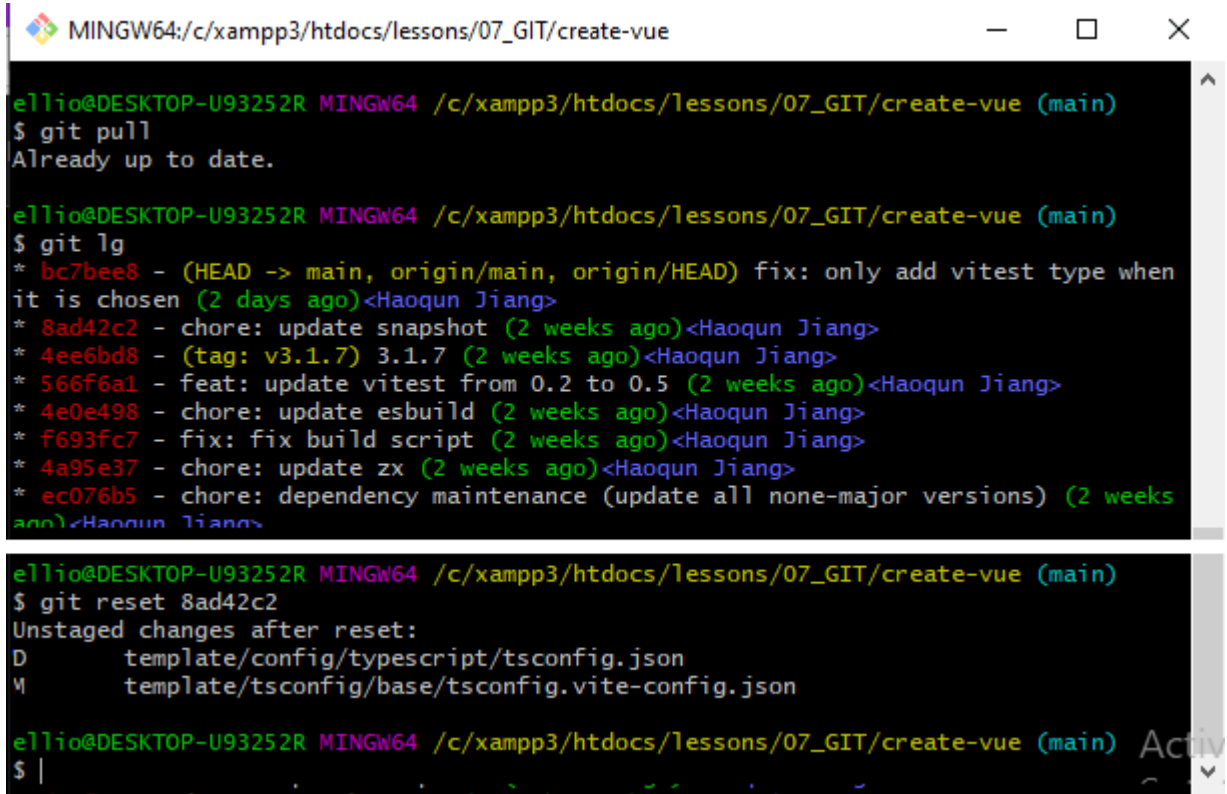
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log --no-merges --oneline
bc7bee8 (HEAD -> main, origin/main, origin/HEAD) fix: only add vitest type when it i
s chosen
8ad42c2 chore: update snapshot
4ee6bd8 (tag: v3.1.7) 3.1.7
566f6a1 feat: update vitest from 0.2 to 0.5
4e0e498 chore: update esbuild
f693fc7 fix: fix build script
4a95e37 chore: update zx
ec076b5 chore: dependency maintenance (update all none-major versions)
74cafa6 fix(cypress-ct): avoid cypress types polluting normal source files (#62)
3bb2a56 docs: note that the version number must not be omitted
90918d0 chore: update snapshot
2f510be (tag: v3.1.6) 3.1.6
faabf60 fix: vitest tsconfig should include all files under 'src' (#56)
7426bbb chore: update snapshot
d7af8e1 (tag: v3.1.5) 3.1.5
f864083 fix: fix packageName when npm init vue set valid targetDir (#51)
```


Git reset

It is possible to modify the git history but this is a destructive process and is not recommended on any public branches. **Git reset** has three options: **Hard**, **Mixed** (default) and **soft**.

The local git repository is up to date (git pull) and in the git log lets say we want to reset git from the last commit (bc7bee8) to the penultimate commit (8ad42c2)

git reset <SHA1> command resets the repository to the earlier commit and tells us that there are now unstaged changes. In this case one file deleted and one file modified. This operation is relatively safe because changes were kept in working directory.

A screenshot of a terminal window with a dark background. The window title is 'MINGW64: /c/xampp3/htdocs/lessons/07_GIT/create-vue'. The terminal shows the following commands and output:
1. `$ git pull`
Output: `Already up to date.`
2. `$ git lg`
Output: A list of git log entries:
* bc7bee8 - (HEAD -> main, origin/main, origin/HEAD) fix: only add vitest type when it is chosen (2 days ago) <Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago) <Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago) <Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago) <Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago) <Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago) <Haoqun Jiang>
* 4a95e37 - chore: update zx (2 weeks ago) <Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (2 weeks ago) <Haoqun Jiang>
3. `$ git reset 8ad42c2`
Output: `Unstaged changes after reset:`
D template/config/typescript/tsconfig.json
M template/tsconfig/base/tsconfig.vite-config.json
4. The prompt `$ |` is shown at the bottom.

By default git reset applies mixed mode. What this means is that The commit is discarded in the repository and staging area but kept in the working directory.

I have undone these changes by performing a fresh pull from the remote repository.

The local git repository is up to date (git pull) and in the git log lets say we want to reset git from the last commit (bc7bee8) to the commit (566f6a1)

git reset --soft <SHA1> command resets the repository to the earlier commit and tells us that there are now uncommitted changes. Git reset soft only removes the changes from the repository but leaves the staging and working area as they were.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* bc7bee8 - (HEAD -> main, origin/main, origin/HEAD) fix: only add vitest type when
it is chosen (2 days ago)<Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago)<Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>
* 4a95e37 - chore: update zx (2 weeks ago)<Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (2 weeks
ago)<Haoqun Jiang>
* 74cafa6 - fix(cypress-ct): avoid cypress types polluting normal source files (#62)
(3 weeks ago)<Haoqun Jiang>
```

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --soft 566f6a1

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 3 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   package.json
    modified:   playground
    deleted:    template/config/typescript/tsconfig.json
    modified:   template/tsconfig/base/tsconfig.vite-config.json
    new file:   template/tsconfig/vitest/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git commit -m "chages were committed after soft reset"
[main b68d36e] chages were committed after soft reset
5 files changed, 11 insertions(+), 23 deletions(-)
delete mode 100644 template/config/typescript/tsconfig.json
create mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
```

The local git repository is up to date (git pull) and in the git log lets say we want to reset git from the last commit (bc7bee8) to the commit (4a95e37)

git reset --hard <SHA1> command resets the repository to the earlier commit and tells us that the head has moved to the specified commit. Git reset hard removes the changes from the working area, staging area and local directory so is very destructive. Commits after the specified commit will be lost and files deleted, or reverted to the version in the specified commit.

These files can only be recovered with a pull from the remote directory.

Note that none of the git reset options were pushed to remote repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* b68d36e - (HEAD -> main) chages were committed after soft reset (2 minutes ago)<EL-BCN>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>
* 4a95e37 - chore: update zx (2 weeks ago)<Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (2 weeks ago)<Haoqun Jiang>
* 74cafa6 - fix(cypress-ct): avoid cypress types polluting normal source files (#62) (3 weeks ago)<Haoqun Jiang>
* 3bb2a56 - docs: note that the version number must not be omitted (3 weeks ago)<Haoqun Jiang>
* 90918d0 - chore: update snapshot (3 weeks ago)<Haoqun Jiang>

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --hard 4a95e37
HEAD is now at 4a95e37 chore: update zx

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 6 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git pull
Updating 4a95e37..bc7bee8
Fast-forward
 package.json | 4 +-
 playground | 2 +-
 pnpm-lock.yaml | 376 ++++++-----
 scripts/build.mjs | 2 +-
 template/config/typescript/tsconfig.json | 20 --
 template/config/vitest/package.json | 2 +-
 template/tsconfig/base/tsconfig.vite-config.json | 2 +-
 template/tsconfig/vitest/tsconfig.vite-config.json | 8 +
 8 files changed, 122 insertions(+), 294 deletions(-)
 delete mode 100644 template/config/typescript/tsconfig.json
 create mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$
```

The local git repository is up to date (git pull) and in the git log lets say we want to reset git from the last commit (bc7bee8) to the commit (4a95e37)

git reset HEAD-n command resets the repository head pointer back the number specified.

The default option of mixed reset is used so the changes are applied to the repository and staging area and can be reverted by adding and committing the changes.

To revert my local repository to its original state I performed a hard reset to a commit before I modified it then performed a pull. Now it is as was originally.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset HEAD~5
Unstaged changes after reset:
M   package.json
M   pnpm-lock.yaml
D   template/config/typescript/tsconfig.json
M   template/config/vitest/package.json
M   template/tsconfig/base/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git commit -m "changes were made after git reset"
[main 64c2171] changes were made after git reset
6 files changed, 120 insertions(+), 292 deletions(-)
delete mode 100644 template/config/typescript/tsconfig.json
create mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$
```

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --hard bc7bee87d
HEAD is now at bc7bee87d fix: only add vitest type when it is chosen

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git pull
Already up to date.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log
commit bc7bee87de1769967343e35c49984f4bcb608726 (HEAD -> main, origin/main, origin/H
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen

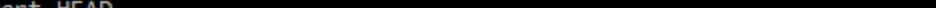
commit 8ad42c20ed7c98413e31f6e6dabc48dff06b068a
Author: Haoqun Jiang <haoqunjiang@gmail.com>
```

Git Revert

Git revert is not destructive but creates a new commit to revert all changes made in a specific commit.

Because it is creating a new commit git asks us for a message.

We can keep the default message and exit using :wq (write, quit)



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
e1llo@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git revert HEAD
```

[illegible]

```
[main aeaddel] Revert "fix: only add vitest type when it is chosen"
3 files changed, 21 insertions(+), 9 deletions(-)
create mode 100644 template/config/typescript/tsconfig.json
delete mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* aeaddel - (HEAD -> main) Revert "fix: only add vitest type when it is chosen" (5 m
inutes ago)<EL-EL-BCN>
* bc7bee8 - (origin/main, origin/HEAD) fix: only add vitest type when it is chosen (
2 days ago)<Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago)<Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update es (2 weeks ago)<Haoqun Jiang>
```

Notice the new commit created.

git log -p command shows all the changes to each file that were made with the git revert.

This command is really useful when you have already pushed changes to remote repository but you really need to undo them.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
$ git log -p
commit aeaddde11c5c66cdb63ec033d762cd824a0f4617a (HEAD -> main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Thu Mar 10 22:37:52 2022 +0100

    Revert "fix: only add vitest type when it is chosen"

    This reverts commit bc7bee87de1769967343e35c49984f4bcb608726.

diff --git a/template/config/typescript/tsconfig.json b/template/config/typescript/tsconfig.json
new file mode 100644
index 00000000..ca1e1ee
--- /dev/null
+++ b/template/config/typescript/tsconfig.json
@@ -0,0 +1,20 @@
+{
+  "compilerOptions": {
+    "target": "esnext",
+    "useDefineForClassFields": true,
+    "module": "esnext",
+    "moduleResolution": "node",
+    "isolatedModules": true,
+    "strict": true,
+    "jsx": "preserve",
+    "sourceMap": true,
+    "resolveJsonModule": true,
+    "esModuleInterop": true,
+    "paths": {
+      "@/*": [".src/*"]
+    },
+    "lib": ["esnext", "dom", "dom.iterable", "scripthost"],
+    "skipLibCheck": true
+  },
+  "include": ["vite.config.*", "env.d.ts", "src/**/*.vue"]
+}

diff --git a/template/tsconfig/base/tsconfig.vite-config.json b/template/tsconfig/base/tsconfig.vite-config.json
index 93fe584..d20d872 100644
--- a/template/tsconfig/base/tsconfig.vite-config.json
+++ b/template/tsconfig/base/tsconfig.vite-config.json
@@ -3,6 +3,6 @@
  "include": ["vite.config.*"],
  "compilerOptions": {
    "composite": true,
-    "types": ["node"]
+    "types": ["node", "vitest"]
  },
}
```

Git Amend

This option is useful when you have made an error or typo on the last commit. It is a destructive process so should only be done on non public branch or repository.

The last commit is overwritten with the amended commit.

Note that we can also amend author for the last commit.

```
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git commit --amend -m "new message for the last commit"
[main 7cd64d2] new message for the last commit
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800
3 files changed, 9 insertions(+), 21 deletions(-)
delete mode 100644 template/config/typescript/tsconfig.json
create mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* 7cd64d2 - (HEAD -> main) new message for the last commit (19 seconds ago)<Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago)<Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git commit --amend --author="EL-EL-BCN <eliottbcn@gmail.com>"

MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
new message for the last commit

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author:      EL-EL-BCN <eliottbcn@gmail.com>
# Date:       Tue Mar 8 21:13:34 2022 +0800
#
# On branch main
# Your branch and 'origin/main' have diverged,
# and have 1 and 1 different commits each, respectively.
#   (use "git pull" to merge the remote branch into yours)
#
# Changes to be committed:
#       deleted:    template/config/typescript/tsconfig.json
#       modified:   template/tsconfig/base/tsconfig.vite-config.json
#       new file:   template/tsconfig/vitest/tsconfig.vite-config.json
#
~
~
~
<xampp3/htdocs/lessons/07_GIT/create-vue/.git/COMMIT_EDITMSG [unix] (23:03 10/03/2022)18,1 All
:wq|

[main 82d6dbf] new message for the last commit
Author: EL-EL-BCN <eliottbcn@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800
3 files changed, 9 insertions(+), 21 deletions(-)
delete mode 100644 template/config/typescript/tsconfig.json
create mode 100644 template/tsconfig/vitest/tsconfig.vite-config.json

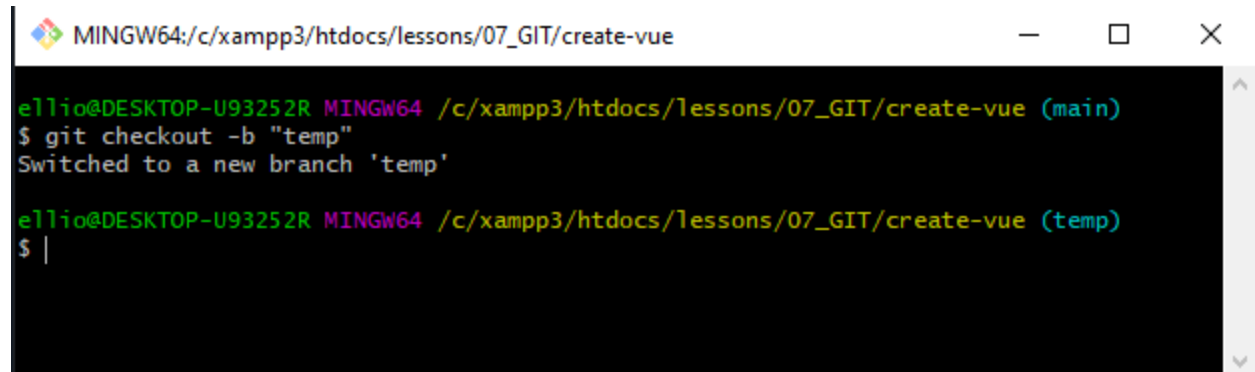
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$
```

Git Cherry pick

Cherry pick allows us to take any commit and insert it into currently checked out branch as latest commit. This is useful if we are working on feature branch and want to take one commit from that branch such as a bug fix and move it to the current branch.

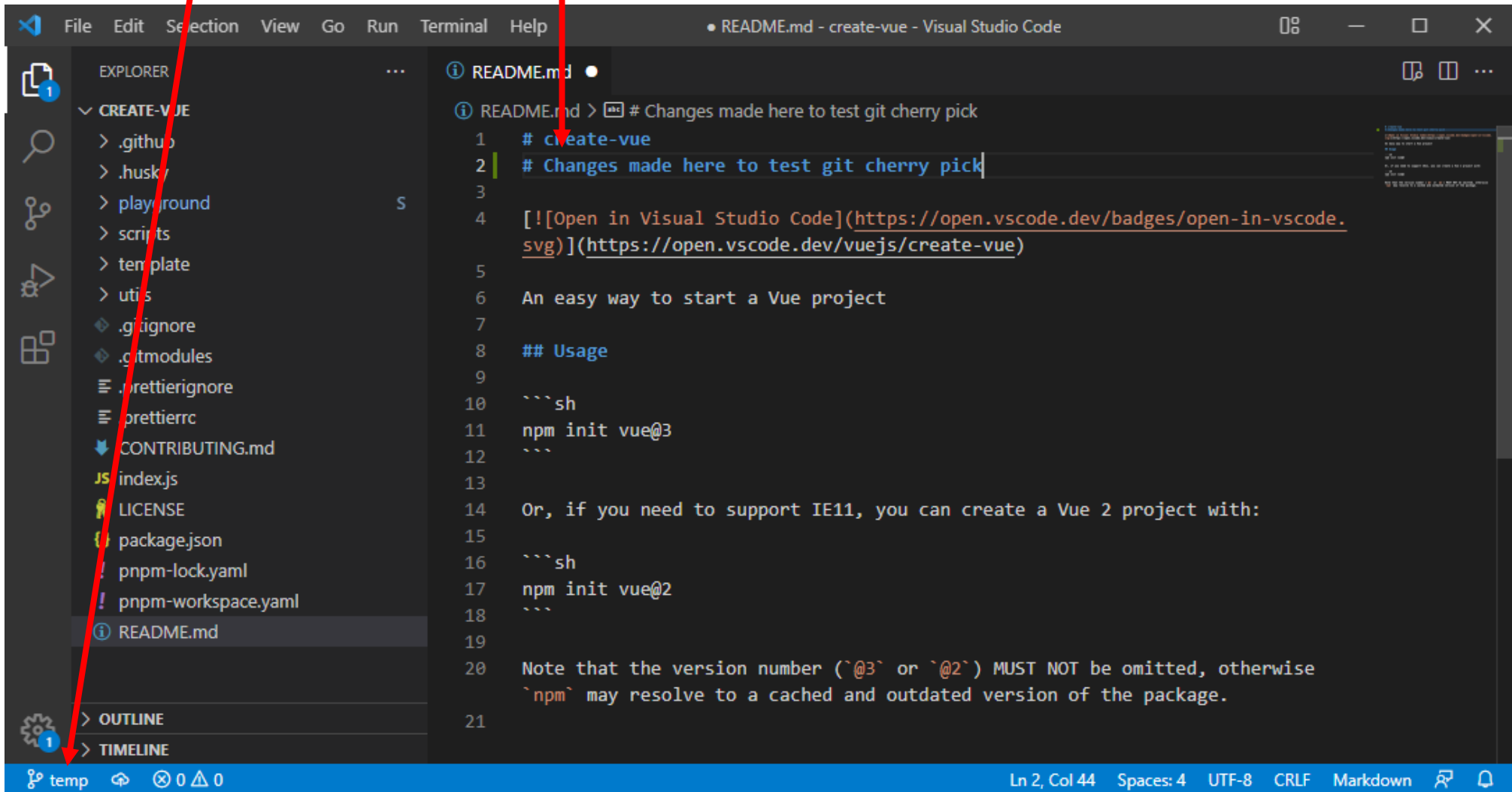
This can also be used to take one or two commits from detached head state and put them in the checked out branch rather than creating a whole new branch and merging it.

In this example we have created and checked out a new branch called temp

A screenshot of a Windows terminal window with a black background and green text. The title bar at the top reads 'MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue'. The terminal shows a user prompt 'e11io@DESKTOP-U93252R' followed by the command '\$ git checkout -b "temp"'. The output is 'Switched to a new branch \'temp\''. The prompt changes to '(temp)' for the next line, which shows '\$ |' indicating the cursor is ready for input.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue  
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)  
$ git checkout -b "temp"  
Switched to a new branch 'temp'  
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)  
$ |
```


In the temp branch I have made changes to the README file and saved it



The changes were staged and committed.

Now we see the commit for the changes in the temp branch.

Back in the main branch we can **git cherry-pick <SHA1>** the commit to import into the main branch.

Note that the sha1 of the commit in temp branch is 2e92dee but when it is cherry-picked into main branch the sha1 is 3f9e7b. This is because cherry picking creates a new commit so the date time will be different.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git commit -a -m "Some changes to README to test cherry pick"
[temp 2e92dee] Some changes to README to test cherry pick
1 file changed, 1 insertion(+)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git log
commit 2e92deefd0fcf17d37bbb48b05059fac81273b91 (HEAD -> temp)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Thu Mar 10 23:33:46 2022 +0100

    Some changes to README to test cherry pick

commit bc7bee87de1769967343e35c49984f4bcb608726 (origin/main, origin/HEAD, main)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git cherry-pick 2e92deefd0
[main 3f9e7bb] Some changes to README to test cherry pick
Date: Thu Mar 10 23:33:46 2022 +0100
1 file changed, 1 insertion(+)

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git log
commit 3f9e7bb09529ab8f7ac3f1ffdeec692c22cf974c (HEAD -> main)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Thu Mar 10 23:33:46 2022 +0100

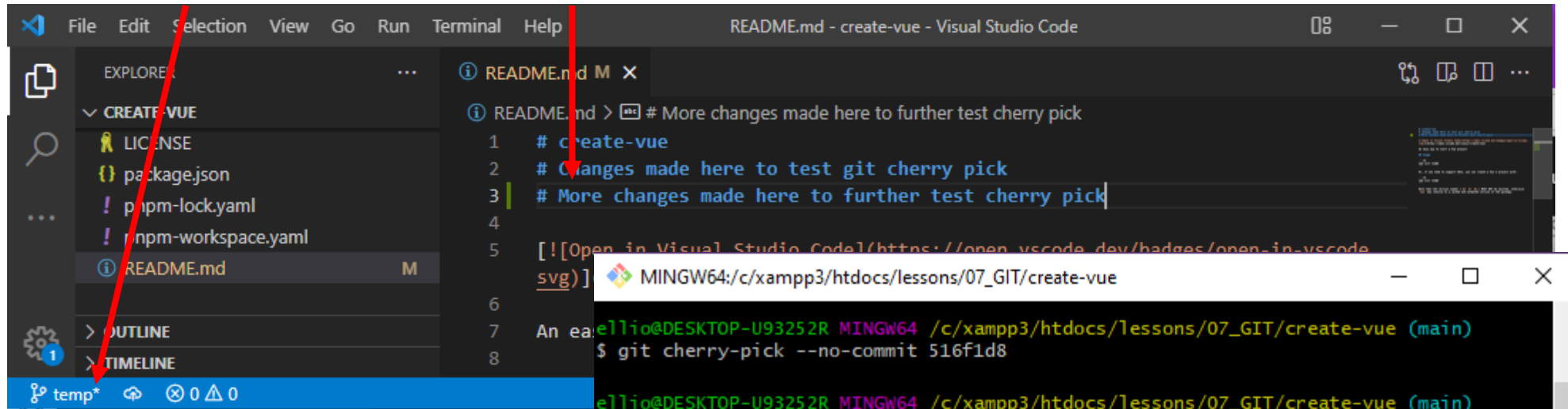
    Some changes to README to test cherry pick

commit bc7bee87de1769967343e35c49984f4bcb608726 (origin/main, origin/HEAD)
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date: Tue Mar 8 21:13:34 2022 +0800

    fix: only add vitest type when it is chosen

commit 8ad42c20ed7c98413e31f6e6dabc48dfe06b068a
```

In the temp branch I have made more changes to the README file and saved it



Back in the main branch we can **git cherry-pick --no-commit <SHA1>** to stage this commit from the temp branch into the main branch.

git status -v will show the changes made and to want files.

Finally we can manually commit the changes.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git cherry-pick --no-commit 516f1d8

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git status -v
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

diff --git a/README.md b/README.md
index 074a763..d5f3abf 100644
--- a/README.md
+++ b/README.md
@@ -1,5 +1,6 @@
 # create-vue
 # Changes made here to test git cherry pick
+# More changes made here to further test cherry pick

[![Open in Visual Studio Code](https://open.vscode.dev/badges/open-in-vscode.svg)](https://open.vscode.dev/vuejs/create-vue)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git commit -m "more changes made in the README file to test cherry pick"
[main 1b3105d] more changes made in the README file to test cherry pick
1 file changed, 1 insertion(+)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$
```

Git Reflog

This will show a **history of all operations on the local repository**. Using results of this command it is possible to revert back to a previous state.

Note that there is an ascending counter **HEAD@{0}** that indicates where the head was.

git reflog show temp will show all local commits for that branch.

I can pick a particular SHA1 and check it out in detached head state.

Then checkout main branch.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/create-vue
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reflog
1b3105d (HEAD -> main) HEAD@{0}: commit: more changes made in the README file to test
cherry pick
3f9e7bb HEAD@{1}: checkout: moving from temp to main
516f1d8 (temp) HEAD@{2}: commit: Some more changes to README to test cherry pick
2e92dee HEAD@{3}: checkout: moving from main to temp
3f9e7bb HEAD@{4}: cherry-pick: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) HEAD@{5}: checkout: moving from temp to main
2e92dee HEAD@{6}: commit: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) HEAD@{7}: checkout: moving from main to temp
bc7bee8 (origin/main, origin/HEAD) HEAD@{8}: pull: Fast-forward
4e0e498 HEAD@{9}: reset: moving to 4e0e498
82d6dbf HEAD@{10}: commit (amend): new message for the last commit
7cd64d2 HEAD@{11}: commit (amend): new message for the last commit
bc7bee8 (origin/main, origin/HEAD) HEAD@{12}: pull: Fast-forward

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reflog show temp
516f1d8 (temp) temp@{0}: commit: Some more changes to README to test cherry pick
2e92dee temp@{1}: commit: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) temp@{2}: branch: Created from HEAD

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git checkout bc7bee8
Note: switching to 'bc7bee8'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at bc7bee8 fix: only add vitest type when it is chosen

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue ((bc7bee8...))
$ git checkout main
Previous HEAD position was bc7bee8 fix: only add vitest type when it is chosen
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
```

Note that the ascending counter **HEAD@{0}** is dynamic, i.e position zero moves up to latest commit etc..

We can checkout one of these counters, i.e. six operations back using **git-checkout HEAD@{6}** and now we are in detached head state.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reflog
1b3105d (HEAD -> main) HEAD@{0}: checkout: moving from bc7bee87de1769967343e35c49984f4
bcb608726 to main
bc7bee8 (origin/main, origin/HEAD) HEAD@{1}: checkout: moving from main to bc7bee8
1b3105d (HEAD -> main) HEAD@{2}: commit: more changes made in the README file to test
cherry pick
3f9e7bb HEAD@{3}: checkout: moving from temp to main
516f1d8 (temp) HEAD@{4}: commit: Some more changes to README to test cherry pick
2e92dee HEAD@{5}: checkout: moving from main to temp
3f9e7bb HEAD@{6}: cherry-pick: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) HEAD@{7}: checkout: moving from temp to main

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git checkout HEAD@{6}
Note: switching to 'HEAD@{6}'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 3f9e7bb Some changes to README to test cherry pick
```


git lg shows us all the commits and we can chose one from 5 weeks ago to hard reset to.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* 1b3105d - (HEAD -> main) more changes made in the README file to test cherry pick (3
2 minutes ago)<EL-EL-BCN>
* 3f9e7bb - Some changes to README to test charry pick (48 minutes ago)<EL-EL-BCN>
* bc7bee8 - (origin/main, origin/HEAD) fix: only add vitest type when it is chosen (2
days ago)<Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (2 weeks ago)<Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (2 weeks ago)<Haoqun Jiang>
* 4e0e498 - chore: update esbuild (2 weeks ago)<Haoqun Jiang>
* f693fc7 - fix: fix build script (2 weeks ago)<Haoqun Jiang>
* 4a95e37 - chore: update zx (2 weeks ago)<Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (3 weeks ag
o)<Haoqun Jiang>
* 74cafa6 - fix(cypress-ct): avoid cypress types polluting normal source files (#62) (
3 weeks ago)<Haoqun Jiang>
* 3bb2a56 - docs: note that the version number must not be omitted (3 weeks ago)<Haoqu
n Jiang>
* 90918d0 - chore: update snapshot (3 weeks ago)<Haoqun Jiang>
* 2f510be - (tag: v3.1.6) 3.1.6 (3 weeks ago)<Haoqun Jiang>
* faabf60 - fix: vitest tsconfig should include all files under 'src' (#56) (3 weeks a
go)<Haoqun Jiang>
* 7426bbb - chore: update snapshot (4 weeks ago)<Haoqun Jiang>
* d7af8e1 - (tag: v3.1.5) 3.1.5 (4 weeks ago)<Haoqun Jiang>
* f864083 - fix: fix packageName when npm init vue set valid targetDir (#51) (4 weeks
ago)<chenpeng991105>
* 9cdc708 - ci: ignore changes in markdown files (4 weeks ago)<Haoqun Jiang>
* fbff1e2 - fix(vitest): allow to typecheck tests (#47) (4 weeks ago)<Cédric Exbrayat>
* 307e19f - fix: typo (#49) (4 weeks ago)<chenpeng991105>
* d8723fb - fix(vitest): allow to use path mapping in tests (#46) (4 weeks ago)<Cédric
Exbrayat>
* e3ea96d - chore: update snapshot (5 weeks ago)<Haoqun Jiang>
* 11dc16a - (tag: v3.1.4) 3.1.4 (5 weeks ago)<Haoqun Jiang>
```

Now head pointer is at the commit from 5 weeks ago.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --hard e3ea96d
HEAD is now at e3ea96d chore: update snapshot

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* e3ea96d - (HEAD -> main) chore: update snapshot (5 weeks ago)<Haoqun Jiang>
* 11dc16a - (tag: v3.1.4) 3.1.4 (5 weeks ago)<Haoqun Jiang>
* 2811834 - chore: update sponsor link (5 weeks ago)<Haoqun Jiang>
* 51b2188 - chore: update links as Vue 3 becomes the new default (5 weeks ago)<Haoqun
Jiang>
* 3a11b80 - chore: update snapshot (5 weeks ago)<Haoqun Jiang>
* a40c52b - (tag: v3.1.3) 3.1.3 (5 weeks ago)<Haoqun Jiang>
* aae0620 - chore: update dependencies (5 weeks ago)<Haoqun Jiang>
* 5f207d4 - chore: update snapshot (5 weeks ago)<Haoqun Jiang>
```

git reflog shows us all the commits and recent resets

If we do not like the results of a reset we can chose the SHA1, for example with counter HEAD@{1}, and reset to that.

This is particularly useful to undo any destructive changes in the repository.

Note that git reflog only lists operations performed locally in repository and does not look at remote repository or other computers of other collaborators.

Note that messages in git reflog are normally kept only 90 days so a revert over 3 months is not viable from git reflog.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reflog
e3ea96d (HEAD -> main) HEAD@{0}: reset: moving to e3ea96d
1b3105d HEAD@{1}: checkout: moving from 3f9e7bb09529ab8f7ac3f1ffdeec692c22cf974c to main
3f9e7bb HEAD@{2}: checkout: moving from main to HEAD@{6}
1b3105d HEAD@{3}: checkout: moving from bc7bee87de1769967343e35c49984f4bcb608726 to main
bc7bee8 (origin/main, origin/HEAD) HEAD@{4}: checkout: moving from main to bc7bee8
1b3105d HEAD@{5}: commit: more changes made in the README file to test cherry pick
3f9e7bb HEAD@{6}: checkout: moving from temp to main
516f1d8 (temp) HEAD@{7}: commit: Some more changes to README to test cherry pick
2e92dee HEAD@{8}: checkout: moving from main to temp
3f9e7bb HEAD@{9}: cherry-pick: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) HEAD@{10}: checkout: moving from temp to main
2e92dee HEAD@{11}: commit: Some changes to README to test cherry pick
bc7bee8 (origin/main, origin/HEAD) HEAD@{12}: checkout: moving from main to temp
bc7bee8 (origin/main, origin/HEAD) HEAD@{13}: pull: Fast-forward
4e0e498 HEAD@{14}: reset: moving to 4e0e498
82d6dbf HEAD@{15}: commit (amend): new message for the last commit
7cd64d2 HEAD@{16}: commit (amend): new message for the last commit
bc7bee8 (origin/main, origin/HEAD) HEAD@{17}: pull: Fast-forward
4e0e498 HEAD@{18}: reset: moving to 4e0e498
aeaddel HEAD@{19}: revert: Revert "fix: only add vitest type when it is chosen"
bc7bee8 (origin/main, origin/HEAD) HEAD@{20}: reset: moving to bc7bee87d
18a1e88 HEAD@{21}: pull: Merge made by the 'ort' strategy.

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --hard 1b3105d
HEAD is now at 1b3105d more changes made in the README file to test cherry pick

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* 1b3105d - (HEAD -> main) more changes made in the README file to test cherry pick (4
1 minutes ago)<EL-EL-BCN>
* 3f9e7bb - Some changes to README to test cherry pick (57 minutes ago)<EL-EL-BCN>
* bc7bee8 - (origin/main, origin/HEAD) fix: only add vitest type when it is chosen (2
days ago)<Haoqun Jiang>
* 8ed42e2 - chore: update snapshot (2 weeks ago)<Haoqun Jiang>
```

Git Stashing

```
File Edit Selection View Go Run Terminal Help
README.md - create-vue - Visual Studio Code

EXPLORER
CREATE-VUE
CONTRIBUTING.md
JS index.js
LICENSE M
package.json
pnpm-lock.yaml
pnpm-workspace.yaml
README.md M

OUTLINE
TIMELINE

temp* 0 0 0
Ln 4, Col 39 Spaces: 4 UTF-8 CRLF Markdown
```

README.md > # Changes to test stashing git feature

```
1 # create-vue
2 # Changes made here to test git cherry pick
3 # More changes made here to further test cherry pick
4 # Changes to test stashing git feature
5
6 [[Open in Visual Studio Code](https://open.vscode.dev/badges/open-in-vscode.svg)](https://open.vscode.dev/vuejs/create-vue)
7
8 An easy way to start a Vue project
9
10 ## Usage
```

README.md file and LICENSE file modified in temp branch and bot files saved.

```
File Edit Selection View Go Run Terminal Help
LICENSE - create-vue - Visual Studio Code

EXPLORER
CREATE-VUE
CONTRIBUTING.md
JS index.js
LICENSE M
package.json
pnpm-lock.yaml
pnpm-workspace.yaml
README.md M

OUTLINE
TIMELINE

temp* 0 0 0
Ln 2, Col 28 Spaces: 4 UTF-8 CRLF Markdown
```

LICENSE > # Test git stashing feature

```
1 # create-vue core license
2 # Test git stashing feature
3
4 create-vue is released under the MIT license:
5
6 MIT License
7
8 Copyright (c) 2021-present vuejs
9
10 Permission is hereby granted, free of charge, to any person obtaining a copy
11 of this software and associated documentation files (the "Software"), to deal
```


git status shows two files pending to be staged and committed.

I have staged only the LICENSE file but now I need to get back to the main branch to do some work there.

I can stash the files using **git stash** command. Note the output.

But if I look again at the files in visual studio code I can see that the changes I made are no longer there so where have the changes been saved/stashed?

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue

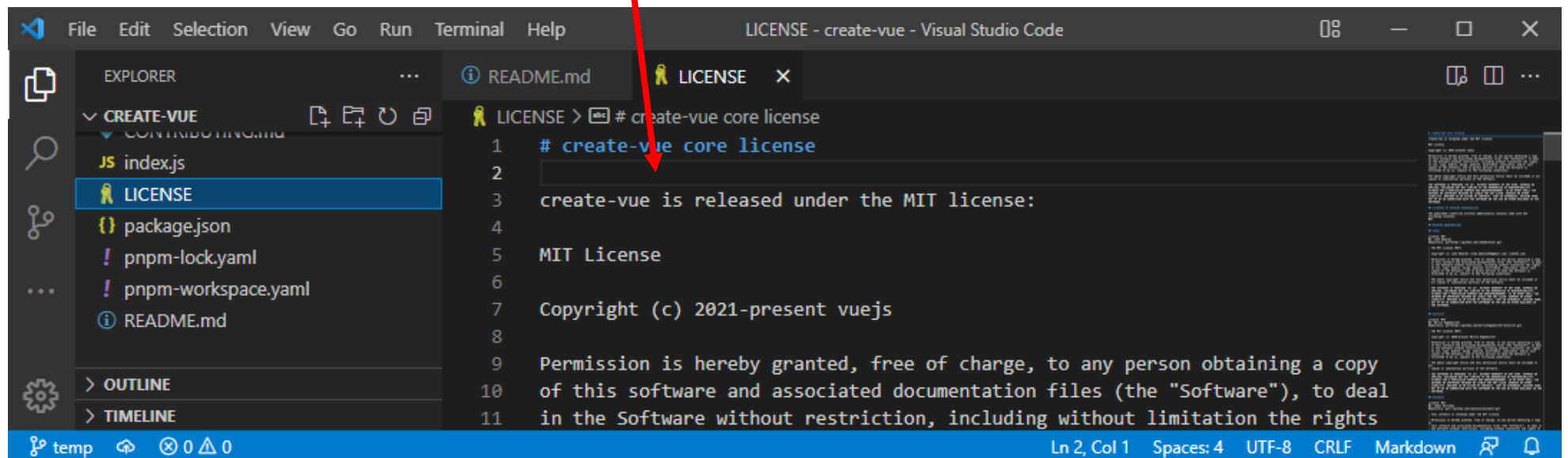
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git status
On branch temp
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   LICENSE
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git add LICENSE

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git stash
Saved working directory and index state WIP on temp: 516f1d8 Some more changes t
o README to test charry pick

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$
```



Git stores stashes in the directory
.git/refs/stash

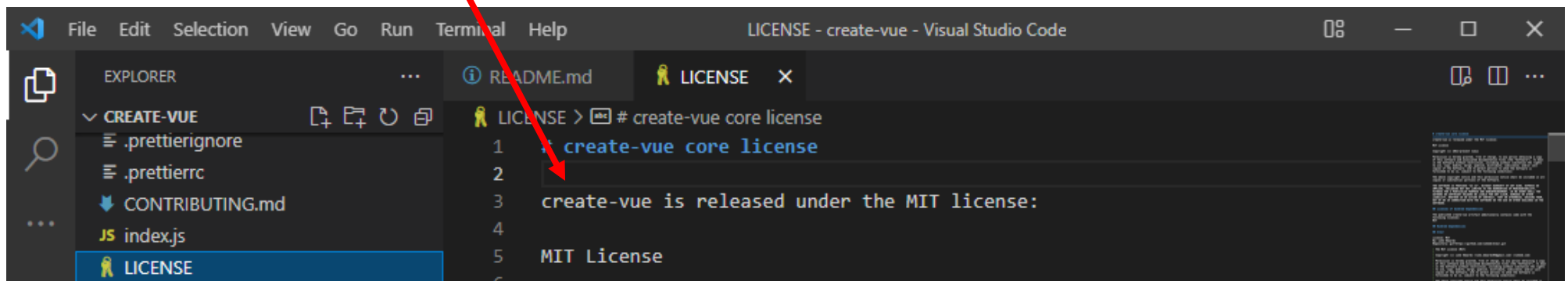
The contents of the stash file are a SHA1

The contents of the stash SHA1 is a
temporary commit which contains
information about the stash changes.
Note the test at the end is the same as
the output of the git stash command.

If I move into main branch then back
into temp branch, I see that in Visual
studio code the files are still missing the
lines that were stashed.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ cat .git/refs/stash
fd94d5040f30dbec970d91505925f496859c94
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git cat-file -p fd94d5040f
tree 0e92f7a47c4b13b083c787e02ccb4c0417ae0637
parent 516f1d8c6839794493f522d3ee9a35310f0f1001
parent 73e2c75b4b7bf719f6986b570239772a22f7be19
author EL-EL-BCN <elliottbcn@gmail.com> 1647117183 +0100
committer EL-EL-BCN <elliottbcn@gmail.com> 1647117183 +0100

WIP on temp: 516f1d8 Some more changes to README to test charry pick
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git cat-file -t fd94d5040f
commit
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git checkout main
Switched to branch 'main'
Your branch and 'origin/main' have diverged,
and have 2 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git checkout temp
Switched to branch 'temp'
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$
```



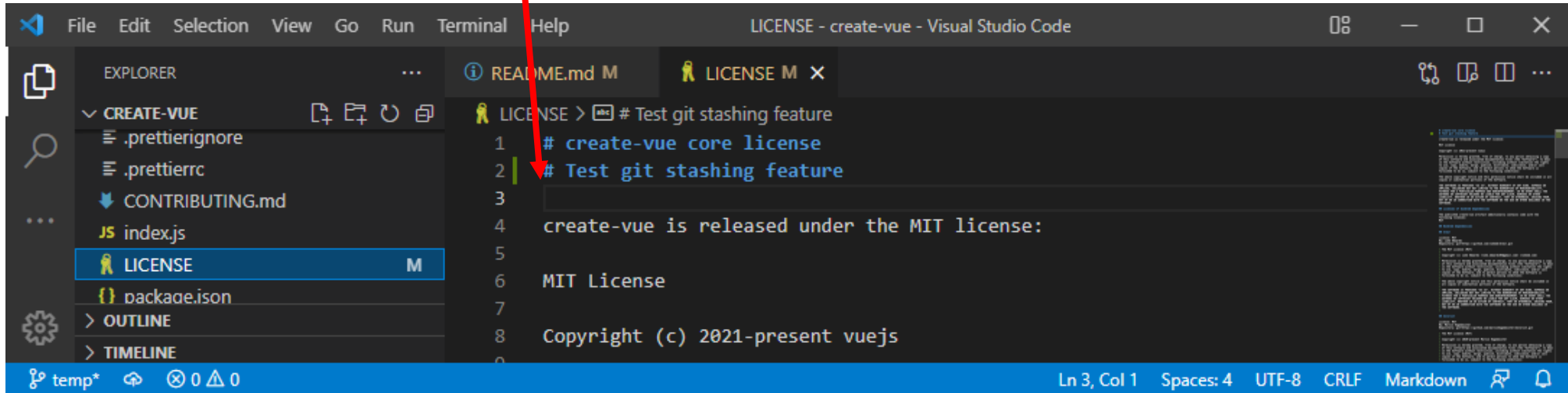
I can activate the changes made in the stash with **git stash pop** command and now we see two modified files.

Now the changes reappear in the files

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git stash pop
On branch temp
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   LICENSE
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (fd94d5040f30dbec970d91505925f9be496859c94)

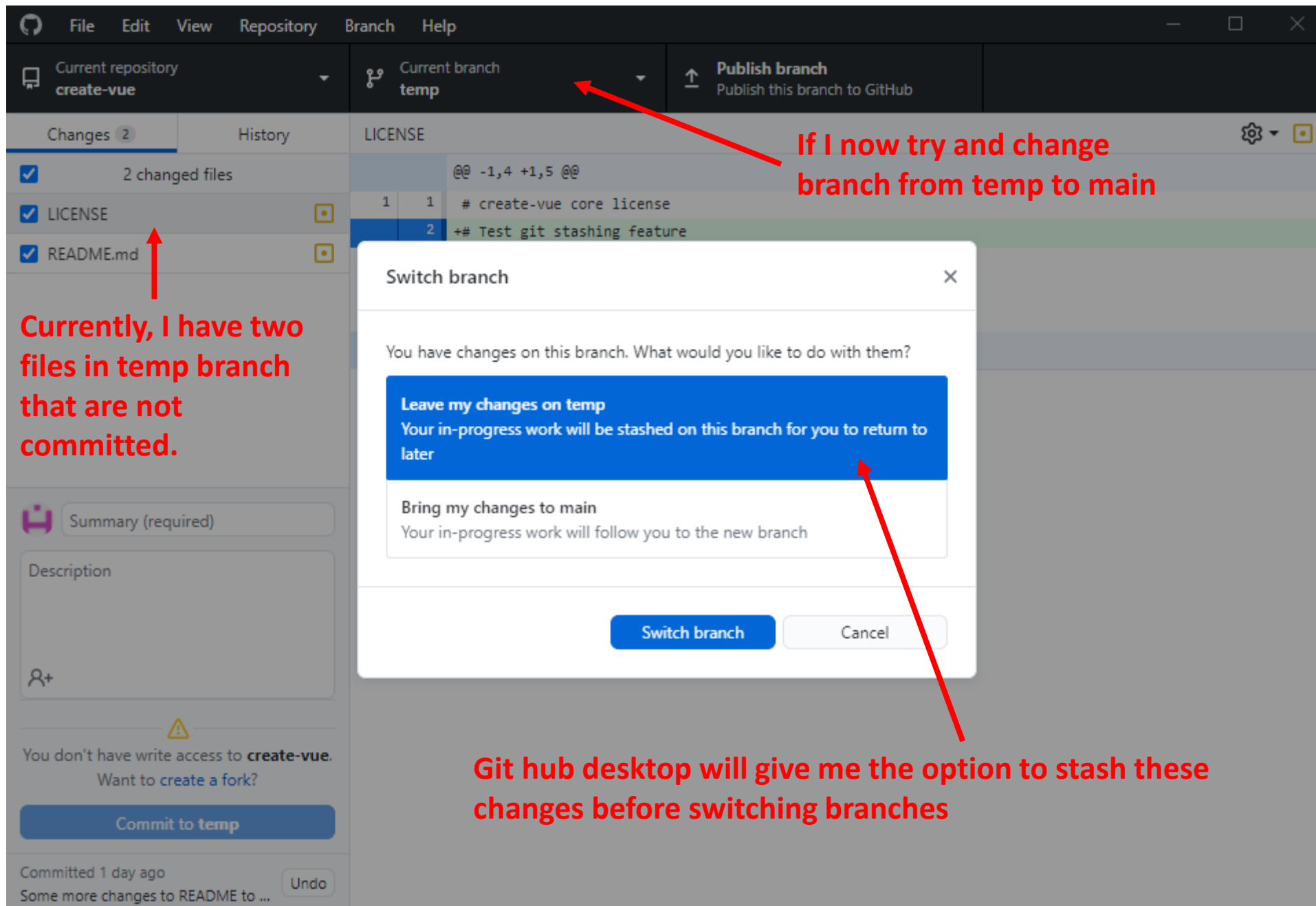
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$
```



After applying this temporary stash with a **git stash pop** command, git has deleted the stash file.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ cat .git/refs/stash
cat: .git/refs/stash: No such file or directory

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$
```



The screenshot shows the GitHub Desktop application window. The top bar includes a menu (File, Edit, View, Repository, Branch, Help) and status information: 'Current repository create-vue', 'Current branch temp', and a 'Publish branch' button. The left sidebar has tabs for 'Changes' and 'History', with 'Changes' selected. Below this, it shows '0 changed files'. At the bottom of the sidebar, there is a 'Stashed Changes' section with a 'Summary (required)' field, a 'Description' field, and a 'Commit to temp' button. The main area displays 'No local changes' and offers suggestions for next steps: 'View your stashed changes', 'Open the repository in your external editor', 'View the files of your repository in Explorer', and 'Open the repository page on GitHub in your browser'. Red annotations highlight the 'temp' branch dropdown and the 'View your stashed changes' section.

File Edit View Repository Branch Help

Current repository create-vue

Current branch temp

Publish branch Publish this branch to GitHub

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

View your stashed changes
You have 2 changes in progress that you have not yet committed.
When a stash exists, access it at the bottom of the Changes tab to the left.

View stash

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl** **Shift** **A**

Open in Visual Studio Code

View the files of your repository in Explorer
Repository menu or **Ctrl** **Shift** **F**

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or **Ctrl** **Shift** **G**

View on GitHub

Stashed Changes

Summary (required)

Description

Commit to temp

Committed 1 day ago
Some more changes to README to ... **Undo**

Switching back to temp branch.

Github desktop tells me I have stashed changes that I can view.

Changes History

0 changed files

Stashed changes

Restore Discard Restore will move your stashed files to the Changes list.

LICENSE		@@ -1,4 +1,5 @@
README.md	1	1 # create-vue core license
	2	2 +# Test git stashing feature
	3	3
	4	4 create-vue is released under the MIT license:
	5	5
	+	+

GitHub desktop allows me to restore or discard these changes.

GitHub desktop shows me the lines that have changed in each file

Stashing is useful if you have upstaged / uncommitted changes and you need to switch to another branch to work on something else but do not want to loose these changes.

Stashed Changes

Summary (required)

Description

Commit to temp

Committed 1 day ago Some more changes to README to ... Undo

Git garbage Collection

Garbage collection in git is an automatic process from time to time to clean the git repository of unused and unreachable commits. It will also place git objects in a pak file to make it smaller. Pack is zip for git!

If we look at the contents of `.git/objects` directory we see some 2hex directories which are the files and a pack directory.

The `.git/objects/pack` directory contains the pack file and an idx file.

git gc command (git garbage collection) will compress files that have not been used for a while into the pack file.

Now we see the SHA1 of the pack and idx files have changed.

And that there are less items in the `.git/objects` directory because some items have been compressed to git pack.

Also with this command git will clean unreachable commits and old reflog records.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ ls .git/objects
07/ 18/ 2e/ 3f/ 47/ 51/ 64/ 73/ 82/ ae/ cc/ d5/ e8/ info/
0e/ 1b/ 36/ 43/ 4c/ 55/ 6d/ 7c/ 9b/ b6/ d1/ db/ fd/ pack/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ ls .git/objects/pack
pack-a112fa22da85b94110262d1941a8a55d5ba854c6.idx
pack-a112fa22da85b94110262d1941a8a55d5ba854c6.pack

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ git gc
Enumerating objects: 3989, done.
Counting objects: 100% (3989/3989), done.
Delta compression using up to 8 threads
Compressing objects: 100% (906/906), done.
Writing objects: 100% (3989/3989), done.
Total 3989 (delta 2920), reused 3964 (delta 2908), pack-reused 0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ ls .git/objects/pack
pack-6914c245f2a6faa1ae09251599cee97015006043.idx
pack-6914c245f2a6faa1ae09251599cee97015006043.pack

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ ls .git/objects
43/ 64/ 6d/ 73/ fd/ info/ pack/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (temp)
$ |
```


Squashing of commits at GitHub

If we look at this particular **git log** message we can see that there are two comments to the 1 commit. This is because it was made by combining two commits.

This is because the administrator is using rebasing with squashing to keep the repository clean of many development merges and branches so that the tree is very linear.

```
commit 81b1f4783374e0f1b017553962bc8bffa98b08b59
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date:   Wed Jan 19 12:20:56 2022 +0800
```

```
feat!: add Vitest template (#29)
```

```
commit bb14f5cd8fdd825bb82bd5e54bb5a3fe5b2d0526
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date:   Mon Jan 17 16:11:08 2022 +0800
```

```
chore: update the link for take over mode
```

```
The new link contains instructions with screenshots, which is more
user-friendly.
```

```
commit 16bba79a6aabe53d967f6261ce8c8f8abccaf61f
Author: Haoqun Jiang <haoqunjiang@gmail.com>
Date:   Mon Jan 17 14:24:16 2022 +0800
```


Lets create a public testing repository in github.com

Owner * EL-EL-BCN / Repository name * rebasing-with-squashing ✓

Great repository names are short and memorable. Need inspiration? How about [furry-waffle?](#)

Description (optional)
Let's make hostory clean by creating just single commit for every feature branch

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

And copy the URL for cloning

main 1 branch 0 tags

EL-EL-BCN Initial commit

README.md Initial commit

README.md

rebasing-with-squashing

Go to file

Add file

Code

Clone

HTTPS SSH GitHub CLI

https://github.com/EL-EL-BCN/rebasing-witl ✓

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Copied!

About

Let's make hostory clean by creating just single commit for every feature branch

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Three files were created and staged and committed to the feature1 branch of the local repository then pushed to the remote repository.

In github.com a pull request was opened for the changes in feature1 branch.

The screenshot shows a GitHub pull request page for the repository 'EL-EL-BCN/rebasing-with-squashing'. The pull request is titled '[Feature1] branch with some files' and is comparing the 'feature1' branch against the 'main' branch. The status is 'Able to merge'. The description of the pull request states: 'Several files created in the feature1 branch' and lists 'file1.txt', 'file2.txt', and 'file3.txt'. The 'Create pull request' button is visible at the bottom right. The right sidebar shows 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development'.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ git clone https://github.com/EL-EL-BCN/rebasing-with-squashing.git
Cloning into 'rebasing-with-squashing'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd rebasing-with-squashing/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (main)
$ ls
README.md

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (main)
$ git checkout -b "feature1"
Switched to a new branch 'feature1'

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ touch file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git add file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git commit -m "first commit in feature1 branch"
[feature1 f1e1477] first commit in feature1 branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ touch file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git add file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git commit -m "second commit in feature1 branch"
[feature1 b77a702] second commit in feature1 branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file2.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ touch file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git add file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git commit -m "third file in feature1 branch"
[feature1 1e060bb] third file in feature1 branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (feature1)
$ git push -u origin feature1
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 665 bytes | 665.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'feature1' on GitHub by visiting:
remote:   https://github.com/EL-EL-BCN/rebasing-with-squashing/pull/new/feature1
remote:
To https://github.com/EL-EL-BCN/rebasing-with-squashing.git
 * [new branch]      feature1 -> feature1
branch 'feature1' set up to track 'origin/feature1'.
```

github.com/EL-EL-BCN/rebasing-with-squashing/pull/1

[Feature1] branch with some files #1
EL-EL-BCN wants to merge 3 commits into `main` from `feature1`

Add more commits by pushing to the `feature1` branch on EL-EL-BCN/rebasing-with-squashing.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request
You can also open this in GitHub Desktop or view command line instructions.

✓ Create a merge commit
All commits from this branch will be added to the base branch via a merge commit.

Squash and merge
The 3 commits from this branch will be combined into one commit in the base branch.

Rebase and merge
The 3 commits from this branch will be rebased and added to the base branch.

Close pull request Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications
Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

Now when we merge the pull request we can select the dropdown to chose additional features of merge. This time we will squash and merge which will combine the three commits from this branch into one commit. Note that it will also prompt us to confirm the quash and merge before actioning it.

The screenshot shows the GitHub interface for the repository `EL-EL-BCN/rebasing-with-squashing`. The main branch is `main`. The commit history for `main` is displayed, showing two commits. The top commit is titled `[Feature1] branch with some files (#1)` and is marked as `Verified` with a commit hash of `b5853e4`. The bottom commit is titled `Initial commit` and is also marked as `Verified` with a commit hash of `3775736`. A red arrow points to the top commit, indicating that it represents three squashed commits.

Commits on Mar 12, 2022

- `[Feature1] branch with some files (#1)` `Verified` `b5853e4`
 - * first commit in feature1 branch
 - * second commit in feature1 branch
 - * third file in feature1 branch
- `Initial commit` `Verified` `3775736`
 - EL-EL-BCN committed 2 minutes ago

`Newer` `Older`

Note that after the merge the main branch only has two commits because we can see the ultimate commit is actually three commits squashed into 1.

If we **git checkout main** branch and perform a **git pull**.

The **git log** only shows 2 commits with the ultimate commit being squashed 3 commits.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 726 bytes | 181.00 KiB/s, done.
From https://github.com/EL-EL-BCN/rebasing-with-squashing
 3775736..b5853e4  main      -> origin/main
Updating 3775736..b5853e4
Fast-forward
 file1.txt | 0
 file2.txt | 0
 file3.txt | 0
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (main)
$ git log
commit b5853e41e233a9a6b8c4d900658e726ab0092d5a (HEAD -> main, origin/main, origin/HEAD)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 12 23:04:49 2022 +0100

    [Feature1] branch with some files (#1)

    * first commit in feature1 branch

    * second commit in feature1 branch

    * third file in feature1 branch

commit 3775736c94bf545de78091a29d951f1bf0d80da1
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 12 22:43:56 2022 +0100

    Initial commit

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing (main)
$
```

File4.txt, file5.txt and file6.txt were staged and committed in feature2 branch.

In git log we see the 3 commits for the three files added in feature2 branch.

To perform **git rebasing** with squashing we need to use the **-i** option (interactive) and include the SHA1 of the commit before the three that we want to squash.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$ ls -la
total 13
drwxr-xr-x 1 ellio 197609  0 Mar 12 23:20 ./
drwxr-xr-x 1 ellio 197609  0 Mar 12 22:46 ../
drwxr-xr-x 1 ellio 197609  0 Mar 12 23:21 .git/
-rw-r--r-- 1 ellio 197609 110 Mar 12 22:46 README.md
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:09 file1.txt
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:09 file2.txt
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:09 file3.txt
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:18 file4.txt
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:19 file5.txt
-rw-r--r-- 1 ellio 197609  0 Mar 12 23:20 file6.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$ git lg
* e110e15 - (HEAD -> feature2) sixth file in feature2 branch (2 minutes ago)<EL-EL-B
CN>
* ea192d0 - fith file in feaure2 branch (3 minutes ago)<EL-EL-BCN>
* 0269a08 - forth file in feature2 branch (4 minutes ago)<EL-EL-BCN>
* b5853e4 - (origin/main, origin/HEAD, main) [Feature1] branch with some files (#1)
(19 minutes ago)<EL-EL-BCN>
* 3775736 - Initial commit (40 minutes ago)<EL-EL-BCN>

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$ git rebase -i b5853e4|
```

git rebase -i <SHA1> command will generate this file output which contains the 3 commits that we want to quash into 1.

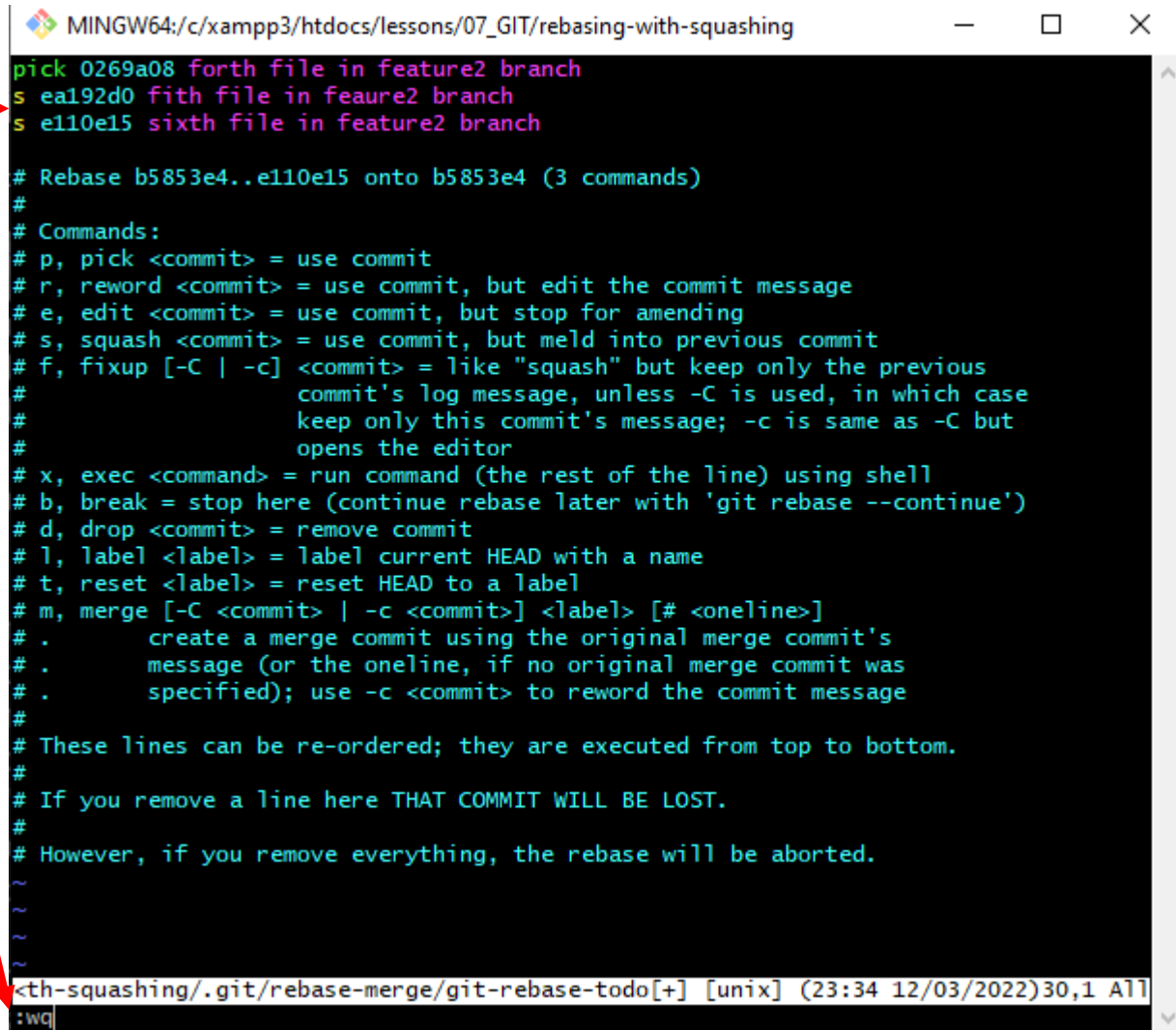
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
pick 0269a08 forth file in feature2 branch
pick ea192d0 fith file in feaure2 branch
pick e110e15 sixth file in feature2 branch

# Rebase b5853e4..e110e15 onto b5853e4 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .      create a merge commit using the original merge commit's
# .      message (or the oneline, if no original merge commit was
# .      specified); use -c <commit> to reword the commit message
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
~
<g-with-squashing/.git/rebase-merge/git-rebase-todo [unix] (23:34 12/03/2022)1,1 All
```

In the file output we need to pick one commit that we want to represent the three then change the other two from pick to **squash** or just **s**.

To edit this file we can use **control + i** (insert)

When we have made the changes we can use **escape** then **:wq** to write and quit.



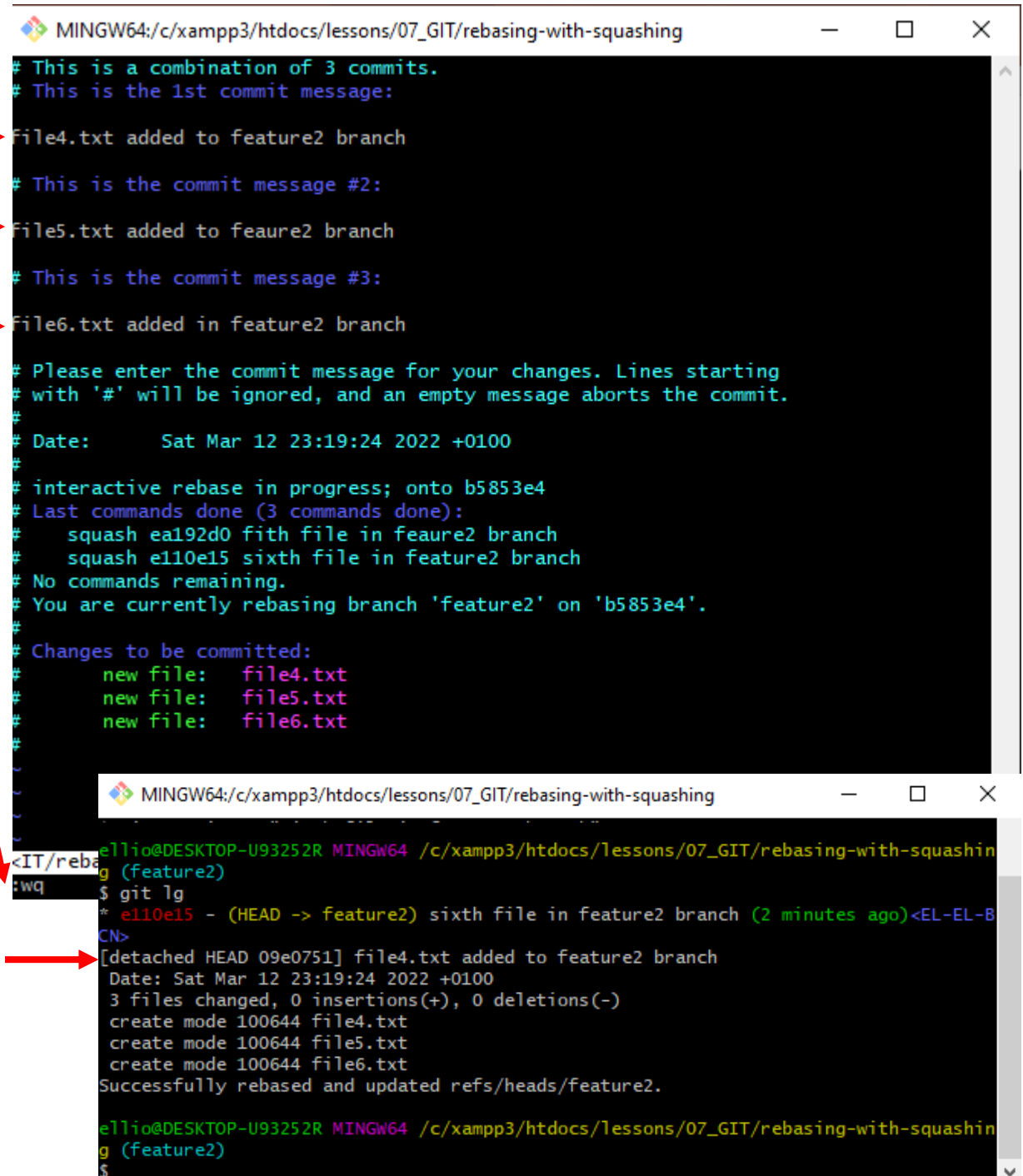
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
pick 0269a08 forth file in feature2 branch
s ea192d0 fith file in feaure2 branch
s e110e15 sixth file in feature2 branch

# Rebase b5853e4..e110e15 onto b5853e4 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified); use -c <commit> to reword the commit message
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
~
~
~
<th-squashing/.git/rebase-merge/git-rebase-todo[+] [unix] (23:34 12/03/2022)30,1 All
:wq
```


We get a conformation file output where we can individually modify the commit comments.

[illegible]

In this case I have changed the comments of the three commits then using **escape** then **:wq** I can write and quit this file.



```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashing
# This is a combination of 3 commits.
# This is the 1st commit message:
file4.txt added to feature2 branch
# This is the commit message #2:
file5.txt added to feaure2 branch
# This is the commit message #3:
file6.txt added in feature2 branch

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sat Mar 12 23:19:24 2022 +0100
#
# interactive rebase in progress; onto b5853e4
# Last commands done (3 commands done):
#   squash ea192d0 fith file in feaure2 branch
#   squash e110e15 sixth file in feature2 branch
# No commands remaining.
# You are currently rebasing branch 'feature2' on 'b5853e4'.
#
# Changes to be committed:
#   new file:   file4.txt
#   new file:   file5.txt
#   new file:   file6.txt
#
<IT/reba
:wq

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$ git lg
* e110e15 - (HEAD -> feature2) sixth file in feature2 branch (2 minutes ago)
CN>
[detached HEAD 09e0751] file4.txt added to feature2 branch
Date: Sat Mar 12 23:19:24 2022 +0100
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4.txt
create mode 100644 file5.txt
create mode 100644 file6.txt
Successfully rebased and updated refs/heads/feature2.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$
```

Git executes the contents of these files creating one commit to replace the three.

Git log shows us the squashed commit with three comments.

```
g (feature2)
$ git log
commit 09e0751bb45c84330fc656ad966d54cc41a7cb3b (HEAD -> feature2)
Author: EL-EL-BCN <elliottbcn@gmail.com>
Date: Sat Mar 12 23:19:24 2022 +0100

    file4.txt added to feature2 branch

    file5.txt added to feaure2 branch

    file6.txt added in feature2 branch

commit b5853e41e233a9a6b8c4d900658e726ab0092d5a (origin/main, origin/HEAD, main)
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 12 23:04:49 2022 +0100

    [Feature1] branch with some files (#1)

    * first commit in feature1 branch

    * second commit in feature1 branch

    * third file in feature1 branch

commit 3775736c94bf545de78091a29d951f1bf0d80da1
Author: EL-EL-BCN <101018935+EL-EL-BCN@users.noreply.github.com>
Date: Sat Mar 12 22:43:56 2022 +0100

    Initial commit
```

I can now **git checkout main** branch

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (feature2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

And **git merge -v feature2** branch into the main branch. Note that git has performed fast-forwarding merge as per rebasing process.

```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/rebasing-with-squashin
g (main)
$ git merge -v feature2
Updating b5853e4..09e0751
Fast-forward
 file4.txt | 0
 file5.txt | 0
 file6.txt | 0
3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file4.txt
 create mode 100644 file5.txt
 create mode 100644 file6.txt
```

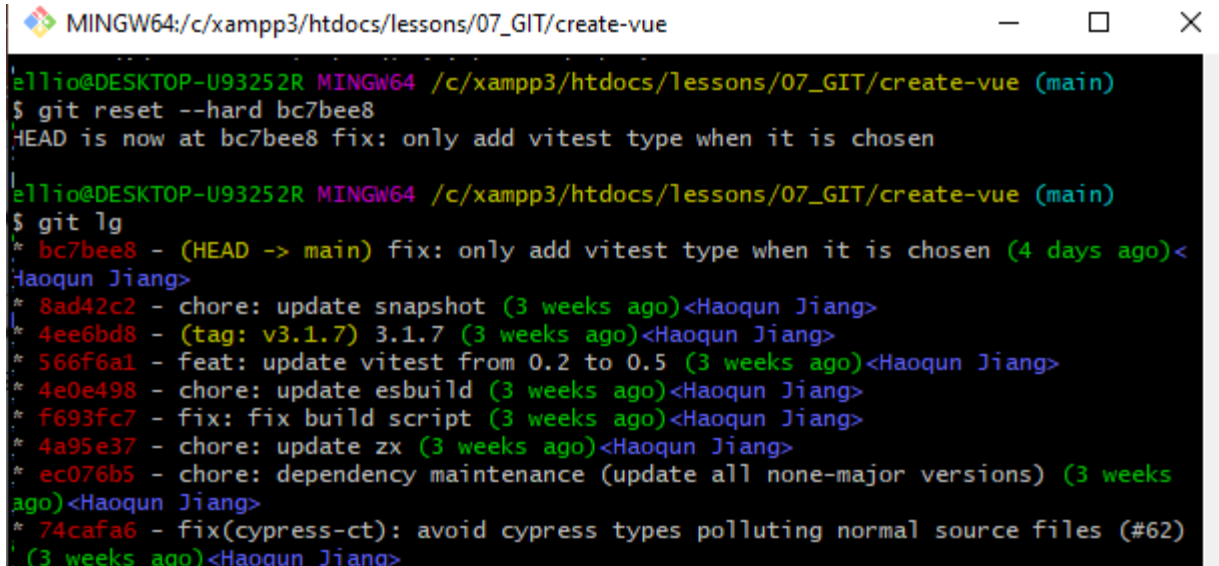
Finally I can git push this to the remote repository.

Git Development workflow

A repository usually contains several public branches such as dev, release and main and these branches are usually selected as protected so that owner can only accept and perform pull requests.

Often you will need to be up to date with public branches while working on new features in private branches.

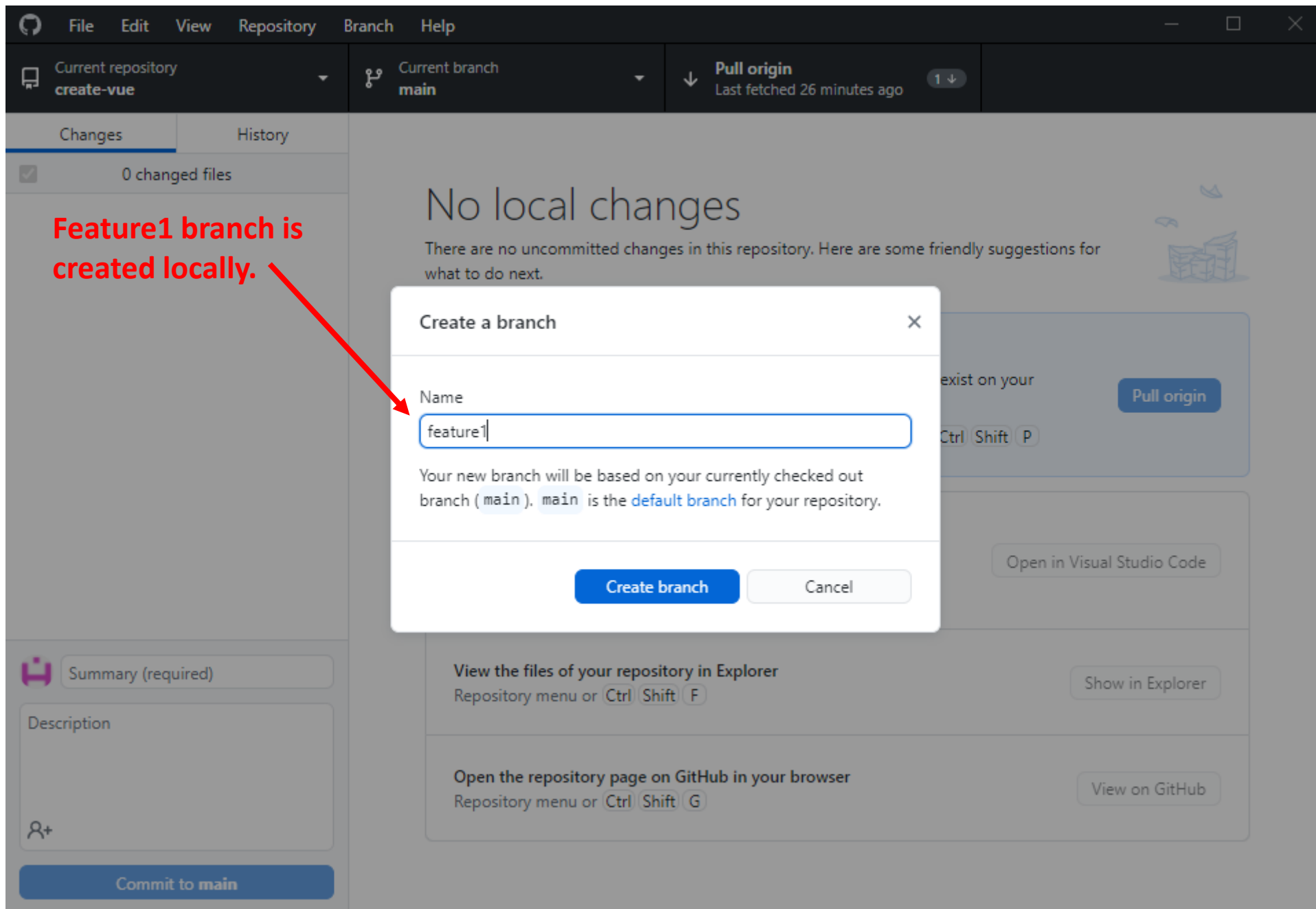
I have performed a hard reset of the local repository to remove all the changes that I made previously.

A terminal window titled 'MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue' with standard window controls. The terminal shows a user performing a hard reset and then listing the commit history. The output of 'git reset --hard bc7bee8' shows the HEAD is now at bc7bee8 with a commit message 'fix: only add vitest type when it is chosen'. The output of 'git lg' shows a list of commits from the HEAD back to the root of the repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/create-vue

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git reset --hard bc7bee8
HEAD is now at bc7bee8 fix: only add vitest type when it is chosen

e1lio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/create-vue (main)
$ git lg
* bc7bee8 - (HEAD -> main) fix: only add vitest type when it is chosen (4 days ago) <Haoqun Jiang>
* 8ad42c2 - chore: update snapshot (3 weeks ago) <Haoqun Jiang>
* 4ee6bd8 - (tag: v3.1.7) 3.1.7 (3 weeks ago) <Haoqun Jiang>
* 566f6a1 - feat: update vitest from 0.2 to 0.5 (3 weeks ago) <Haoqun Jiang>
* 4e0e498 - chore: update esbuild (3 weeks ago) <Haoqun Jiang>
* f693fc7 - fix: fix build script (3 weeks ago) <Haoqun Jiang>
* 4a95e37 - chore: update zx (3 weeks ago) <Haoqun Jiang>
* ec076b5 - chore: dependency maintenance (update all none-major versions) (3 weeks ago) <Haoqun Jiang>
* 74cafa6 - fix(cypress-ct): avoid cypress types polluting normal source files (#62) (3 weeks ago) <Haoqun Jiang>
```



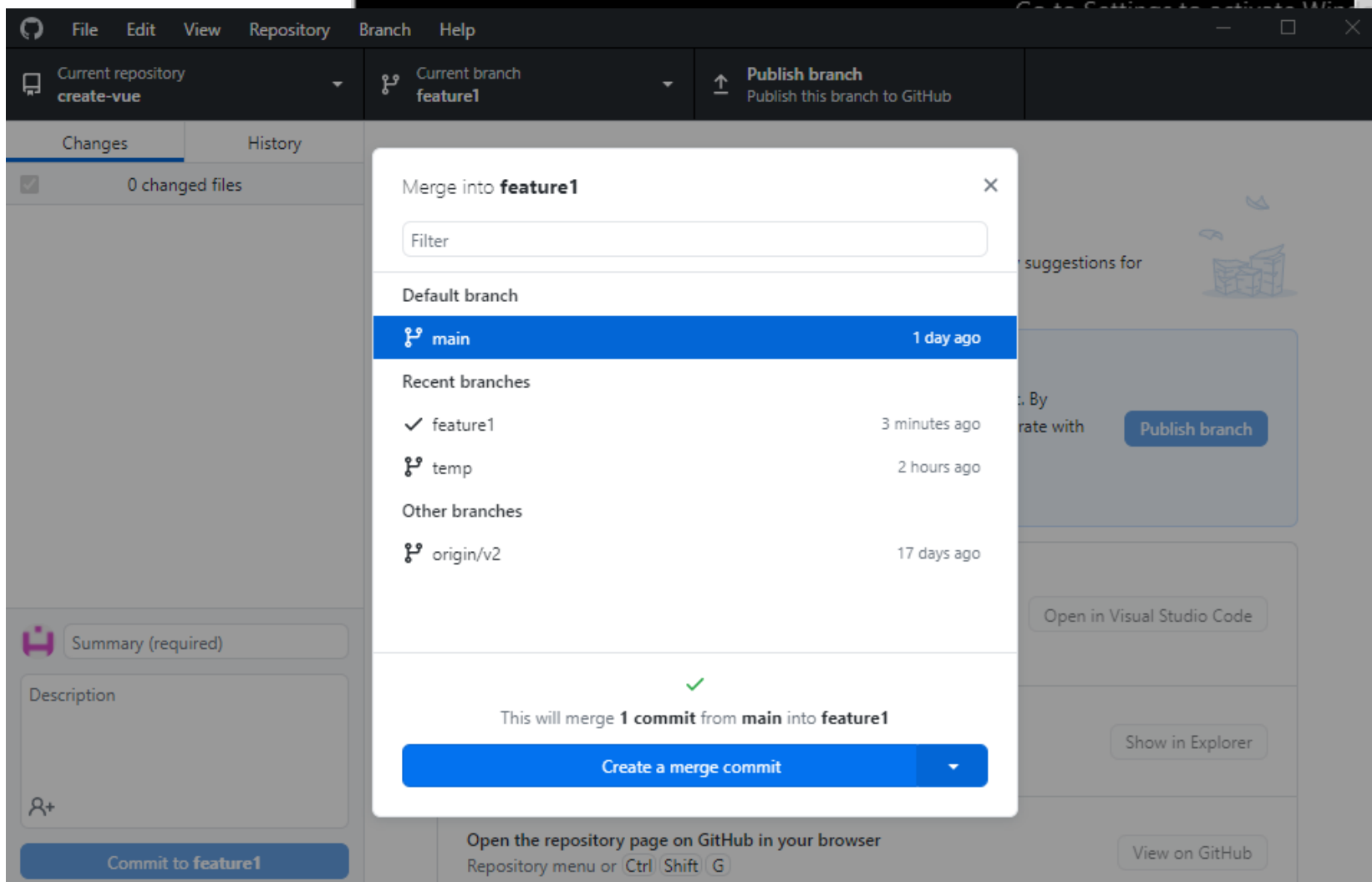
I have stages and committed file1.txt to feature1 branch.

To keep up to date I can merge main branch into feature1 branch.

```
e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (feature1)
$ touch file1.txt

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (feature1)
$ git add file1.txt

e1lio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/create-vue (feature1)
$ git commit -m "file1.txt added to feature1 branch"
[feature1 cc8a319] file1.txt added to feature1 branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt
```



Summary of Shell Commands

git lg	<i>Summary of git log with pretty colours</i>
git log --oneline	<i>Outputs git log with one commit per line</i>
git log --graph	<i>Outputs git log showing parents of each commit</i>
git log --graph --oneline	<i>Outputs git log showing parents of each commit on one line</i>
git log --stat	<i>Outputs git log showing number of lines modified in each file</i>
git log -p	<i>Outputs git log showing exactly what was changed in each file</i>
git log -4 --graph	<i>Outputs last 4 git log commits showing parents</i>
git log -4 --oneline	<i>Outputs last 4 git log commits on one line</i>
git log -4 --graph --oneline	<i>Outputs last 4 git log commits showing parents on one line</i>
git shortlog -n	<i>Outputs number of commits sorted by author alphabetically</i>
git shortlog -n -s	<i>Outputs author, number of commits sorted by total commits</i>
git shortlog -n -s -e	<i>Outputs author, email, number of commits sorted by total commits</i>
git log --author <name>	<i>Filter log by author name</i>
git log --grep <word>	<i>A wild card search of git log to find commits that contain word</i>
git log --pretty=format:"%H"	<i>Show only SHA1 hashes of all commits in the git log</i>
git log --pretty=format:"%cn %H"	<i>Show only author and SHA1 hashes of all commits in the git log</i>
git log --pretty=format:"%cn %h"	<i>Show only author and short SHA1 hashes of all commits in git log</i>

git log –pretty=format:”author of commit: %cn SHA1 hash: %h”	<i>Pretty author & SHA1 text string</i>
git log –pretty=format:”author: %cn; SHA1 hash: %h; Date: %cd”	<i>Pretty author; SHA1; Date</i>
git log --merges --oneline	<i>Show only merge commits</i>
git log –no-merges --oneline	<i>Show all minus merge commits</i>
git reset <SHA1>	<i>Perform mixed reset to commit (only repository)</i>
git reset --soft<SHA1>	<i>Perform soft reset to commit (repository & staging)</i>
git reset --hard<SHA1>	<i>Perform hard reset to commit (repository, staging & working)</i>
git reset HEAD-n	<i>Reset head to n commits backwards (mixed reset)</i>
git commit --amend -m “message”	<i>Overwrite last commit with new commit</i>
git commit --amend – author=“username <email>”	<i>Amend author of last commit</i>
git cherry-pick <SHA1>	<i>Cherry-pick a commit from another branch or detached head and commit it to the current branch</i>
git cherry-pick --no-commit <SHA1>	<i>Cherry-pick a commit from another branch and stage it into current branch</i>
git reflog	<i>Show history of all operations on the local repository</i>

git stash

Make a temporary save commit (stash)

git stash pop

Apply changes made in stash

git gc

Manual clean up unreachable commits and old reflog records and compress to pack

git rebase -i <SHA1>

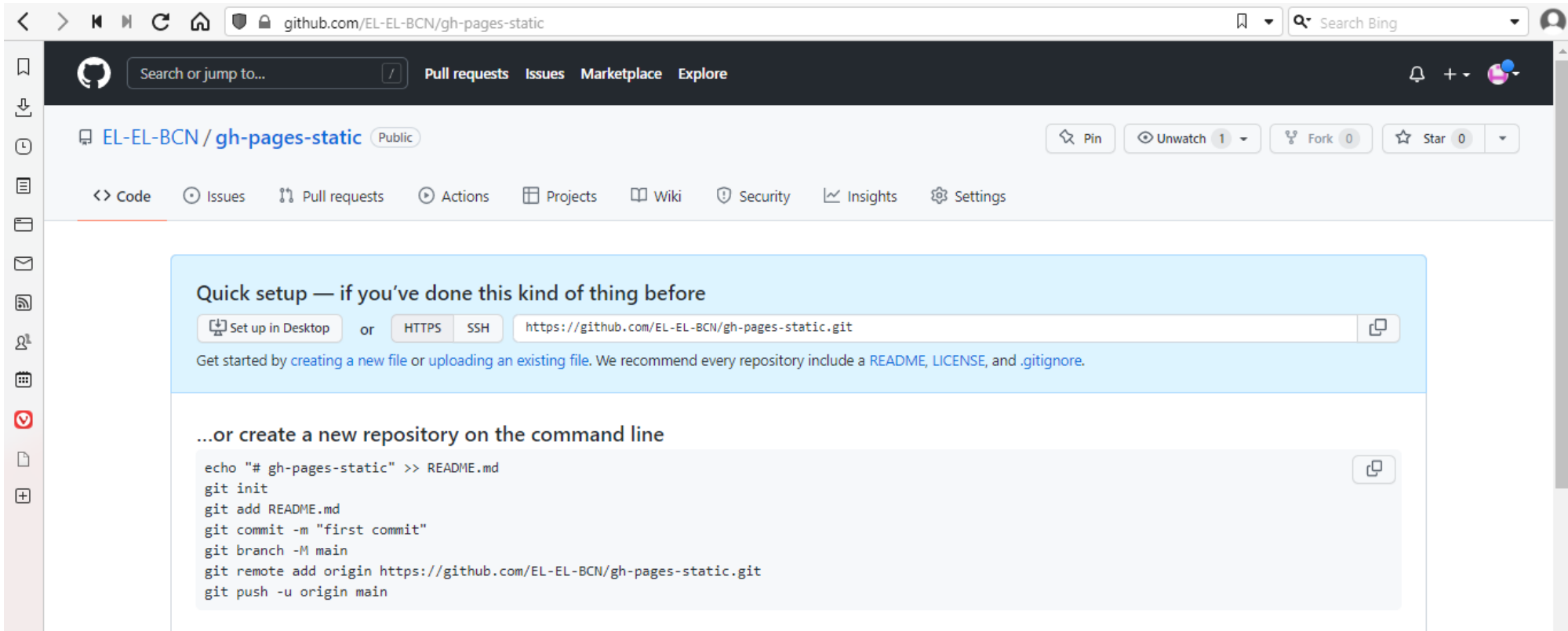
Perform manual rebasing with squashing of all commits after the SHA1 specified.

GIT HUB PAGES

GitHub pages

A git hub page is a way to easily host a website with a GitHub sub domain. The website can be created from the GitHub repository.

A standard repository is created in GitHub with the following settings: Repo name – gh-pages-static, Repo description – Another test static page, Repo is public with no README.me file.



The screenshot shows the GitHub web interface for the repository `EL-EL-BCN / gh-pages-static`. The repository is public and has 1 watcher, 0 forks, and 0 stars. The main content area displays a "Quick setup" section for users who have done this before, providing a link to the repository and instructions to get started by creating a new file or uploading an existing file. Below this, there is a section titled "...or create a new repository on the command line" which includes a code block with the following commands:

```
echo "# gh-pages-static" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/EL-EL-BCN/gh-pages-static.git
git push -u origin main
```

The repository is created locally with the same name and a basic html file added.

The html file is staged and added then the changes are pushed to the remote repository which is manually configured.

Git may have cached credentials set locally. So returns a 403 error when pushing. This can be solved with **git config credential helper ""**

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/gh-pages-static
e11io@DESKTOP-U93252R MINGW64 ~
$ cd /c/xampp3/htdocs/lessons/07_GIT/

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ mkdir gh-pages-static

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd gh-pages-static/

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/gh-pages-static/.git/

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ echo "Another Static GitHub Page" > index.html

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ cat index.html
Another Static GitHub Page

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git config user.name EL-EL-BCN

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git config user.email elliottbcn@gmail.com

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git commit -m "First commit with index.html file"
[master (root-commit) d928016] First commit with index.html file
1 file changed, 1 insertion(+)
create mode 100644 index.html

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git remote add origin https://github.com/EL-EL-BCN/gh-pages-static.git

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 258 bytes | 258.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EL-EL-BCN/gh-pages-static.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/gh-pages-static (master)
$ |
```

github.com/EL-EL-BCN/gh-pages-static

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / gh-pages-static Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

About

Another test static website

0 stars 1 watching 0 forks

Releases

No releases published

EL-EL-BCN First commit with index.html file d928016 9 minutes ago 1 commit

index.html First commit with index.html file 9 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

In GitHub remote repository we now see the index.html file.

github.com/EL-EL-BCN/gh-pages-static/settings

In GitHub remote repository choose settings then scroll down to the bottom to GitHub pages

Archives

When creating source code archives, you can choose to include files stored using Git LFS in the archive.

☐ Include Git LFS objects in archives
Git LFS usage in archives is billed at the same rate as usage with the client.

GitHub Pages

Pages settings now has its own dedicated tab! [Check it out here!](#)

github.com/EL-EL-BCN/gh-pages-static/settings/pages

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / gh-pages-static Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None Save

Select branch

Select branch

master

✓ None

Try risk-free for 30 days using a GitHub Enterprise organization, or [learn more about changing the visibility of your GitHub Pages site.](#)

From the None button, we select the branch that we want to use for the website. In this case master and select save.

github.com/EL-EL-BCN/gh-pages-static/settings/pages

Search or jump to... Pull requests Issues Marketplace Explore

GitHub Pages source saved.

EL-EL-BCN / gh-pages-static Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

ⓘ Your site is ready to be published at <https://el-el-bcn.github.io/gh-pages-static/>

Source
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

Branch: master / (root) Save

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Actions
Webhooks

Now we are given a URL which is similar to our GitHub repository.

Facebook YouTube The Complete Git Guide: U Pages https://el-el-bcn.github.io/gh-pages-static/

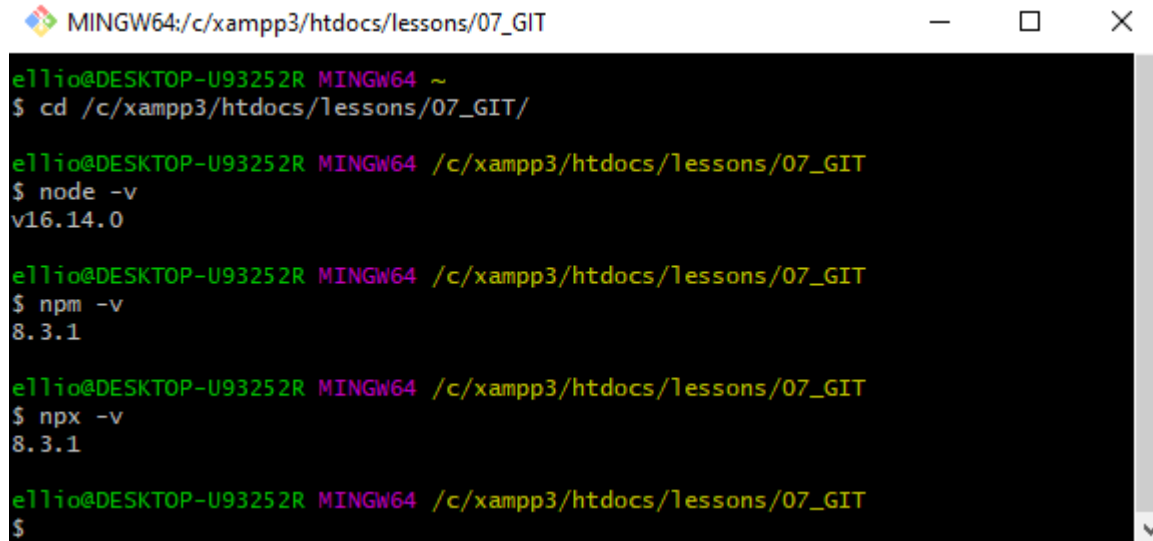
el-el-bcn.github.io/gh-pages-static/

Another Static GitHub Page

**This URL is globally reachable from explorer so is a hosted website!
Note that it can take up to 2 minutes before the URL is live and reachable.**

Creating basic React application

Download latest version of node js from <https://nodejs.org/en/>

A terminal window titled 'MINGW64:/c/xampp3/htdocs/lessons/07_GIT' with standard window controls. The terminal shows a series of commands and their outputs: 'cd /c/xampp3/htdocs/lessons/07_GIT/' is followed by a new prompt; 'node -v' returns 'v16.14.0'; 'npm -v' returns '8.3.1'; 'npx -v' returns '8.3.1'; and the final prompt '\$' is shown without further output.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT
ellio@DESKTOP-U93252R MINGW64 ~
$ cd /c/xampp3/htdocs/lessons/07_GIT/
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ node -v
v16.14.0
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ npm -v
8.3.1
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ npx -v
8.3.1
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$
```

Select add chocolatey tools to add all the node js dependencies automatically.

From our habitual git folder I can check the availability of Node js and the version, NPM and NPX.

I can cd into the new directory and run npm.

It will **npm start** and tell me that I can view the page locally in the web browser and also open a browser to that page.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd gh-pages-react

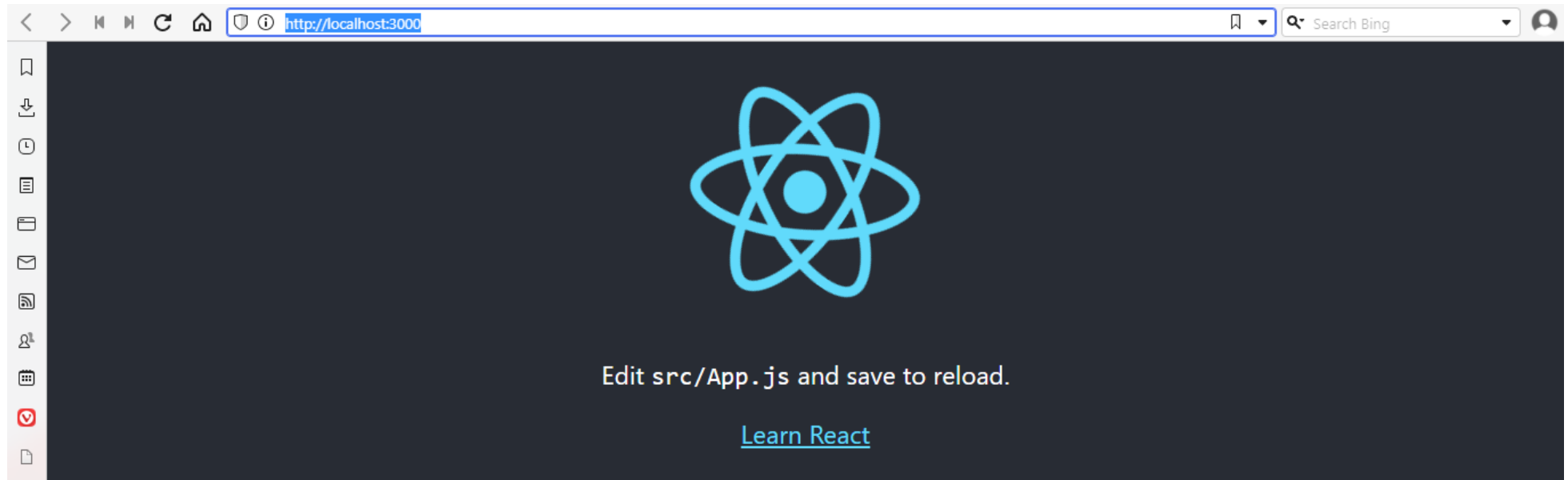
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
$ npm start

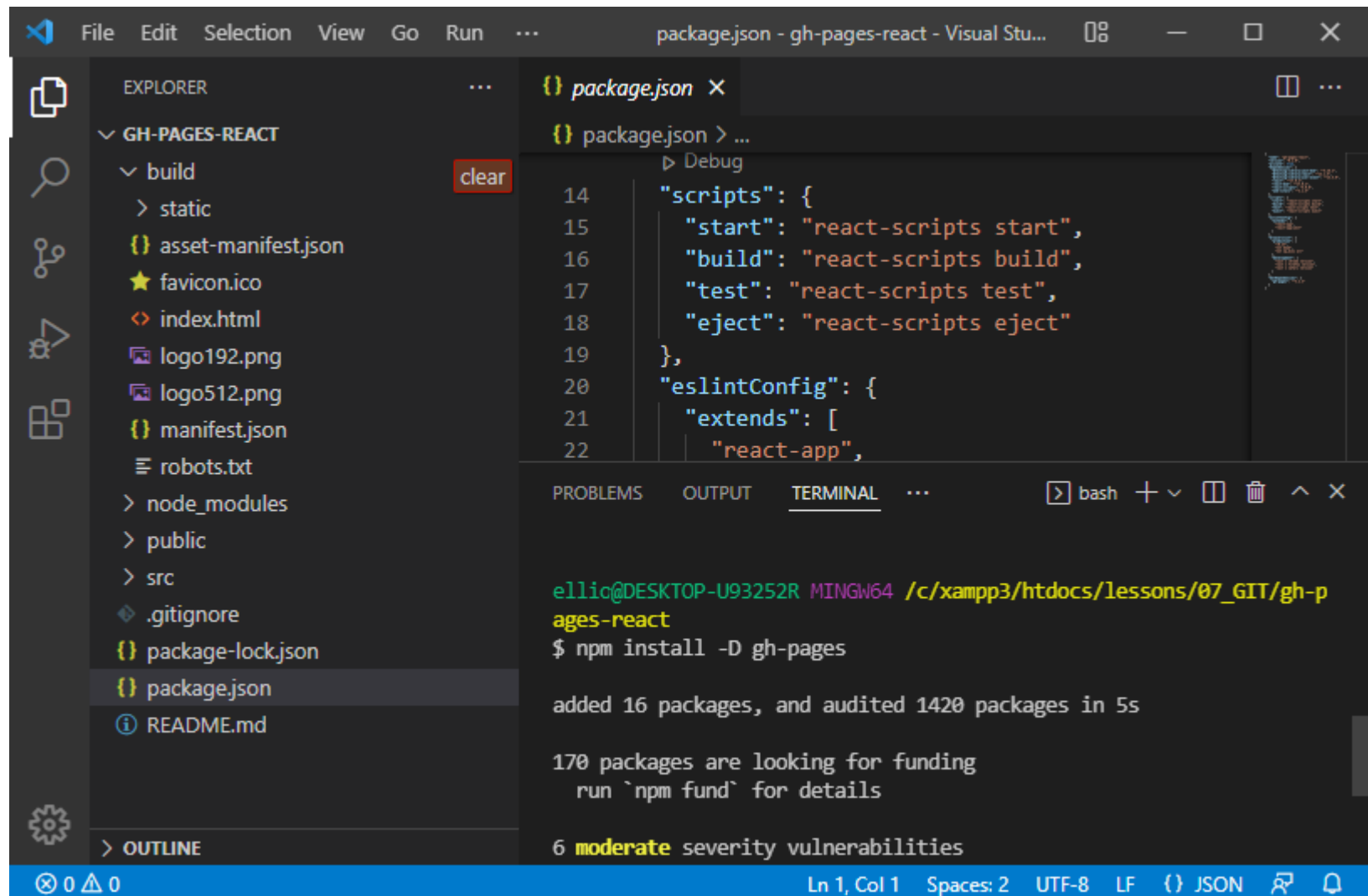
> gh-pages-react@0.1.0 start
> react-scripts start

(node:9316) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewar
```

You can now view gh-pages-react in the browser.

Local:	<code>http://localhost:3000</code>
On Your Network:	<code>http://192.168.1.152:3000</code>





I need to add another npm package called gh-pages from <https://www.npmjs.com/package/gh-pages> with a `npm install -D gh-pages` command

github.com/EL-EL-BCN/gh-pages-react/settings/pages

Search or jump to... Pull requests Issues Marketplace Explore

GitHub Pages source saved.

EL-EL-BCN / gh-pages-react Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://el-el-bcn.github.io/gh-pages-react/>

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

Branch: master / (root) Save

Theme Chooser

Copy the URL

Under the settings tab for this repository we create a GitHub page and set the master branch to root and save.

Homepage is added manually to package.json file

The screenshot shows the Visual Studio Code editor with the `package.json` file open. The Explorer panel on the left shows the project structure for `GH-PAGES-REACT`, including a `build` directory and various files like `asset-manifest.json`, `favicon.ico`, `index.html`, `logo192.png`, `logo512.png`, `manifest.json`, `robots.txt`, `node_modules`, `public`, `src`, `.gitignore`, `package-lock.json`, `package.json`, and `README.md`. The `package.json` file is selected and highlighted in the Explorer panel. The main editor area shows the content of `package.json` with the following structure:

```
{
  "name": "gh-pages-react",
  "version": "0.1.0",
  "private": true,
  "homepage": "https://el-el-bcn.github.io/gh-pages-react/",
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.2",
    "@testing-library/react": "^12.1.4",
    "@testing-library/user-event": "^13.5.0",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "5.0.0",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
    "predeploy": "react-scripts build",
    "deploy": "gh-pages -d build"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  }
}
```

Two red arrows are present: one pointing to the `homepage` field in the `dependencies` section, and another pointing to the `scripts` section.

Predeploy and deploy scripts are added manually to the scripts section of package.json

We can initialise git and stage and commit the project locally.

Next we set the remote repository and push to it.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
drwxr-xr-x 1 ellio 197609    0 Mar 14 03:36 src/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/gh-pages-react/.git/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git remote add origin https://github.com/EL-EL-BCN/gh-pages-react.git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git config user.name EL-EL-BCN

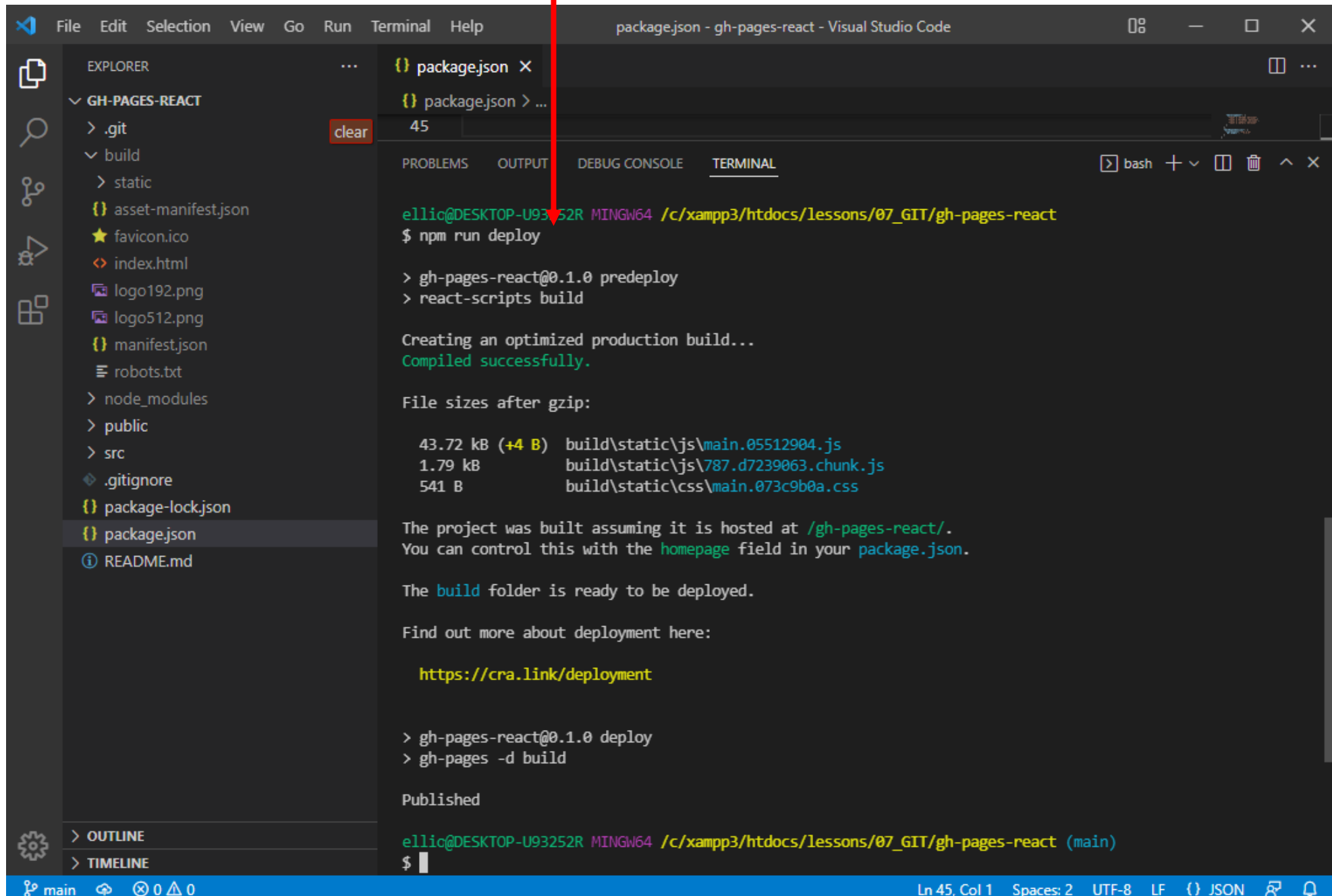
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git config user.email elliotbnc@gmail.com

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git commit -m "Initial react app"
[master (root-commit) 34e09f8] Initial react app
18 files changed, 28160 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 public/favicon.ico
create mode 100644 public/index.html
create mode 100644 public/logo192.png
create mode 100644 public/logo512.png
create mode 100644 public/manifest.json
create mode 100644 public/robots.txt
create mode 100644 src/App.css
create mode 100644 src/App.js
create mode 100644 src/App.test.js
create mode 100644 src/index.css
create mode 100644 src/index.js
create mode 100644 src/logo.svg
create mode 100644 src/reportWebVitals.js
create mode 100644 src/setupTests.js

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git push -u origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (22/22), 285.74 KiB | 10.99 MiB/s, done.
Total 22 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EL-EL-BCN/gh-pages-react.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ |
```

I now run the deploy script in terminal



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the file structure of the 'GH-PAGES-REACT' project. The file list includes .git, build, static, asset-manifest.json, favicon.ico, index.html, logo192.png, logo512.png, manifest.json, robots.txt, node_modules, public, src, .gitignore, package-lock.json, package.json, and README.md. The package.json file is selected. The Terminal view on the right shows the execution of the 'npm run deploy' command. The output indicates a successful build and deployment to GitHub Pages. A red vertical line highlights the 'package.json' file in the Explorer and the 'package.json' tab in the editor.

```
package.json - gh-pages-react - Visual Studio Code

EXPLORER
  GH-PAGES-REACT
    .git
    build
    static
    {} asset-manifest.json
    ★ favicon.ico
    <> index.html
    🖼 logo192.png
    🖼 logo512.png
    {} manifest.json
    📄 robots.txt
    > node_modules
    > public
    > src
    💎 .gitignore
    {} package-lock.json
    {} package.json
    ⓘ README.md

package.json > ...
45

clear

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + v [ ] [ ] ^ x

ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
$ npm run deploy

> gh-pages-react@0.1.0 predeploy
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  43.72 kB (+4 B) build/static/js/main.05512904.js
  1.79 kB      build/static/js/787.d7239063.chunk.js
  541 B       build/static/css/main.073c9b0a.css

The project was built assuming it is hosted at /gh-pages-react/.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.

Find out more about deployment here:

  https://cra.link/deployment

> gh-pages-react@0.1.0 deploy
> gh-pages -d build

Published

ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (main)
$
```

main 0 0 0 Ln 45, Col 1 Spaces: 2 UTF-8 LF {} JSON

Configure SSH key

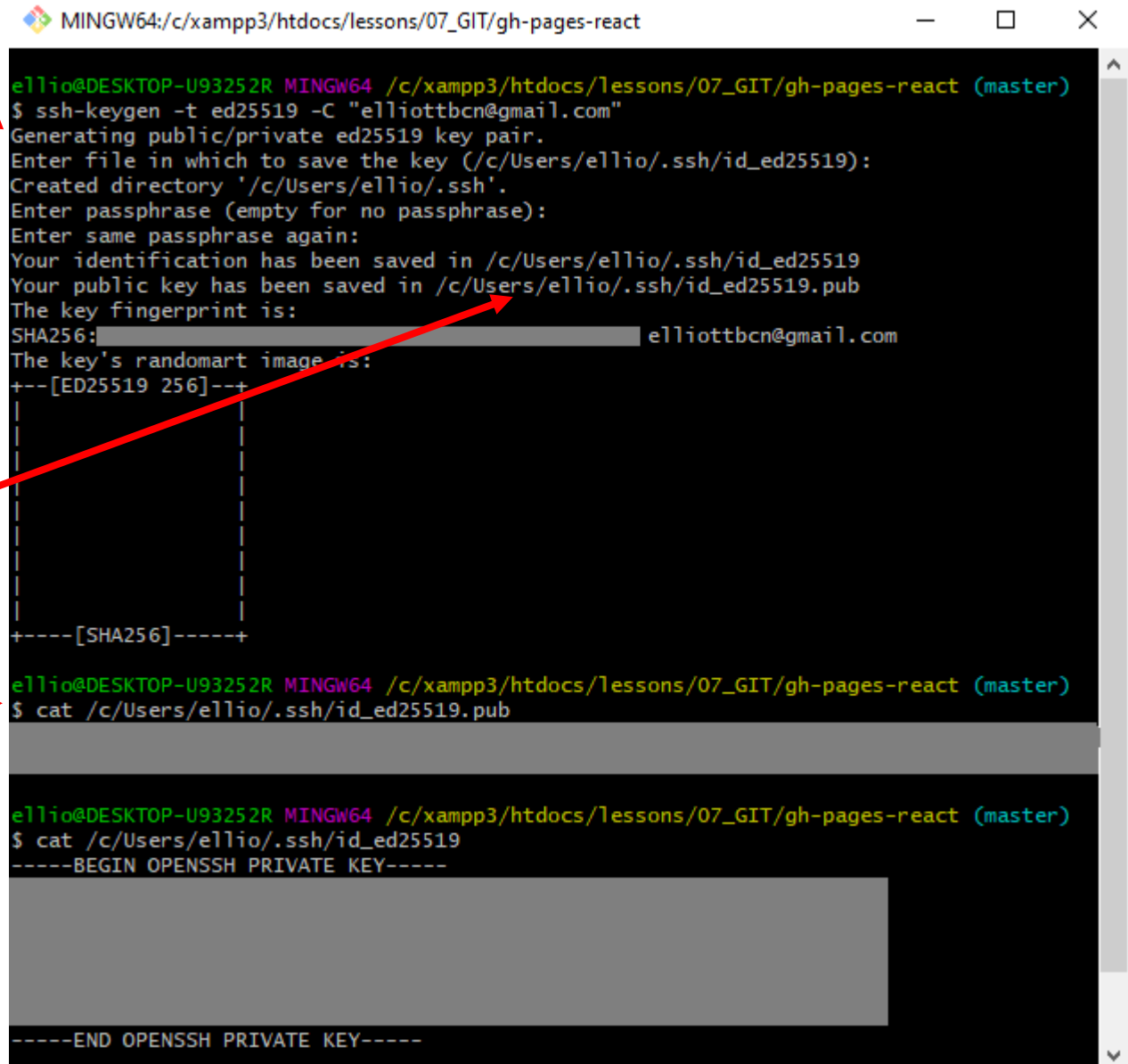
We need to make to connection between local repository and remote repository secure using SSH key.

First we generate a key locally with the command

Ssh-keygen -t ed25519 -C "email"

This will give us a location where the key is saved.

We open the file with a cat command and copy the contents.

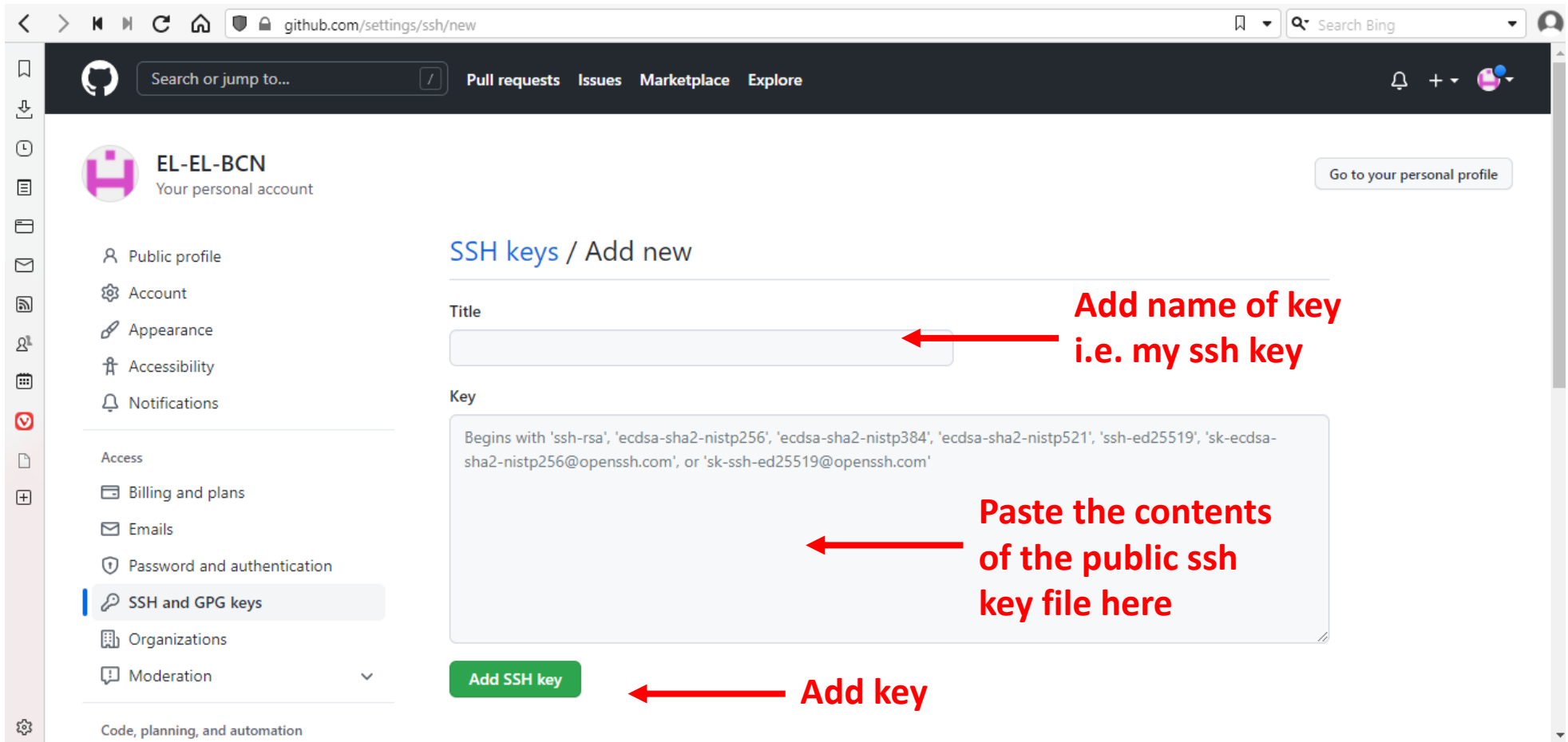
A terminal window titled 'MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react' shows the process of generating an SSH key. The user runs 'ssh-keygen -t ed25519 -C "elliottbcn@gmail.com"'. The terminal prompts for a file to save the key, a passphrase, and confirmation of the passphrase. It then displays the key's fingerprint and a randomart image. A red arrow points from the text 'First we generate a key locally with the command' to the command line. Another red arrow points from 'This will give us a location where the key is saved.' to the file path in the 'cat' command. The terminal shows the public key file being viewed with 'cat /c:/Users/ellio/.ssh/id_ed25519.pub' and the private key file being viewed with 'cat /c:/Users/ellio/.ssh/id_ed25519'.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ ssh-keygen -t ed25519 -C "elliottbcn@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c:/Users/ellio/.ssh/id_ed25519):
Created directory '/c:/Users/ellio/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/ellio/.ssh/id_ed25519
Your public key has been saved in /c:/Users/ellio/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256: [redacted] elliottbcn@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|
|
|
|
|
|
+-----[SHA256]-----+

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ cat /c:/Users/ellio/.ssh/id_ed25519.pub

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ cat /c:/Users/ellio/.ssh/id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
[redacted]
-----END OPENSSH PRIVATE KEY-----
```

Then we go to github.com and in the top right we select user settings, SSH and GPG keys.



The screenshot shows the GitHub 'Add new SSH key' page for user 'EL-EL-BCN'. The page has a dark header with navigation links and a sidebar with account settings. The main content area is titled 'SSH keys / Add new' and contains a form with two fields: 'Title' and 'Key'. The 'Title' field is a text input, and the 'Key' field is a large text area with a placeholder text. A green 'Add SSH key' button is at the bottom. Three red arrows point to these elements with text annotations: one to the 'Title' field, one to the 'Key' field, and one to the 'Add SSH key' button.

github.com/settings/ssh/new

Search or jump to...

Pull requests Issues Marketplace Explore

EL-EL-BCN
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access
Billing and plans
Emails
Password and authentication
SSH and GPG keys
Organizations
Moderation

Code, planning, and automation

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Add name of key
i.e. my ssh key

Paste the contents
of the public ssh
key file here

Add key

Note that the SSH key is created against the user not the repository so the key can be used to access any remote repository under the user.

github.com/EL-EL-BCN/gh-pages-react

Search or jump to... Pull requests Issues Marketplace Explore

EL-EL-BCN / gh-pages-react Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

gh-pages had recent pushes about 1 hour ago [Compare & pull request](#)

master 2 branches 0 tags


Go to file Add file Code

EL-EL-BCN packages.json updated to include homepage

public	Initial react app
src	Initial react app
.gitignore	Initial react app
README.md	Initial react app
package-lock.json	Initial react app
package.json	packages.json updated to include homepage 1 hour ago

Clone ?

HTTPS SSH GitHub CLI

git@github.com:EL-EL-BCN/gh-pages-react.g: 

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

About

My awesome react website

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

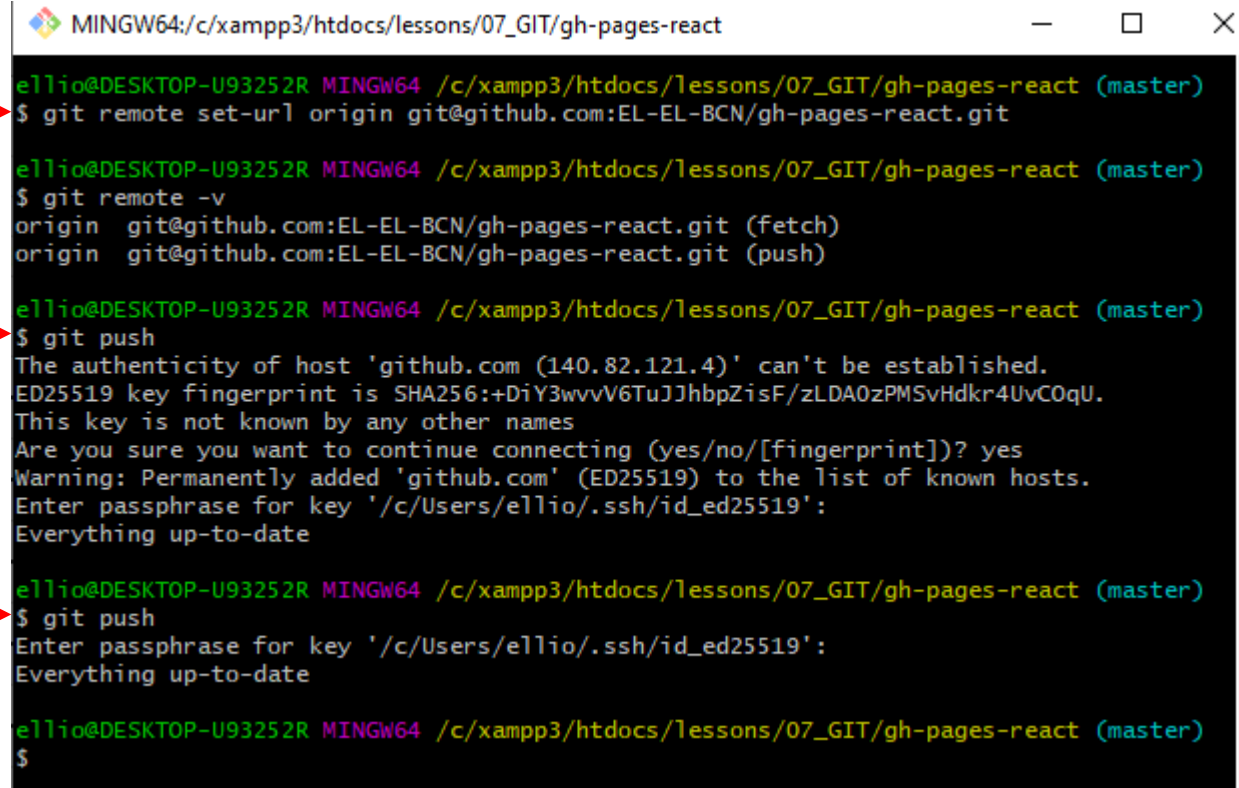
Packages

Now we need to get the remote repository name using SSH.

Now we need to set the local repository to use the SSH url.

Now when we git push it will ask us the first time if want to continue connecting.

I set a passphrase when configuring the SSH key so it will also ask me for this when I push.

A terminal window titled 'MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react' with standard window controls. It shows a series of Git commands and their outputs. Red arrows point from the text on the left to specific lines in the terminal: the first arrow points to the 'git remote set-url' command, the second to the 'git push' command, and the third to the 'git push' command again. The terminal output shows the remote URL being set, the remote list being updated, and then a push attempt that triggers an SSH warning about a host fingerprint. The user responds 'yes' to the warning and enters a passphrase when prompted.

```
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git remote set-url origin git@github.com:EL-EL-BCN/gh-pages-react.git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git remote -v
origin  git@github.com:EL-EL-BCN/gh-pages-react.git (fetch)
origin  git@github.com:EL-EL-BCN/gh-pages-react.git (push)

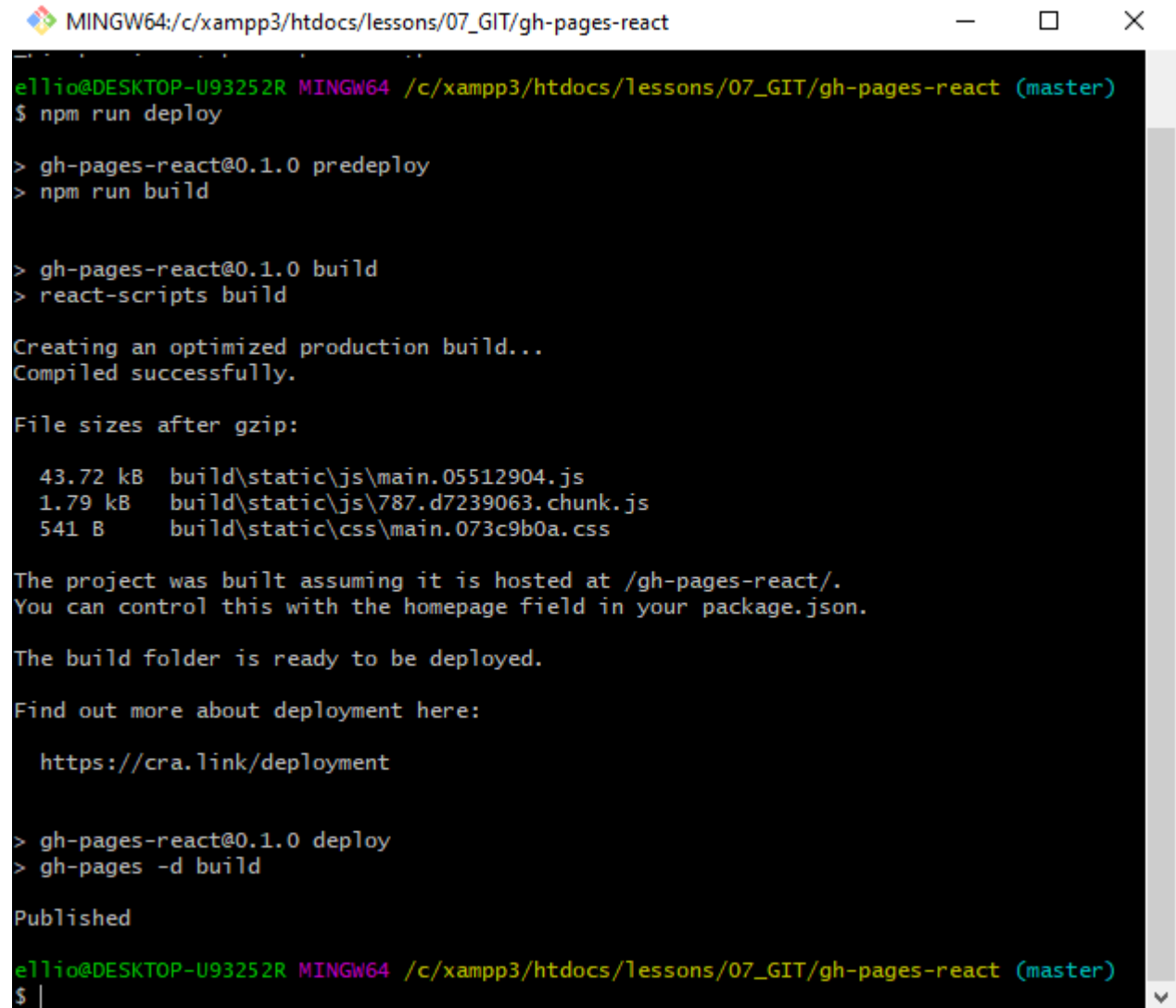
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git push
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/c:/Users/ellio/.ssh/id_ed25519':
Everything up-to-date

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ git push
Enter passphrase for key '/c:/Users/ellio/.ssh/id_ed25519':
Everything up-to-date

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$
```

I rerun the **npm run deploy** command and enter the ssh passphrase where prompted.

Now we have a node js react webpage hosted on github.io that is updated by git from a local repository using SSH key pair.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react
e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ npm run deploy

> gh-pages-react@0.1.0 predeploy
> npm run build

> gh-pages-react@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  43.72 kB   build\static\js\main.05512904.js
  1.79 kB   build\static\js\787.d7239063.chunk.js
  541 B     build\static\css\main.073c9b0a.css

The project was built assuming it is hosted at /gh-pages-react/.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.

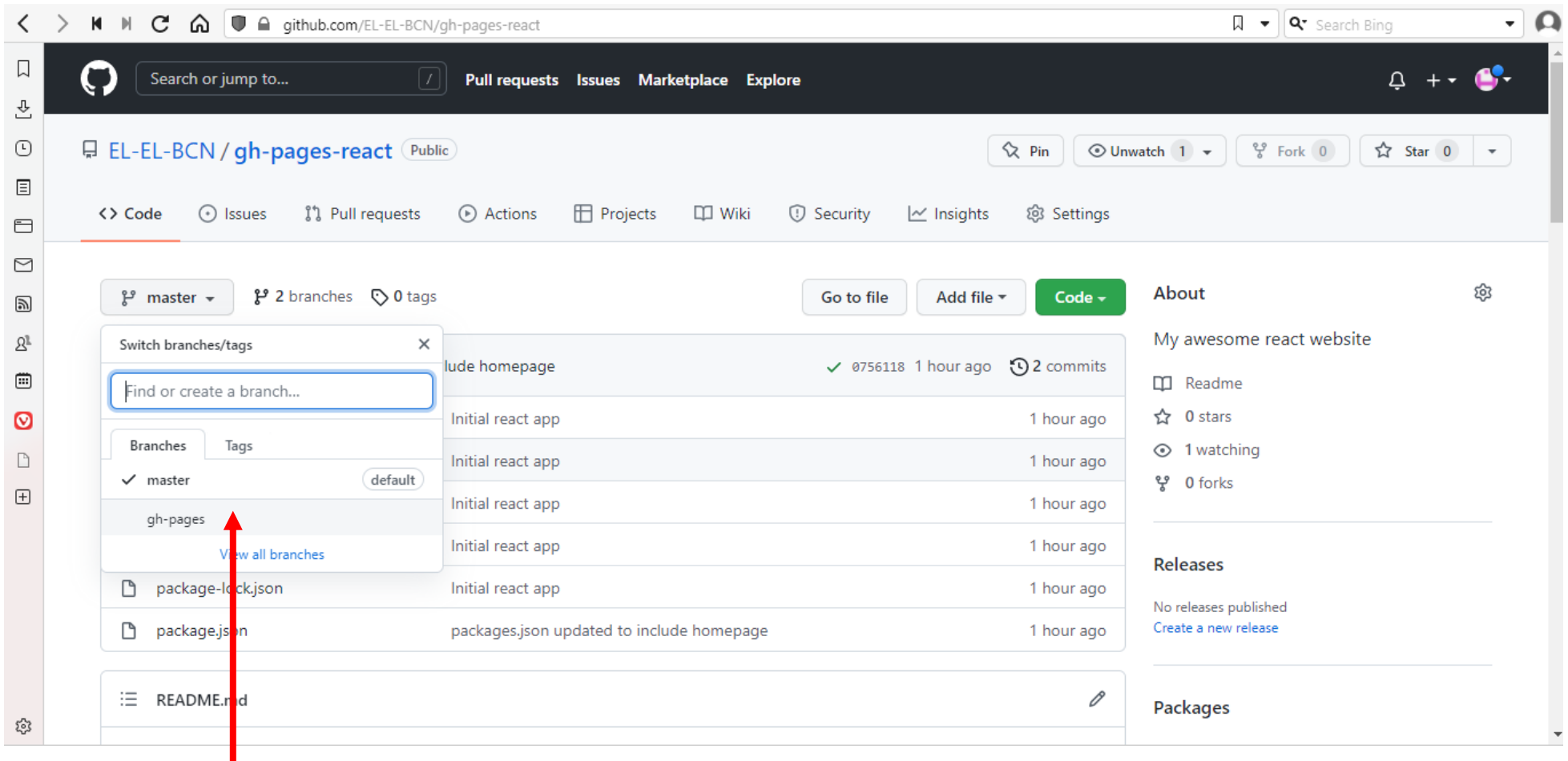
Find out more about deployment here:

  https://cra.link/deployment

> gh-pages-react@0.1.0 deploy
> gh-pages -d build

Published

e11io@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/gh-pages-react (master)
$ |
```

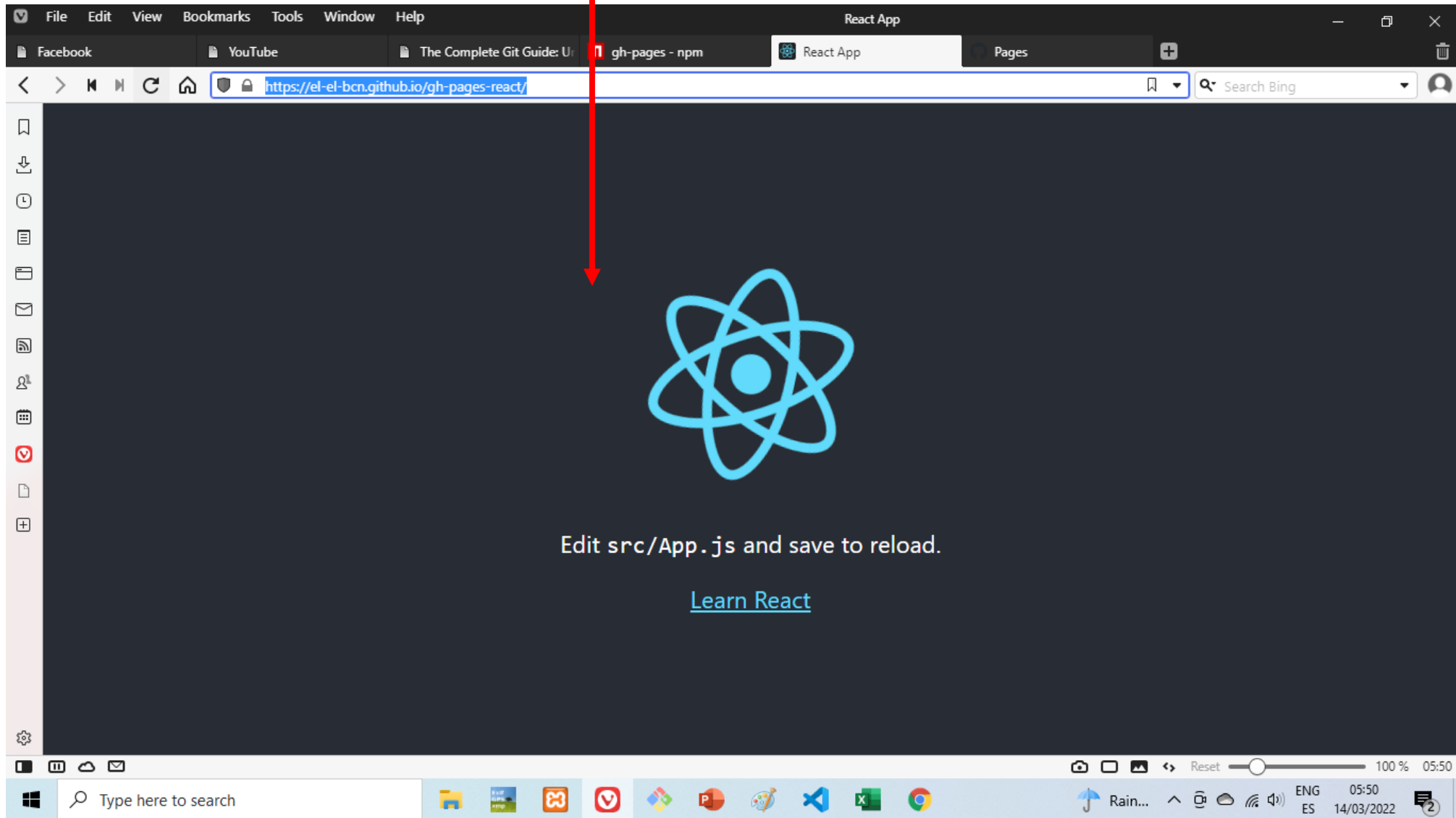


Note that npm deploy script creates a new branch called gh-pages which is the live version of the webpage.

The screenshot shows the GitHub web interface for the repository 'EL-EL-BCN / gh-pages-react'. The 'Settings' tab is selected, and the 'Pages' sub-tab is active in the left sidebar. The main content area is titled 'GitHub Pages' and includes a message stating the site is ready to be published at <https://el-el-bcn.github.io/gh-pages-react/>. Under the 'Source' section, the 'Branch: gh-pages' dropdown is highlighted with a red arrow. Other sections visible include 'Theme Chooser' with a 'Choose a theme' button, and 'Custom domain' with a 'Learn more' link. The bottom of the page shows a status bar with 'Reset', '100%', and '05:48'.

We need to set the github pages to the gh-pages branch.

And now we refresh the URL and it has changed to the contents of the default node js index page.

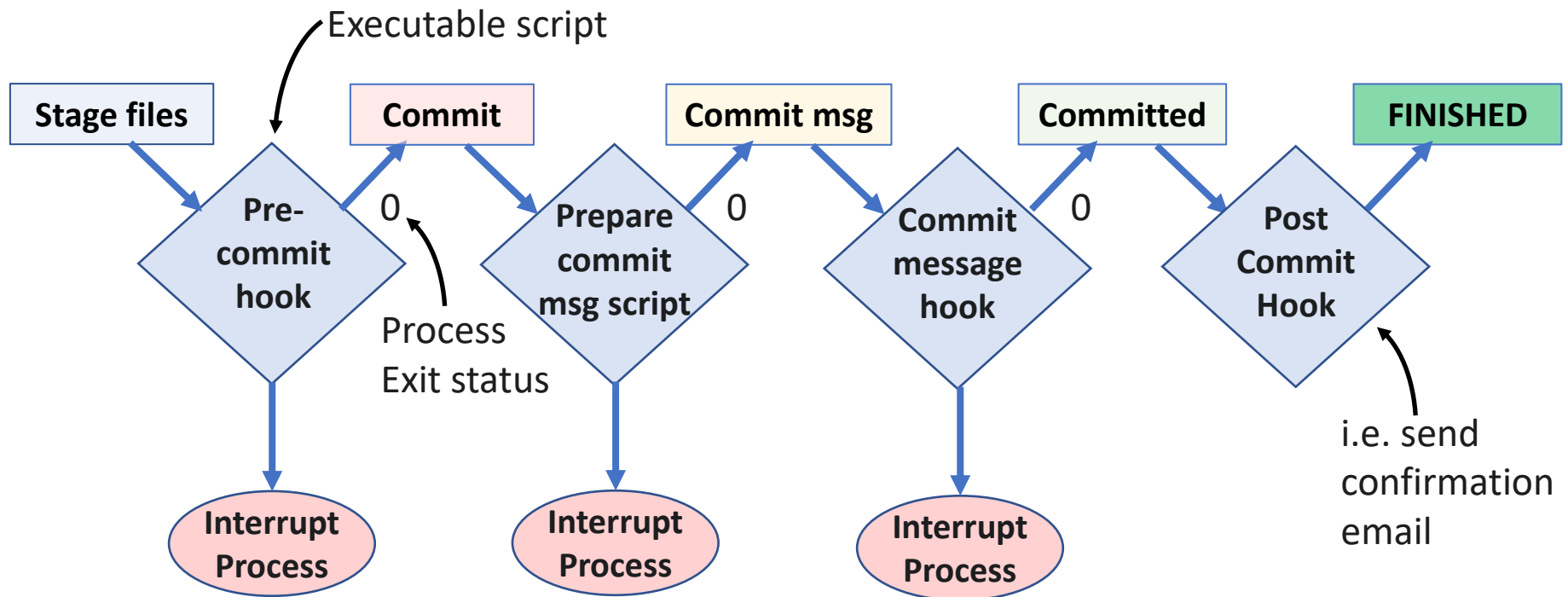


GIT HOOKS

Normal workflow without git hooks



By default git allows you to commit any file, enter any commit message and does not perform any checks such as validity of commit message or syntax of any code files. However git automatically creates hooks by default out of the box but they are disabled unless otherwise activated.



In this diagram there are executable scripts that check the content of the commit and if output value is zero then the process proceeds to the next step, any other value than zero goes to an interrupt process. There are many more types of git hook and not all need to be used.. There are client and server hooks.

How git hooks could be used in the real world

- Verify or edit commit message
- Run application test before each commit or before pushing to remote
- Verify and potentially fix code syntax mistakes using linters like Eslint or Pylint
- Generate notifications after successful committing
- Deploy updated application to production on the server

Note: that deploying hooks will slow down git because the hook script has to be run so depending on the complexity of the hook script it could take several minutes or more.

Code syntax mistakes could route to an interrupt process if errors found notifying the user to fix the errors and try the commit again.

Notifications could be an email, or terminal output to screen etc.

Deploy updated application to server requires a server side hook.

Using pre-commit hook

Starting from the habitual directory

A new directory called **git-hooks-local** is created

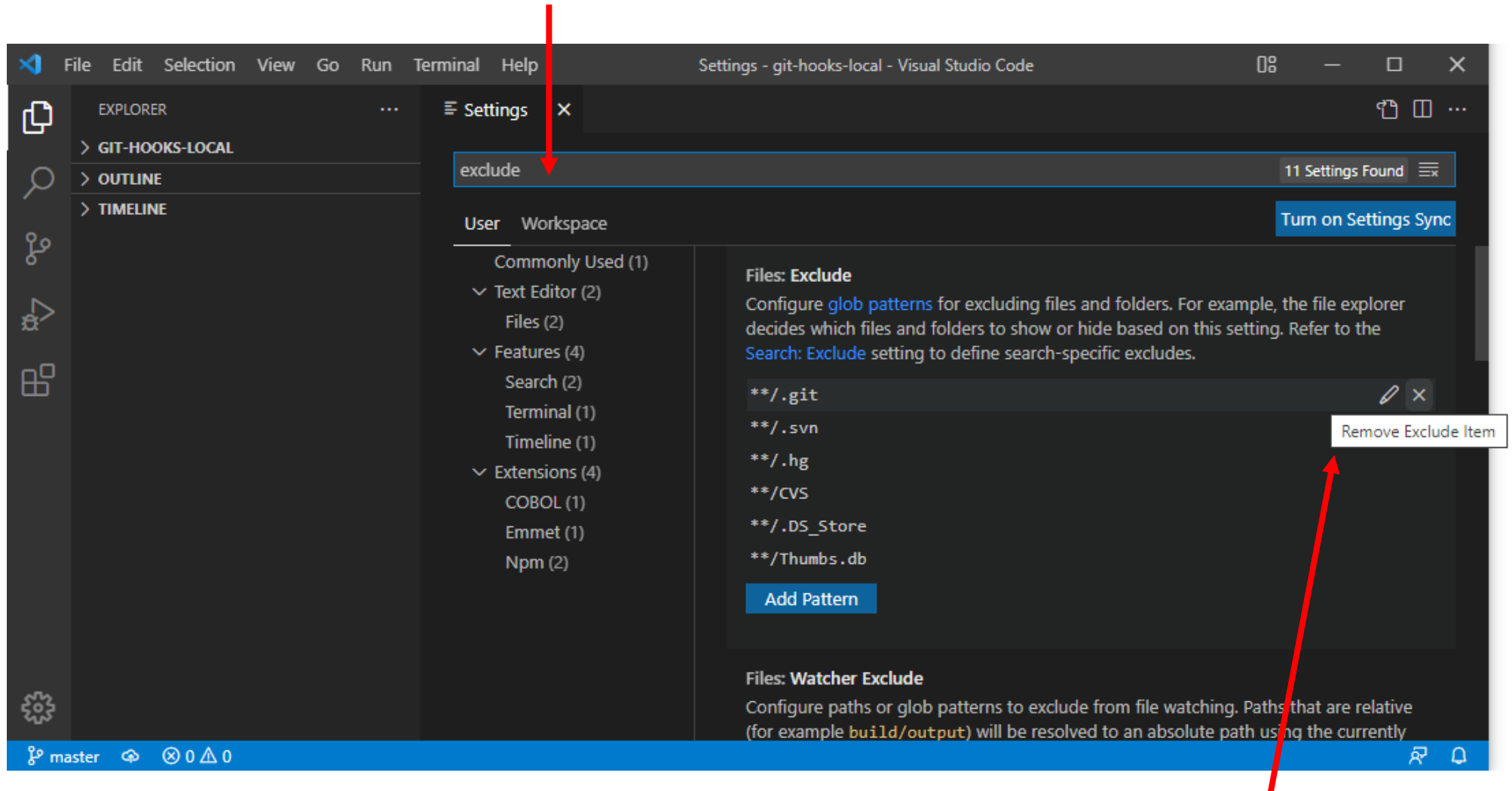
From within **git-hooks-local** git is initialised

Immediately as out of the box configuration we see a git directory called hooks.

The hooks directory contains pre-configured default hooks ready to use. These are a series of files with .sample extension. The .sample extension is disabled on git by default so to utilise one of these pre-configured files just remove the .sample extension.

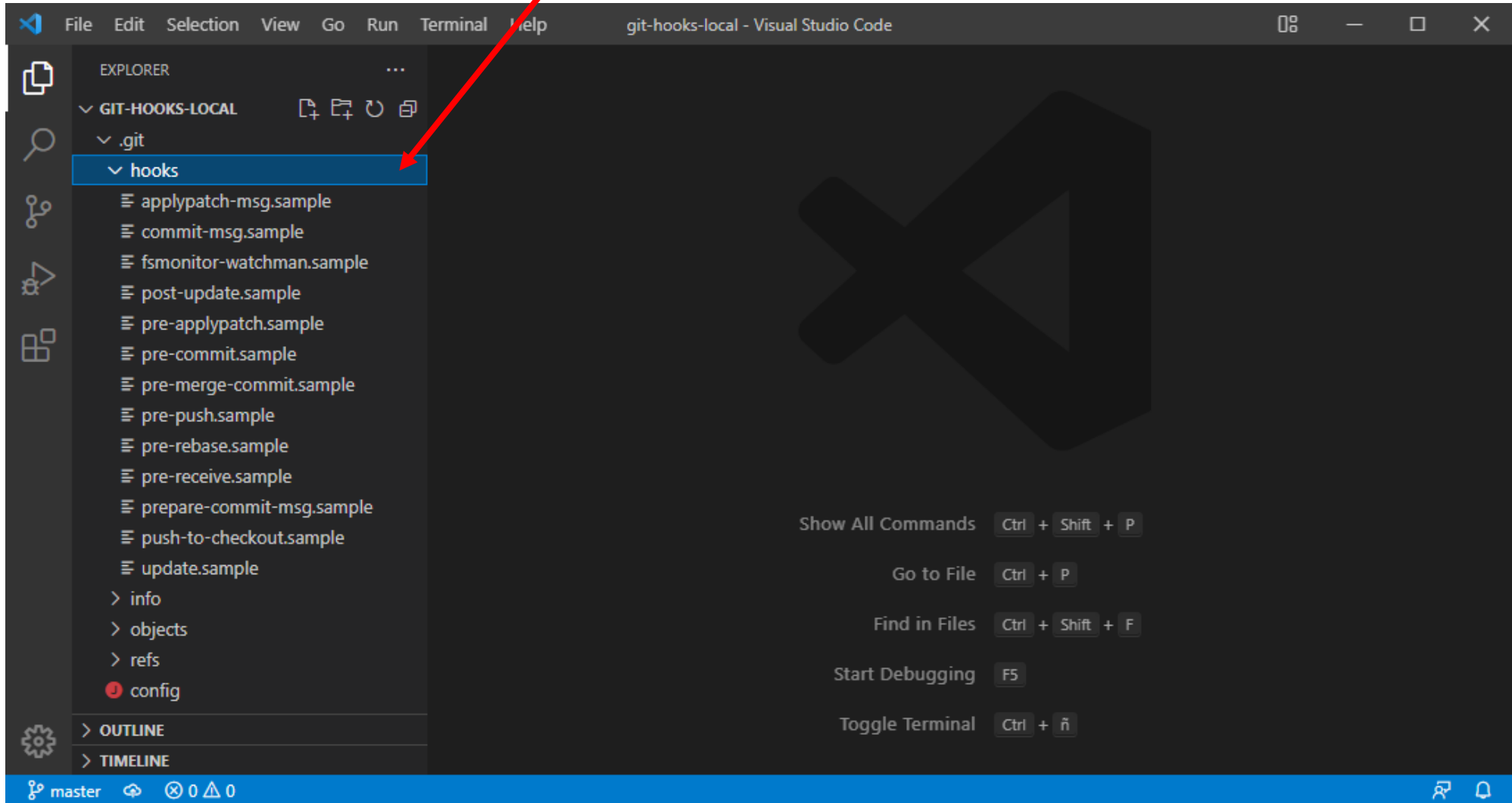
```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local
ellio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ mkdir git-hooks-local
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd git-hooks-local/
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/git-hooks-local/.git/
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ ls .git
HEAD  config  description  hooks/  info/  objects/  refs/
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ ls -la .git/hooks
total 53
drwxr-xr-x 1 ellio 197609  0 Mar 15 05:45 ./
drwxr-xr-x 1 ellio 197609  0 Mar 15 05:45 ../
-rwxr-xr-x 1 ellio 197609 478 Mar 15 05:45 applypatch-msg.sample*
-rwxr-xr-x 1 ellio 197609 896 Mar 15 05:45 commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 4655 Mar 15 05:45 fsmonitor-watchman.sample*
-rwxr-xr-x 1 ellio 197609 189 Mar 15 05:45 post-update.sample*
-rwxr-xr-x 1 ellio 197609 424 Mar 15 05:45 pre-applypatch.sample*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 05:45 pre-commit.sample*
-rwxr-xr-x 1 ellio 197609 416 Mar 15 05:45 pre-merge-commit.sample*
-rwxr-xr-x 1 ellio 197609 1374 Mar 15 05:45 pre-push.sample*
-rwxr-xr-x 1 ellio 197609 4898 Mar 15 05:45 pre-rebase.sample*
-rwxr-xr-x 1 ellio 197609 544 Mar 15 05:45 pre-receive.sample*
-rwxr-xr-x 1 ellio 197609 1492 Mar 15 05:45 prepare-commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 2783 Mar 15 05:45 push-to-checkout.sample*
-rwxr-xr-x 1 ellio 197609 3650 Mar 15 05:45 update.sample*
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$
```

By default visual studio code does not display hidden files so we need to go to file->preferences->settings and type exclude where we will see a list of files and folders that are excluded by default.



If we mouse over the right of the column for `**/.git` we can use the x to remove excluded item.

Now the hidden .git folder is visible and we can explore the contents of the hooks sub-folder.



The pre-commit script first line comment `#!/bin/sh` tells us what interpreter should be used to read the script, in this case shell. Note that this script is a sample file that can be used for learning purposes.

pre-commits.sample - git-hooks-local - Visual Studio Code

EXPLORER

- GIT-HOOKS-LOCAL
 - .git
 - hooks
 - applypatch-msg.sample
 - commit-msg.sample
 - fsmonitor-watchman.sample
 - post-update.sample
 - pre-applypatch.sample
 - \$ pre-commit.sample**
 - pre-merge-commit.sample
 - pre-push.sample
 - pre-rebase.sample
 - pre-receive.sample
 - prepare-commit-msg.sample
 - push-to-checkout.sample
 - update.sample
 - info
 - objects
 - refs
 - config
 - description
 - FETCH_HEAD
 - HEAD
 - OUTLINE
 - TIMELINE

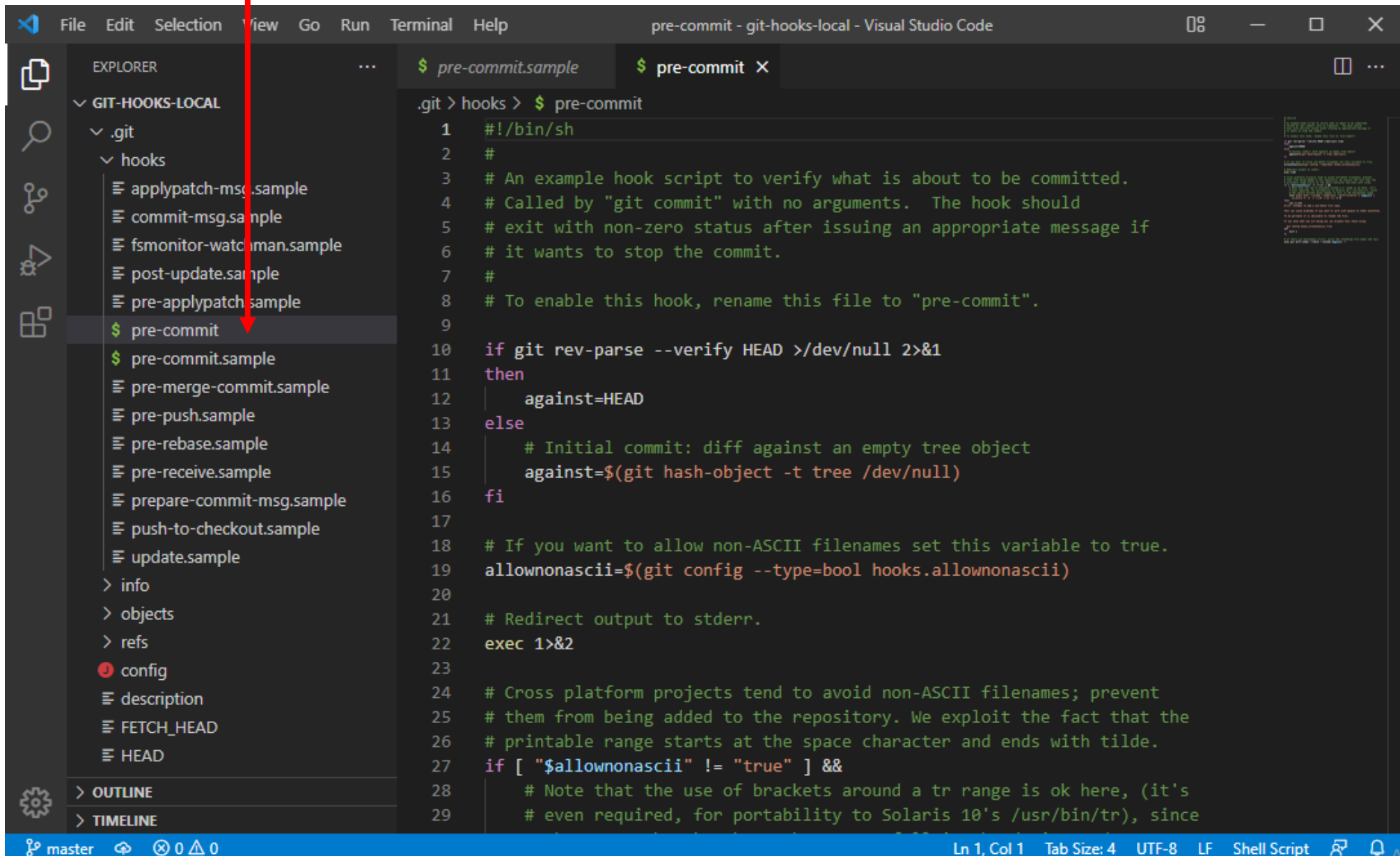
pre-commit.sample

```
1  #!/bin/sh
2  #
3  # An example hook script to verify what is about to be committed.
4  # Called by "git commit" with no arguments. The hook should
5  # exit with non-zero status after issuing an appropriate message if
6  # it wants to stop the commit.
7  #
8  # To enable this hook, rename this file to "pre-commit"
9
10 if git rev-parse --verify HEAD >/dev/null 2>&1
11 then
12     against=HEAD
13 else
14     # Initial commit: diff against an empty tree object
15     against=$(git hash-object -t tree /dev/null)
16 fi
17
18 # If you want to allow non-ASCII filenames set this variable to true.
19 allownonascii=$(git config --type=bool hooks.allownonascii)
20
21 # Redirect output to stderr.
22 exec 1>&2
23
24 # Cross platform projects tend to avoid non-ASCII filenames; prevent
25 # them from being added to the repository. We exploit the fact that the
26 # printable range starts at the space character and ends with tilde.
27 if [ "$allownonascii" != "true" ] &&
28     # Note that the use of brackets around a tr range is ok here, (it's
29     # even required, for portability to Solaris 10's /usr/bin/tr), since
```

The comments also detail what the script does, for example it checks for non ASCII filenames.

Ln 1, Col 1 Tab Size: 4 UTF-8 LF Shell Script

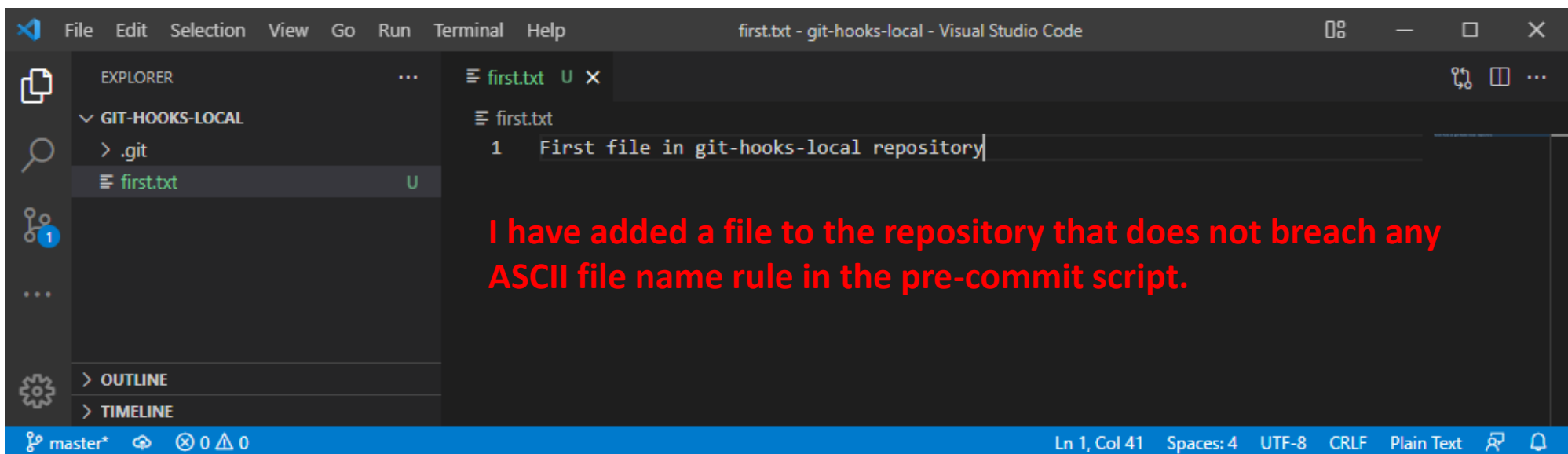
To use this pre-commit script as is we just copy it then rename it to remove the .sample extension



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor on the right. The Explorer sidebar shows the file structure of a repository, with the `.git` directory expanded to show the `hooks` directory. The `pre-commit.sample` file is highlighted in the Explorer sidebar, and a red arrow points to it. The Editor shows the content of the `pre-commit.sample` file, which is a shell script. The script is a pre-commit hook that checks for non-ASCII filenames. The script is as follows:

```
.git > hooks > $ pre-commit
1  #!/bin/sh
2  #
3  # An example hook script to verify what is about to be committed.
4  # Called by "git commit" with no arguments. The hook should
5  # exit with non-zero status after issuing an appropriate message if
6  # it wants to stop the commit.
7  #
8  # To enable this hook, rename this file to "pre-commit".
9
10 if git rev-parse --verify HEAD >/dev/null 2>&1
11 then
12     against=HEAD
13 else
14     # Initial commit: diff against an empty tree object
15     against=$(git hash-object -t tree /dev/null)
16 fi
17
18 # If you want to allow non-ASCII filenames set this variable to true.
19 allownonascii=$(git config --type=bool hooks.allownonascii)
20
21 # Redirect output to stderr.
22 exec 1>&2
23
24 # Cross platform projects tend to avoid non-ASCII filenames; prevent
25 # them from being added to the repository. We exploit the fact that the
26 # printable range starts at the space character and ends with tilde.
27 if [ "$allownonascii" != "true" ] &&
28     # Note that the use of brackets around a tr range is ok here, (it's
29     # even required, for portability to Solaris 10's /usr/bin/tr), since
```

The status bar at the bottom shows the current file is `pre-commit.sample` and the editor is in `Shell Script` mode.



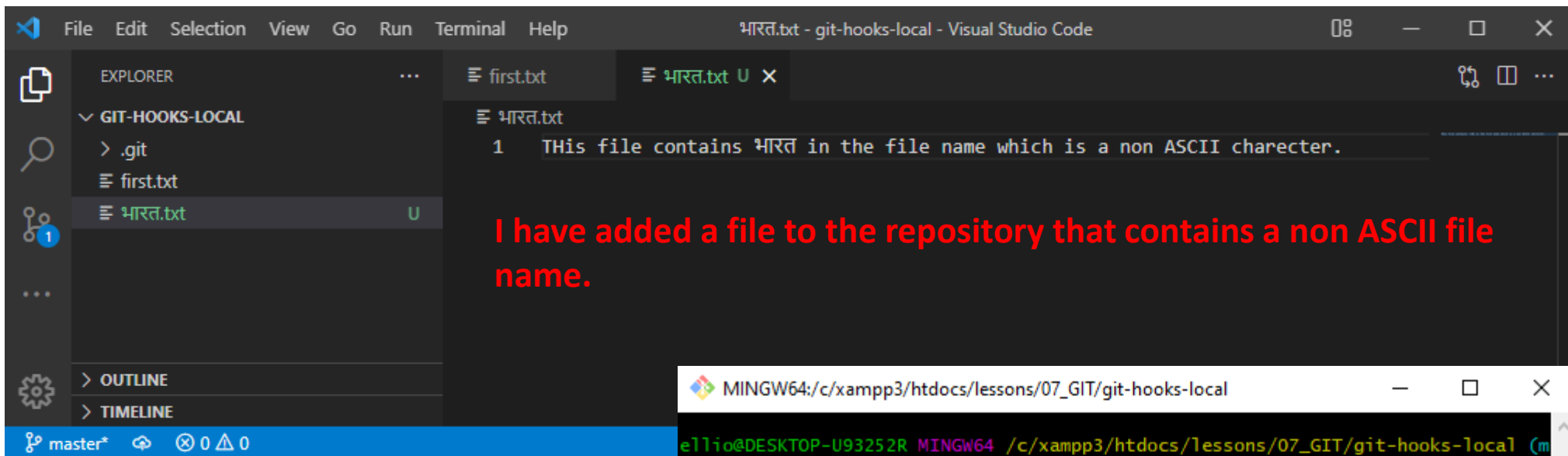
The file is staged and committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/git-hooks-local

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "first commit"
[master (root-commit) a218c1f] first commit
1 file changed, 1 insertion(+)
create mode 100644 first.txt

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ |
```



The file is staged but gives the error message from the pre-commit script because it contains non ASCII characters in the file name.

WE can see from git status that the file is in the staging area but was blocked from being committed to the repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "non ascii file name"
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.
To be portable it is advisable to rename the file.
If you know what you are doing you can disable this check using:

    git config hooks.allownonascii true

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "\340\244\255\340\244\276\340\244\260\340\244\244.txt"

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$
```

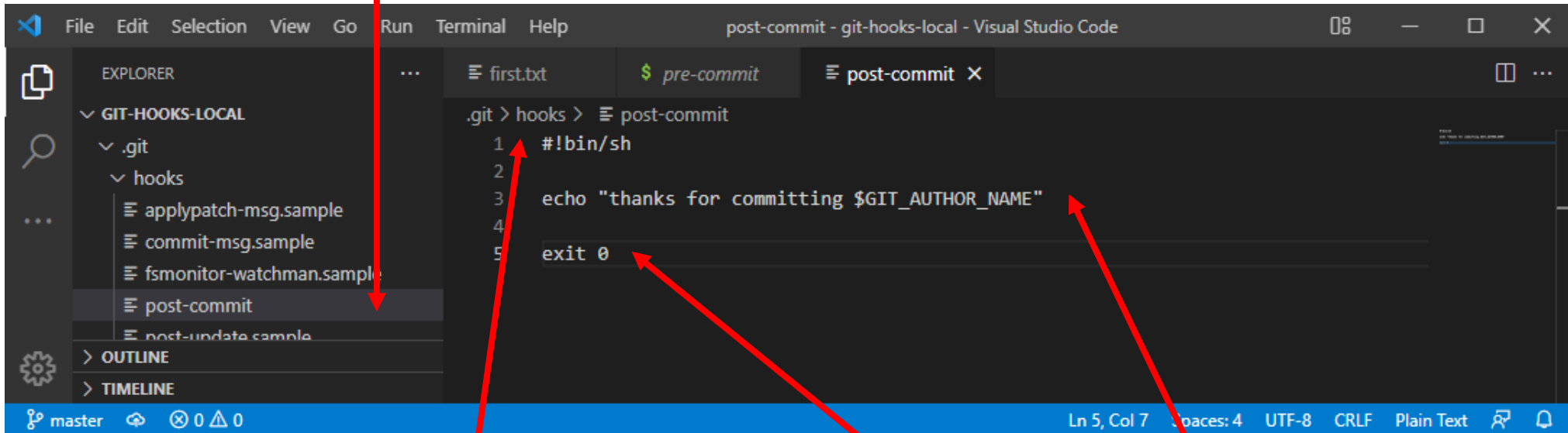

The screenshot shows the Visual Studio Code interface with the 'pre-commit' hook script open. The Explorer sidebar on the left shows the file structure under '.git > hooks > \$ pre-commit', with 'pre-commit' selected. The main editor displays the script content, which includes instructions on how to disable the check using 'git config hooks.allownonascii true' and an 'exit 1' statement at the end of the script. A red arrow points to the 'exit 1' line. The status bar at the bottom indicates 'Ln 1, Col 1' and 'Tab Size: 4 UTF-8 LF Shell Script'.

```
38
39 To be portable it is advisable to rename the file.
40
41 If you know what you are doing you can disable this check using:
42
43     git config hooks.allownonascii true
44 EOF
45     exit 1
46 fi
47
48 # If there are whitespace errors, print the offending file names and fail
```

At the bottom of the pre-commit script we see the exit code of 1 for an error, remember 0 is a pass, anything other than 0 is a fail.

Enabling post-commit hook

There is no sample post-commit hook but we can create one by adding a file to the `.git/hooks` folder called "post-commit". It must specifically be named post-commit.

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows the file structure: 'GIT-HOOKS-LOCAL' > '.git' > 'hooks'. The 'post-commit' file is selected and highlighted. The main editor area shows the content of 'post-commit' with line numbers 1 to 5. The code is: 1 #!/bin/sh, 2, 3 echo "thanks for committing \$GIT_AUTHOR_NAME", 4, 5 exit 0. Red arrows point from the text annotations below to specific parts of the code: one to the 'post-commit' file in the Explorer, one to the first line '#!/bin/sh', one to the 'echo' command, and one to the 'exit 0' command.

```
post-commit - git-hooks-local - Visual Studio Code

EXPLORER
  GIT-HOOKS-LOCAL
    .git
      hooks
        appypatch-msg.sample
        commit-msg.sample
        fsmonitor-watchman.sample
        post-commit
        post-update.sample
      > OUTLINE
      > TIMELINE

.git > hooks > post-commit
1  #!/bin/sh
2
3  echo "thanks for committing $GIT_AUTHOR_NAME"
4
5  exit 0

Ln 5, Col 7  Spaces: 4  UTF-8  CRLF  Plain Text
```

The post hooks first line must be the interpreter used to run the script, in this case it is a shell script

All the script does is output the committer author name

We do not need to specify exit 0 because the script does not make any decisions about validity of files, folders etc.

In the .git/hooks directory we see the newly create post-commit file.
Note the permissions of **-rwxr-xr-x**.
If the hooks file permissions are not set to this then it will not be executable.

Now I have created a third file and staged it then committed it.

Notice the message *"thanks for committing EL-EL-BCN"*. This is the message we configured in the post-commit hook file.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ ls -la .git/hooks
total 58
drwxr-xr-x 1 ellio 197609  0 Mar 15 06:53 ./
drwxr-xr-x 1 ellio 197609  0 Mar 15 06:41 ../
-rwxr-xr-x 1 ellio 197609 478 Mar 15 05:45 applypatch-msg.sample*
-rwxr-xr-x 1 ellio 197609 896 Mar 15 05:45 commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 4655 Mar 15 05:45 fsmonitor-watchman.sample*
-rwxr-xr-x 1 ellio 197609  67 Mar 15 06:57 post-commit*
-rwxr-xr-x 1 ellio 197609 189 Mar 15 05:45 post-update.sample*
-rwxr-xr-x 1 ellio 197609 424 Mar 15 05:45 pre-applypatch.sample*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 06:12 pre-commit*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 05:45 pre-commit.sample*
-rwxr-xr-x 1 ellio 197609 416 Mar 15 05:45 pre-merge-commit.sample*
-rwxr-xr-x 1 ellio 197609 1374 Mar 15 05:45 pre-push.sample*
-rwxr-xr-x 1 ellio 197609 4898 Mar 15 05:45 pre-rebase.sample*
-rwxr-xr-x 1 ellio 197609  544 Mar 15 05:45 pre-receive.sample*
-rwxr-xr-x 1 ellio 197609 1492 Mar 15 05:45 prepare-commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 2783 Mar 15 05:45 push-to-checkout.sample*
-rwxr-xr-x 1 ellio 197609 3650 Mar 15 05:45 update.sample*

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ touch third.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "third file committed"
thanks for committing EL-EL-BCN
[master 5ec66df] third file committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third.txt

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$
```

```

→ git-hooks-local git:(master) x git commit -m 'Third file'
hint: The '.git/hooks/post-commit' hook was ignored because it's not set as executable.
hint: You can disable this warning with `git config advice.ignoredHook false`.
[master 96631e0] Third file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third.txt
→ git-hooks-local git:(master)

```

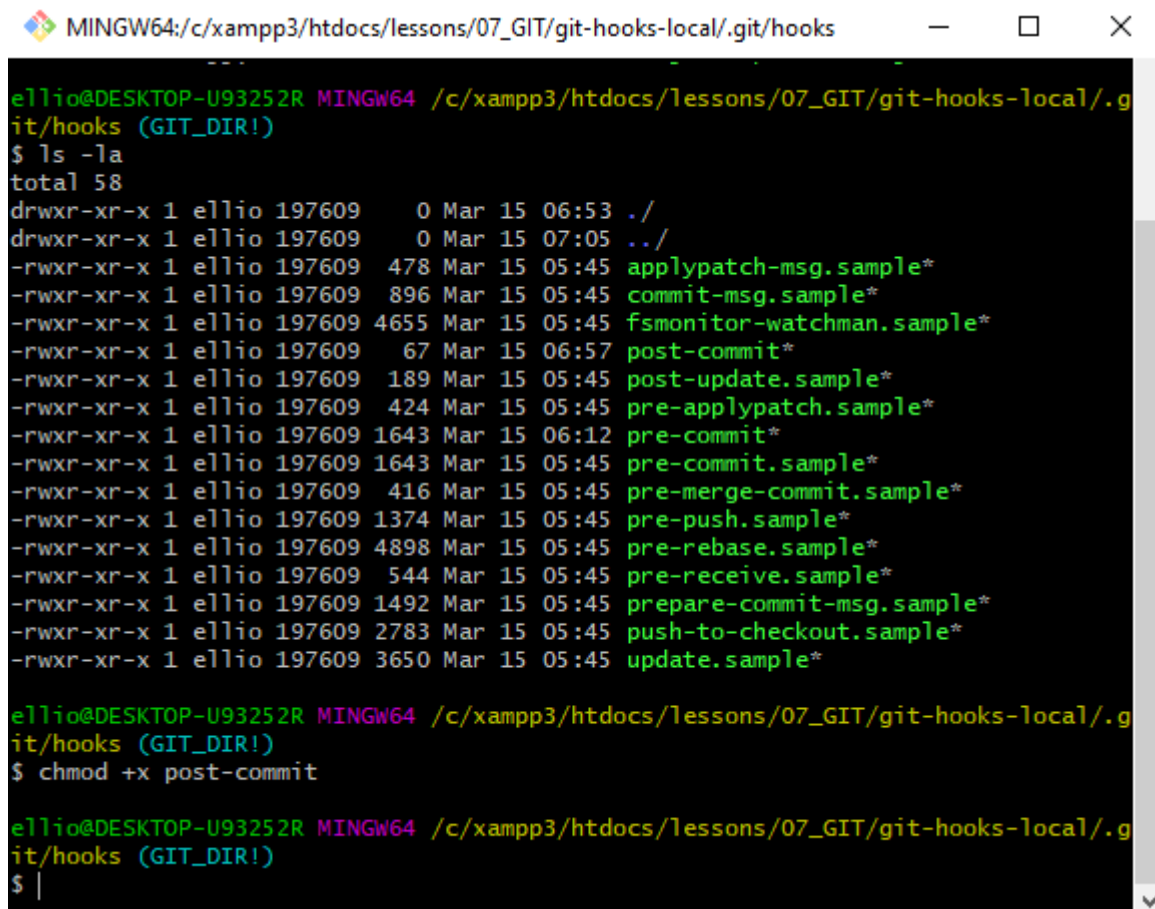
If .git/hooks file is not executable it will throw an error similar to the one above shown in yellow.

The file permission can be edited by

cd .git/hooks

chmod +x <filename>

In the case of this exercise git has automatically created the hooks file with the correct permissions.



```

MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local/.git/hooks
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local/.git/hooks (GIT_DIR!)
$ ls -la
total 58
drwxr-xr-x 1 ellio 197609  0 Mar 15 06:53 ./
drwxr-xr-x 1 ellio 197609  0 Mar 15 07:05 ../
-rwxr-xr-x 1 ellio 197609 478 Mar 15 05:45 applypatch-msg.sample*
-rwxr-xr-x 1 ellio 197609 896 Mar 15 05:45 commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 4655 Mar 15 05:45 fsmonitor-watchman.sample*
-rwxr-xr-x 1 ellio 197609  67 Mar 15 06:57 post-commit*
-rwxr-xr-x 1 ellio 197609 189 Mar 15 05:45 post-update.sample*
-rwxr-xr-x 1 ellio 197609 424 Mar 15 05:45 pre-applypatch.sample*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 06:12 pre-commit*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 05:45 pre-commit.sample*
-rwxr-xr-x 1 ellio 197609 416 Mar 15 05:45 pre-merge-commit.sample*
-rwxr-xr-x 1 ellio 197609 1374 Mar 15 05:45 pre-push.sample*
-rwxr-xr-x 1 ellio 197609 4898 Mar 15 05:45 pre-rebase.sample*
-rwxr-xr-x 1 ellio 197609 544 Mar 15 05:45 pre-receive.sample*
-rwxr-xr-x 1 ellio 197609 1492 Mar 15 05:45 prepare-commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 2783 Mar 15 05:45 push-to-checkout.sample*
-rwxr-xr-x 1 ellio 197609 3650 Mar 15 05:45 update.sample*

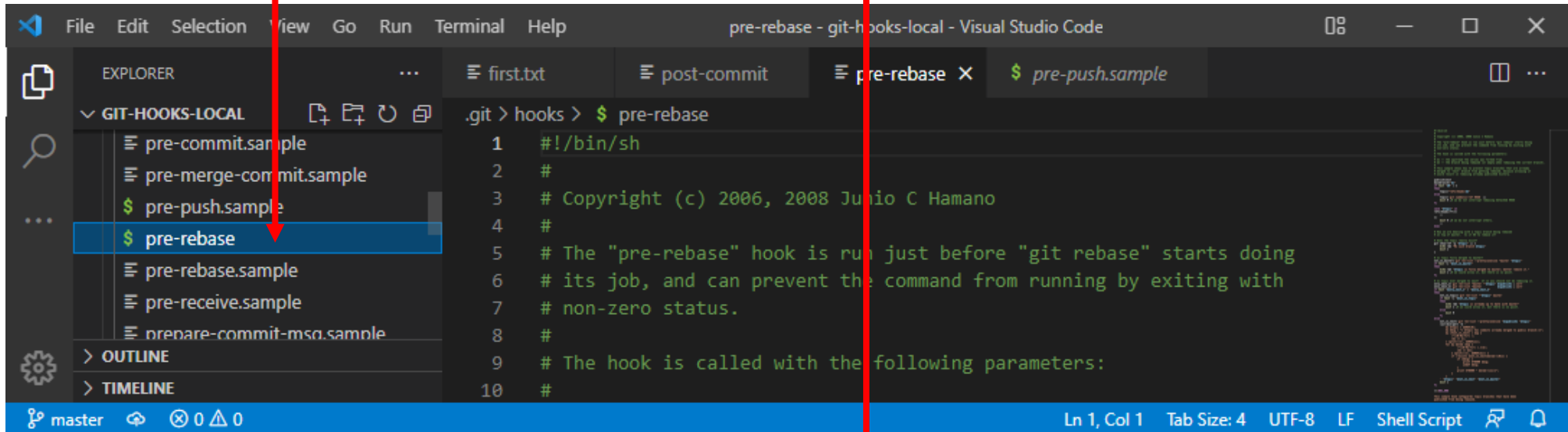
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local/.git/hooks (GIT_DIR!)
$ chmod +x post-commit

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local/.git/hooks (GIT_DIR!)
$ |

```

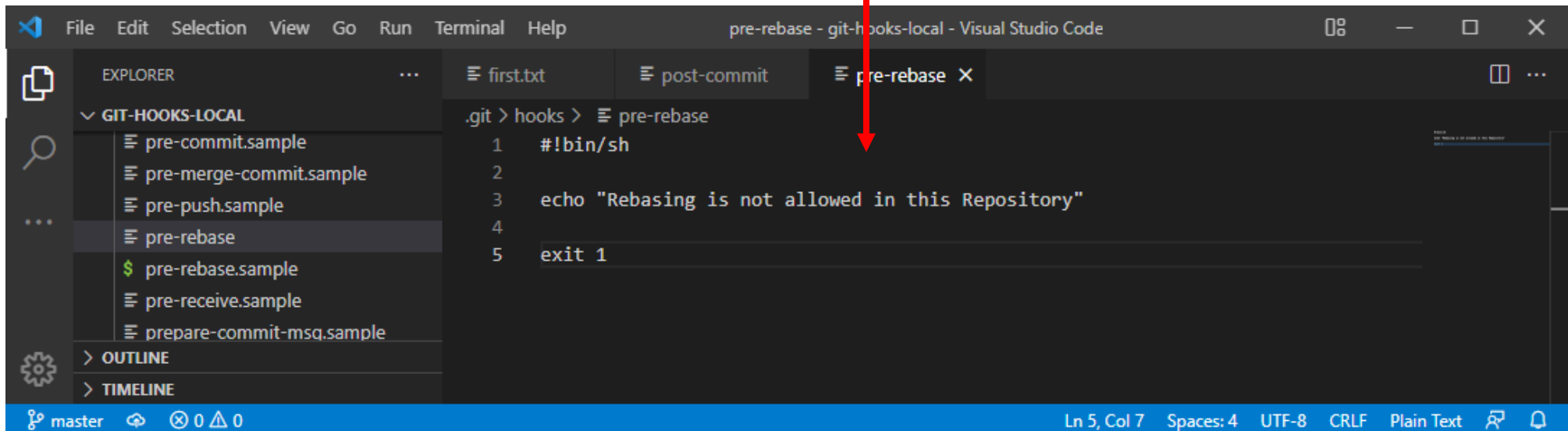
Disabling Rebasing with pre-rebasing hook

There is a default pre-rebasing hook already created so we can just copy it and rename the copy removing the .sample from the filename or create our own script.



This screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the 'GIT-HOOKS-LOCAL' directory. The file 'pre-rebase.sample' is selected and highlighted in blue. A red arrow points from this file to the main editor window. The editor window shows the content of the 'pre-rebase.sample' file, which is a shell script. The script starts with a shebang, a copyright notice, and a comment explaining its purpose: to prevent rebase operations by exiting with a non-zero status. The script is 10 lines long. The status bar at the bottom indicates the current file is 'pre-rebase.sample' and it is a shell script.

```
.git > hooks > $ pre-rebase
1  #!/bin/sh
2  #
3  # Copyright (c) 2006, 2008 Junio C Hamano
4  #
5  # The "pre-rebase" hook is run just before "git rebase" starts doing
6  # its job, and can prevent the command from running by exiting with
7  # non-zero status.
8  #
9  # The hook is called with the following parameters:
10 #
```



This screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the 'GIT-HOOKS-LOCAL' directory. The file 'pre-rebase' is selected and highlighted in blue. A red arrow points from this file to the main editor window. The editor window shows the content of the 'pre-rebase' file, which is a custom shell script designed to disable rebasing. The script is 5 lines long, starting with a shebang, followed by an echo statement that displays 'Rebasing is not allowed in this Repository', and ending with an exit 1 command. The status bar at the bottom indicates the current file is 'pre-rebase' and it is plain text.

```
.git > hooks > pre-rebase
1  #!/bin/sh
2
3  echo "Rebasing is not allowed in this Repository"
4
5  exit 1
```

To test the disable rebasing hook:

1. feature1 branch created and checked out
2. file4.txt created and staged and committed
3. Master branch checked out
4. file5.txt created and staged and committed
5. Feature 1 branch checked out
6. **git rebase master** command issued

Note how when we try to rebase we get the error message “got rebasing is not allowed in this repository” message which we configured previously.

Also we get fatal error: the pre-rebase hook refused to rebase.

To continue the exercises I removed feture1 branch from master branch with **git branch -D feature1**

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/git-hooks-local
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git checkout -b feature1
Switched to a new branch 'feature1'

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$ touch file4.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$ git add .

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$ git commit -m "fourth file added to feature1 branch"
thanks for committing EL-EL-BCN
[feature1 5519c8e] fourth file added to feature1 branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$ git checkout master
Switched to branch 'master'

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ touch file5.txt

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "fith file added to master branch"
thanks for committing EL-EL-BCN
[master 50c0600] fith file added to master branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file5.txt

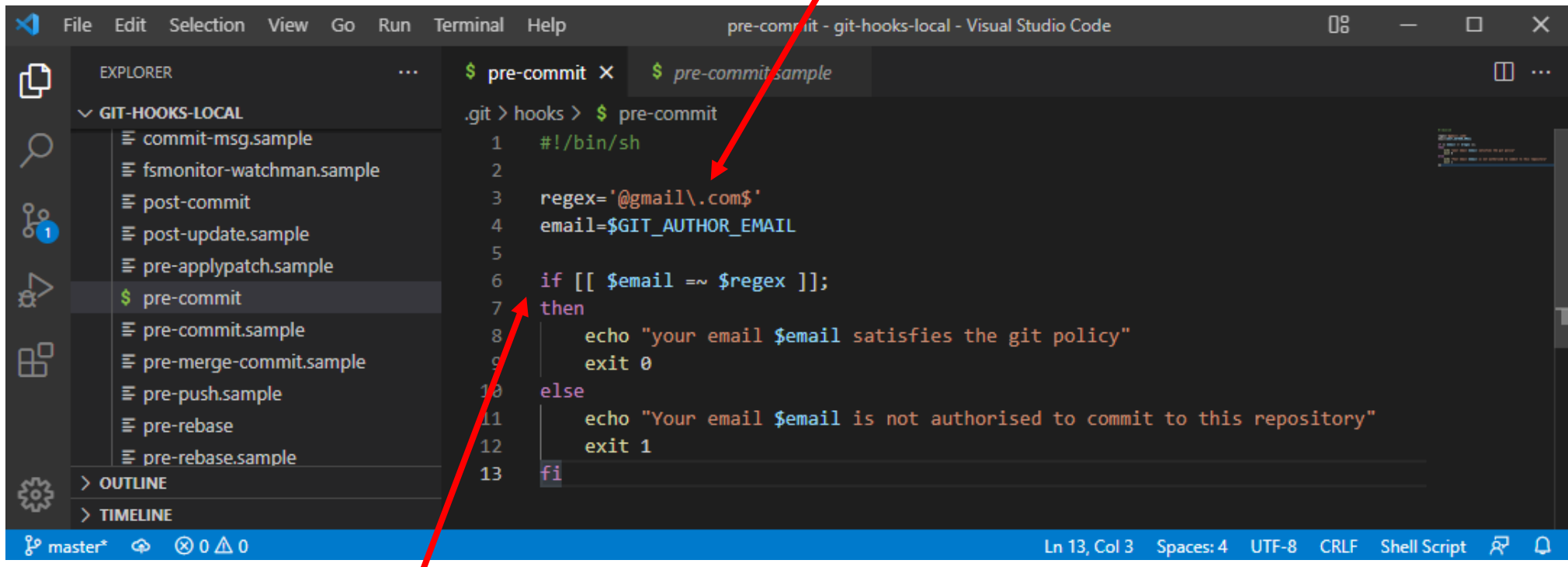
e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git checkout feature1
Switched to branch 'feature1'

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$ git rebase master
Rebasing is not allowed in this Repository
fatal: The pre-rebase hook refused to rebase.

e11io@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (feature1)
$
```

Match authors email against regular expression

The pre-commit script is modified to include an email matching regex that looks for emails only from @gmail.com. We also assign git author email to an email variable.



```
.git > hooks > $ pre-commit
1  #!/bin/sh
2
3  regex='@gmail\.com$'
4  email=$GIT_AUTHOR_EMAIL
5
6  if [[ $email =~ $regex ]];
7  then
8      echo "your email $email satisfies the git policy"
9      exit 0
10 else
11     echo "Your email $email is not authorised to commit to this repository"
12     exit 1
13 fi
```

We then use a simple if statement to check if git author email matches the regular expression. The hook value of zero is used for positive match and hook value of 1 for negative match so that an invalid email is dumped into an interrupt process. fi at the end signifies finalise.

The git hook lets us commit the seventh file because elliottbcn@gmail.com matches the regex in the pre-commit file.

Note the commit message “your email satisfies the git policy”

Now if I modify the regex in the pre-commit hook and test again.....

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local
elliottbcn@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ touch seven.txt

elliottbcn@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

elliottbcn@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "seventh file committed"
your email elliottbcn@gmail.com satisfies the git policy
thanks for committing EL-EL-BCN
[master f60de8e] seventh file committed
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 seven.txt
create mode 100644 six.txt

elliottbcn@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ |
```

```
File Edit Selection View Go Run Terminal Help pre-commit - git-hooks-local - Visual Studio Code

EXPLORER
GIT-HOOKS-LOCAL
  commit-msg.sample
  fsmonitor-watchman.sample
  post-commit
  post-update.sample
  pre-applypatch.sample
  $ pre-commit
  pre-commit.sample
  pre-merge-commit.sample
  pre-push.sample
  pre-rebase
  pre-rebase.sample
  > OUTLINE
  > TIMELINE

pre-commit X pre-commit.sample
.git > hooks > $ pre-commit
1  #!/bin/sh
2
3  regex='@company\.com$'
4  email=$GIT_AUTHOR_EMAIL
5
6  if [[ $email =~ $regex ]];
7  then
8      echo "your email $email satisfies the git policy"
9      exit 0
10 else
11     echo "Your email $email is not authorised to commit to this repository"
12     exit 1
13 fi
```


The git hook denies elliottbcn@gmail.com from committing file eight.com with the configured error message.

A git status command shows that this file is still pending to be committed.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/git-hooks-local
elliottbcn@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ touch eight.txt

elliottbcn@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git add .

elliottbcn@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git commit -m "file eight.txt committed"
Your email elliottbcn@gmail.com is not authorised to commit to this repository

elliottbcn@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   eight.txt

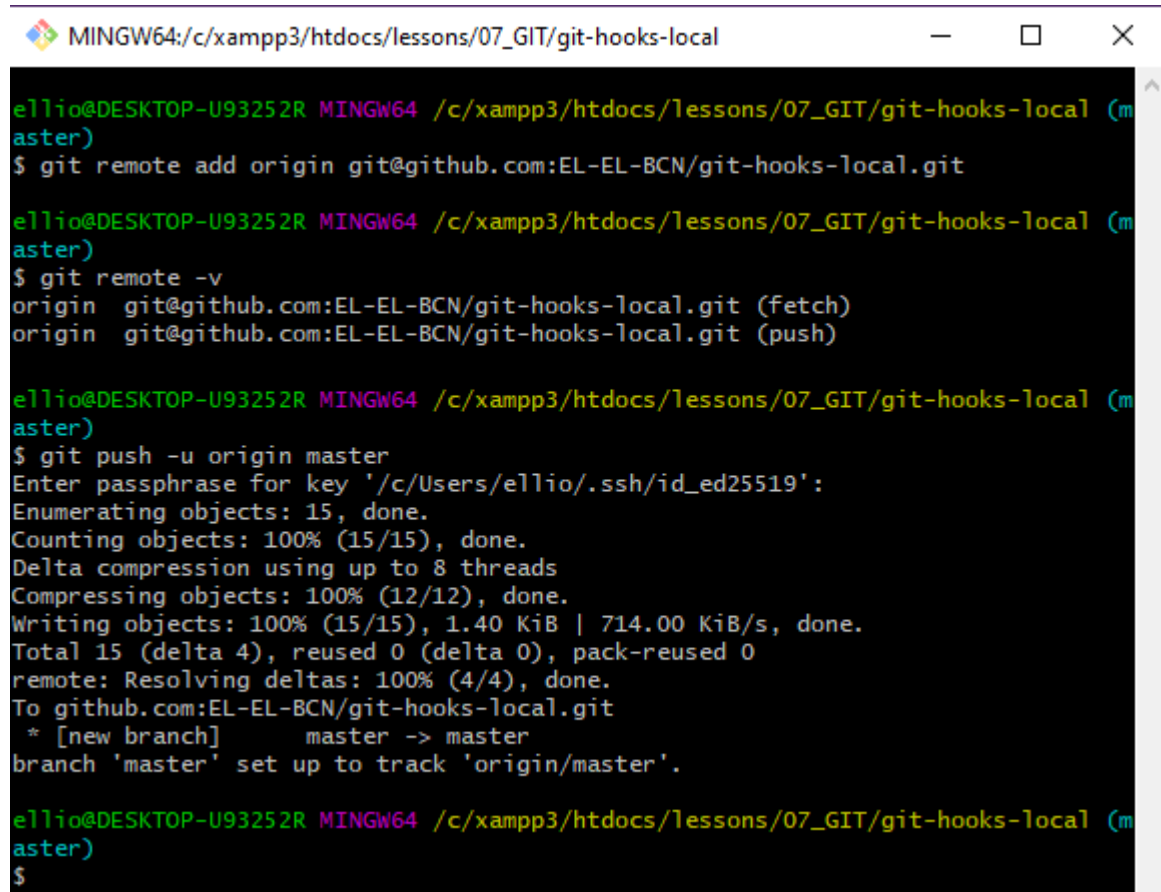
elliottbcn@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$
```

Git hooks are NOT pushed to the remote repository

A remote repository was created in github.com with the following features:

- Name: git-hooks-local
- Description: Experimenting with git local hooks
- No README.me file

The remote repository is configured for the local repository and verified then local is pushed to remote.



```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git remote add origin git@github.com:EL-EL-BCN/git-hooks-local.git

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git remote -v
origin    git@github.com:EL-EL-BCN/git-hooks-local.git (fetch)
origin    git@github.com:EL-EL-BCN/git-hooks-local.git (push)

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ git push -u origin master
Enter passphrase for key '/c:/Users/ellio/.ssh/id_ed25519':
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (15/15), 1.40 KiB | 714.00 KiB/s, done.
Total 15 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To github.com:EL-EL-BCN/git-hooks-local.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$
```

The easiest way to see if .git/hooks files are transferred to remote repository is to clone the remote repository and analyse files.

Note that we have to give the cloned repository a different name.

In the .git/hooks directory of the cloned repository we only see the default hooks and none of the custom files previously created proving that .git/hooks files are not pushed to remote.

HOOKS ARE ONLY LOCAL AND NOT PUSHED TO REMOTE.

A workaround for this is to create the hooks in working directory the create symbolic links in the hooks directory pointing to the hook files in working directory, However each collaborator must then add the symbolic links in their local repository.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local-from-remote
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local (master)
$ cd ..

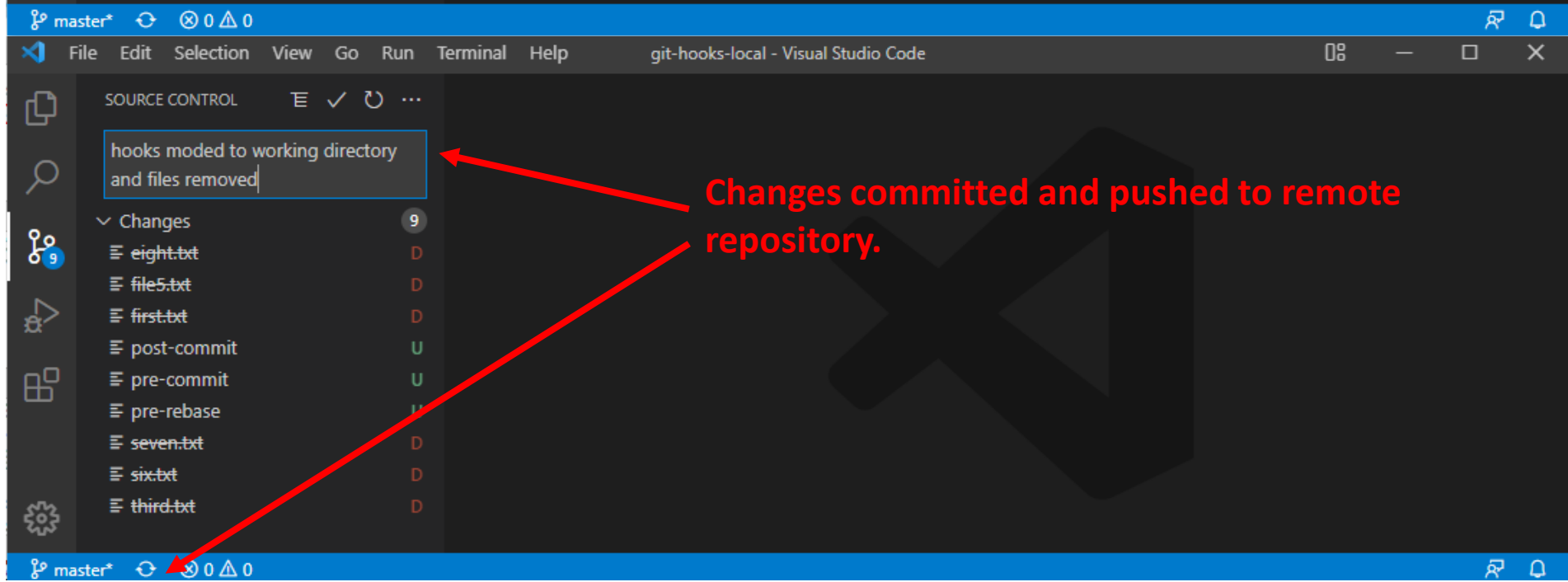
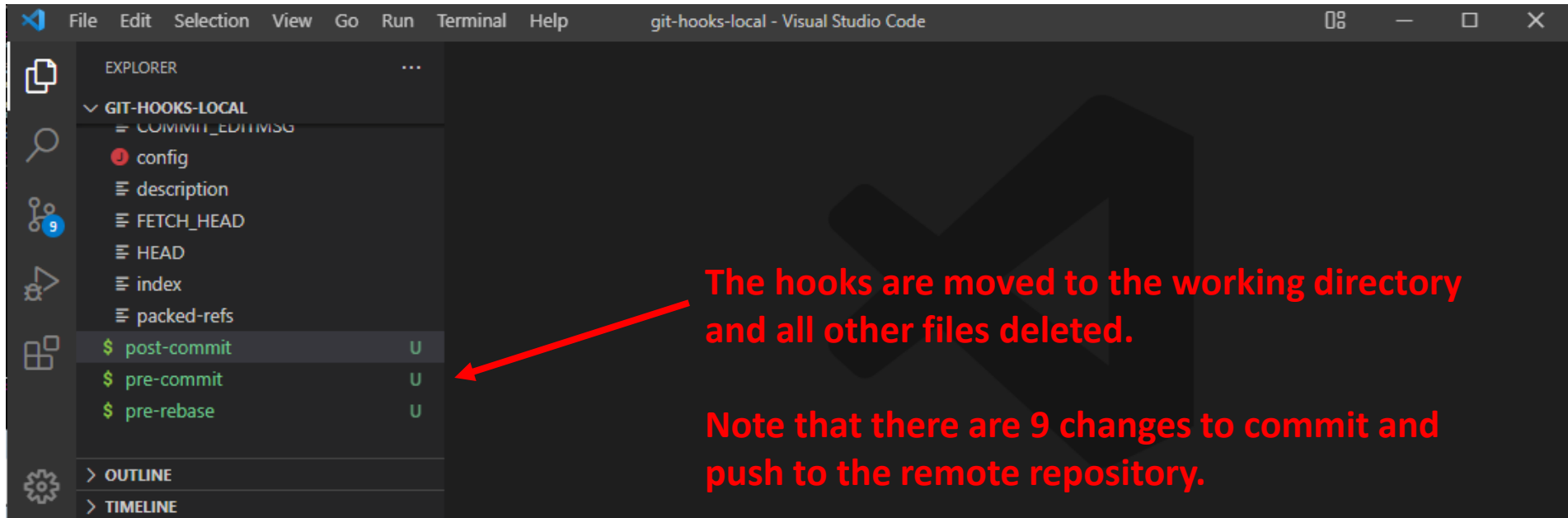
ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone git@github.com:EL-EL-BCN/git-hooks-local.git
fatal: destination path 'git-hooks-local' already exists and is not an empty directory.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone git@github.com:EL-EL-BCN/git-hooks-local.git git-hooks-local-from-remote
Cloning into 'git-hooks-local-from-remote'...
Enter passphrase for key '/c:/Users/ellio/.ssh/id_ed25519':
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (8/8), done.
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (4/4), done.
remote: Total 15 (delta 4), reused 15 (delta 4), pack-reused 0

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd git-hooks-local-from-remote/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-local-from-remote (master)
$ ls -la .git/hooks
total 53
drwxr-xr-x 1 ellio 197609  0 Mar 15 09:07 ./
drwxr-xr-x 1 ellio 197609  0 Mar 15 09:08 ../
-rwxr-xr-x 1 ellio 197609 478 Mar 15 09:07 applypatch-msg.sample*
-rwxr-xr-x 1 ellio 197609 896 Mar 15 09:07 commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 4655 Mar 15 09:07 fsmonitor-watchman.sample*
-rwxr-xr-x 1 ellio 197609 189 Mar 15 09:07 post-update.sample*
-rwxr-xr-x 1 ellio 197609 424 Mar 15 09:07 pre-applypatch.sample*
-rwxr-xr-x 1 ellio 197609 1643 Mar 15 09:07 pre-commit.sample*
-rwxr-xr-x 1 ellio 197609 416 Mar 15 09:07 pre-merge-commit.sample*
-rwxr-xr-x 1 ellio 197609 1374 Mar 15 09:07 pre-push.sample*
-rwxr-xr-x 1 ellio 197609 4898 Mar 15 09:07 pre-rebase.sample*
-rwxr-xr-x 1 ellio 197609 544 Mar 15 09:07 pre-receive.sample*
-rwxr-xr-x 1 ellio 197609 1492 Mar 15 09:07 prepare-commit-msg.sample*
-rwxr-xr-x 1 ellio 197609 2783 Mar 15 09:07 push-to-checkout.sample*
-rwxr-xr-x 1 ellio 197609 3650 Mar 15 09:07 update.sample*
```

Cleaning up the hooks in the local repository



Initialising Node js project

A remote repository was created in github.com with the following features:

- Name: git-hooks-nodejs
- Description: Nodejs application with Husky and commitlint
- No README.me file

A local folder is created with the same name and git is initialised.

Npm is also initialised with a **npm-init -y** command.

Now we see an empty directory with a .git directory and a package.json file.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ mkdir git-hooks-nodejs

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT
$ cd git-hooks-nodejs/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs
$ git init
Initialized empty Git repository in C:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs/.git/

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ node -v
v16.14.0

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ npm -v
8.3.1

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ npm init -y
Wrote to C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\package.json:

{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ ls -la
total 9
drwxr-xr-x 1 ellio 197609  0 Mar 15 10:38 ./
drwxr-xr-x 1 ellio 197609  0 Mar 15 10:37 ../
drwxr-xr-x 1 ellio 197609  0 Mar 15 10:37 .git/
-rw-r--r-- 1 ellio 197609 230 Mar 15 10:38 package.json

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$
```

Remote repository was added.

Master branch set to main locally.

Username set for this repository.

Email set for this repository.

Changes staged

Changes committed locally.

Local repository pushed to remote repository.

```
MINGW64:/c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ git remote add origin git@github.com:EL-EL-BCN/git-hooks-nodejs.git

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (master)
$ git branch -M main

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git config user.name EL-EL-BCN

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git config user.email elliottbcn@gmail.com

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git add .
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "initial commit"
[main (root-commit) 1adba5d] initial commit
1 file changed, 12 insertions(+)
create mode 100644 package.json

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git push -u origin main
Enter passphrase for key '/c/Users/ellio/.ssh/id_ed25519':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 376 bytes | 376.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:EL-EL-BCN/git-hooks-nodejs.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows the file structure of the 'GIT-HOOKS-NODEJS' project, with 'package.json' selected. The main editor displays the contents of 'package.json':

```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Below the editor, the TERMINAL panel is active, showing a bash shell. The user has run the command 'npm test', which has executed the script defined in package.json, resulting in the output: 'Error: no test specified'.

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm test

> git-hooks-nodejs@1.0.0 test
> echo "Error: no test specified" && exit 1

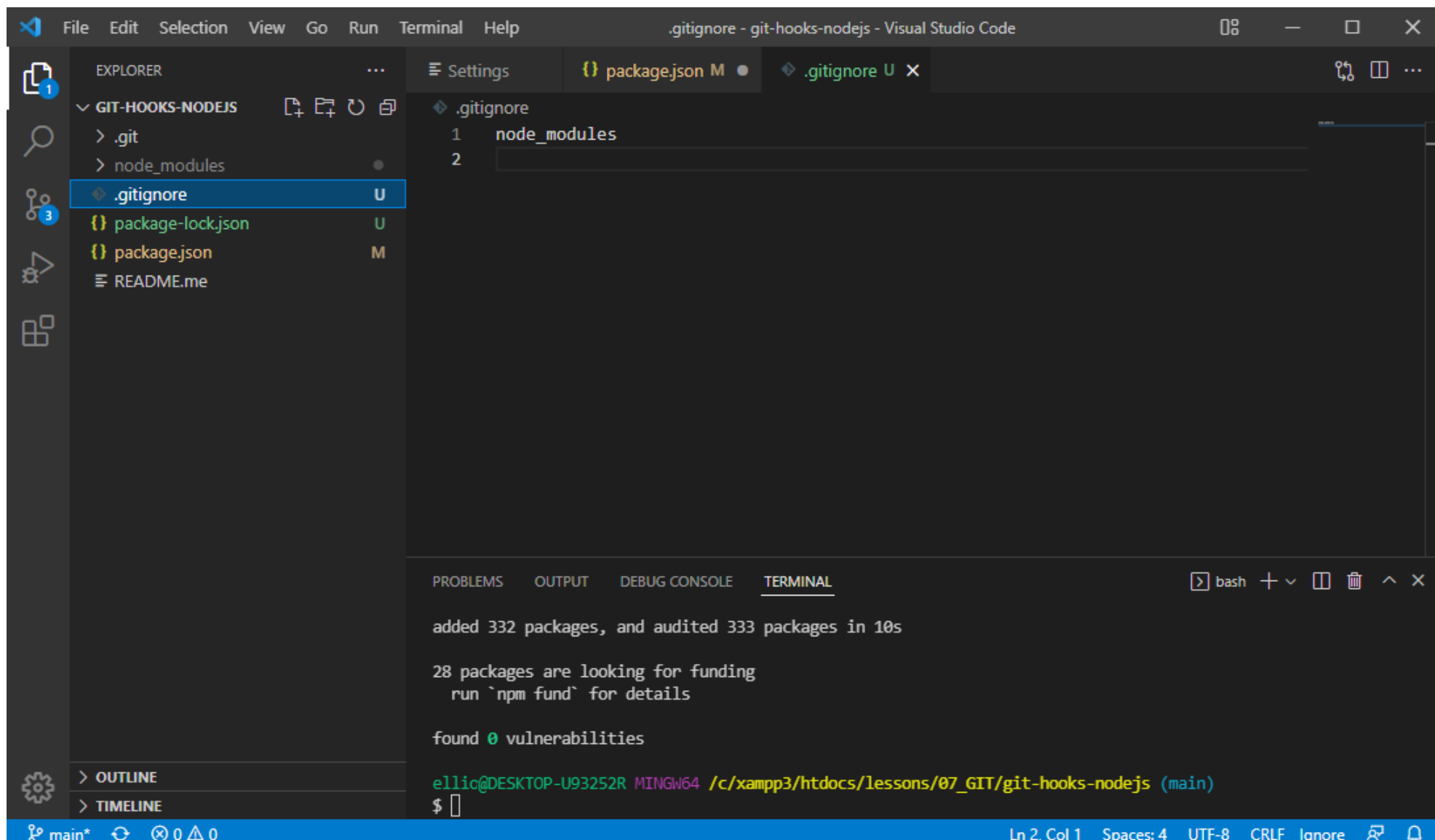
"Error: no test specified"

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

The status bar at the bottom indicates the current file is 'main', there are 0 errors and 0 warnings, and the cursor is at line 1, column 1.

If we try an npm test from the terminal it returns the error message specified in the package.json file. We need to install jest npm from the terminal with **npm install -D jest** command.

A gitignore file is added to exclude node-modules from synchronisation with the remote repository because these packages can be downloaded from npm by anyone that clones this repository.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of the 'GIT-HOOKS-NODEJS' project, with the '.gitignore' file selected. The main editor area shows the content of the '.gitignore' file, which contains the text 'node_modules'. The bottom panel shows the Terminal window with the following output:

```
added 332 packages, and audited 333 packages in 10s

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

The status bar at the bottom indicates the current file is 'main*' and shows 0 errors, 0 warnings, and 0 info messages. The right side of the status bar shows the current cursor position (Ln 2, Col 1), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), and the option to ignore the file.

Also package.json has been modified to include jest so we can change the scripts to jest.

The screenshot shows the Visual Studio Code interface with the `package.json` file open in the Editor. The Explorer on the left shows the file structure of the `git-hooks-nodejs` project, with `package.json` selected. The Editor displays the following JSON content:

```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "jest": "^27.5.1"
  }
}
```

Two red arrows point from the text above to the `"test": "jest"` line in the `scripts` section and the `"jest": "^27.5.1"` line in the `devDependencies` section.

The terminal at the bottom shows the output of an `npm audit` command:

```
added 332 packages, and audited 333 packages in 10s

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellie@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

The status bar at the bottom indicates the current file is `main*`, the encoding is `UTF-8`, and the line/col is `Ln 7, Col 18`.

Now npm test runs with jest.

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the file structure of the project 'GIT-HOOKS-NODEJS'. The files listed are `.git`, `node_modules`, `.gitignore`, `package-lock.json`, `package.json` (selected), and `README.me`.
- package.json:** The selected file is open in the editor. It contains the following JSON structure:

```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest"
  },
  "keywords": [],
}
```
- TERMINAL:** The terminal window shows the command `$ npm test` being executed. The output is as follows:

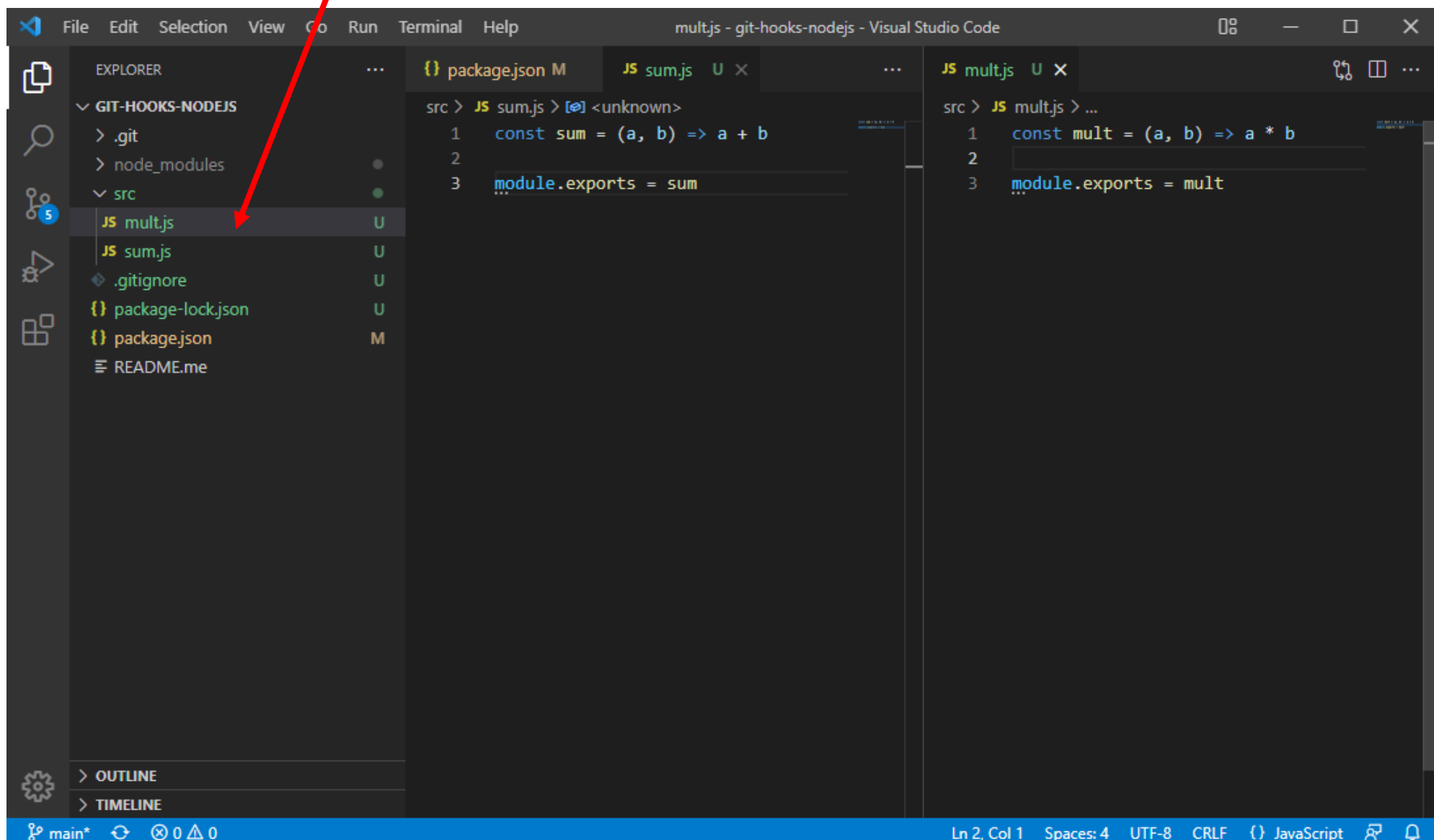
```
ellio@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm test


> git-hooks-nodejs@1.0.0 test
> jest

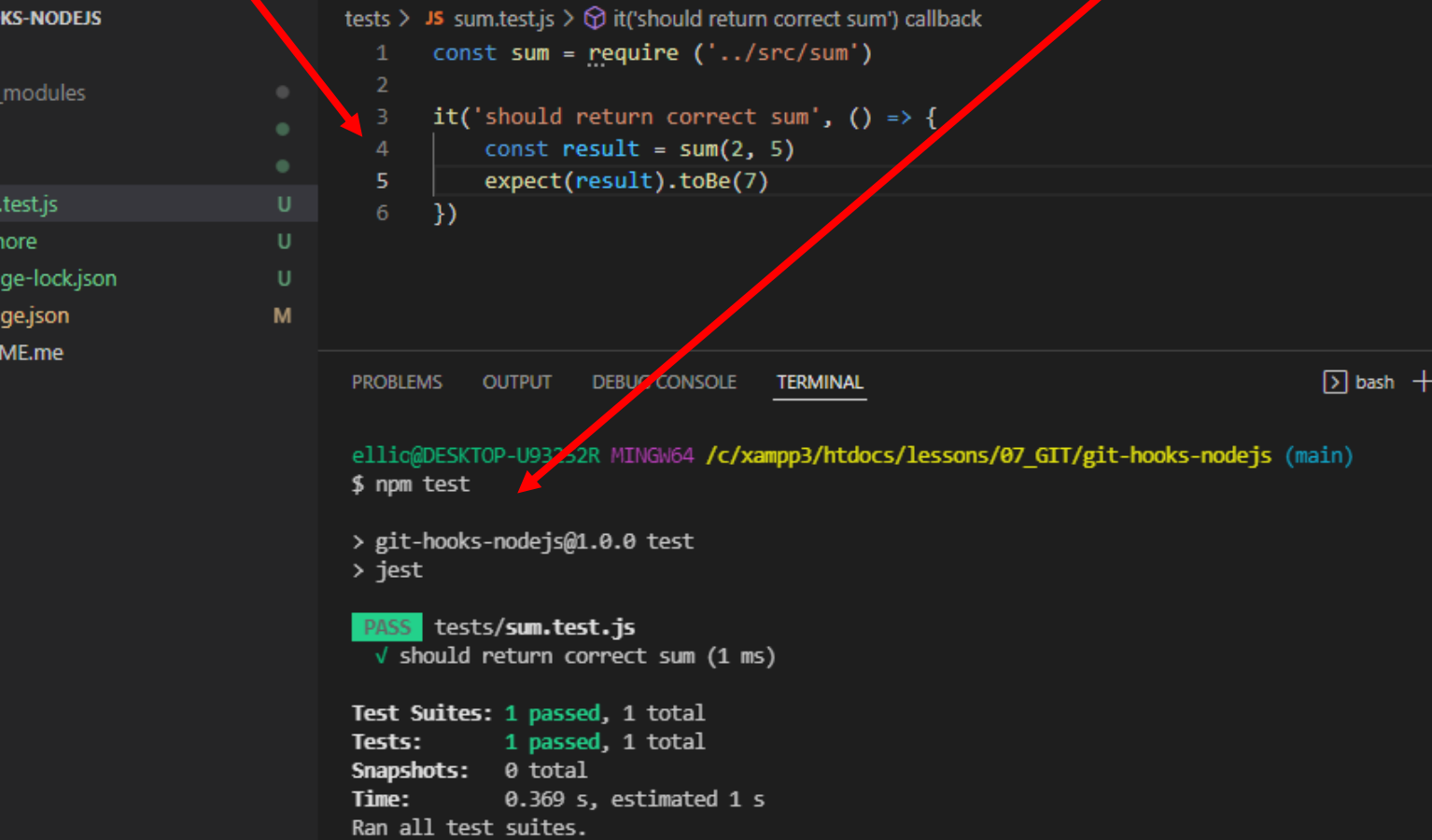
No tests found, exiting with code 1
Run with `--passWithNoTests` to exit with code 0
In C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs
  2 files checked.
    testMatch: **/_tests_/**/*.[jt]s?(x), **/?(*.)+(spec|test).[tj]s?(x) - 0 matches
    testPathIgnorePatterns: \\node_modules\\ - 2 matches
    testRegex: - 0 matches
    Pattern: - 0 matches
```

A red arrow points from the text 'Now npm test runs with jest.' to the `test` script in `package.json` and the terminal output.

Next a folder is created called src and two js files added called sum.js and mult.js. Each file has some simple js code to either sum or multiply two variables a and b.



Next a folder is created called tests and a js file that has a constant of sum that has values 2 and 5 that are passed to the src/sum.js file to get summed. The expected result is 7. When we now run NPM test jest will call this js file and run the test script. 



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Run and Debug panels. The Explorer panel shows the file structure of the 'git-hooks-nodejs' project, with 'sum.test.js' selected. The Source Control panel shows the changes in 'sum.test.js'. The Run and Debug panel shows the terminal output of the test command.

The terminal output shows the command `npm test` being executed, which runs the Jest test suite. The test suite 'tests/sum.test.js' passes, with the message 'should return correct sum (1 ms)'. The terminal also shows the Jest summary: 'Test Suites: 1 passed, 1 total', 'Tests: 1 passed, 1 total', 'Snapshots: 0 total', 'Time: 0.369 s, estimated 1 s', and 'Ran all test suites.'

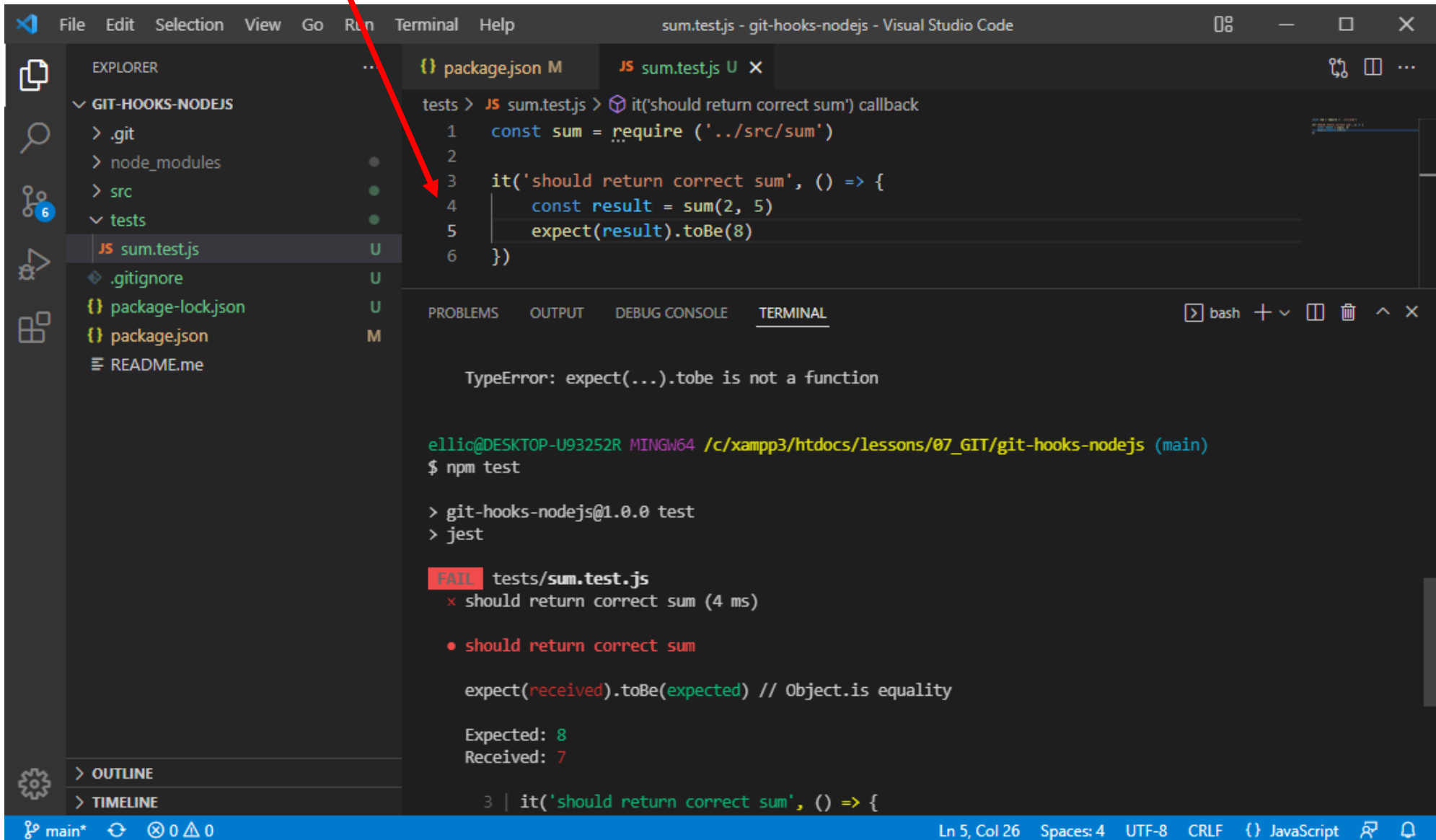
```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm test

> git-hooks-nodejs@1.0.0 test
> jest

PASS tests/sum.test.js
  ✓ should return correct sum (1 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.369 s, estimated 1 s
Ran all test suites.
```

If I modify the expected value to 8 and run npm test then it returns a fail.



```
File Edit Selection View Go Run Terminal Help
sum.test.js - git-hooks-nodejs - Visual Studio Code

EXPLORER
GIT-HOOKS-NODEJS
  .git
  node_modules
  src
  tests
    JS sum.test.js
  .gitignore
  package-lock.json
  package.json
  README.me

tests > JS sum.test.js > it('should return correct sum') callback
1 const sum = require ('../src/sum')
2
3 it('should return correct sum', () => {
4   const result = sum(2, 5)
5   expect(result).toBe(8)
6 })

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + v [ ] [ ] ^ x

TypeError: expect(...).tobe is not a function

ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm test

> git-hooks-nodejs@1.0.0 test
> jest

FAIL tests/sum.test.js
  x should return correct sum (4 ms)

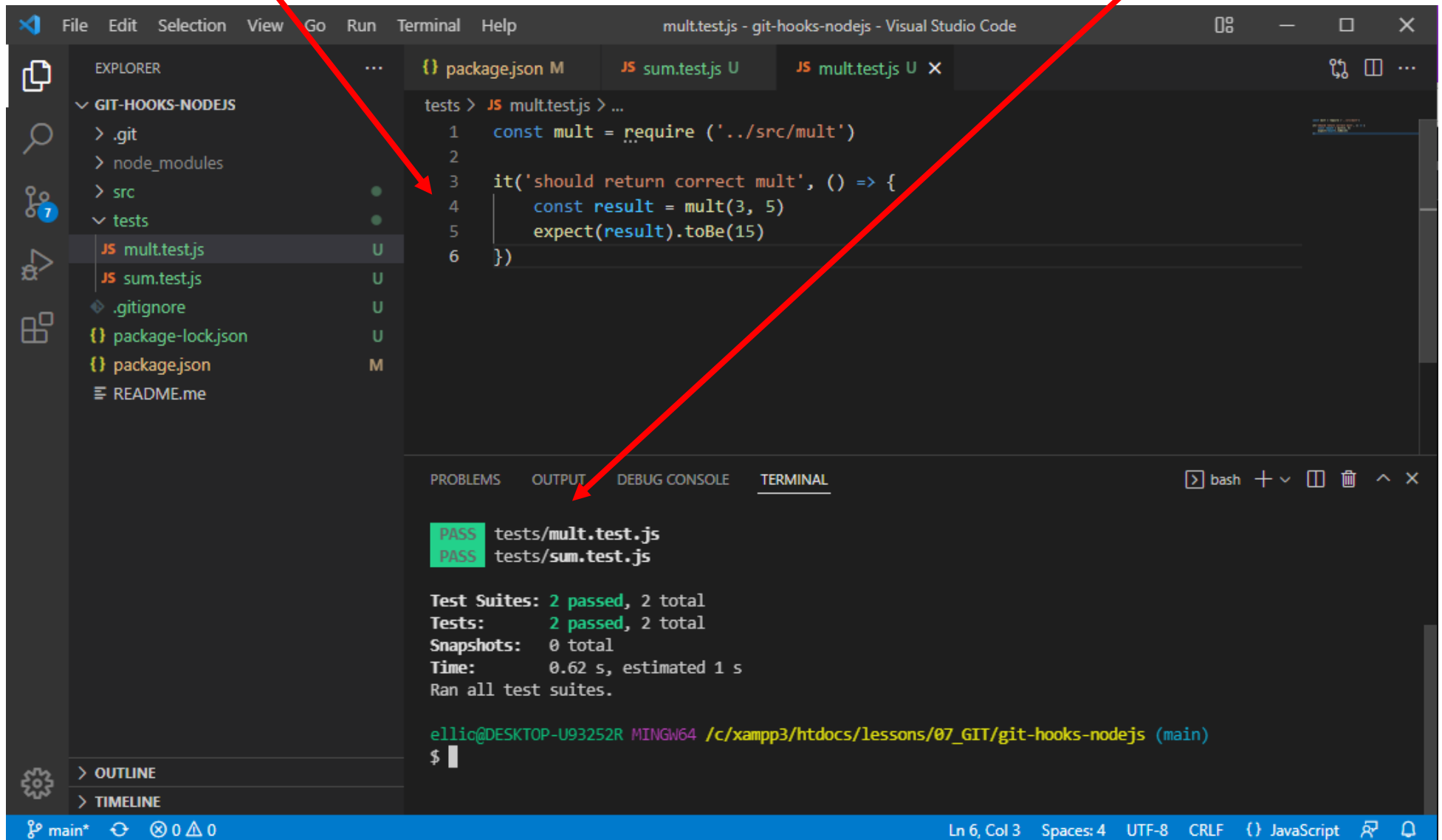
    • should return correct sum

      expect(received).toBe(expected) // Object.is equality

      Expected: 8
      Received: 7

3 | it('should return correct sum', () => {
```

In the tests folder a js file that has a constant of mult that has values 2 and 5 that are passed to the src/sum.js file to get summed. The expected result is 10. When I now run NPM test jest will call both js files and perform the scripts.



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Run and Debug panels. The Explorer panel shows the file structure of the 'git-hooks-nodejs' project, with the 'tests' folder expanded. The Source Control panel shows the 'mult.test.js' file. The Run and Debug panel shows the output of the Jest test, which passed. A red arrow points from the text above to the 'mult.test.js' file in the Explorer. Another red arrow points from the text above to the 'OUTPUT' tab in the Run and Debug panel.

```
File Edit Selection View Go Run Terminal Help
mult.test.js - git-hooks-nodejs - Visual Studio Code

EXPLORER
  GIT-HOOKS-NODEJS
    > .git
    > node_modules
    > src
    > tests
      JS mult.test.js
      JS sum.test.js
    .gitignore
    package-lock.json
    package.json
    README.me

package.json M
JS sum.test.js U
JS mult.test.js U X

tests > JS mult.test.js > ...
1  const mult = require ('../src/mult')
2
3  it('should return correct mult', () => {
4    const result = mult(3, 5)
5    expect(result).toBe(15)
6  })

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + v [] [] ^ x

PASS tests/mult.test.js
PASS tests/sum.test.js

Test Suites: 2 passed, 2 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 0.62 s, estimated 1 s
Ran all test suites.

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

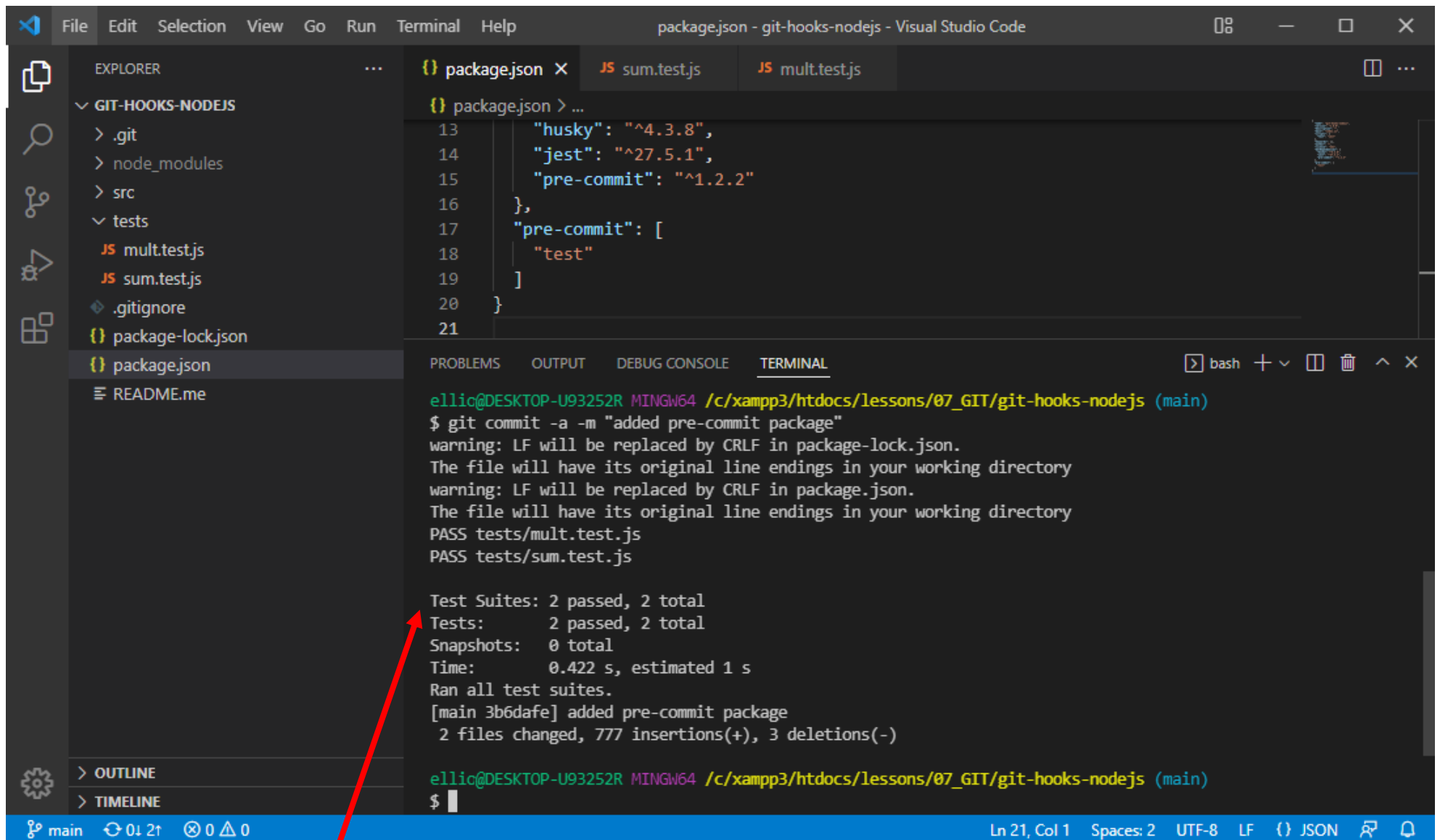
2) I then modify the package.json file to direct the pre-commit dependency to the script tests. Note that this code is an array (square brackets) of all the scripts we want to run for the pre-commit.

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure of the 'GIT-HOOKS-NODEJS' project, with 'package.json' selected. The main editor displays the 'package.json' file, which contains the following configuration:

```
{  
  "name": "git-hooks-nodejs",  
  "version": "1.0.0",  
  "description": "A collection of git hooks for nodejs projects",  
  "author": "Ellic",  
  "license": "ISC",  
  "devDependencies": {  
    "husky": "^4.3.8",  
    "jest": "^27.5.1",  
    "pre-commit": "^1.2.2"  
  },  
  "pre-commit": [  
    "test"  
  ]  
}
```

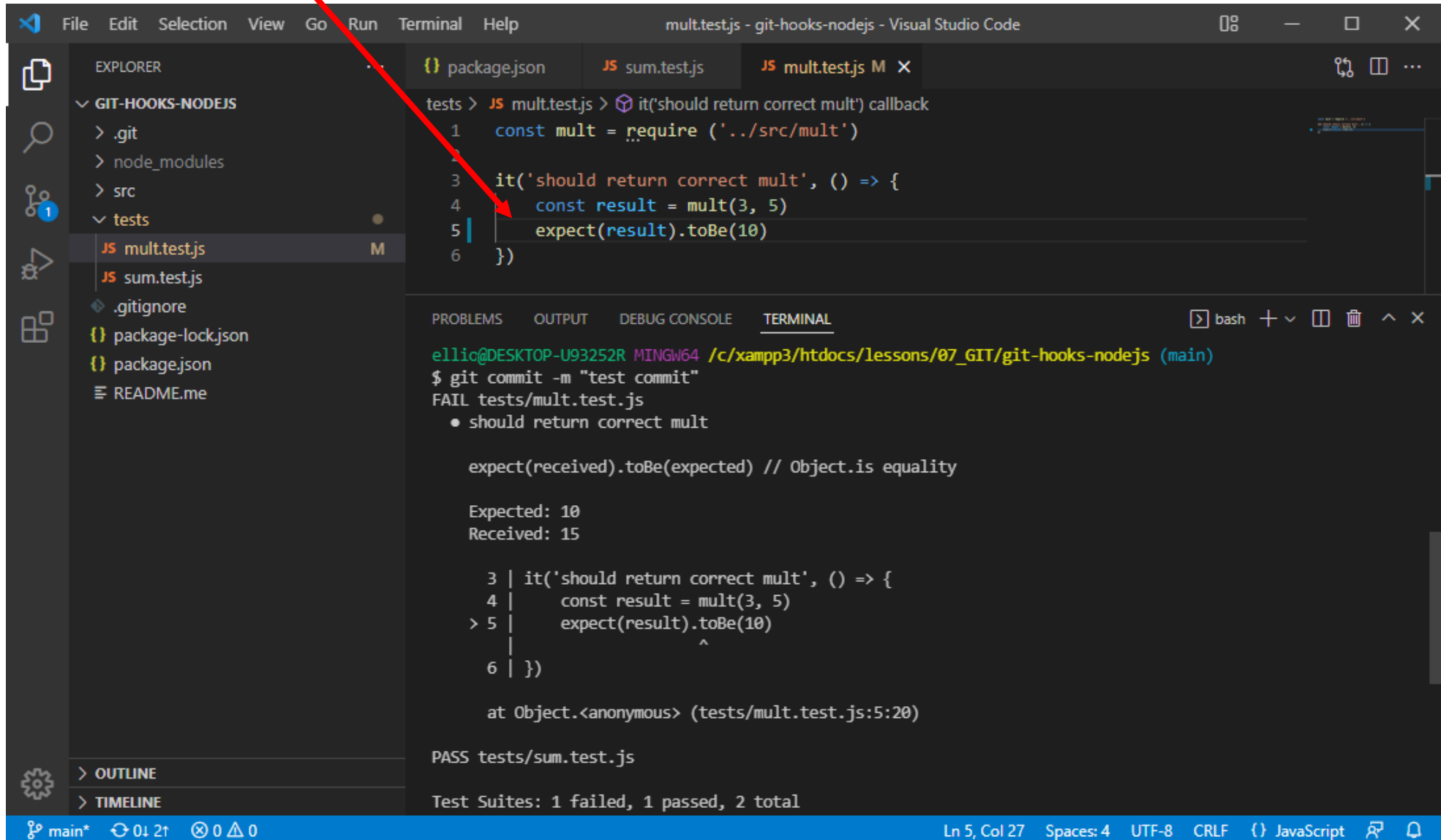
The terminal window at the bottom shows the command `$ npm i -D pre-commit` being executed. The output indicates that 18 packages were added and 377 packages were audited in 2 seconds. The terminal also shows the command `$ npm i -D pre-commit` being executed again.

1) I have installed a package called pre-commit from <https://www.npmjs.com/package/pre-commit>



Now when I commit it runs tests script which currently has two test scripts in it.

If I modify the expected result of the sum file an then try to commit the commit fails.



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the file structure of the 'GIT-HOOKS-NODEJS' project, including `.git`, `node_modules`, `src`, and `tests`. The `tests` folder is expanded, showing `mult.test.js` (marked with 'M' for modified) and `sum.test.js`.
- Editor:** Displays the content of `mult.test.js`. The code is as follows:

```
1 const mult = require ('../src/mult')
2
3 it('should return correct mult', () => {
4   const result = mult(3, 5)
5   expect(result).toBe(10)
6 })
```
- TERMINAL:** Shows the output of a `git commit` command. The commit fails due to a Jest test failure in `tests/mult.test.js`. The failure message is:

```
FAIL tests/mult.test.js
  ● should return correct mult

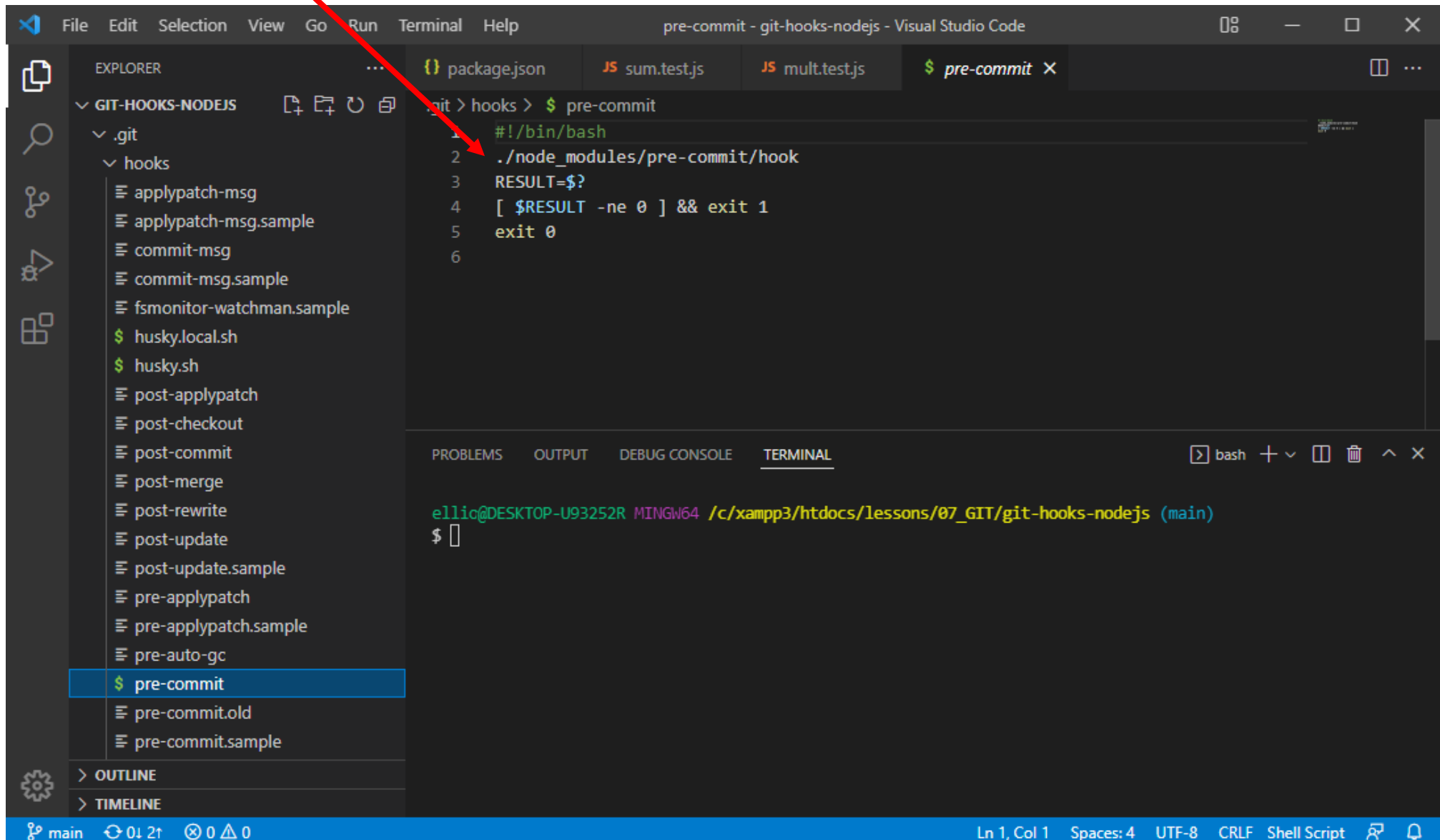
    expect(received).toBe(expected) // Object.is equality

    Expected: 10
    Received: 15

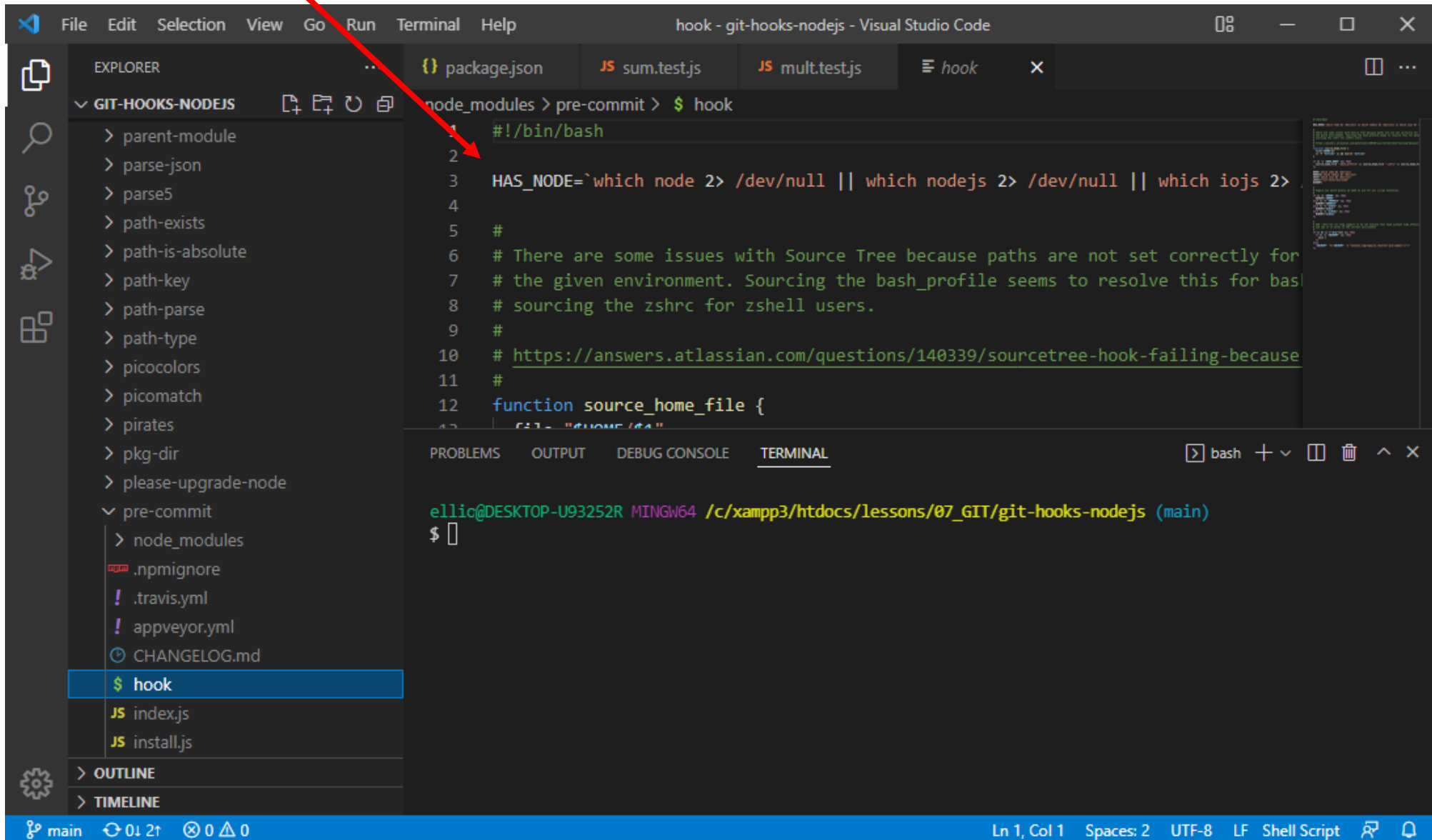
    3 | it('should return correct mult', () => {
    4 |   const result = mult(3, 5)
  > 5 |   expect(result).toBe(10)
      |                   ^
    6 | })

    at Object.<anonymous> (tests/mult.test.js:5:20)
```
- STATUS BAR:** Shows the current file is `main*`, with 0 errors and 0 warnings.

If I check in the `.git/hooks` directory I see a new file, `pre-commit` which was added. This file redirects to `./node_modules/pre-commit/hook`



The contents of the `./node_modules/pre-commit/hook` file is a bash script



The screenshot shows the Visual Studio Code interface with the file `hook` open in the editor. The file's contents are as follows:

```
#!/bin/bash
HAS_NODE=`which node 2> /dev/null || which nodejs 2> /dev/null || which iojs 2> /dev/null`
#
# There are some issues with Source Tree because paths are not set correctly for
# the given environment. Sourcing the bash_profile seems to resolve this for bas
# sourcing the zshrc for zshell users.
#
# https://answers.atlassian.com/questions/140339/sourcetree-hook-failing-because
#
function source_home_file {
    if [ -f "$HOME/.bashrc" ]; then
        source "$HOME/.bashrc"
    fi
    if [ -f "$HOME/.zshrc" ]; then
        source "$HOME/.zshrc"
    fi
}
```

The terminal at the bottom shows the command `hook` being executed, resulting in a prompt `$`.

Cloning remote repository & verifying pre-commit hook

The git-hooks-nodejs local repository has had all changes committed and pushed to remote. (Not shown)

The git-hooks-nodejs is cloned from remote and named git-hooks-nodejs-from-remote.

Note that the src, tests and json files are present but not the ***node_modules*** directory. This is because it is excluded in gitignore.

```
MINGW64:/c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs-from-remote
ellio@DESKTOP-U93252R MINGW64 ~
$ cd /c:/xampp3/htdocs/lessons/07_GIT/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ ls
-my-first-git-hub-repository/  gitignore/
JavaScript-Bible-Basics/      hello-world/
create-vue/                   my-first-git-hub-repository/
detached-head/                my-learning/
first-project/                pull-requests/
git-hooks-local/              rebasing/
git-hooks-local-from-remote/  rebasing-with-squashing/
git-hooks-nodejs/             tags/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ git clone git@github.com:EL-EL-BCN/git-hooks-nodejs.git git-hooks-nodejs-from-remote
Cloning into 'git-hooks-nodejs-from-remote'...
Enter passphrase for key '/c:/Users/ellio/.ssh/id_ed25519':
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 21 (delta 4), reused 17 (delta 3), pack-reused 0
Receiving objects: 100% (21/21), 78.09 KiB | 195.00 KiB/s, done.
Resolving deltas: 100% (4/4), done.

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT
$ cd git-hooks-nodejs-from-remote/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs-from-remote (main)
$ ls -la
total 315
drwxr-xr-x 1 ellio 197609      0 Mar 16 08:15 ./
drwxr-xr-x 1 ellio 197609      0 Mar 16 08:14 ../
drwxr-xr-x 1 ellio 197609      0 Mar 16 08:15 .git/
-rw-r--r-- 1 ellio 197609     14 Mar 16 08:15 .gitignore
-rw-r--r-- 1 ellio 197609    211 Mar 16 08:15 README.me
-rw-r--r-- 1 ellio 197609  305280 Mar 16 08:15 package-lock.json
-rw-r--r-- 1 ellio 197609    346 Mar 16 08:15 package.json
drwxr-xr-x 1 ellio 197609      0 Mar 16 08:15 src/
drwxr-xr-x 1 ellio 197609      0 Mar 16 08:15 tests/

ellio@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs-from-remote (main)
$ |
```

npm install command can be used to install the node-modules folder rather than including it in the git files.

The screenshot shows the Visual Studio Code interface with the Explorer panel on the left, the Editor panel in the center, and the Terminal panel at the bottom. The Explorer panel shows the file structure of a project named 'GIT-HOOKS-NODEJS-FROM-REMOTE'. The Editor panel shows the 'package.json' file. The Terminal panel shows the execution of the 'npm install' command.

EXPLORER

- ✓ GIT-HOOKS-NODEJS-FROM-REMOTE
 - > .git
 - > node_modules
 - > src
 - > tests
 - ◆ .gitignore
 - { package-lock.json
 - { package.json
 - ≡ README.me

package.json

```
10  "author": " ",
11  "license": "ISC",
12  "devDependencies": {
13    "husky": "^4.3.8",
14    "jest": "^27.5.1",
15    "pre-commit": "^1.2.2"
16  },
17  "pre-commit": [
18    "test"
19  ]
20  }
21
```

TERMINAL

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs-from-remote (main)
$ npm install

added 372 packages, and audited 373 packages in 3s

37 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs-from-remote (main)
$
```

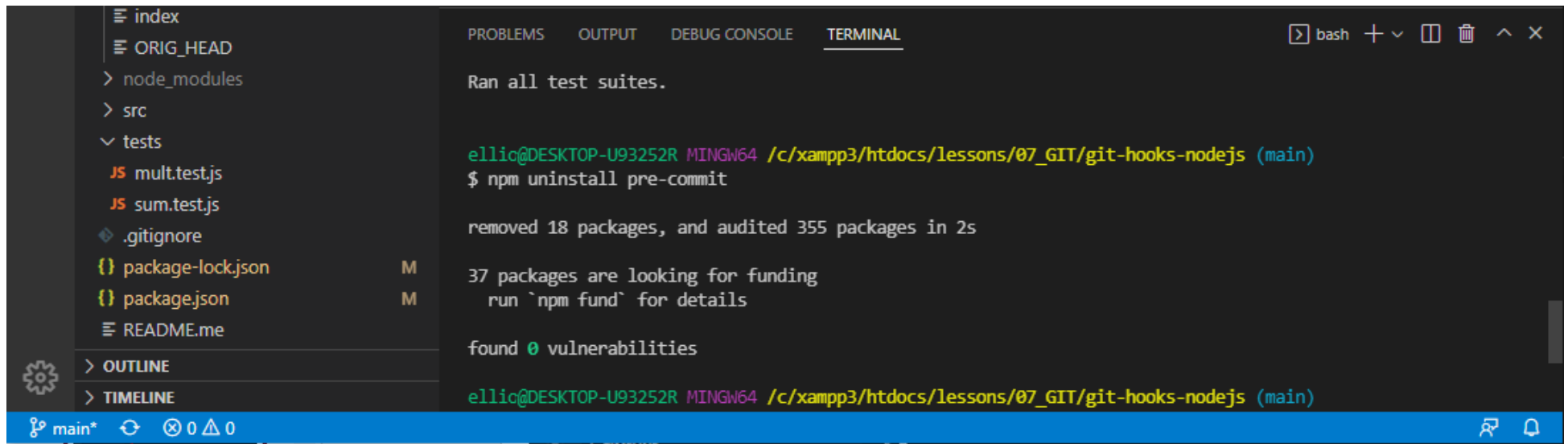
Now the configured .git/hooks/pre-hooks file is present and will run the pre-commit tests as before.

Replace pre-commit package with Husky

The NPM pre-commit package only does pre-commit hooks whereas husky lets us define many more hooks such as pre-commit, pre-push etc..

```
// package.json
{
  "husky": {
    "hooks": {
      "pre-commit": "npm test",
      "pre-push": "npm test",
      "...": "..."
    }
  }
}
```

The NPM pre-commit package is uninstalled with a **npm uninstall pre-commit** command



The screenshot shows a VS Code interface with a file explorer on the left and a terminal window on the right. The file explorer shows a project structure with files like `index`, `ORIG_HEAD`, `node_modules`, `src`, `tests` (containing `mult.test.js` and `sum.test.js`), `.gitignore`, `package-lock.json`, `package.json`, and `README.me`. The terminal window has tabs for `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, and `TERMINAL`. The `TERMINAL` tab is active, showing the command `npm uninstall pre-commit` and its output: `Ran all test suites.`, `removed 18 packages, and audited 355 packages in 2s`, `37 packages are looking for funding`, `run 'npm fund' for details`, and `found 0 vulnerabilities`. The terminal prompt is `ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)`.

```
index
ORIG_HEAD
> node_modules
> src
> tests
  JS mult.test.js
  JS sum.test.js
  .gitignore
  {} package-lock.json M
  {} package.json M
  README.me
> OUTLINE
> TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + v [] [X] ^ X

Ran all test suites.

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm uninstall pre-commit

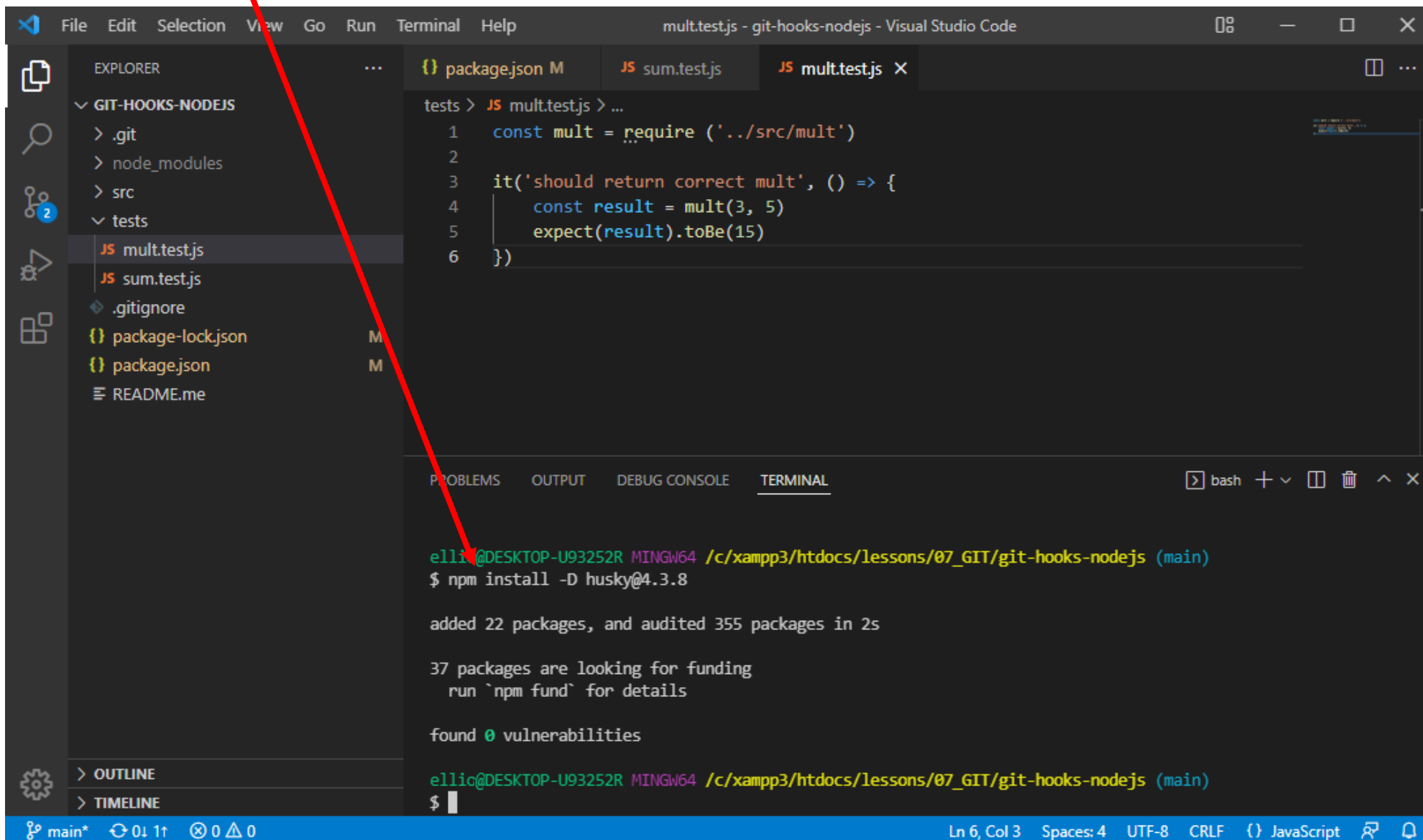
removed 18 packages, and audited 355 packages in 2s

37 packages are looking for funding
  run `npm fund` for details

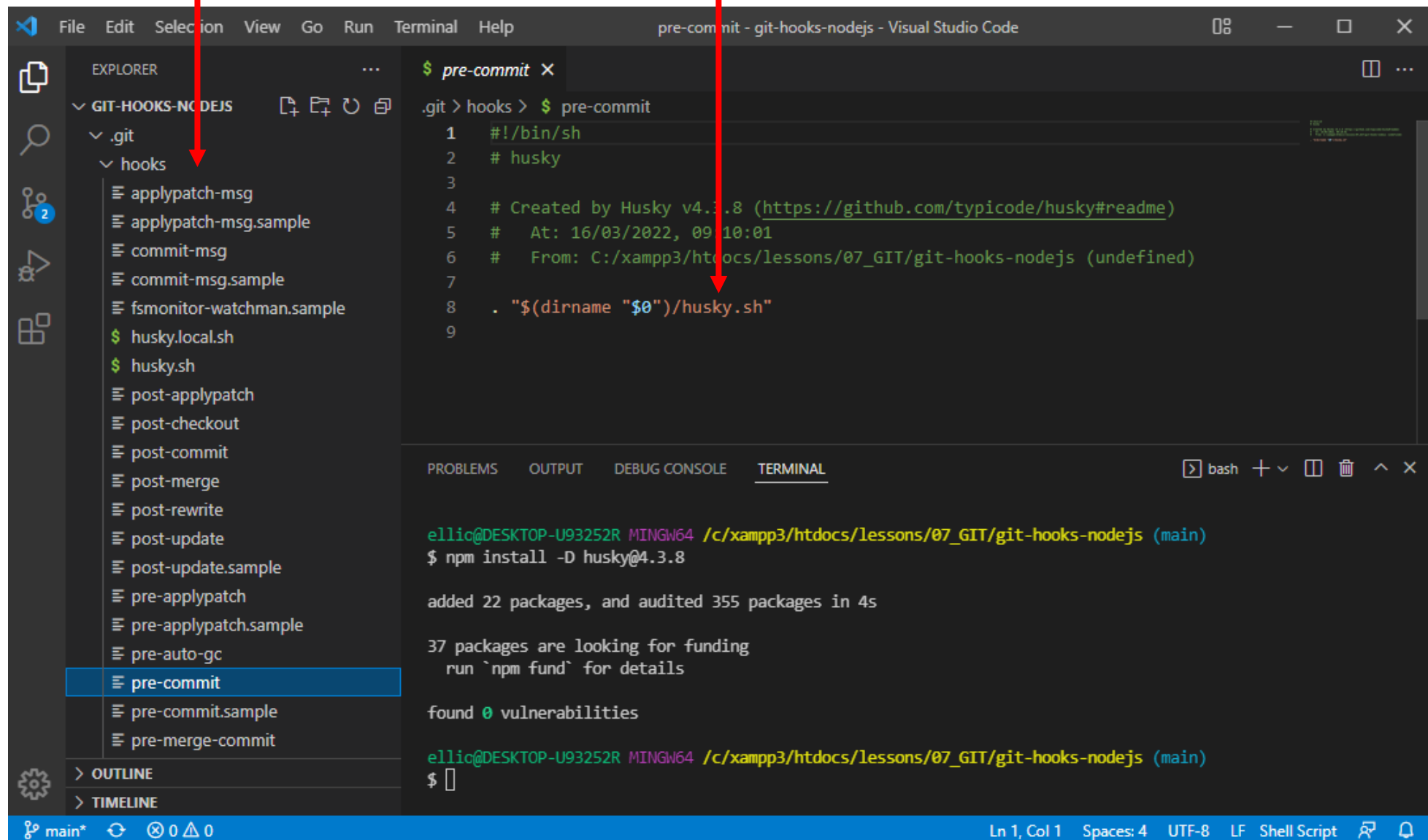
found 0 vulnerabilities

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
```

Next husky 4.3.8 is installed from <https://www.npmjs.com/package/husky/v/4.3.8>
With `npm install -D husky@4.3.8` command



Husky installs a package of additional husky hooks, which point to a husky.sh script which takes control of all the hooks.



The screenshot shows the Visual Studio Code interface with the Explorer panel on the left, the Editor panel in the center, and the Terminal panel at the bottom.

Explorer Panel: The file tree shows the project structure. The `.git` directory is expanded, and the `hooks` subdirectory is selected. The `pre-commit` file is highlighted.

Editor Panel: The `pre-commit` hook script is displayed. It contains the following content:

```
#!/bin/sh
# husky
# Created by Husky v4.3.8 (https://github.com/typicode/husky#readme)
# At: 16/03/2022, 09:10:01
# From: C:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (undefined)
. "$(dirname "$0")/husky.sh"
```

Terminal Panel: The terminal shows the command `npm install -D husky@4.3.8` being executed. The output indicates that 22 packages were added and 355 packages were audited. It also shows a list of vulnerabilities found by npm fund.

The package.json file is modified to replace pre-commit with husky and define the hooks with one hook of pre-commit added. Now when we commit, husky runs the same test scripts as before.

The screenshot shows the Visual Studio Code interface with the `package.json` file open. The file content is as follows:

```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest"
  },
  "husky": {
    "hooks": {
      "pre-commit": "npm test"
    }
  },
  "devDependencies": {
    "jest": "^27.5.1"
  }
}
```

The terminal window shows the following output:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "test commit"
husky > pre-commit (node v16.14.0)

> git-hooks-nodejs@1.0.0 test
> jest

PASS tests/mult.test.js
PASS tests/sum.test.js

Test Suites: 2 passed, 2 total
Tests:      2 passed, 2 total
Snapshots:  0 total
Time:       0.375 s, estimated 1 s
Ran all test suites.
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
```

If I change the mult.js file to make const mult equal to $a * b * b$ and save this file then try and commit, I get a test condition failure. I know that husky is working fine.

The screenshot shows the Visual Studio Code interface with the following components:

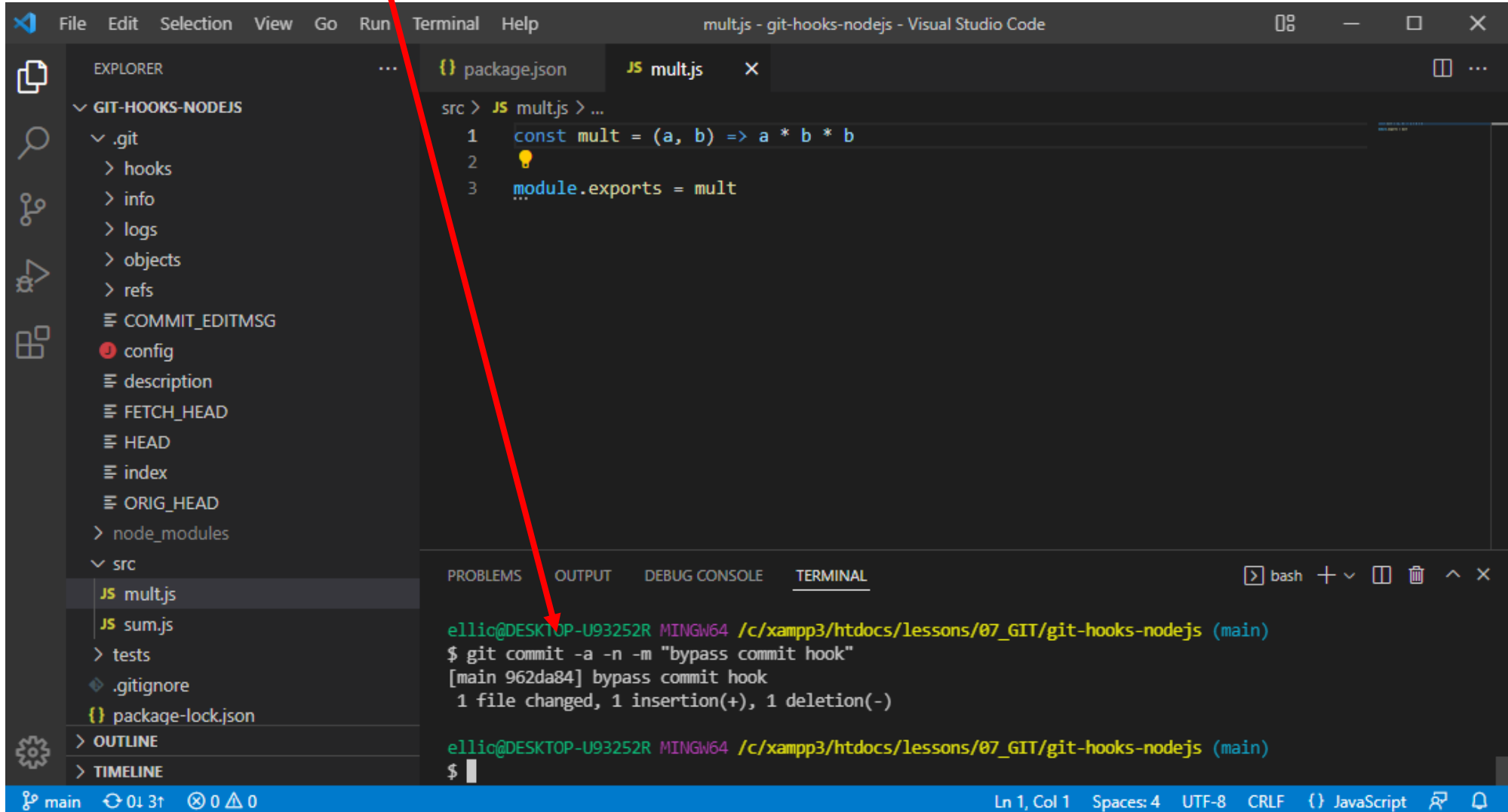
- Explorer:** Shows the file structure of the project, including `src` and `tests` directories. The `mult.js` file is highlighted.
- Source Control:** Shows the status of the `mult.js` file as modified (M).
- Editor:** Displays the content of `mult.js`:

```
src > JS mult.js > ...  
1 | const mult = (a, b) => a * b * b  
2 |  
3 | module.exports = mult
```
- Terminal:** Shows the output of a git commit command, which triggers a pre-commit hook (husky) and runs a test. The test fails:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)  
$ git commit -a -m "test message"  
husky > pre-commit (node v16.14.0)  
  
> git-hooks-nodejs@1.0.0 test  
> jest  
  
FAIL tests/mult.test.js  
  • should return correct mult  
  
    expect(received).toBe(expected) // Object.is equality  
  
    Expected: 15  
    Received: 75  
  
    3 | it('should return correct mult', () => {  
    4 |       const result = mult(3, 5)  
> 5 |       expect(result).toBe(15)  
      |                           ^  
    6 | })
```

A red arrow points from the text above to the `const mult` line in the editor.

It is possible to bypass the commit hooks using a no verify option with the command `git commit -a -n -m "commit message"` Note how husky is not triggered to perform any tests. We can even push these changes to the remote repository.



The screenshot shows the Visual Studio Code interface with the Explorer view on the left, the Editor view in the center, and the Terminal view at the bottom. The Explorer view shows the file structure of a project named 'GIT-HOOKS-NODEJS'. The Editor view shows the file 'mult.js' with the following code:

```
src > JS mult.js > ...  
1 const mult = (a, b) => a * b * b  
2   
3 module.exports = mult
```

The Terminal view shows the command prompt with the following commands and output:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)  
$ git commit -a -n -m "bypass commit hook"  
[main 962da84] bypass commit hook  
1 file changed, 1 insertion(+), 1 deletion(-)  
  
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)  
$
```

A red arrow points from the text in the first block to the command `git commit -a -n -m "bypass commit hook"` in the terminal.

So if the hooks can be bypassed locally and contributors allowed to push then what is the point of git hooks? Well it is expected that contributors do not use the `--no-verify` option with commit messages and abide by the rules specified in the hooks. Below you can see that with a no verify option the commit which might contain bad or unverified code has made it into the remote repository.

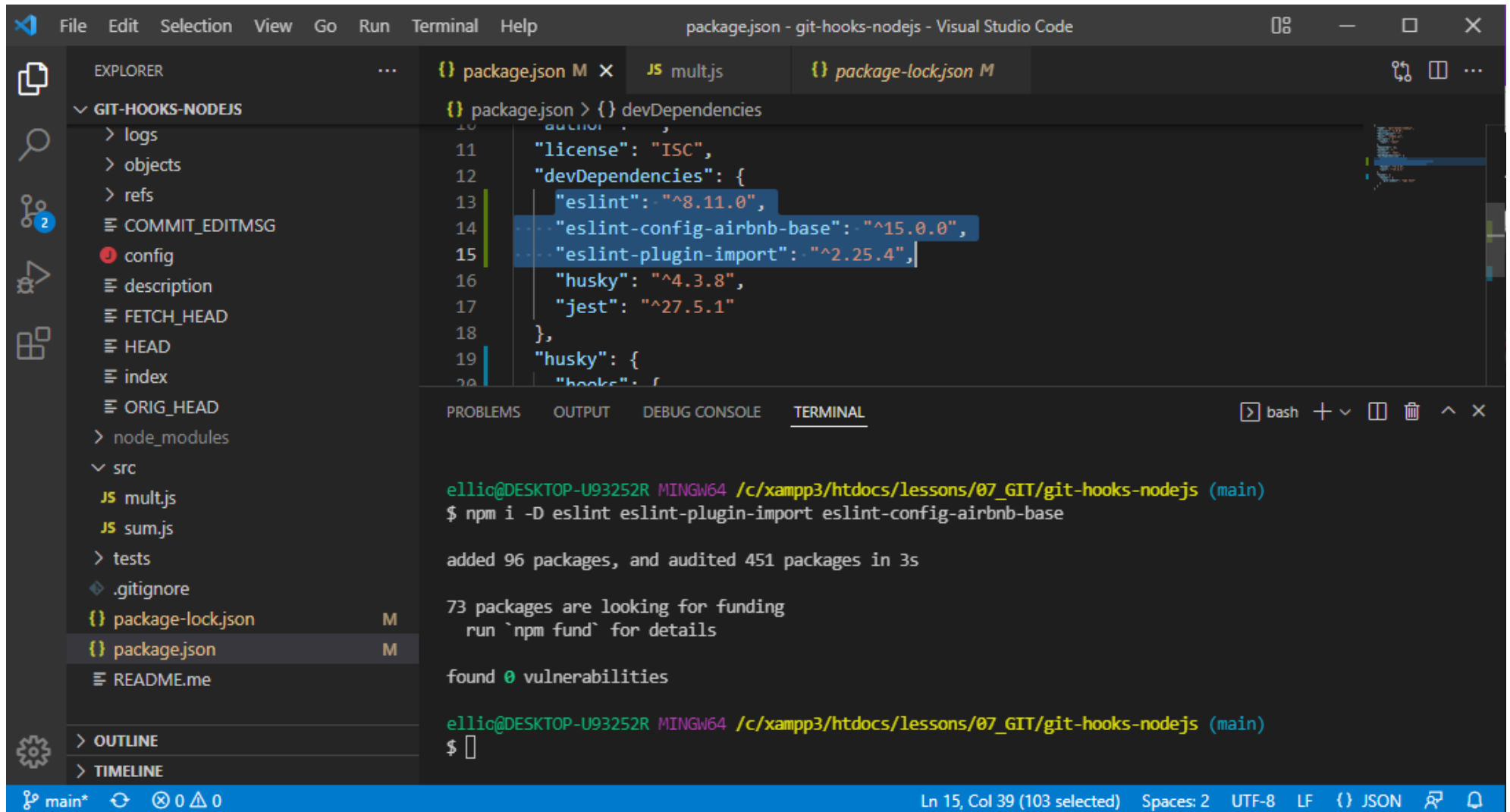
To overcome this we should add an additional layer of protection from the bad code in our central repository and run checks there for every pushed change or every opened pull request. Of course main branch and say release branch should also be protected from unauthorised merges.

The screenshot shows the GitHub interface for the repository 'EL-EL-BCN / git-hooks-nodejs'. The commit history for the 'main' branch is displayed, showing three commits from March 16, 2022. A red arrow points to the first commit, 'bypass commit hook', which was committed 6 minutes ago. The commit message is 'bypass commit hook' and the commit hash is '962da84'. The other two commits are 'pre-commit replaced with husky' (hash 'd4801d8', committed 20 minutes ago) and 'replaced pre-commit with husky' (hash 'd582dec', committed 22 minutes ago).

Commit Message	Author	Committed	Hash
bypass commit hook	EL-EL-BCN	6 minutes ago	962da84
pre-commit replaced with husky	EL-EL-BCN	20 minutes ago	d4801d8
replaced pre-commit with husky	EL-EL-BCN	22 minutes ago	d582dec

Run linter in the pre-commit hook

I will install eslint which checks JS code as well as a couple of other packages using the command `npm i -D eslint eslint-plugin-import eslint-config-airbnb-base`



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Terminal panels. The Explorer panel on the left shows the project structure for 'GIT-HOOKS-NODEJS', including files like 'logs', 'objects', 'refs', 'COMMIT_EDITMSG', 'config', 'description', 'FETCH_HEAD', 'HEAD', 'index', 'ORIG_HEAD', 'node_modules', 'src', 'mult.js', 'sum.js', 'tests', '.gitignore', 'package-lock.json', 'package.json', and 'README.me'. The Source Control panel shows the 'package.json' file with changes. The Terminal panel at the bottom shows the command `npm i -D eslint eslint-plugin-import eslint-config-airbnb-base` being executed, resulting in the installation of 96 packages and 73 packages looking for funding. The status bar at the bottom indicates the current file is 'package.json' at line 15, column 39.

```
package.json > {} devDependencies
10   "author": "ellie",
11   "license": "ISC",
12   "devDependencies": {
13     "eslint": "^8.11.0",
14     "eslint-config-airbnb-base": "^15.0.0",
15     "eslint-plugin-import": "^2.25.4",
16     "husky": "^4.3.8",
17     "jest": "^27.5.1"
18   },
19   "husky": {
20     "hooks": {
```

```
ellie@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm i -D eslint eslint-plugin-import eslint-config-airbnb-base

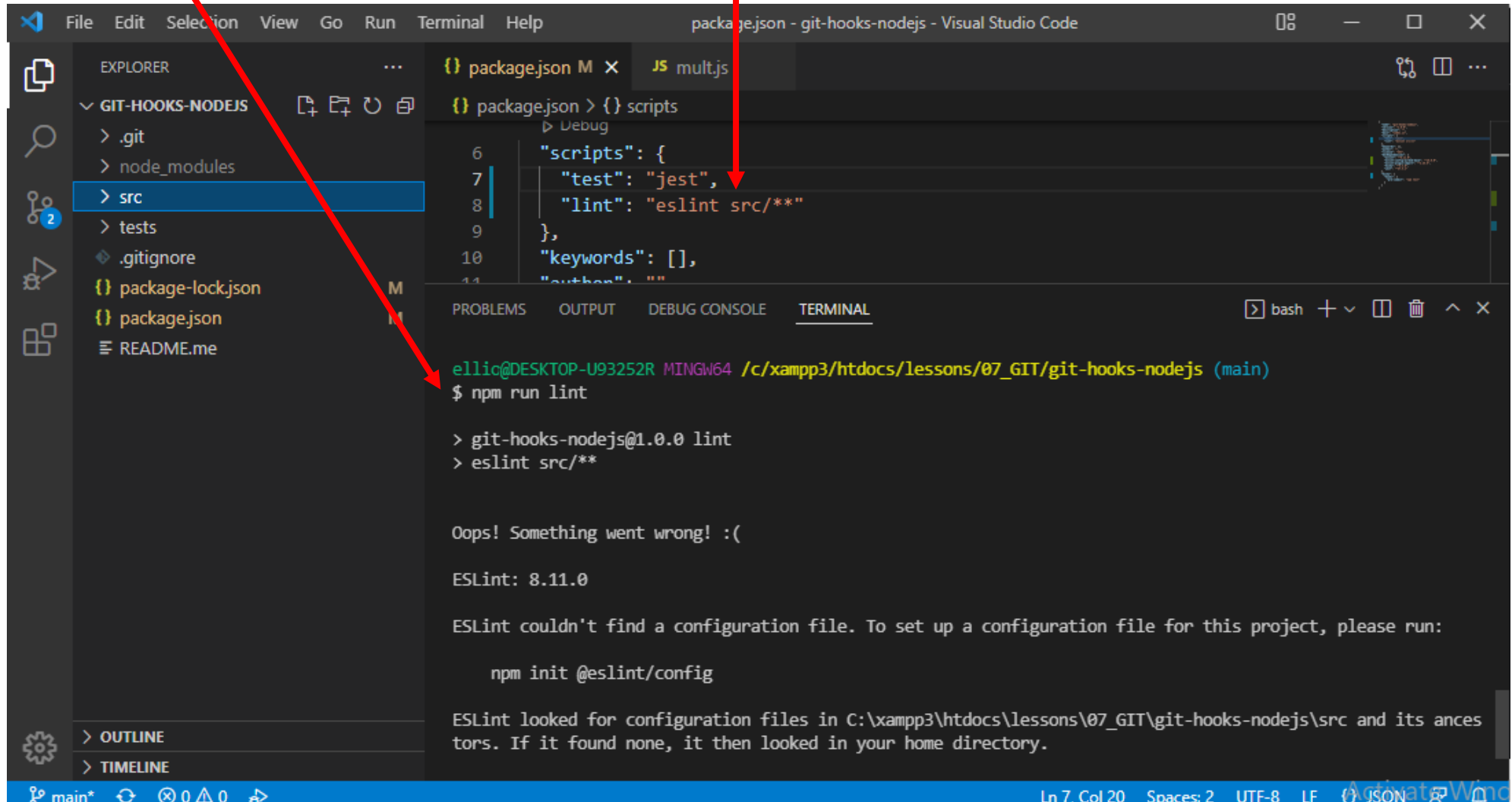
added 96 packages, and audited 451 packages in 3s

73 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellie@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

Now the package.json file is modified to add the eslint script and saved then tested using npm run lint command. Note that it fails because of a missing config file and advises a npm init @eslint/config command.



```
package.json M x JS mult.js
package.json > {} scripts
  > Debug
6   "scripts": {
7     "test": "jest",
8     "lint": "eslint src/**"
9   },
10  "keywords": [],
11  "author": ""

```

```
ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm run lint

> git-hooks-nodejs@1.0.0 lint
> eslint src/**

Oops! Something went wrong! :(

ESLint: 8.11.0

ESLint couldn't find a configuration file. To set up a configuration file for this project, please run:

  npm init @eslint/config

ESLint looked for configuration files in C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src and its ancestors. If it found none, it then looked in your home directory.
```

In the scripts section of package.json I have added lint-init.

I then run `npm init @eslint/config` command which gives me series of questions about this project.
Check syntax and find problems | CommonJS | Framework none | Typescript no | Code runs in node | Format is javascript.

Once setup is complete a `eslint.config.js` file appears in the root.

The screenshot shows the Visual Studio Code interface with the following components:

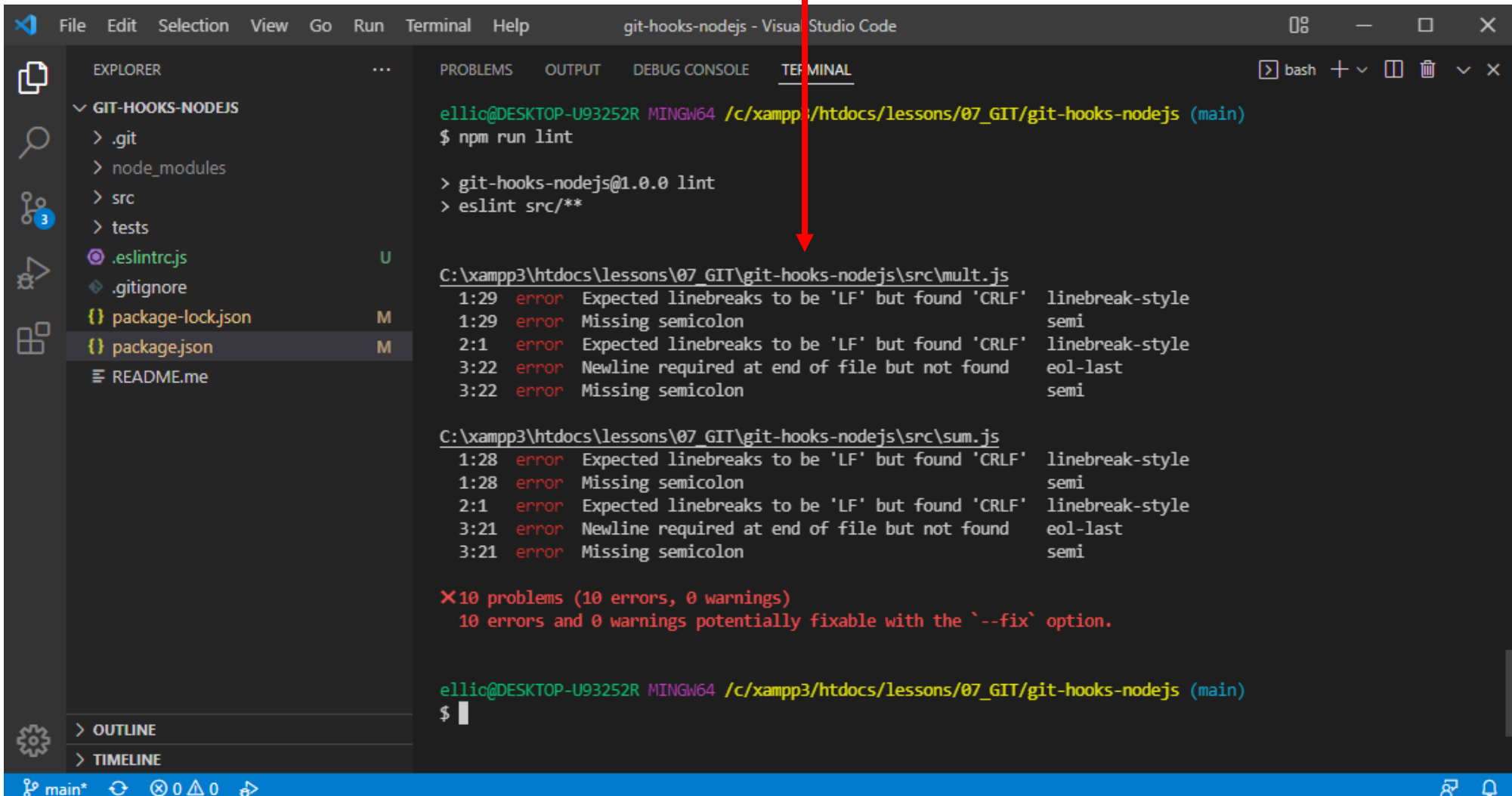
- EXPLORER:** Displays the file structure of the project. The `.git` folder is expanded, showing `node_modules`, `src`, and `tests`. The `.eslint.config.js` file is highlighted with a red arrow, indicating it has been created.
- package.json:** The file is open in the editor, showing the `scripts` section. The `lint-init` script is added, pointing to `eslint --init`, highlighted with a red arrow.
- TERMINAL:** Shows the output of the `npm init @eslint/config` command. The command is highlighted with a red arrow. The terminal displays a series of prompts and answers, confirming the setup of ESLint with the following configuration: `problems` for linting, `commonjs` for modules, `none` for framework, `No` for TypeScript, `browser` for where code runs, and `JavaScript` for the config file format. The final message states: "Successfully created .eslint.config.js file in C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs".

```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest",
    "lint": "eslint src/**",
    "lint-init": "eslint --init"
  },
  "keywords": [],
```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm init @eslint/config
✓ How would you like to use ESLint? - problems
✓ What type of modules does your project use? - commonjs
✓ Which framework does your project use? - none
✓ Does your project use TypeScript? - No / Yes
✓ Where does your code run? - browser
✓ What format do you want your config file to be in? - JavaScript
Successfully created .eslint.config.js file in C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```


Now if I run lint it finds all the errors in the js files.



The screenshot shows the Visual Studio Code interface with the Explorer, Problems, and Terminal panels. The Explorer panel on the left shows the file structure of a project named 'GIT-HOOKS-NODEJS', including files like '.git', 'node_modules', 'src', 'tests', '.eslintrc.js', '.gitignore', 'package-lock.json', 'package.json', and 'README.me'. The Problems panel in the center shows a list of 10 errors found by ESLint. The Terminal panel on the right shows the command prompt where the command 'npm run lint' was executed, resulting in the output of the linting process. A red arrow points from the text 'Now if I run lint it finds all the errors in the js files.' to the terminal output.

```
git-hooks-nodejs - Visual Studio Code

EXPLORER
  GIT-HOOKS-NODEJS
    > .git
    > node_modules
    > src
    > tests
    > .eslintrc.js
    > .gitignore
    > package-lock.json
    > package.json
    > README.me

PROBLEMS
  10 problems (10 errors, 0 warnings)
  10 errors and 0 warnings potentially fixable with the `--fix` option.

TERMINAL
  ellic@DESKTOP-U93252R MINGW64 /c:/xampp/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
  $ npm run lint
  > git-hooks-nodejs@1.0.0 lint
  > eslint src/**

  C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src\mult.js
    1:29 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
    1:29 error Missing semicolon semi
    2:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
    3:22 error Newline required at end of file but not found eol-last
    3:22 error Missing semicolon semi

  C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src\sum.js
    1:28 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
    1:28 error Missing semicolon semi
    2:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
    3:21 error Newline required at end of file but not found eol-last
    3:21 error Missing semicolon semi

  ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
  $
```

I have modified the pre-commit hook to run lint and then npm test.

When I try to commit I get multiple syntax errors because now it is checking the js syntax before allowing a commit.

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure of a project named 'GIT-HOOKS-NODEJS'. The file 'package.json' is selected and open in the editor. The editor shows the following JSON configuration for the pre-commit hook:

```
24 "husky": {  
25   "hooks": {  
26     "pre-commit": "npm run lint && npm test"  
27   }  
28 }  
29 }
```

A red arrow points to the line `"pre-commit": "npm run lint && npm test"`. Below the editor, the TERMINAL panel is active, showing the command `$ git commit -m "lint added"` and the resulting error message:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)  
$ git commit -m "lint added"  
husky > unexpected error SyntaxError: Unexpected token c in JSON at position 159  
    at JSON.parse (<anonymous>)  
    at Object.readPkg (C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\node_modules\husky\lib\read-pkg.js:11:17)  
    at getOldCommand (C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\node_modules\husky\lib\runner\index.js:16:26)  
    at Object.run [as default] (C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\node_modules\husky\lib\runner\index.js:62:24)  
    at run (C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\node_modules\husky\lib\runner\bin.js:11:40)  
    at Object.<anonymous> (C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\node_modules\husky\lib\runner\bin.js:19:1)  
    at Module._compile (node:internal/modules/cjs/loader:1103:14)  
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1155:10)  
    at Module.load (node:internal/modules/cjs/loader:981:32)  
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
```

A red arrow points to the command `$ git commit -m "lint added"`. The status bar at the bottom indicates the current file is 'package.json' with 2 lines modified, 0 lines added, and 0 lines deleted. The status bar also shows the current line is 26, column 38, with 2 spaces, UTF-8 encoding, LF line endings, and JSON syntax highlighting.

Once I have fixed all the code errors identified by lint then I am able to successfully commit.

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Displays the file structure of the 'GIT-HOOKS-NODEJS' project, including `.git`, `node_modules`, `src` (containing `mult.js` and `sum.js`), `tests`, `.eslintrc.js`, `.gitignore`, `package-lock.json`, `package.json`, and `README.me`.
- Editor:** Shows the content of `sum.js` in the `src` directory:

```
src > JS sum.js > ...
1  const sum = (a, b) => a + b;
2
3  module.exports = sum;
```
- TERMINAL:** Displays the output of a git commit and test execution:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -a -m "lint installed"
warning: LF will be replaced by CRLF in .eslintrc.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
husky > pre-commit (node v16.14.0)

> git-hooks-nodejs@1.0.0 lint
> eslint src/**

> git-hooks-nodejs@1.0.0 test
> jest

PASS tests/mult.test.js
PASS tests/sum.test.js

Test Suites: 2 passed, 2 total
Tests:      2 passed, 2 total
Snapshots:  0 total
Time:       0.549 s, estimated 1 s
Ran all test suites.
[main f2c8d31] lint installed
5 files changed, 2870 insertions(+), 104 deletions(-)
create mode 100644 .eslintrc.js

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```
- STATUS BAR:** Shows the current branch as `main`, commit hash `0111`, and file encoding `UTF-8` with `CRLF` line endings.

Eslint has automatic error fixing which can be configured by adding a line to the scripts. Now when I `npm run lint-fix` the `--fix` option auto corrects any syntax errors.

The screenshot shows the Visual Studio Code interface with the `package.json` file open. A red arrow points to the `"lint-fix": "eslint src/** --fix"` line in the `scripts` object. The `package.json` content is as follows:

```
{
  "scripts": {
    "test": "jest",
    "lint": "eslint src/**",
    "lint-fix": "eslint src/** --fix"
  },
  "keywords": []
}
```

The `PROBLEMS` panel shows 6 errors related to linebreak style (CRLF vs LF) in `mult.js` and `sum.js`:

```
C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src\mult.js
1:30 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
2:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
3:23 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style

C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src\sum.js
1:29 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
2:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
3:22 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
```

Below the errors, it states: `6 errors and 0 warnings potentially fixable with the `--fix` option.`

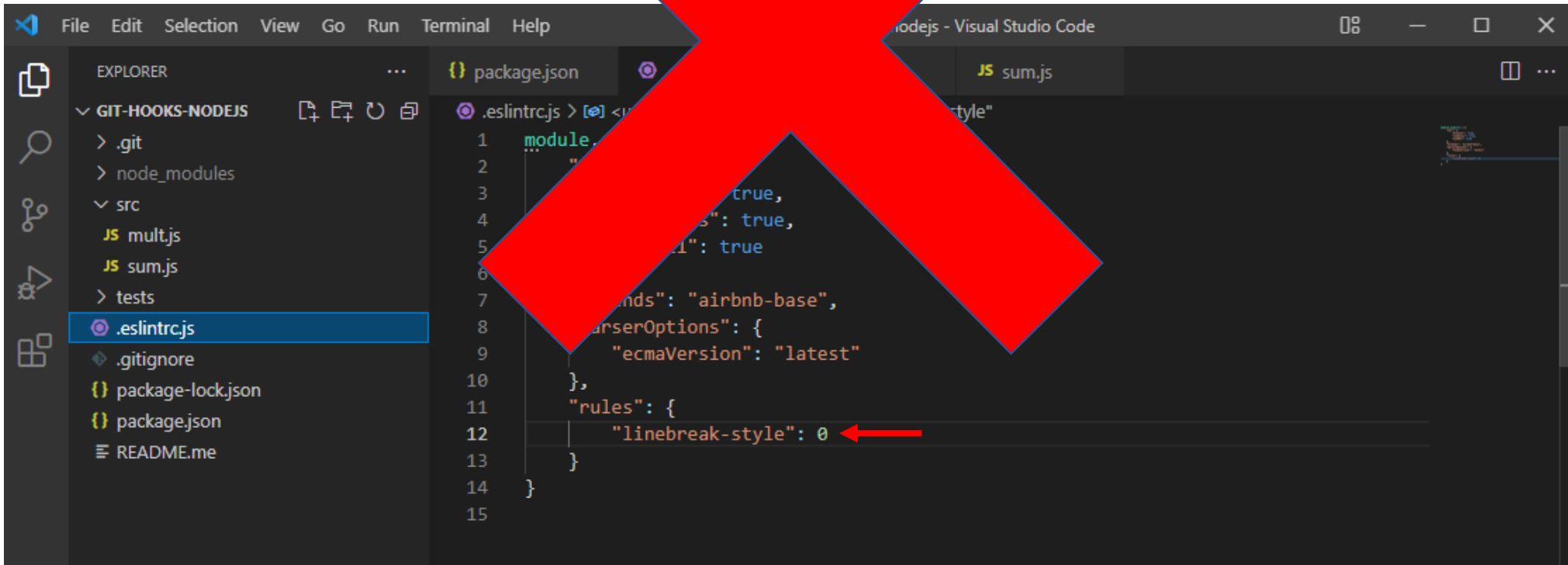
The `TERMINAL` panel shows the command `npm run lint-fix` being executed, which runs `git-hooks-nodejs@1.0.0 lint-fix` and `eslint src/** --fix`.

The status bar at the bottom indicates the current file is `package.json` at line 9, column 38, with 33 characters selected. The encoding is UTF-8 and the line ending is LF.

Lint line break rule workaround:

When developing with a lot of people all having different editors, VCS applications and operating systems it may occur that different line endings are written by either of the mentioned. The linebreaks (new lines) used in windows operating system are usually carriage returns (CR) followed by a line feed (LF) making it a carriage return line feed (CRLF) whereas Linux and Unix use a simple line feed (LF).

In the `eslinttrc.js` file I have added a rule which says ignore linebreak rules. The reason for this is that I do not want to automatically convert linebreaks into unix/linux format when I will always be using a windows machine and an editor that uses CRLF.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists the project files, with `.eslinttrc.js` selected. The main editor area displays the contents of `.eslinttrc.js`. A large red 'X' is drawn over the entire editor area. A red arrow points to the `"linebreak-style": 0` line in the `rules` object.

```
1 module.exports = {
2   "rules": {
3     "linebreak-style": 0,
4     "quotes": true,
5     "semi": true
6   },
7   "extends": "airbnb-base",
8   "parserOptions": {
9     "ecmaVersion": "latest"
10  },
11  "rules": {
12    "linebreak-style": 0
13  }
14 }
15
```

Lint staged package for checks on staged files only

I will install lint staged package with `npm i -D lint-staged` command

The screenshot shows the Visual Studio Code interface with the `package.json` file open. The `devDependencies` section is expanded, showing the installation of `lint-staged` at version `^12.3.6`. A red arrow points to the `"lint-staged": "^12.3.6"` line. A tooltip for `lint-staged` is visible, showing the latest version (12.3.6) and the GitHub repository link (<https://github.com/okonet/lint-staged#readme>). The terminal shows the command `npm i -D lint-staged` being executed, resulting in the package being added to the project.

Note how it is automatically added to dev dependencies

```
{
  "license": "ISC",
  "devDependencies": {
    "eslint": "^8.11.0",
    "eslint-config-airbnb": "^19.0.4",
    "eslint-config-airbnb-base": "^15.0.0",
    "eslint-plugin-import": "^2.25.4",
    "eslint-plugin-jsx-a11y": "^6.5.1",
    "eslint-plugin-react": "^7.31.1",
    "husky": "^4.3.0",
    "jest": "^27.5.1",
    "lint-staged": "^12.3.6"
  },
  "husky": {
    "hooks": {
      "pre-commit": "npm run lint && npm test"
    }
  }
}
```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm i -D lint-staged

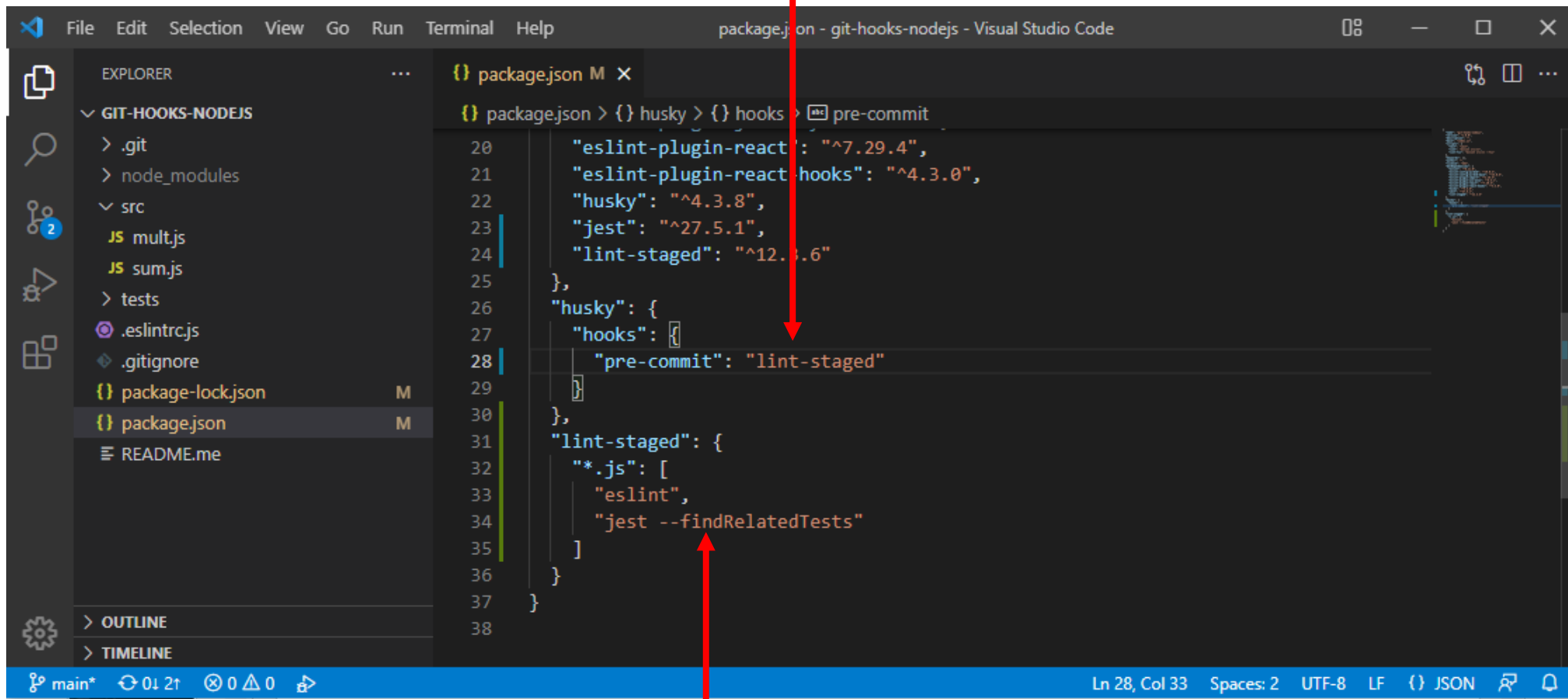
added 33 packages, and audited 512 packages in 5s

93 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

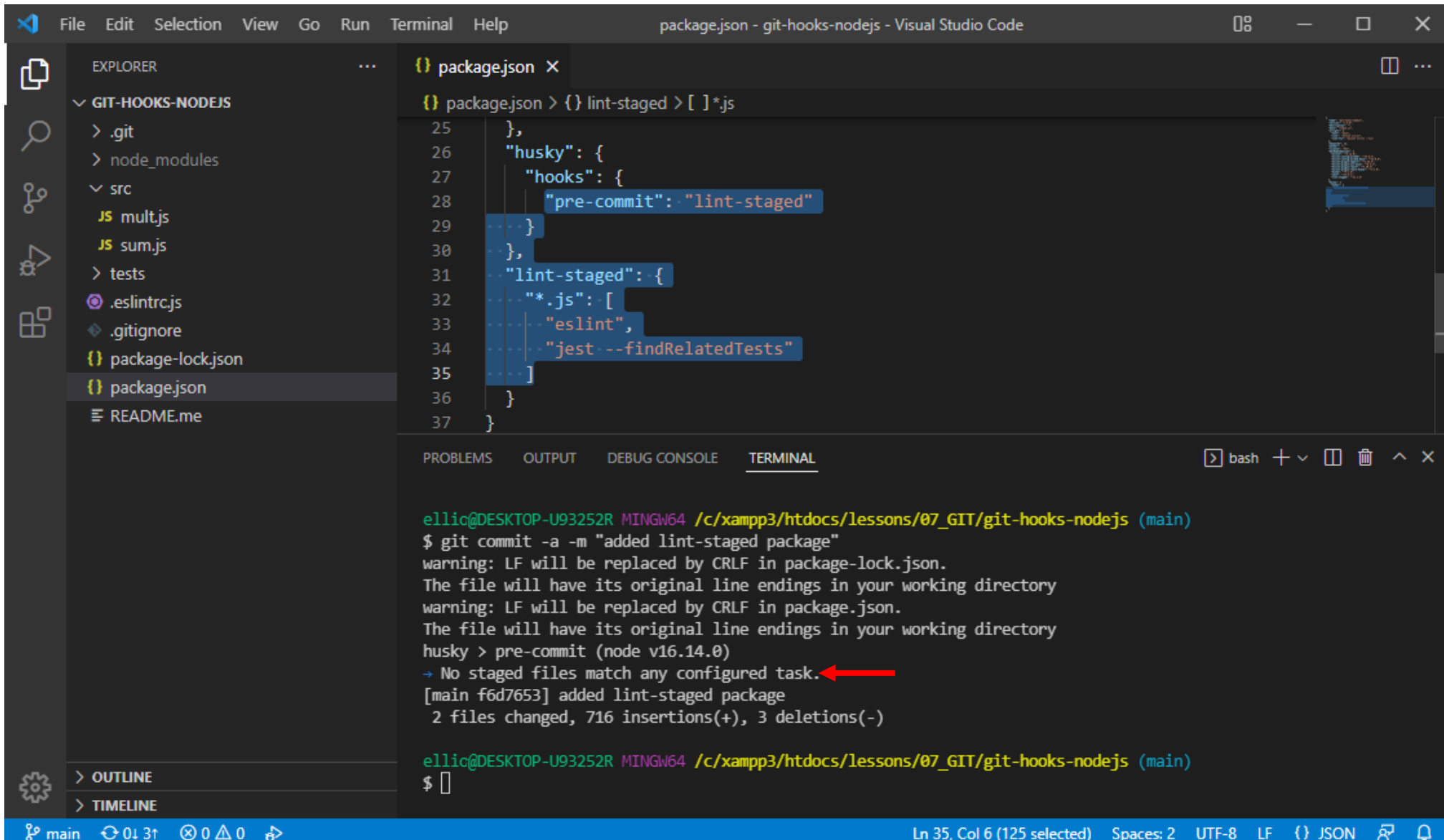
Now we want the pre-commit hook to run lint-staged script.



```
{
  "name": "git-hooks-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "jest",
    "lint": "eslint .",
    "pre-commit": "lint-staged"
  },
  "husky": {
    "hooks": {
      "pre-commit": "lint-staged"
    }
  },
  "lint-staged": {
    "*.js": [
      "eslint",
      "jest --findRelatedTests"
    ]
  }
}
```

I have added some more lines to define the lint-staged part. I want it to check files with a .js extension. In an array, I configure it to run eslint and jest, but this time only run jest on staged files (--findRelatedTests)

Previously, package.json and package-lock.json file were modified and saved. When I commit with staging the output includes the message “no staged files match any configured task” This is because lint-staged is configured to only check .js files.



The screenshot shows the Visual Studio Code interface with the `package.json` file open. The file content is as follows:

```
25 },
26 "husky": {
27   "hooks": {
28     "pre-commit": "lint-staged"
29   }
30 },
31 "lint-staged": {
32   "*.js": [
33     "eslint",
34     "jest --findRelatedTests"
35   ]
36 }
37 }
```

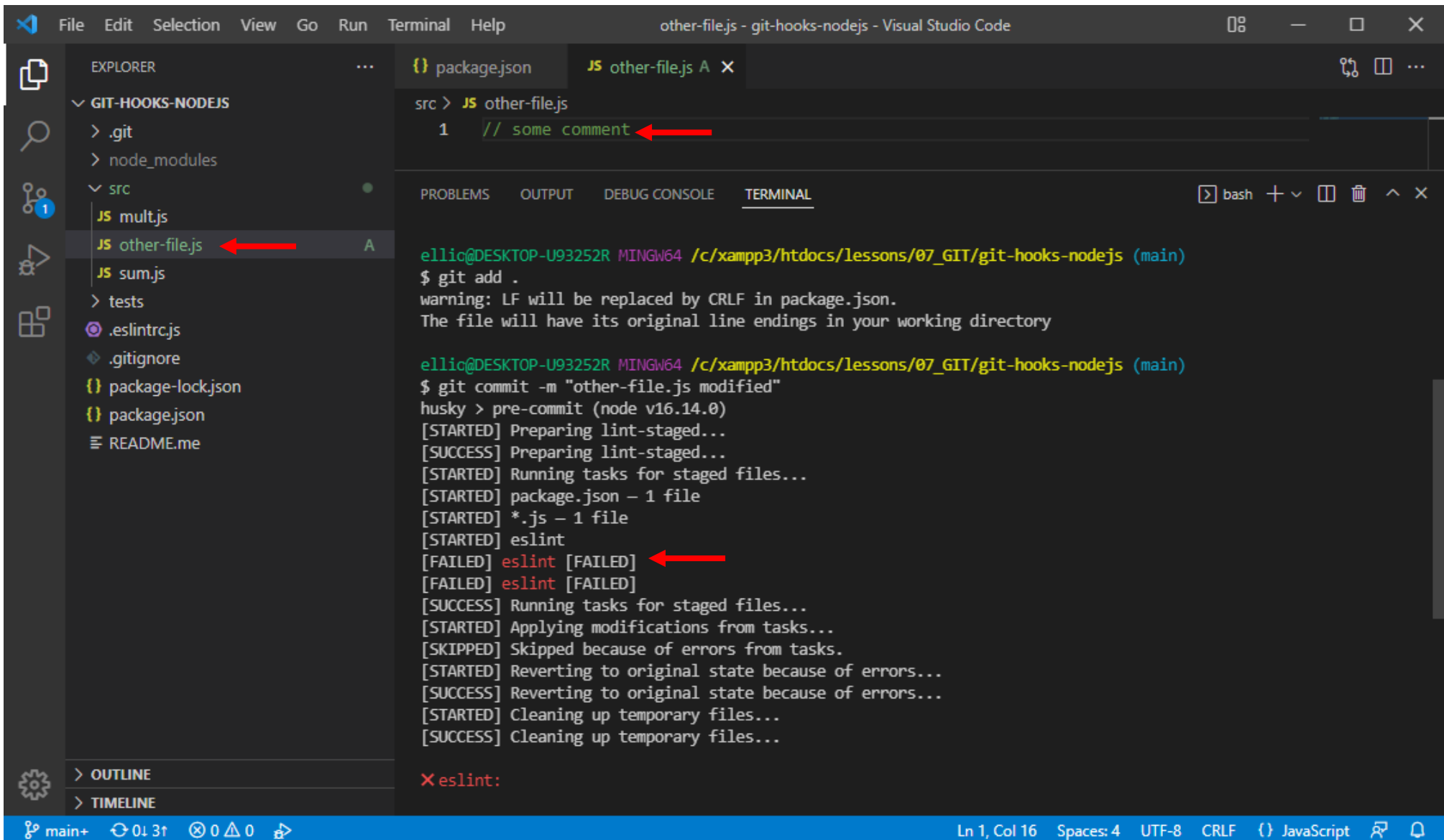
The terminal window shows the output of the command `git commit -a -m "added lint-staged package"`. The output includes warnings about line endings and a message from husky: "No staged files match any configured task." A red arrow points to this message.

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -a -m "added lint-staged package"
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
husky > pre-commit (node v16.14.0)
→ No staged files match any configured task.
[main f6d7653] added lint-staged package
2 files changed, 716 insertions(+), 3 deletions(-)

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

In the src folder other-file.js is created and a comment added.

When I try to commit the changes, husky starts checking but it fails at the eslint stage of the pre-commit hook.



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Terminal panels. The Explorer panel shows the file structure with `src/other-file.js` highlighted. The Source Control panel shows the changes to `package.json` and `other-file.js`. The Terminal panel shows the output of the pre-commit hook, which fails at the ESLint stage.

```
src > JS other-file.js
1 // some comment

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git add .
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "other-file.js modified"
husky > pre-commit (node v16.14.0)
[STARTED] Preparing lint-staged...
[SUCCESS] Preparing lint-staged...
[STARTED] Running tasks for staged files...
[STARTED] package.json - 1 file
[STARTED] *.js - 1 file
[STARTED] eslint
[FAILED] eslint [FAILED]
[FAILED] eslint [FAILED]
[SUCCESS] Running tasks for staged files...
[STARTED] Applying modifications from tasks...
[SKIPPED] Skipped because of errors from tasks.
[STARTED] Reverting to original state because of errors...
[SUCCESS] Reverting to original state because of errors...
[STARTED] Cleaning up temporary files...
[SUCCESS] Cleaning up temporary files...

X eslint:
```

The status bar at the bottom shows the current file is `main+` with 0 changes, 0 deletions, and 0 additions. The language mode is set to JavaScript.

Eslint tells me that the file needs a new line at the end of the file.

The screenshot shows the Visual Studio Code interface with a pre-commit hook failing due to an ESLint error. The Explorer on the left shows the project structure, including a `src` directory with `other-file.js`. The editor shows `other-file.js` with a single line of code: `// some comment`. The Terminal panel at the bottom displays the output of the pre-commit hook, which includes ESLint's output. A red arrow points to the error message: `1:16 error Newline required at end of file but not found eol-last`. The status bar at the bottom indicates the current file is `other-file.js` at line 1, column 16.

```
src > JS other-file.js
1 // some comment
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[STARTED] package.json - 1 file
[STARTED] *.js - 1 file
[STARTED] eslint
[FAILED] eslint [FAILED]
[FAILED] eslint [FAILED]
[SUCCESS] Running tasks for staged files...
[STARTED] Applying modifications from tasks...
[SKIPPED] Skipped because of errors from tasks.
[STARTED] Reverting to original state because of errors...
[SUCCESS] Reverting to original state because of errors...
[STARTED] Cleaning up temporary files...
[SUCCESS] Cleaning up temporary files...

✖ eslint:

C:\xampp3\htdocs\lessons\07_GIT\git-hooks-nodejs\src\other-file.js
1:16 error Newline required at end of file but not found eol-last

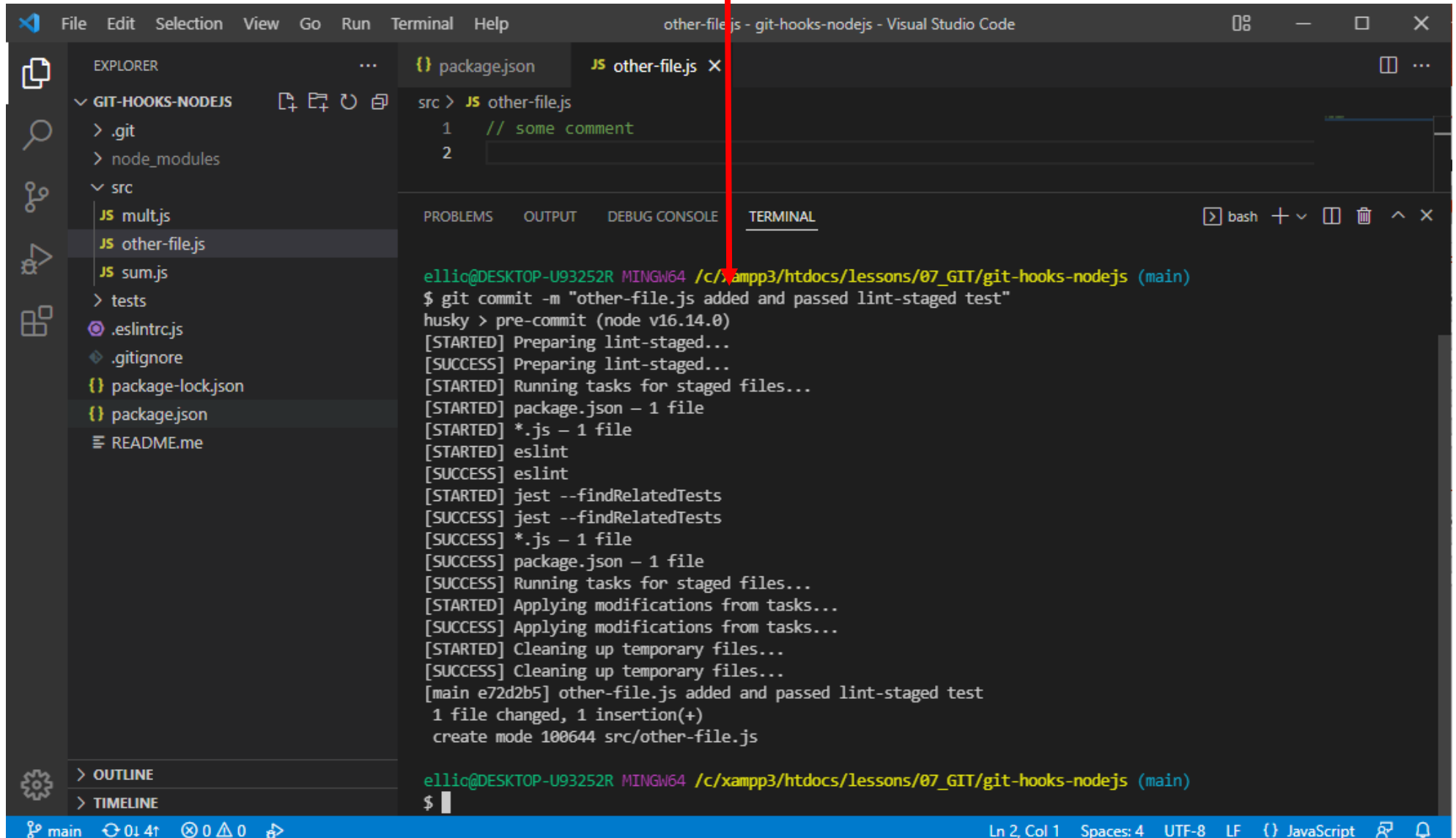
✖ 1 problem (1 error, 0 warnings)
1 error and 0 warnings potentially fixable with the `--fix` option.

husky > pre-commit hook failed (add --no-verify to bypass)

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
\$

Ln 1, Col 16 Spaces: 4 UTF-8 CRLF {} JavaScript

If I correct the error and save and stage the file (not shown) I can try to commit the file. This time it passes the lint-staged test and jest tests and is successfully committed.



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Terminal panels. The Explorer panel shows the file structure of the 'GIT-HOOKS-NODEJS' project, including 'src' and 'tests' directories. The Source Control panel shows the 'other-file.js' file staged for commit. The Terminal panel shows the command prompt output for the commit process, which includes linting and testing steps.

```
other-file.js - git-hooks-nodejs - Visual Studio Code

EXPLORER
  GIT-HOOKS-NODEJS
    .git
    node_modules
    src
      mult.js
      other-file.js
      sum.js
    tests
    .eslintrc.js
    .gitignore
    package-lock.json
    package.json
    README.me

package.json  JS other-file.js X

src > JS other-file.js
1 // some comment
2

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash + v [ ] [ ] ^ x

ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "other-file.js added and passed lint-staged test"
husky > pre-commit (node v16.14.0)
[STARTED] Preparing lint-staged...
[SUCCESS] Preparing lint-staged...
[STARTED] Running tasks for staged files...
[STARTED] package.json - 1 file
[STARTED] *.js - 1 file
[STARTED] eslint
[SUCCESS] eslint
[STARTED] jest --findRelatedTests
[SUCCESS] jest --findRelatedTests
[SUCCESS] *.js - 1 file
[SUCCESS] package.json - 1 file
[SUCCESS] Running tasks for staged files...
[STARTED] Applying modifications from tasks...
[SUCCESS] Applying modifications from tasks...
[STARTED] Cleaning up temporary files...
[SUCCESS] Cleaning up temporary files...
[main e72d2b5] other-file.js added and passed lint-staged test
1 file changed, 1 insertion(+)
create mode 100644 src/other-file.js

ellic@DESKTOP-U93252R MINGW64 /c:/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

This time I have modified the sum.js file to produce a failure.

The screenshot shows the Visual Studio Code interface with the 'sum.js' file open in the editor. The file contains the following code:

```
src > JS sum.js > sum
1  const sum = (a, b) => a + b + b;
2
3  module.exports = sum;
```

The Explorer sidebar on the left shows the project structure, including the 'src' directory with files like 'mult.js', 'other-file.js', and 'sum.js'. The 'sum.js' file is highlighted with a red arrow.

The Terminal panel at the bottom shows the output of a git commit command. The commit message is 'break sum.js'. The terminal output shows the following steps:

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "break sum.js"
husky > pre-commit (node v16.14.0)
[STARTED] Preparing lint-staged...
[SUCCESS] Preparing lint-staged...
[STARTED] Running tasks for staged files...
[STARTED] package.json - 1 file
[STARTED] *.js - 1 file
[STARTED] eslint
[SUCCESS] eslint
[STARTED] jest --findRelatedTests
[FAILED] jest --findRelatedTests [FAILED]
[FAILED] jest --findRelatedTests [FAILED]
[SUCCESS] Running tasks for staged files...
[STARTED] Applying modifications from tasks...
[SKIPPED] Skipped because of errors from tasks.
[STARTED] Reverting to original state because of errors...
[SUCCESS] Reverting to original state because of errors...
[STARTED] Cleaning up temporary files...
[SUCCESS] Cleaning up temporary files...
```

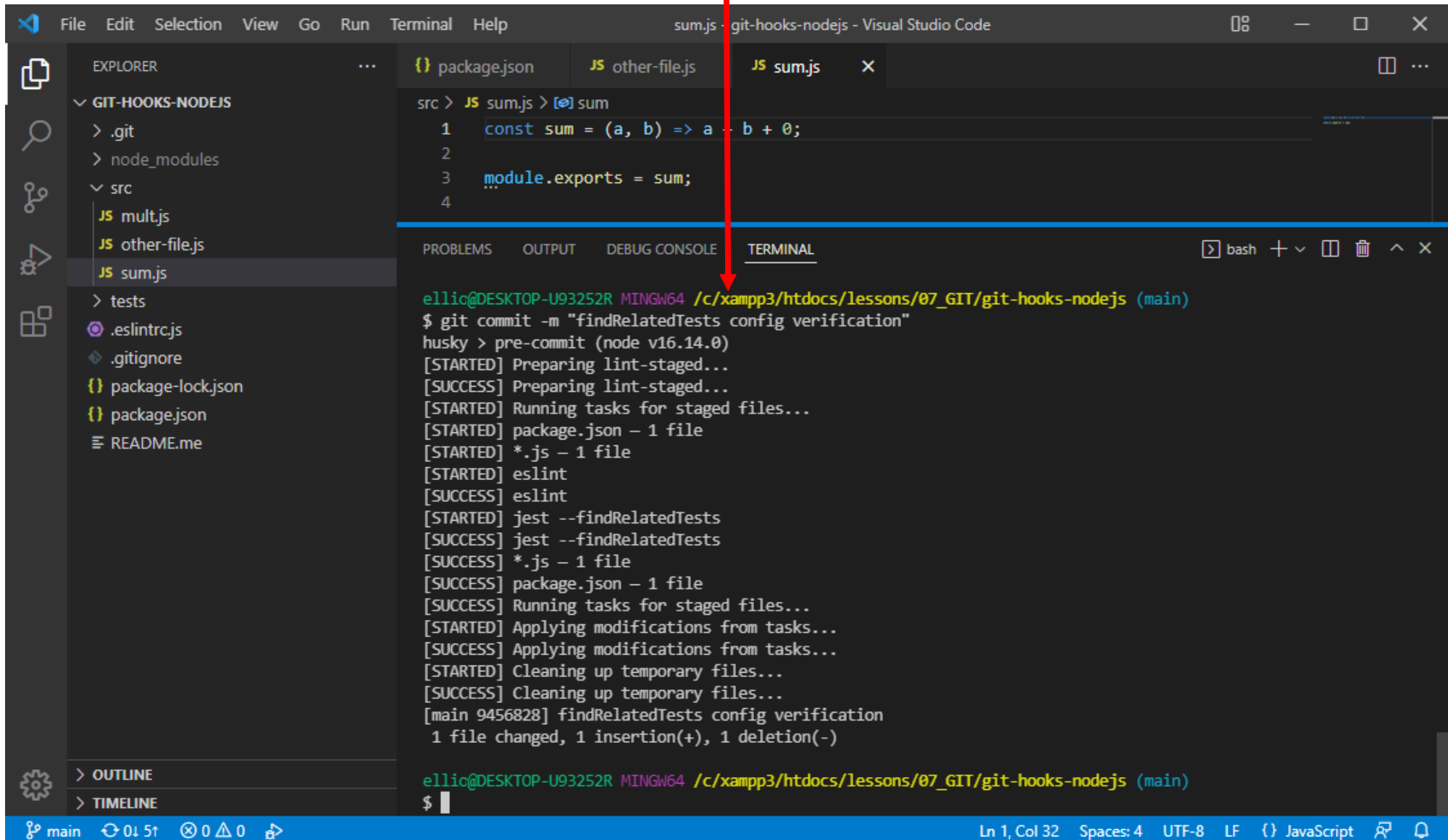
The terminal output shows that the Jest test failed. A red arrow points to the failure message in the terminal:

```
✖ jest --findRelatedTests:
FAIL tests/sum.test.js
  ✖ should return correct sum (3 ms)
```

The status bar at the bottom of the window shows the current file is 'main+' and the editor is in 'JavaScript' mode.

After saving the changes and staging the file when I try and commit it fails the jest test. Note how it only checks staged file and not all files because of the --findRelatedTests

If I correct the error and save and stage the file (not shown) I can try to commit the file. This time it passes the lint-staged test and jest tests and is successfully committed.



The screenshot shows the Visual Studio Code interface with the Explorer, Source Control, and Terminal panels. The Explorer panel on the left shows the file structure of the 'git-hooks-nodejs' project, with 'sum.js' selected. The Source Control panel shows the changes to 'sum.js'. The Terminal panel at the bottom shows the output of a git commit command, which includes the lint-staged and jest test results. A red arrow points from the text above to the 'sum.js' file in the Explorer panel.

```
src > JS sum.js > [?] sum
1  const sum = (a, b) => a + b + 0;
2
3  module.exports = sum;
4
```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "findRelatedTests config verification"
husky > pre-commit (node v16.14.0)
[STARTED] Preparing lint-staged...
[SUCCESS] Preparing lint-staged...
[STARTED] Running tasks for staged files...
[STARTED] package.json - 1 file
[STARTED] *.js - 1 file
[STARTED] eslint
[SUCCESS] eslint
[STARTED] jest --findRelatedTests
[SUCCESS] jest --findRelatedTests
[SUCCESS] *.js - 1 file
[SUCCESS] package.json - 1 file
[SUCCESS] Running tasks for staged files...
[STARTED] Applying modifications from tasks...
[SUCCESS] Applying modifications from tasks...
[STARTED] Cleaning up temporary files...
[SUCCESS] Cleaning up temporary files...
[main 9456828] findRelatedTests config verification
1 file changed, 1 insertion(+), 1 deletion(-)

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

Adding post commit hook

Visual Studio Code interface showing the `package.json` file. The Explorer sidebar on the left shows the project structure. The `package.json` file is open in the editor, showing the `scripts` section. A red arrow points to the `success-message` field in the `scripts` section, with the text "Script added" next to it. Another red arrow points to the `post-commit` field in the `husky` section, with the text "Post commit hook added" next to it. The status bar at the bottom shows "Ln 8, Col 29".

The image shows a Visual Studio Code editor window with the file `package.json` open. The file contains a `scripts` section with a `post-commit` hook that runs `husky > post-commit (node v16.14.0)`. The terminal shows the command `git commit -m "Added post commit hook"` being executed, which triggers the hook. The hook's output is displayed in the terminal, showing the success message and the commit details. A red arrow points to the success message in the terminal output.

```
package.json > {} scripts
8   "lint": "eslint src/**",
9   "lint-fix": "eslint src/** --fix",
10  "success-message": "echo thanks for committing %GIT_AUTHOR_NAME%"
11 },
12 "keywords": [],
13 "author": "",
14 "license": "ISC",
15 "devDependencies": {
16   "eslint": "^8.11.0",
17   "eslint-config-airbnb": "^19.0.4",
18   "eslint-config-airbnb-base": "^15.0.0",
19   "eslint-plugin-import": "^2.26.0"
20 }
```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -m "Added post commit hook"
husky > pre-commit (node v16.14.0)
→ No staged files match any configured task.
husky > post-commit (node v16.14.0)

> git-hooks-nodejs@1.0.0 success-message
> echo thanks for committing %GIT_AUTHOR_NAME%

thanks for committing EL-EL-BCN
[main 5b5bb29] Added post commit hook
1 file changed, 1 insertion(+), 1 deletion(-)

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

If I stage the file (not shown) I can try to commit the file. It echos the post-commit message.

Verify commit message using commit lint

I will install commitlint package with `npm i -D @commitlint/cli` command from <https://www.npmjs.com/package/@commitlint/cli>

The screenshot shows the Visual Studio Code interface with the `package.json` file open. The `devDependencies` section is visible, and the entry `"@commitlint/cli": "^16.2.3",` is highlighted in blue. A red arrow points from the terminal command `npm i -D @commitlint/cli` to this entry. Another red arrow points from the text "Note how it is automatically added to dev dependencies" to the same entry. The terminal output shows the successful installation of 99 packages and 0 vulnerabilities.

Note how it is automatically added to dev dependencies

```
{} package.json M X
{} package.json > {} devDependencies
8   "lint": "eslint src/**",
9   "lint-fix": "eslint src/** --fix",
10  "success-message": "echo thanks for committing %GIT_AUTHOR_NAME%",
11 },
12 "keywords": [],
13 "author": "",
14 "license": "ISC",
15 "devDependencies": {
16   "@commitlint/cli": "^16.2.3",
17   "eslint": "^8.11.0",
18   "eslint-config-airbnb": "^19.0.4",
19   "eslint-config-airbnb-base": "^15.0.0"
20 }
```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ npm i -D @commitlint/cli

added 99 packages, and audited 611 packages in 8s

99 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

Visual Studio Code interface showing the `package.json` file for a project named `git-hooks-nodejs`. The Explorer sidebar on the left shows the project structure, including `.git`, `node_modules`, `src` (with `mult.js`, `other-file.js`, and `sum.js`), `tests`, `.eslintrc.js`, `.gitignore`, `package-lock.json`, `package.json`, and `README.me`.

The main editor displays the `package.json` file with the following configuration:

```
27 },
28 "husky": {
29   "hooks": {
30     "pre-commit": "lint-staged",
31     "commit-msg": "commitlint -E HUSKY_GIT_PARAMS",
32     "post-commit": "npm run success-message"
33   }
34 },
35 "lint-staged": {
36   "*.js": [
37     "eslint",
38     "jest --findRelatedTests"
39   ]
40 },
41 "commitlint": {
42   "rules": {
43     "references-empty": [2, "never"]
44   },
45   "parserPreset": {
46     "parserOpts": {
47       "issuePrefixes": ["ISSUE-"]
48     }
49   }
50 }
51 }
```

Annotations in red text highlight specific changes:

- Husky hook added**: Points to the `commit-msg` hook configuration on line 31: `"commit-msg": "commitlint -E HUSKY_GIT_PARAMS",`.
- commitlint rules added**: A bracket points to the `commitlint` configuration block (lines 41-49), indicating the addition of linting rules.

The status bar at the bottom shows the current file is `package.json` on line 47, column 33, using UTF-8 encoding and LF line endings.

If I add an ISSUE- number to the commit message then the commit is successful. Note the three configured hooks that have fired.

```
package.json > {} commitlint > {} parserPreset > {} parserOpts > [ ] issuePrefixes
43 |   "references-empty": [2, "never"]
44 | },
45 |   "parserPreset": {
46 |     "parserOpts": {
47 |       "issuePrefixes": ["ISSUE-"]
48 |     }
49 |   }

```

```
ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$ git commit -a -m "[ISSUE-001] commitlint added"
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
husky > pre-commit (node v16.14.0)
→ No staged files match any configured task.
husky > commit-msg (node v16.14.0)
husky > post-commit (node v16.14.0)

> git-hooks-nodejs@1.0.0 success-message
> echo thanks for committing %GIT_AUTHOR_NAME%

thanks for committing EL-EL-BCN
[main bf162b1] [ISSUE-001] commitlint added
2 files changed, 2065 insertions(+), 6 deletions(-)

ellic@DESKTOP-U93252R MINGW64 /c/xampp3/htdocs/lessons/07_GIT/git-hooks-nodejs (main)
$
```

Pre-commit hook
No staged files match any configured task (.js files)

commit-msg

post-commit

Use **npm audit fix** command to fix any vulnerabilities warnings that may crop up in lint.

Use **npm audit fix** command to fix any vulnerabilities warnings that may crop up in npm.

Additional resources for learning more about hooks:

<https://githooks.com>

<https://pre-commit.com>

Hooks can be written in any programming language and a relevant interpreter must be used. Only thing to watch is the exit code – 0 for OK, any other number for NOK

Course: [The Complete Git Guide: Understand and master Git and GitHub](#)

Installing git | Git bash terminal | visual studio code installation | Basic shell commands ls clear pwd cd rm mkdir touch echo nano cat > >> autocomplete git init | git directories structure | object types blob tree commit annotated tag | SHA1 hash | git directory objects | Reading git object | Contents of git object | Git tree detail | Git commit detail | Working Area, Staging Area and Git Repository | git hash conflicts | author and email global and local settings | bash terminal git add commit | untracked modified staged unmodified untracked | git status | git cat file -p -s -t | git branches | git checkout | git detached HEAD | git merge 3-way fast forward | resolve merge conflicts | bash git merge abort | .git/refs/heads | bash main branch | tracking branches | bash git log | bash rename branch | bash delete branch | remote & origin repositories | bash git clone | bash git remote | bash git unpack objects | bash git garbage collection | visual studio code | github desktop | hithub.com | sourceTree | bitbucket | bash git pull | bash git push | bash git fetch | git remote prune | PAT SSH key | git pull requests | forked repository | pull request merge | review pull request | pull request approvals | add collaborators | protected branches | git issues | git labels | sync forked repository | configure remote fork | bash add upstream | .git/refs/remotes | git pull from fork | pull request to forked | bash git tags | CI/CD | Semantic Versioning | lightweight annotated | .git/tag/refs | git push --tags | git rebasing | gitignore | gitignore best practices | Gitignore templates | Advanced git bash | log graph oneline shortlog log lg filter log by user email search string | git reset mixed soft hard | git amend | git cherry-pick | git reflog | git stashing | .git/refs/stash | git stash pop | bash Squashing of commits | Git Development workflow | Github pages | basic React application | local git hooks | pre-commit hook | post commit hook | pre-rebasing hook | regex pre-commit hook | NodeJS and NPM hooks in remote repository | NPM Husky | pre-commit linters | NPM lint-staged | commit-msg-lint