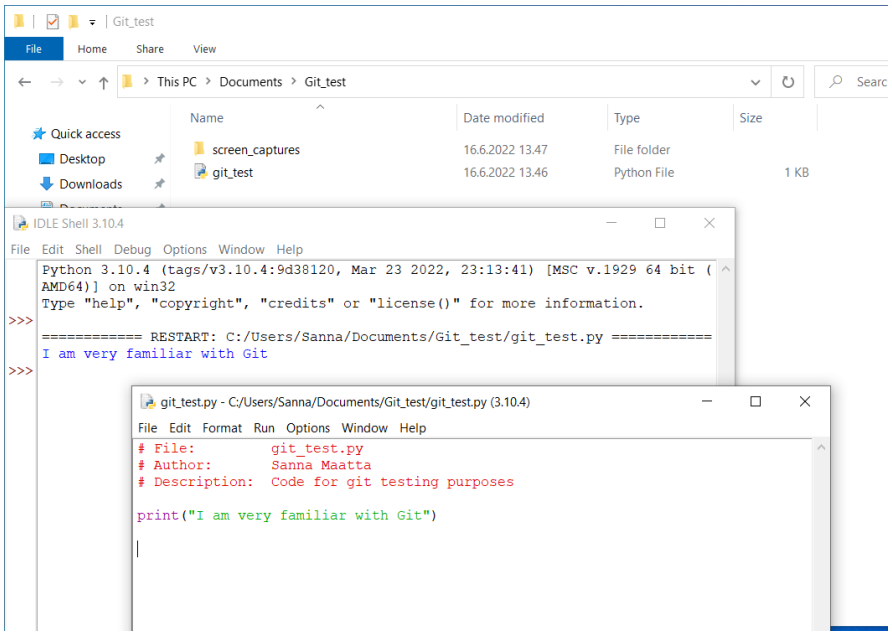
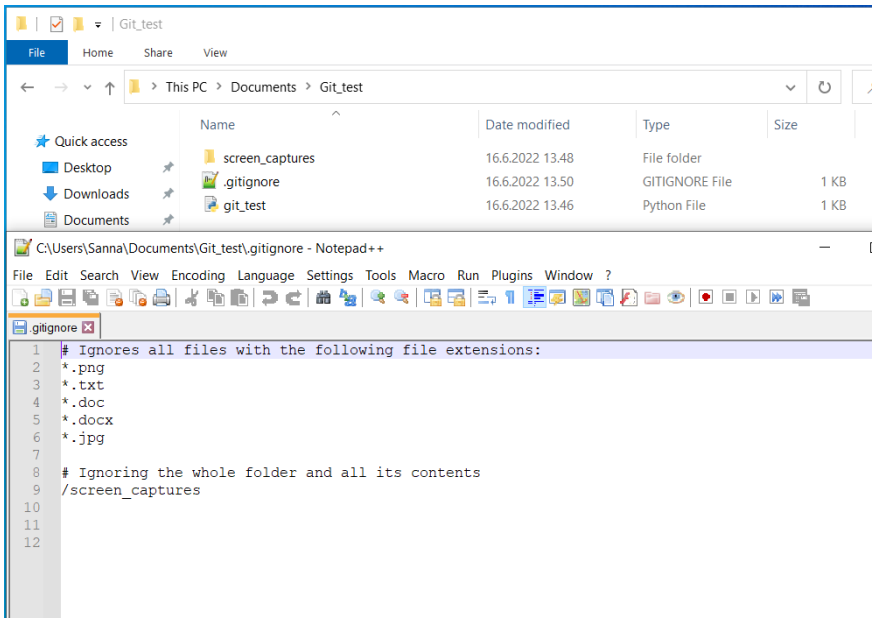
Here you can see examples of the commands and how your git bash etc. should look after each step. So, basically, mimic these. If/when you face problems, make sure that you have typed all the commands right and carefully read the instructions.

Capture 1 – Create folder and add some code

(Notice that the screen_captures folder that you can see in the capture below can be used for e.g. to store the Word document, where you paste the screen captures (that you are instructed to do). But you can store the Word document some other location on your computer as well.)

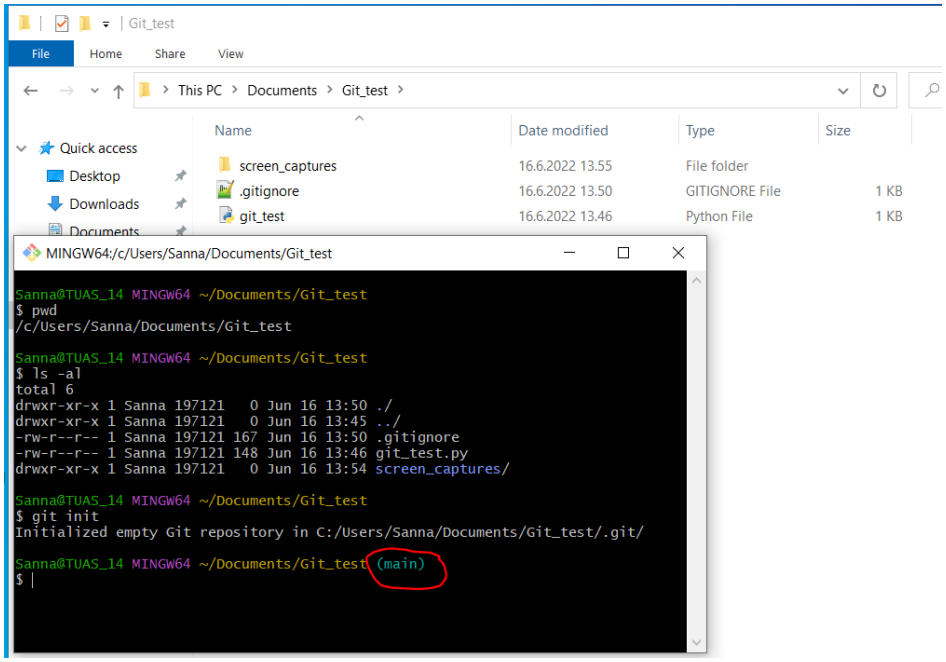


Capture 2 – .gitignore file

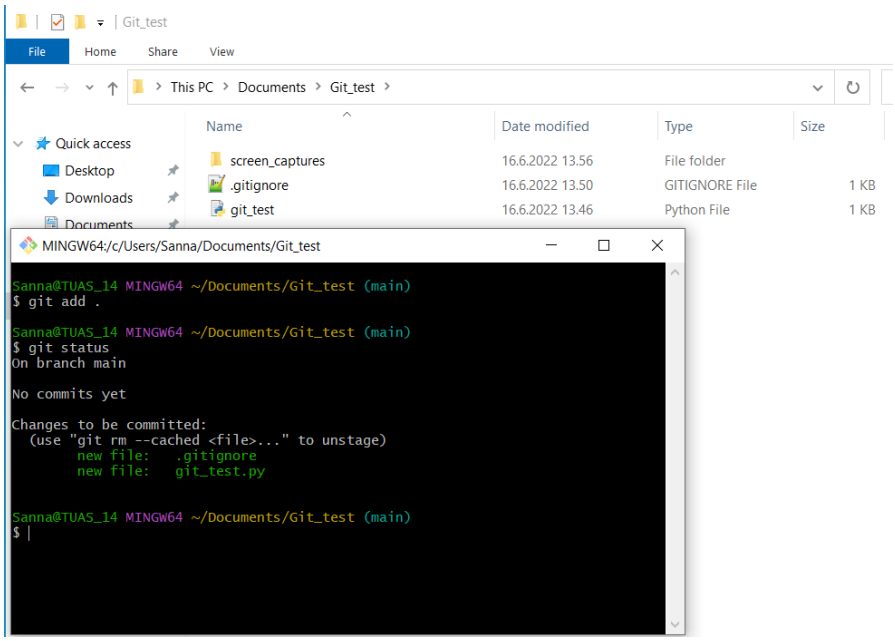


Capture 3 – Open Git Bash, see where you are and initialize the repository

Notice that instead of (main), see the red circle in screen capture, you may have (master). It is ok, just take this into account also later on this course.

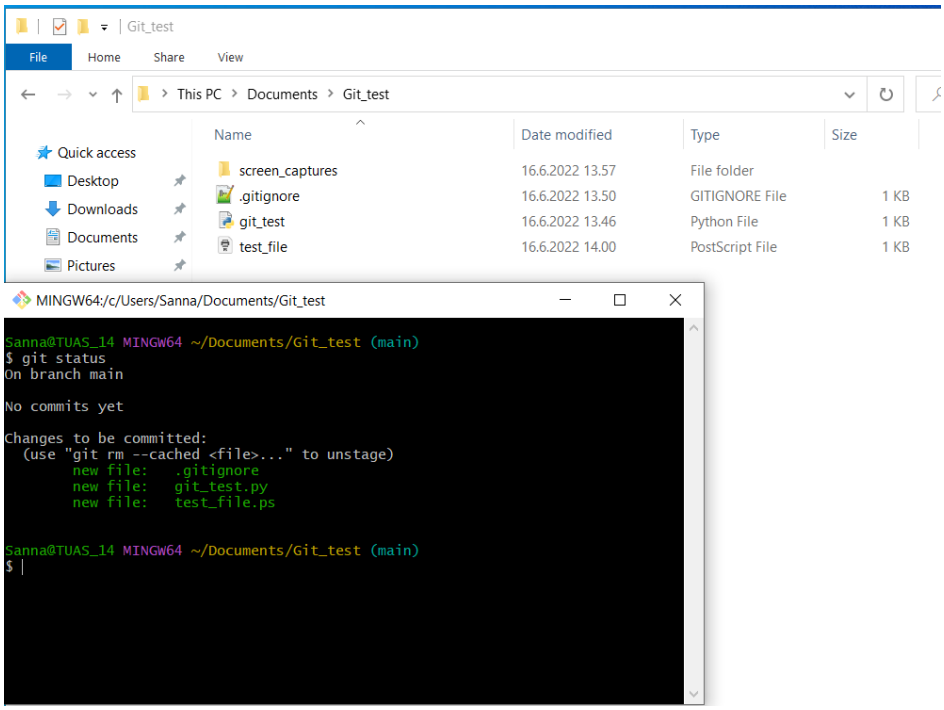


Capture 4 – Git add



[Capture 5 – Undoing git add command (repeat this only if you accidentally added files you were not supposed to or you want to try out the undoing things)]

For demonstration purposes, a test_file.ps is added using git add command and this is how you can undo the git add command.



```
MINGW64:/c/Users/Sanna/Documents/Git_test

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   git_test.py
        new file:   test_file.ps


Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git rm --cached test_file.ps
rm 'test_file.ps'

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   git_test.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test_file.ps


Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

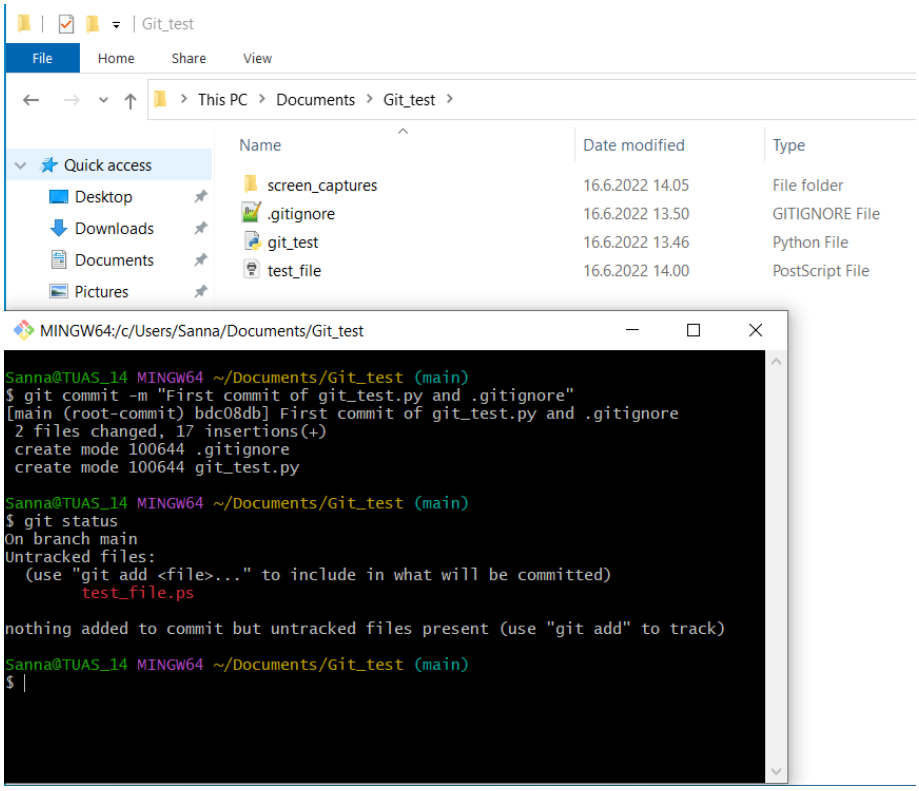
Capture 6 – Setting up your account

```
$ git commit -m "Initial commit of git_test.py"
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.
```

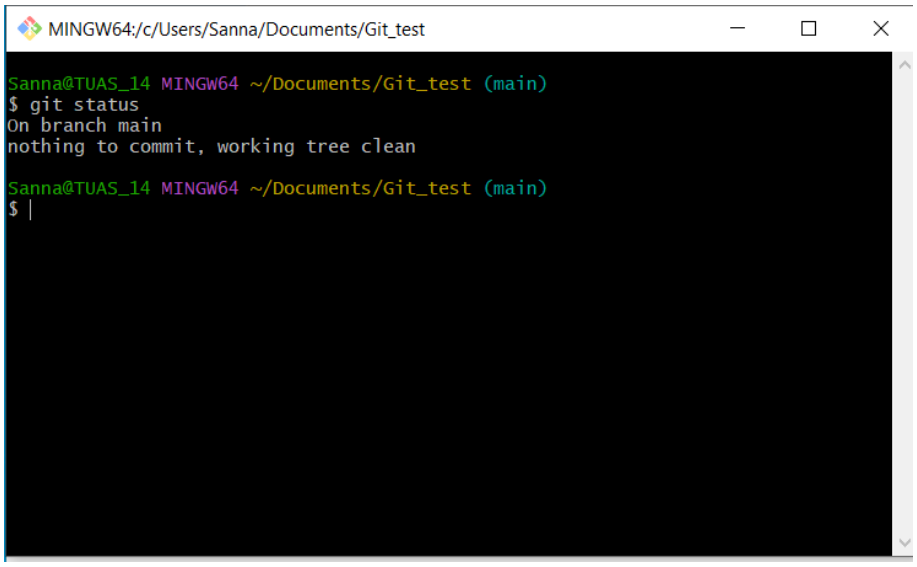
```
MINGW64:/c/Users/Sanna/Documents/Git_test

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git config --global user.email "sanna.maatta@turkuamk.fi"

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git config --global user.name "Sanna Maatta"

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

Capture 7a – Git commit

Notice that the test_file.ps is there for demonstration purposes only (how to undo to the git add command, see Capture 5 for more details)



Capture 7b – Git status



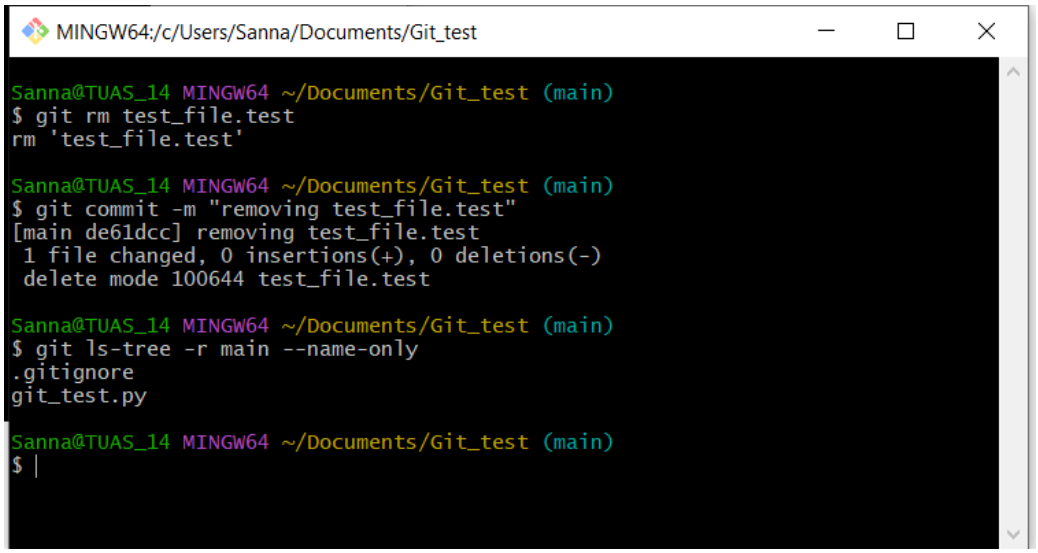
Capture 8 – Git log after committing the changed code

```
MINGW64:/c/Users/Sanna/Documents/Git_test
Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git log
commit 2caf345c173a225d46c913f705759bb53476632e (HEAD -> main)
Author: Sanna Maatta <sanna.maatta@turkuamk.fi>
Date:   Thu Jun 16 14:21:01 2022 +0300

    Added an output print to git_test.py

commit bdc08dbdfa560bed0577c2160ad48d7452579959
Author: Sanna Maatta <sanna.maatta@turkuamk.fi>
Date:   Thu Jun 16 14:06:24 2022 +0300

    First commit of git_test.py and .gitignore

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git ls-tree -r main --name-only
.gitignore
git_test.py

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

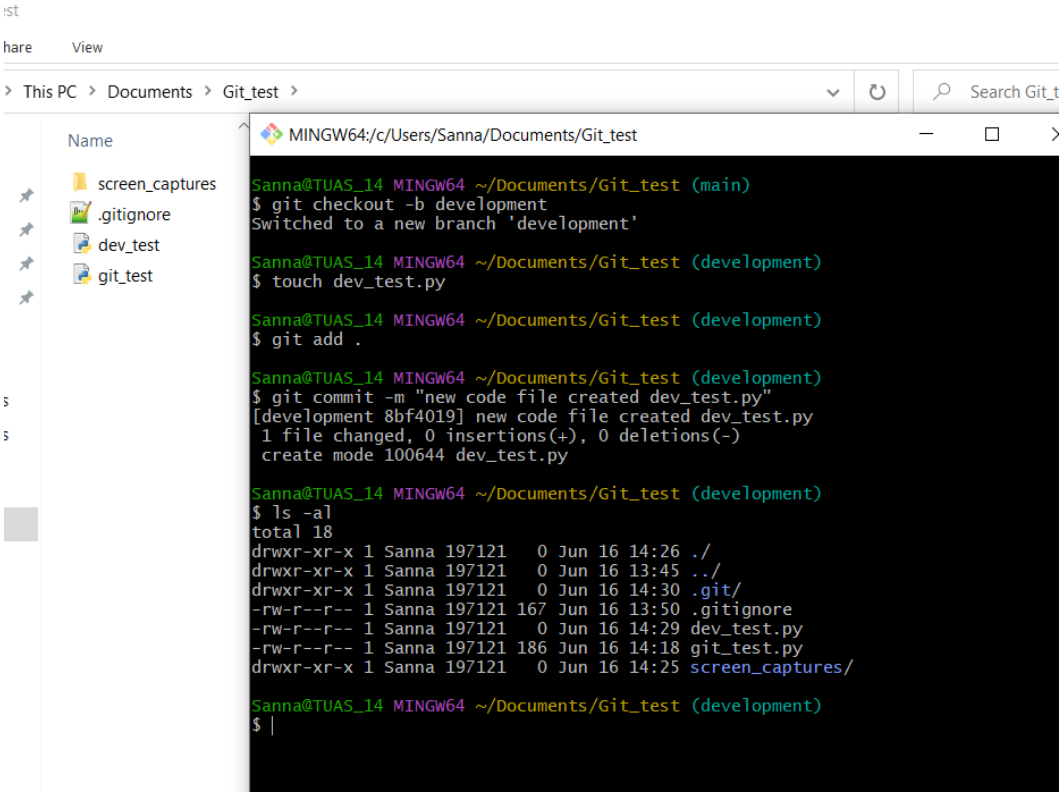
Capture 9 – Showing the versioned files

```
MINGW64:/c/Users/Sanna/Documents/Git_test

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git ls-tree -r main --name-only
.gitignore
git_test.py
test_file.test

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

[Capture 10 – Removing a versioned file from versioning (repeat this only if you accidentally versioned a file you were not supposed to version; or repeat this if you want to try out the undoing things commands)]



Capture 11 a and b – A new branch and a new code file and moving back to main branch



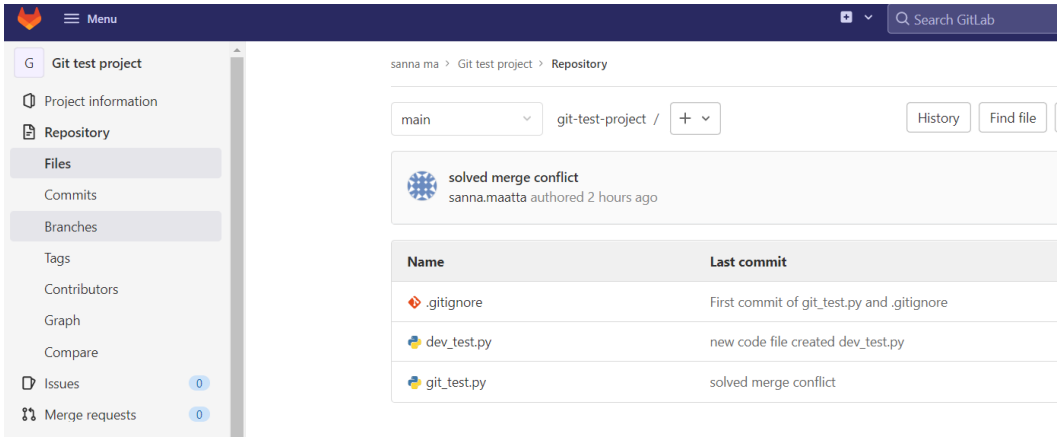
```
Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (development)
$ git checkout main
Switched to branch 'main'

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ ls -al
total 18
drwxr-xr-x 1 Sanna 197121   0 Jun 16 14:31 ./
drwxr-xr-x 1 Sanna 197121   0 Jun 16 13:45 ../
drwxr-xr-x 1 Sanna 197121   0 Jun 16 14:31 .git/
-rw-r--r-- 1 Sanna 197121 167 Jun 16 13:50 .gitignore
-rw-r--r-- 1 Sanna 197121 186 Jun 16 14:18 git_test.py
drwxr-xr-x 1 Sanna 197121   0 Jun 16 14:30 screen_captures/

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

Capture 12 – Working with git remotes

Remember that if you have (master) instead of (main), your command may be: git push -u origin master

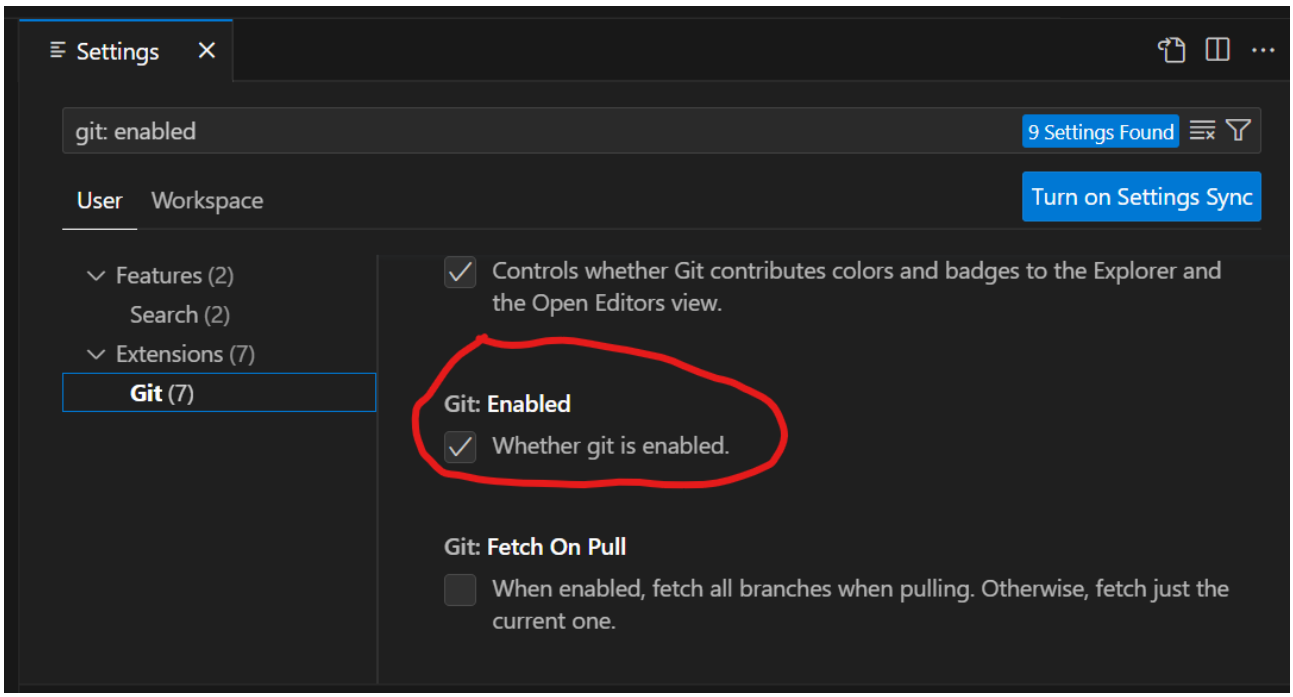
```
MINGW64:/c/Users/Sanna/Documents/Git_test

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git push -u origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (22/22), 2.12 KiB | 1.06 MiB/s, done.
Total 22 (delta 9), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/[illegible]/git-test-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git branch --all
  development
* main
  remotes/origin/main

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ git remote -v
origin  https://gitlab.com/[illegible]/git-test-project.git (fetch)
origin  https://gitlab.com/[illegible]/git-test-project.git (push)

Sanna@TUAS_14 MINGW64 ~/Documents/Git_test (main)
$ |
```

Capture 13 – GitLab



Visual Studio Code

Capture 14 – Git Enabled

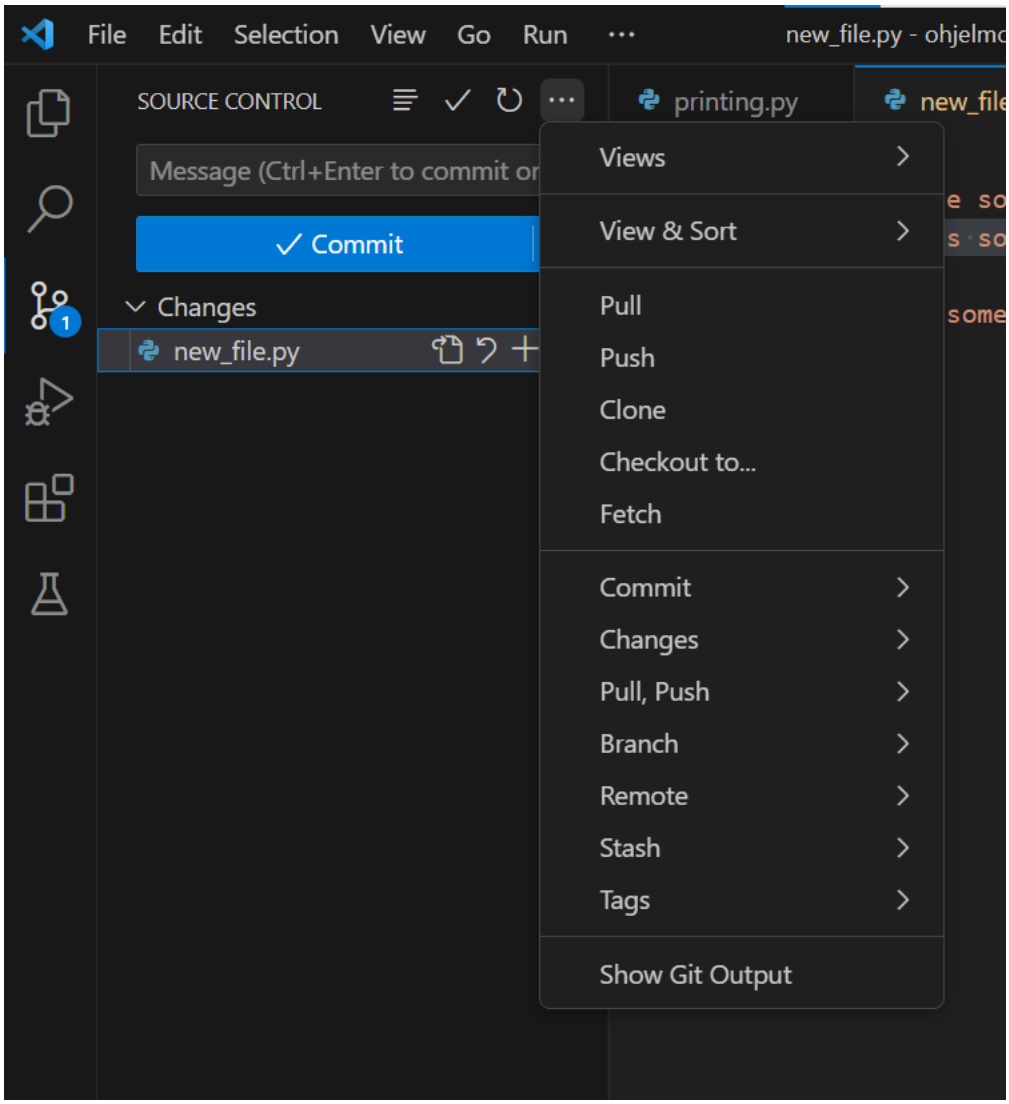


Capture 15 – Type Git in VS Code Terminal

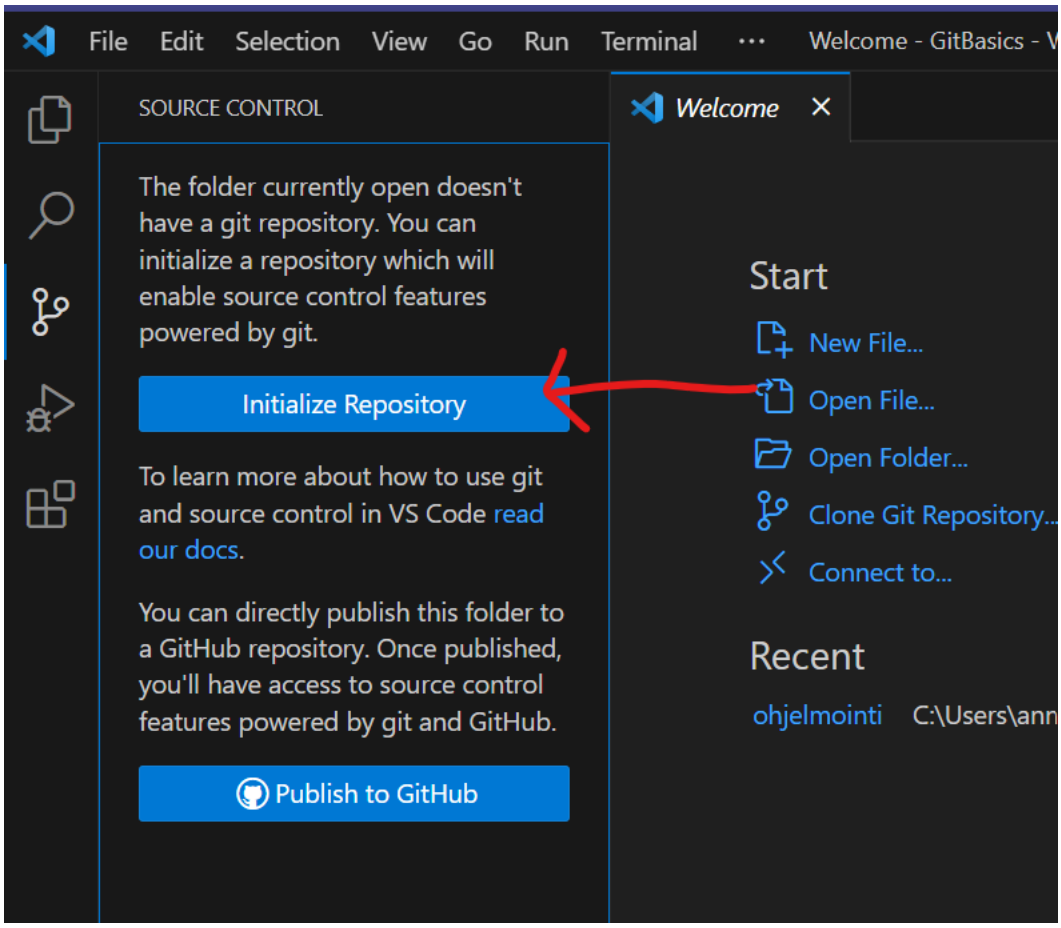
The beginning looks something like this:

```
PS C:\Users\annukka23H\Documents\ohjelmointi> git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>
]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-pat
h]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [-
-bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:
```

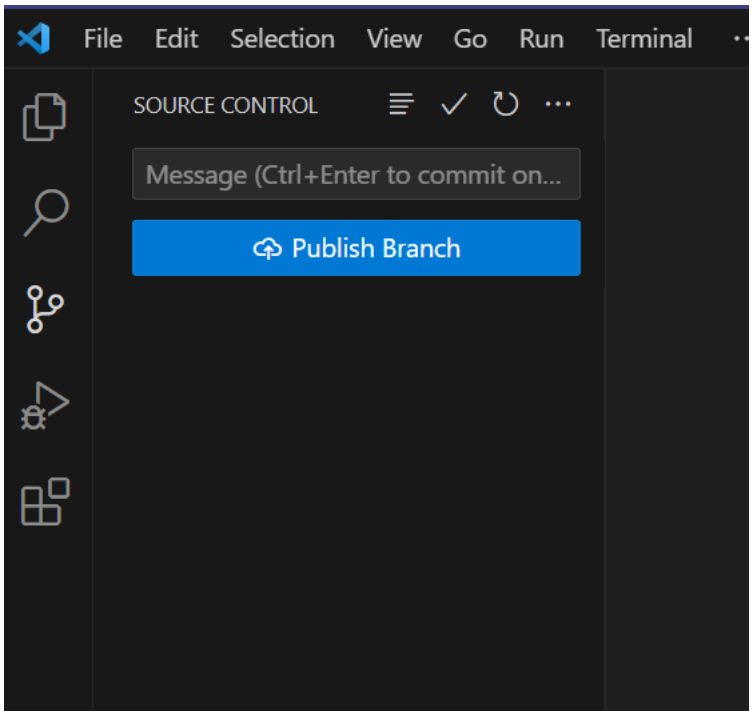
Capture 16 – Git Commands in VS Code



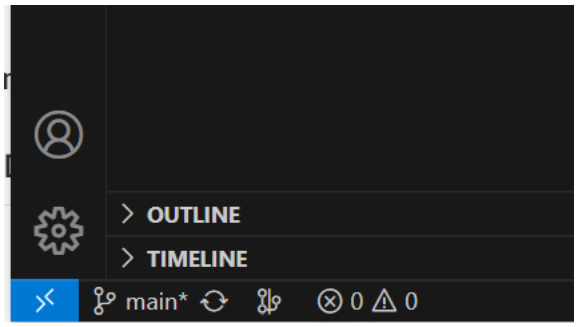
Capture 17 – Initialize Repository



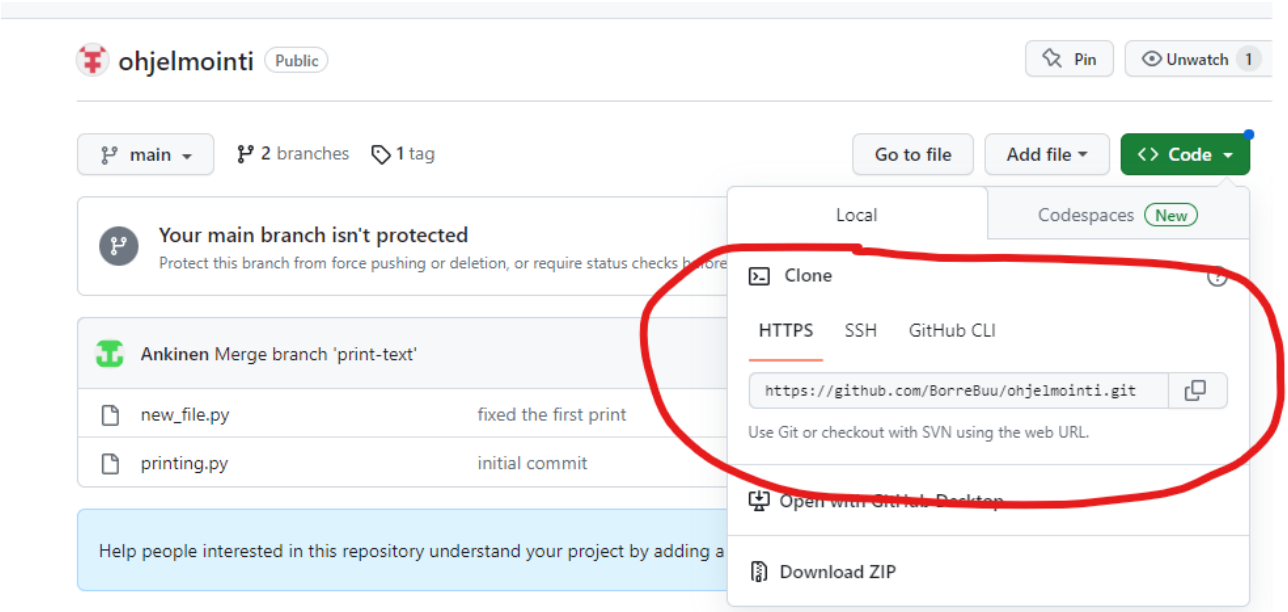
Initialized repository looks like this:



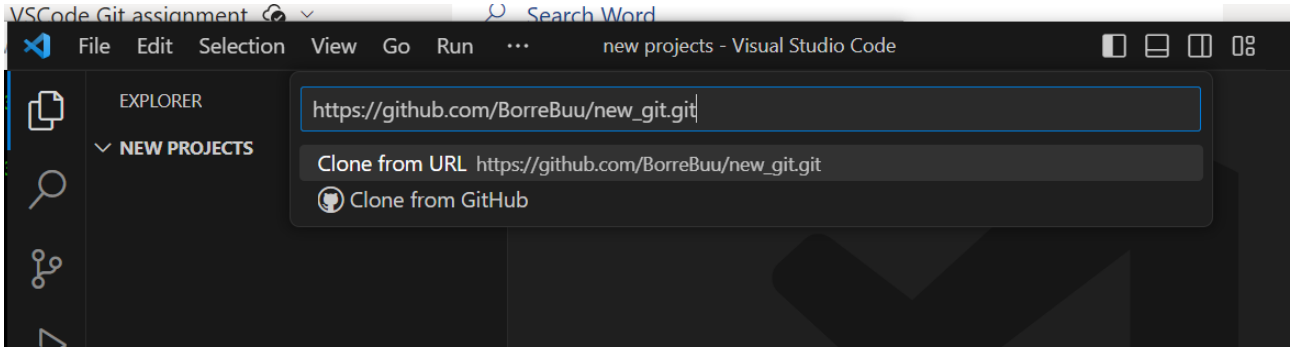
Capture 18 - See the main or master branch on the bottom left corner



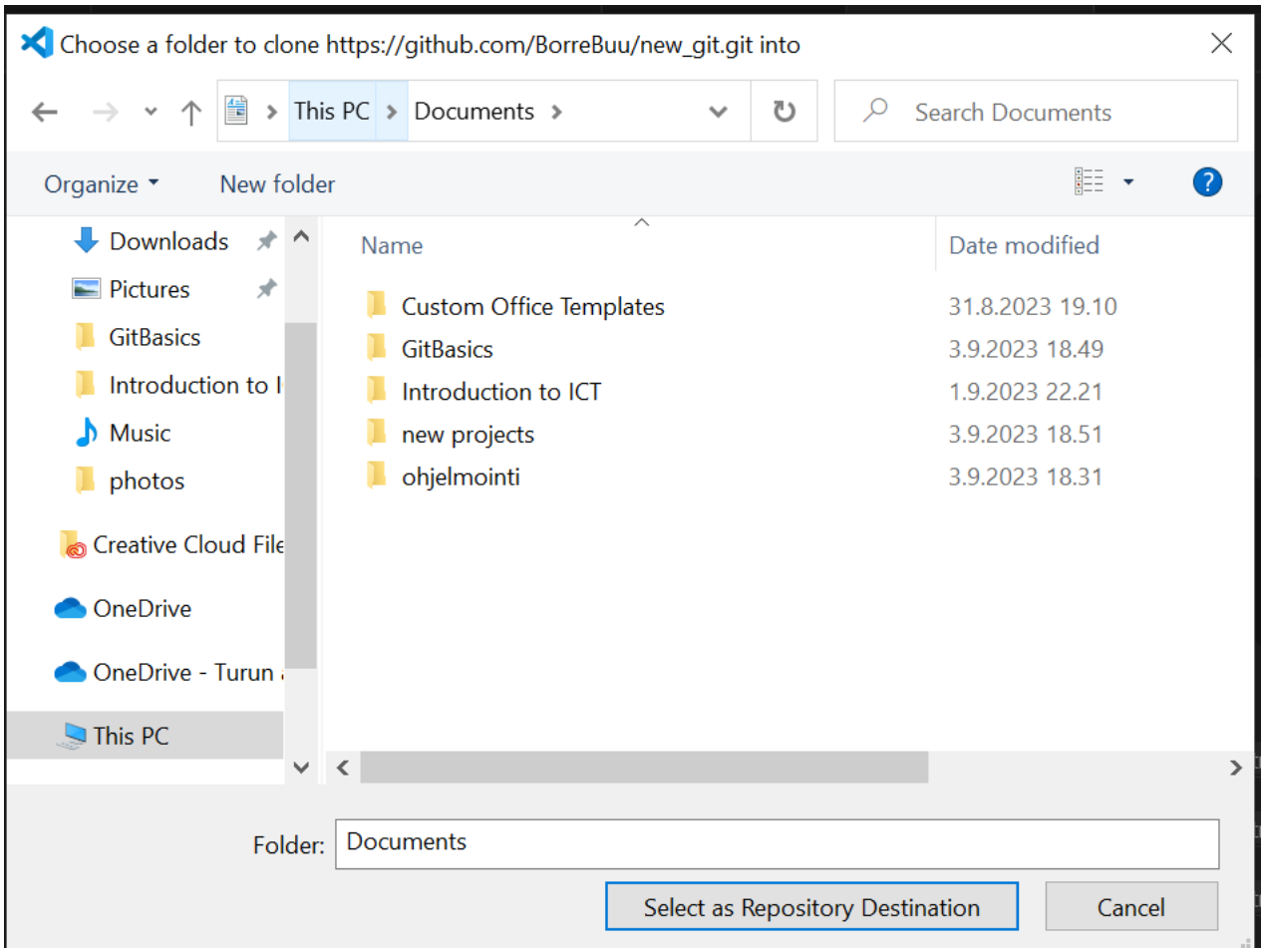
Capture 19 - Copy the URL Address from a Remote Repository



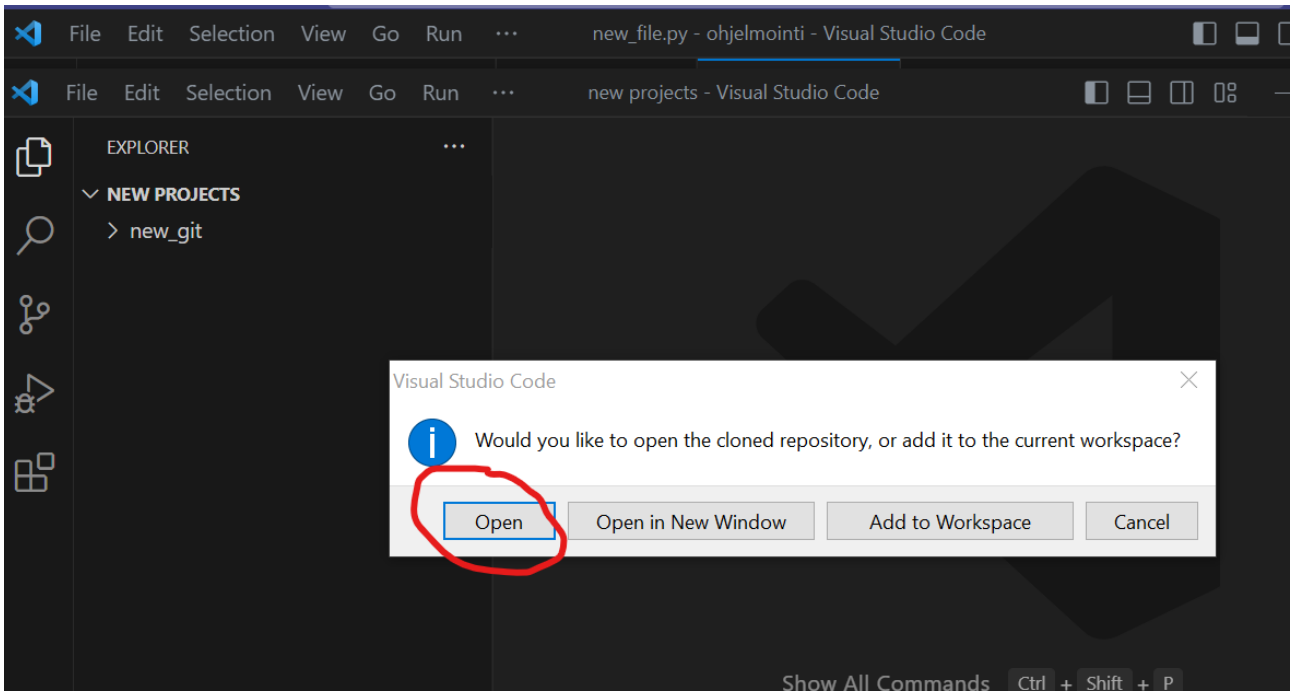
Capture 20 – Clone from URL



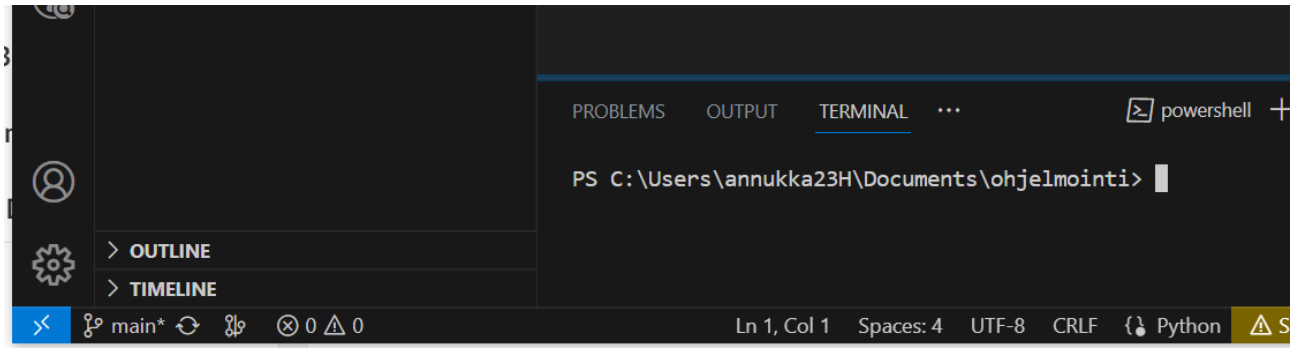
Specify the folder you want the clone to be:



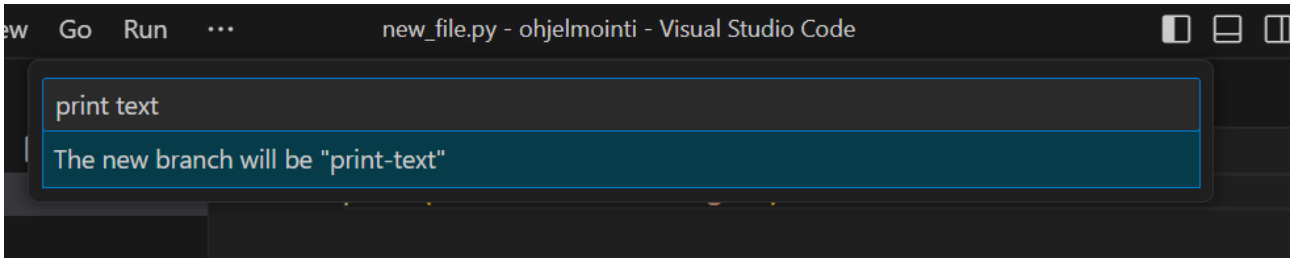
Capture 21 – Open the Cloned Repository

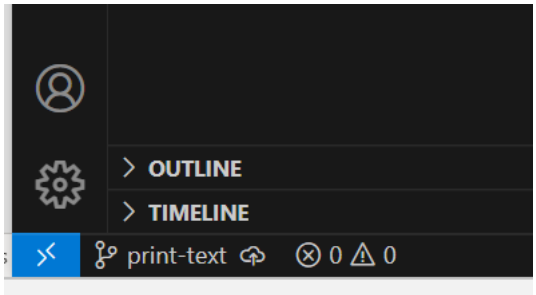


Capture 22 – Example of a Created Repository

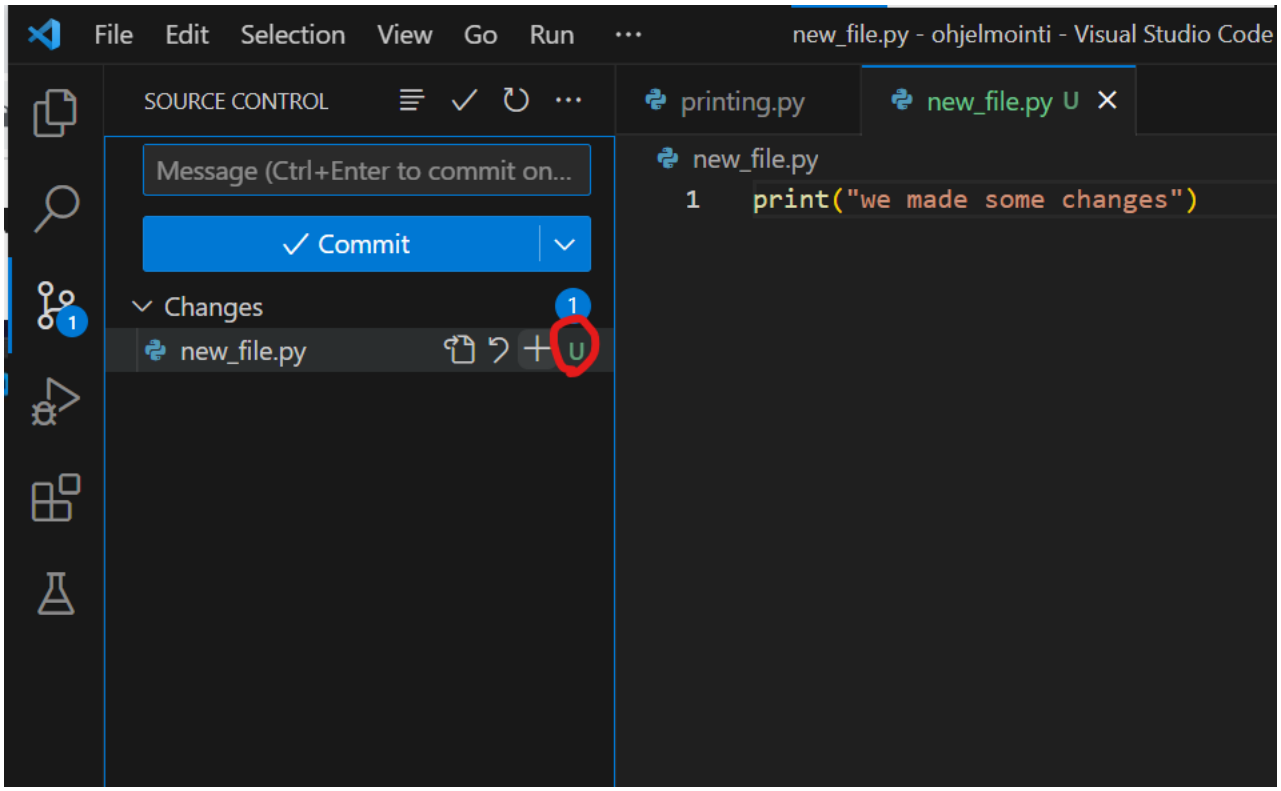


Capture 23 – New Branch Name

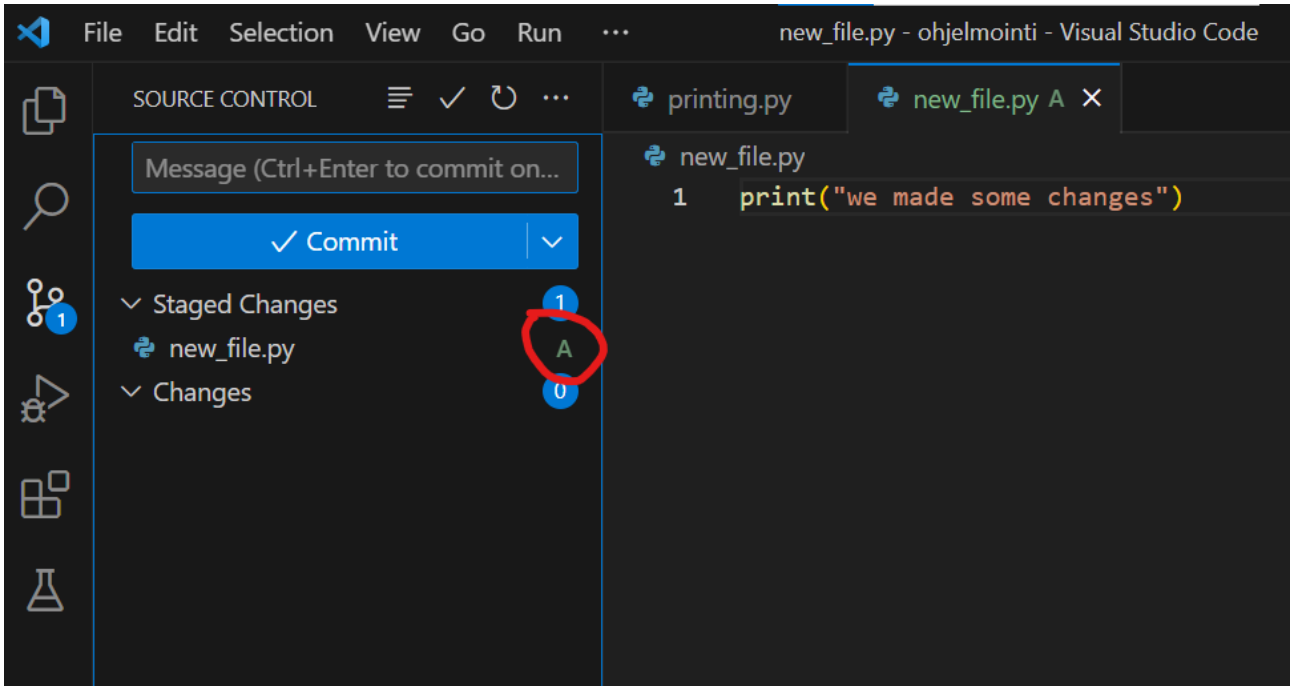




Capture 24 – Untracked File

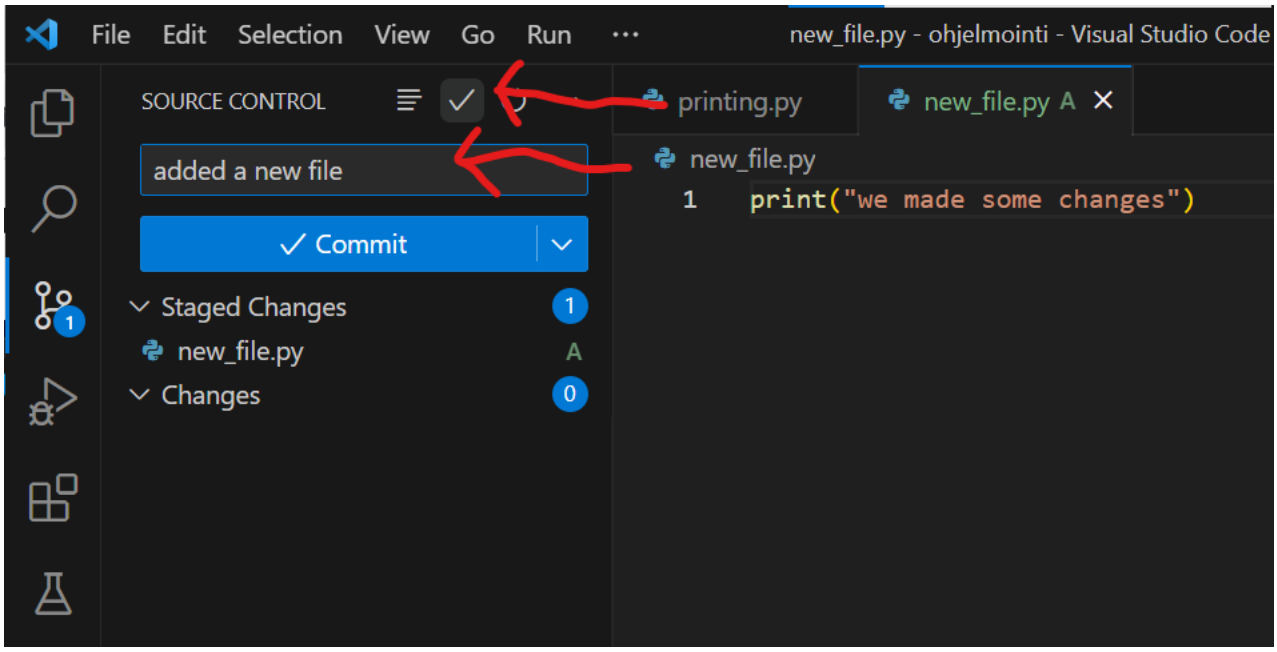


Capture 25 – Staged file

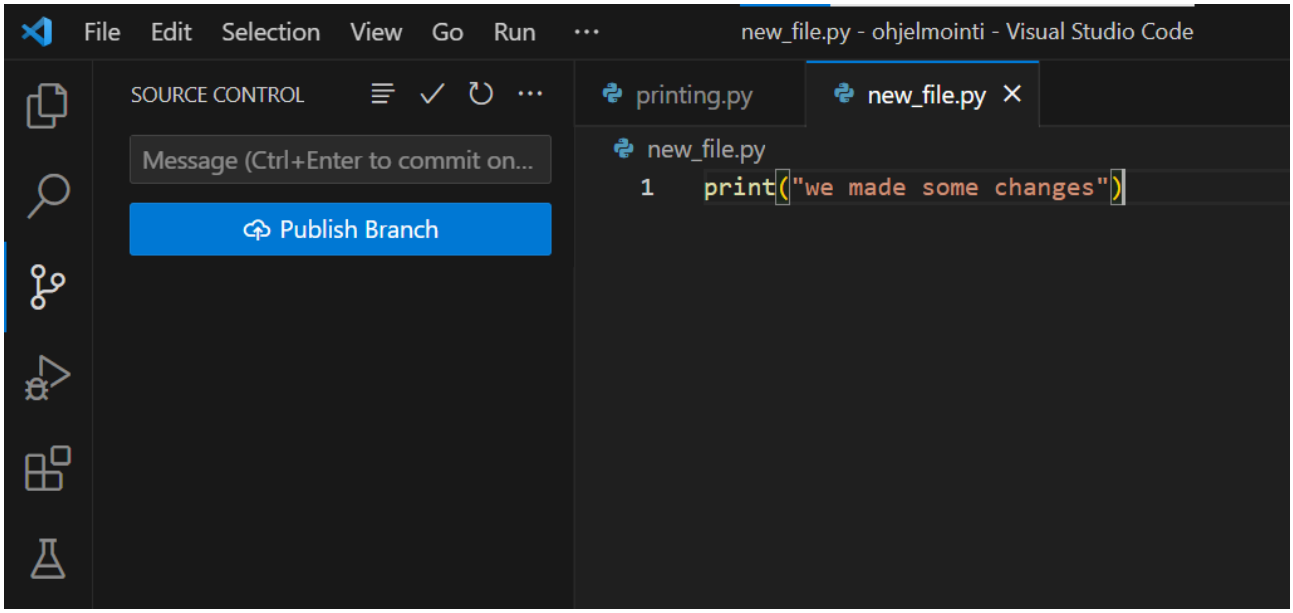


Capture 26 – Commit the Changes

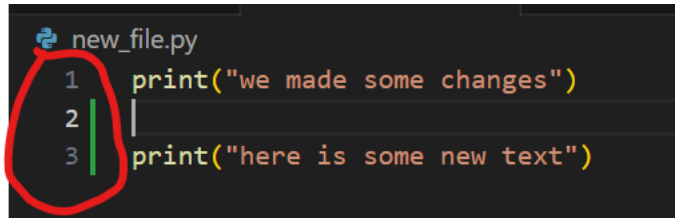
Write a commit message to the text box:



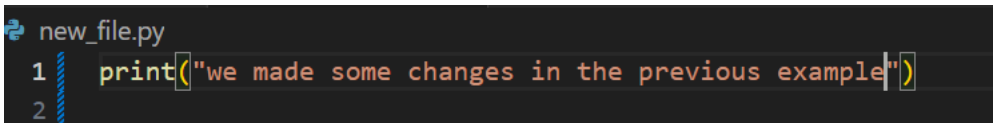
After commit, it should look like this in Source Control:



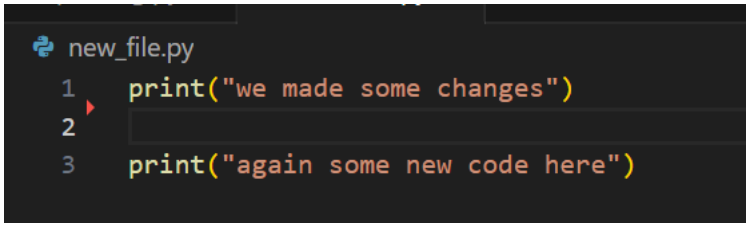
Capture 27 – "Gutter" – New Code Added



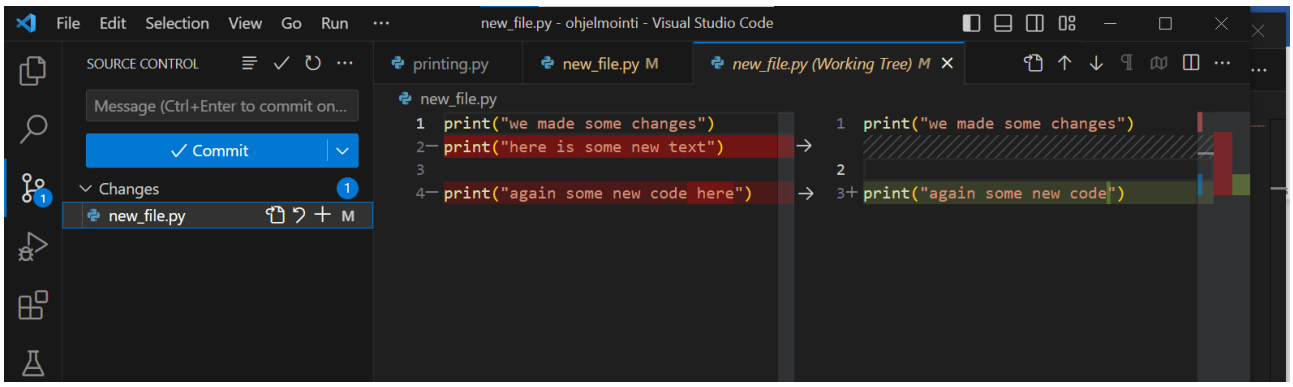
Capture 28 – “Gutter" – Changes made



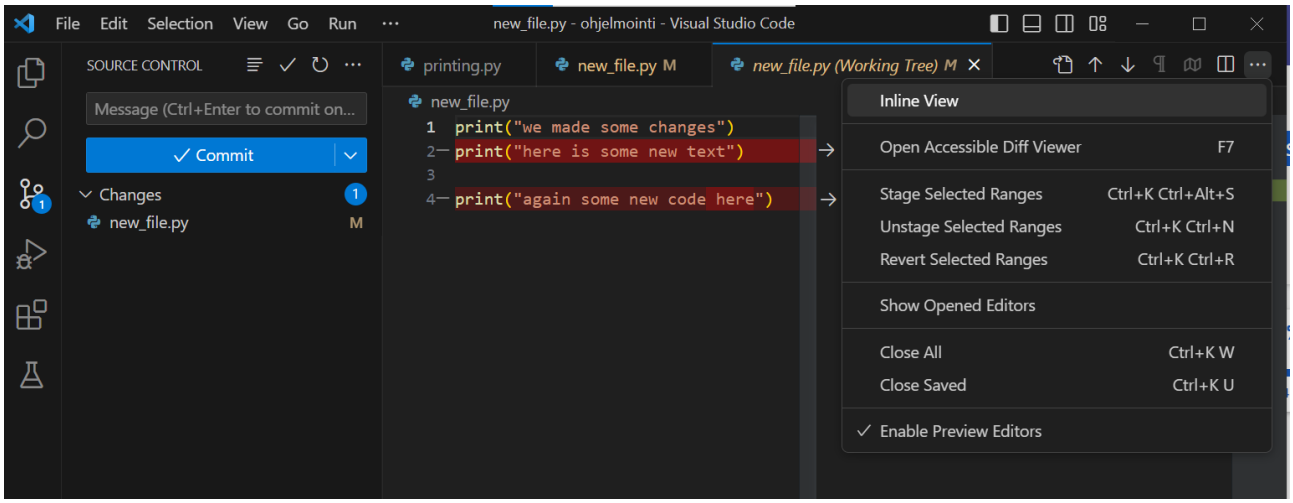
Capture 29 – “Gutter" – Deleted a Line



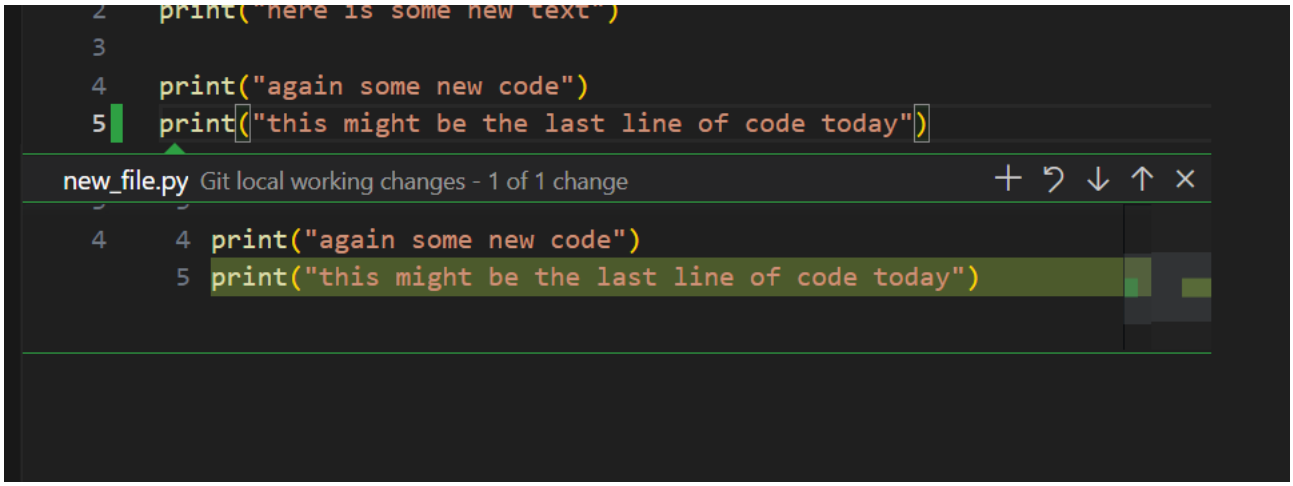
Capture 30 – Select the file name and see the changes you made



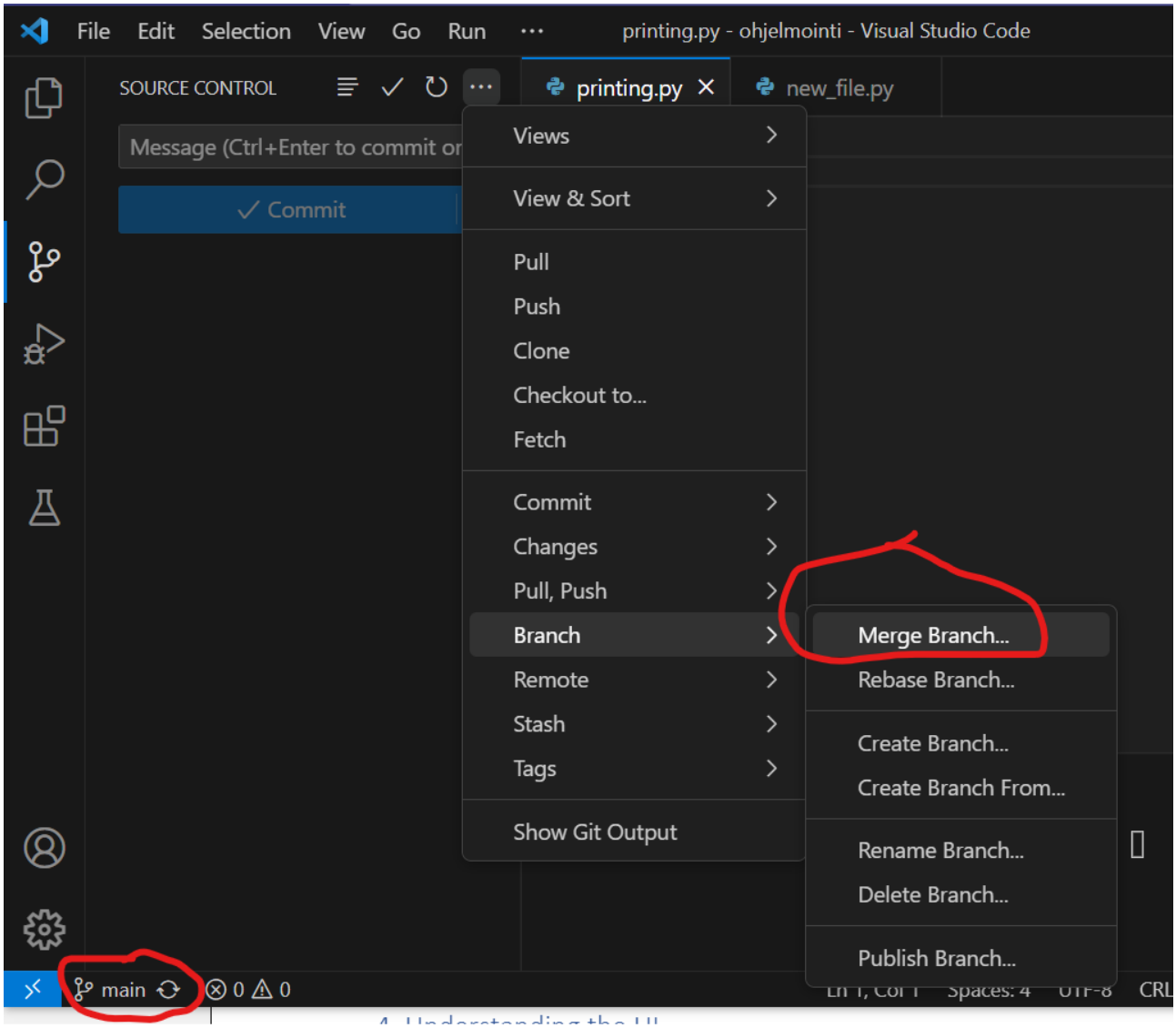
From the upper right corner, you can select Inline View to see the changes in one file:



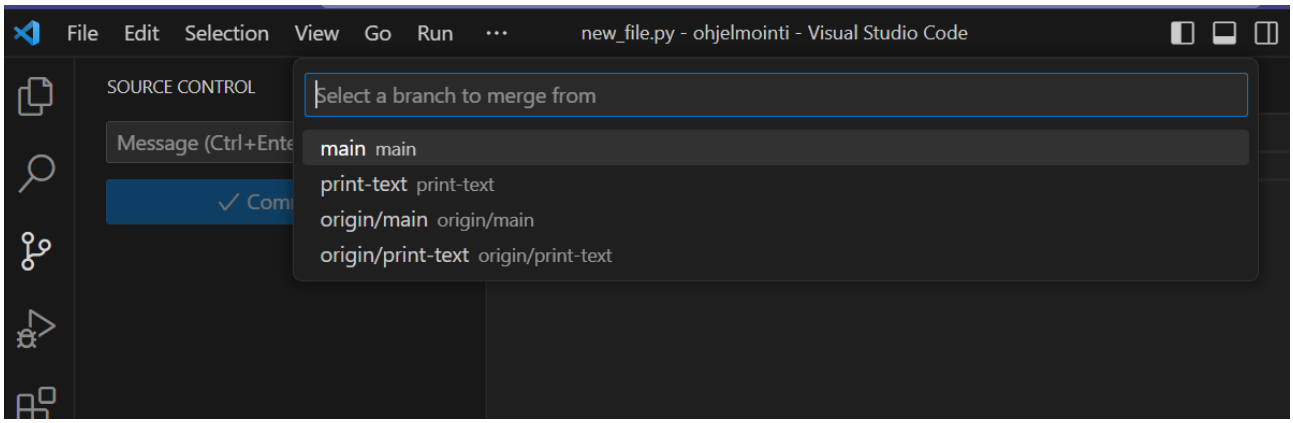
Capture 31 – Click the “Gutter” to See the Changes



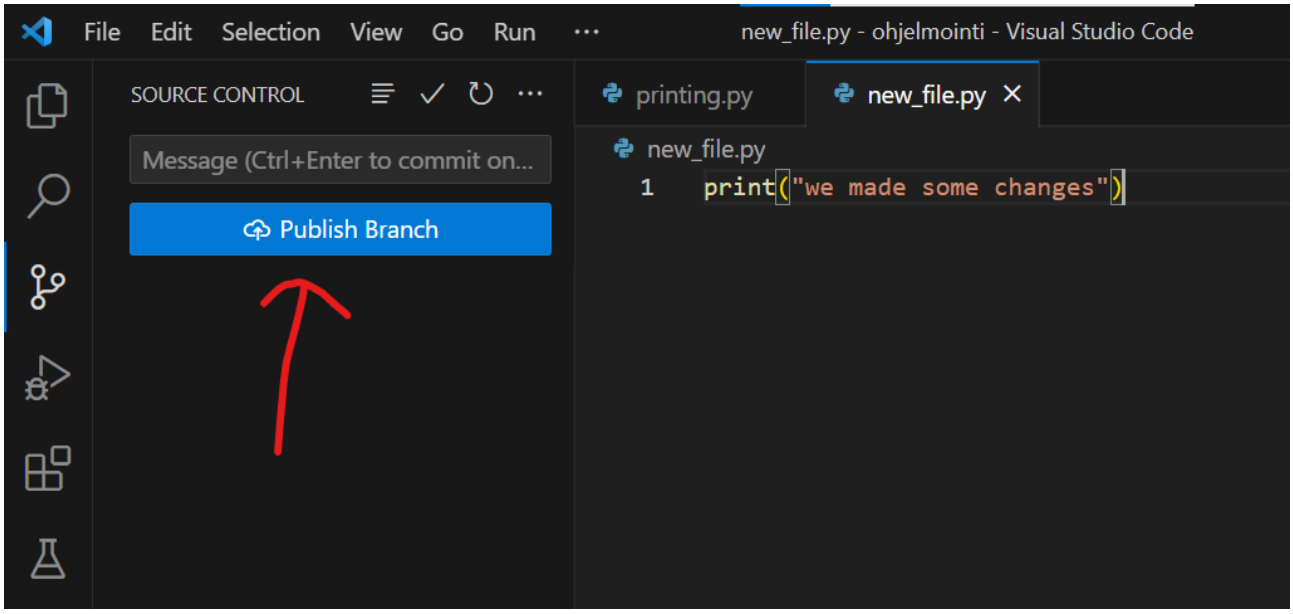
Capture 32 – Select Merge Branch



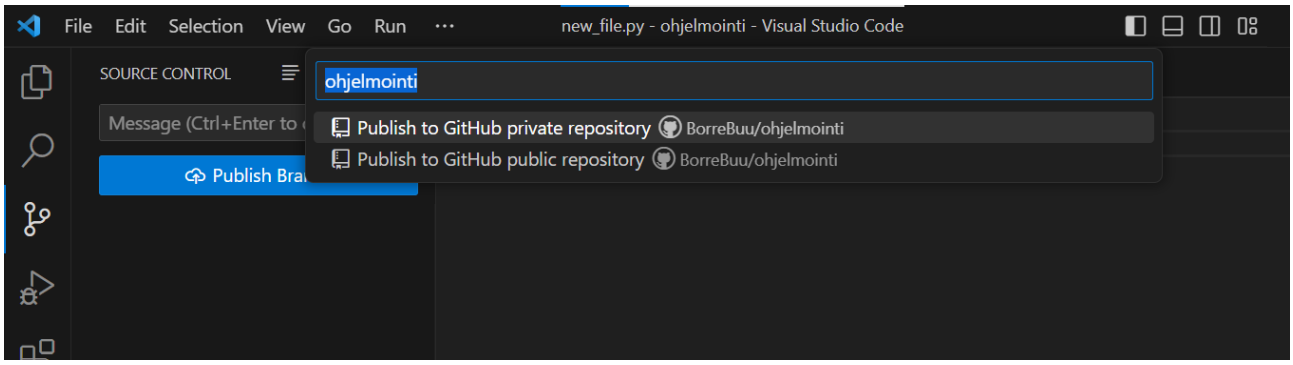
Capture 33 – Select the Branch You Want to Merge from



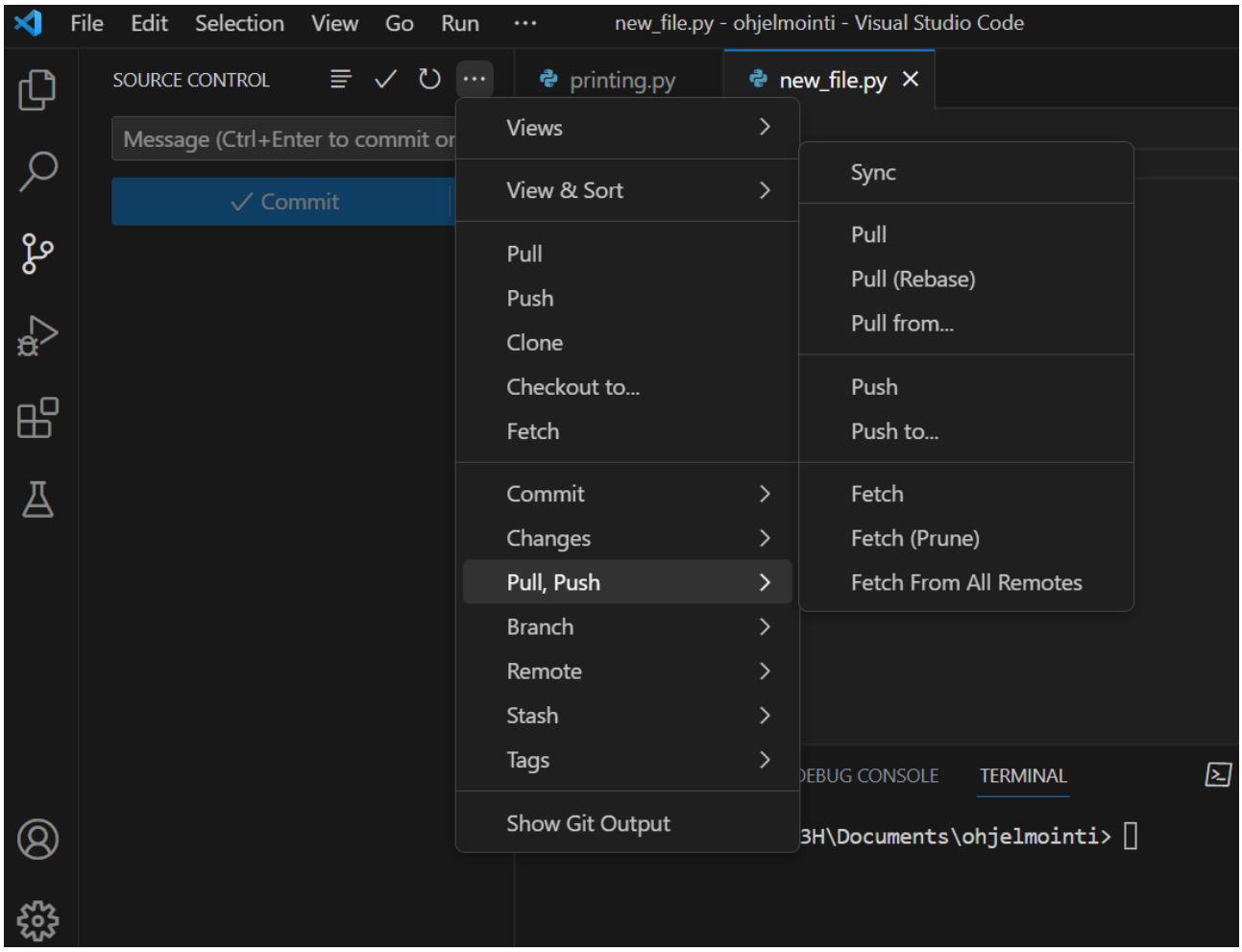
Capture 34 – Publish Branch



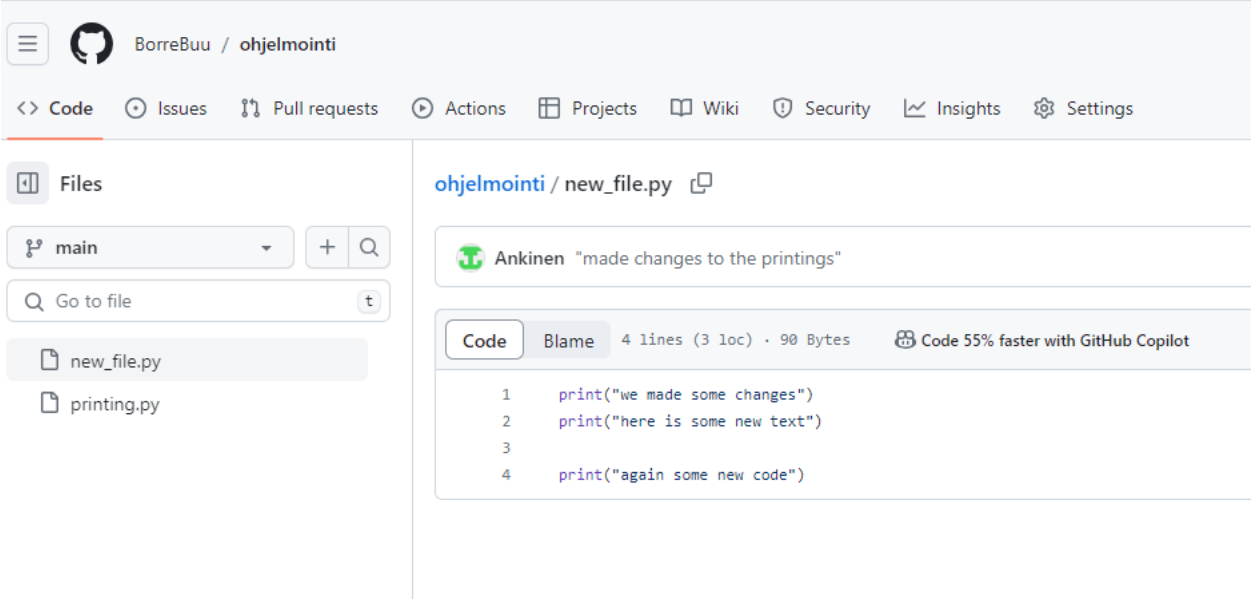
Capture 35 – Choose Public or Private Repository



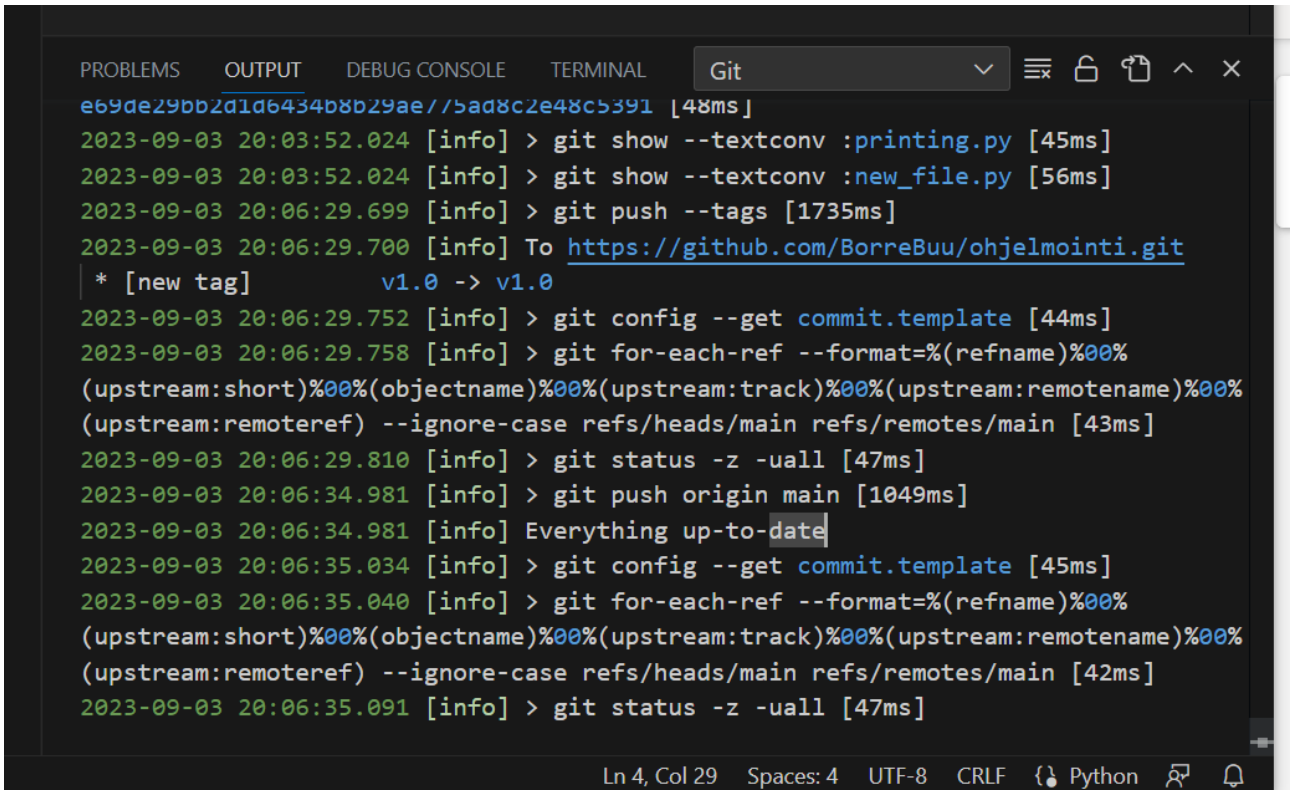
Capture 36 – Push, Pull and Sync



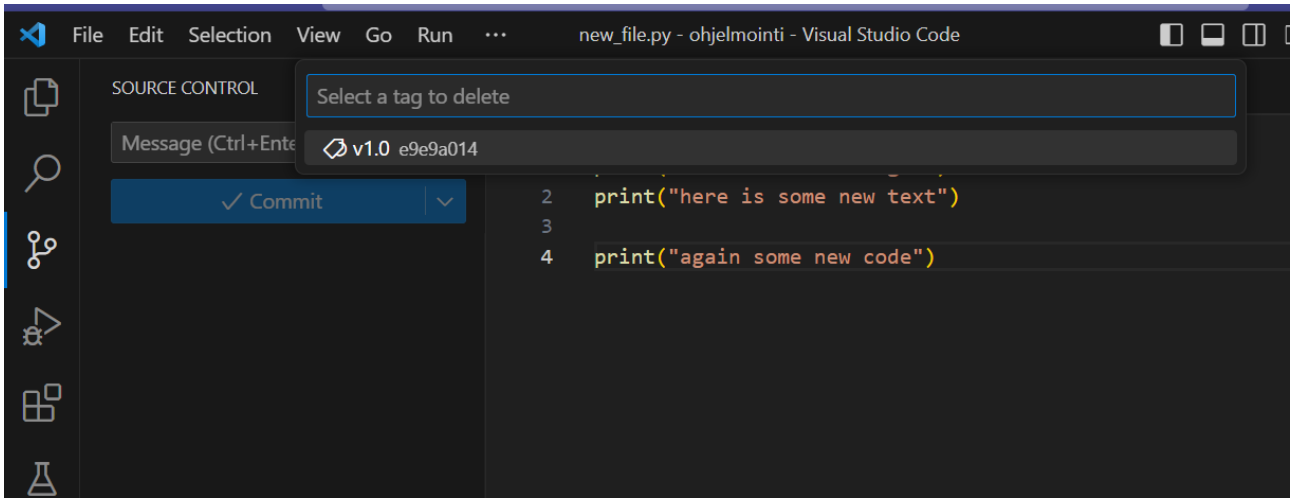
Capture 37 – Example of a Remote Repository Where All the Changes Have Been Successfully Pushed



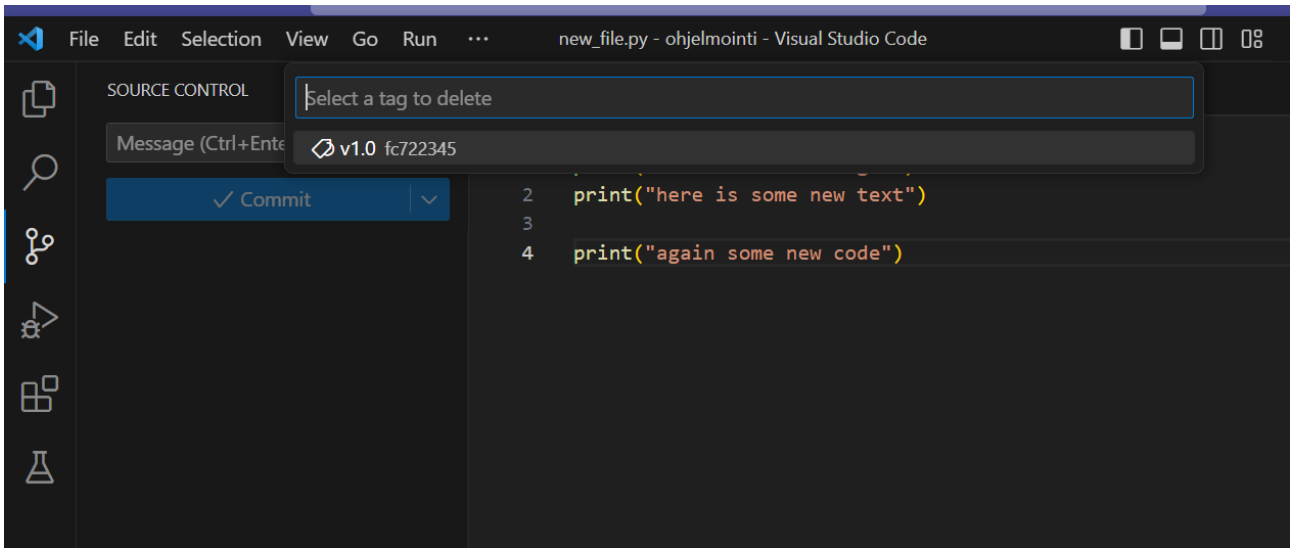
Capture 38 - Example of “Git: Push Tag”



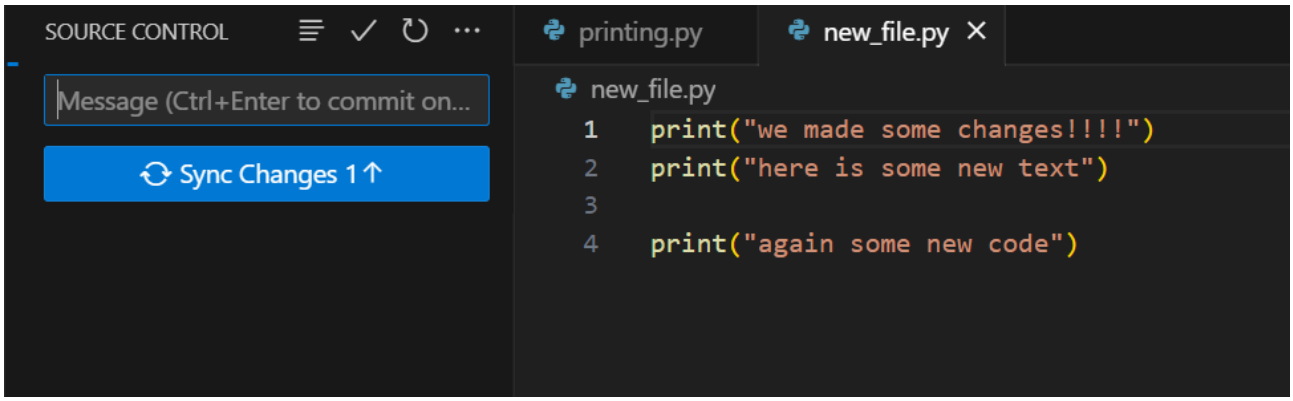
Capture 39 – Delete Remote Tag



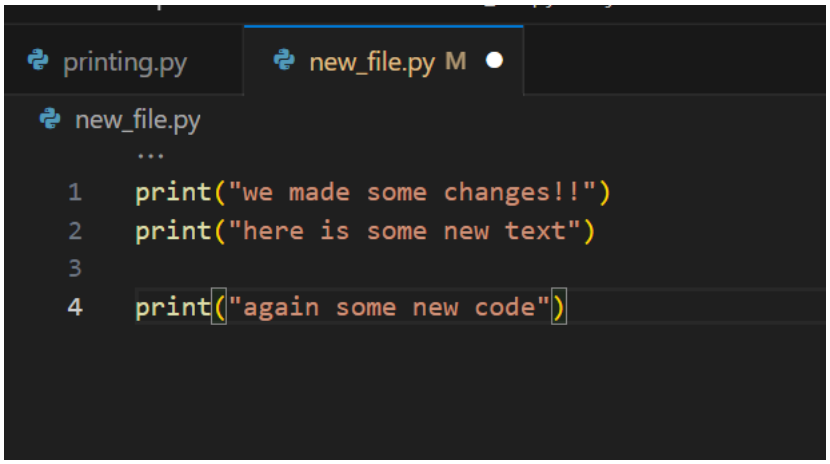
Capture 40 – Delete a Local Tag



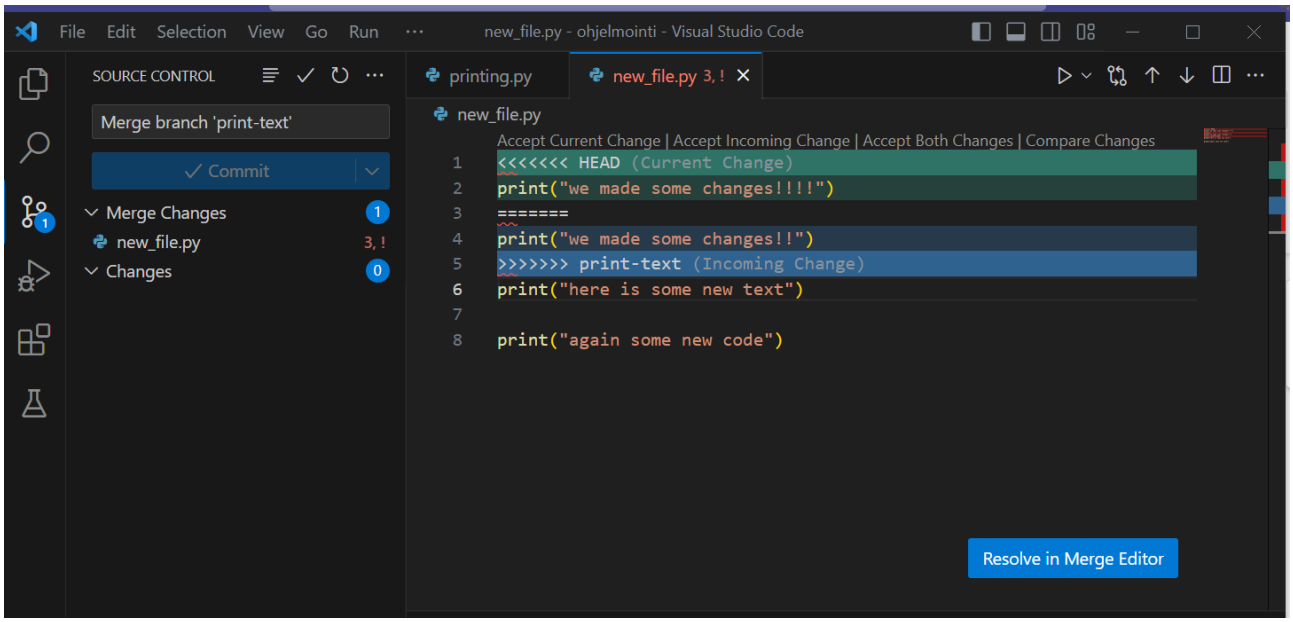
Capture 41 – Causing Conflict – Edit the Main Branch



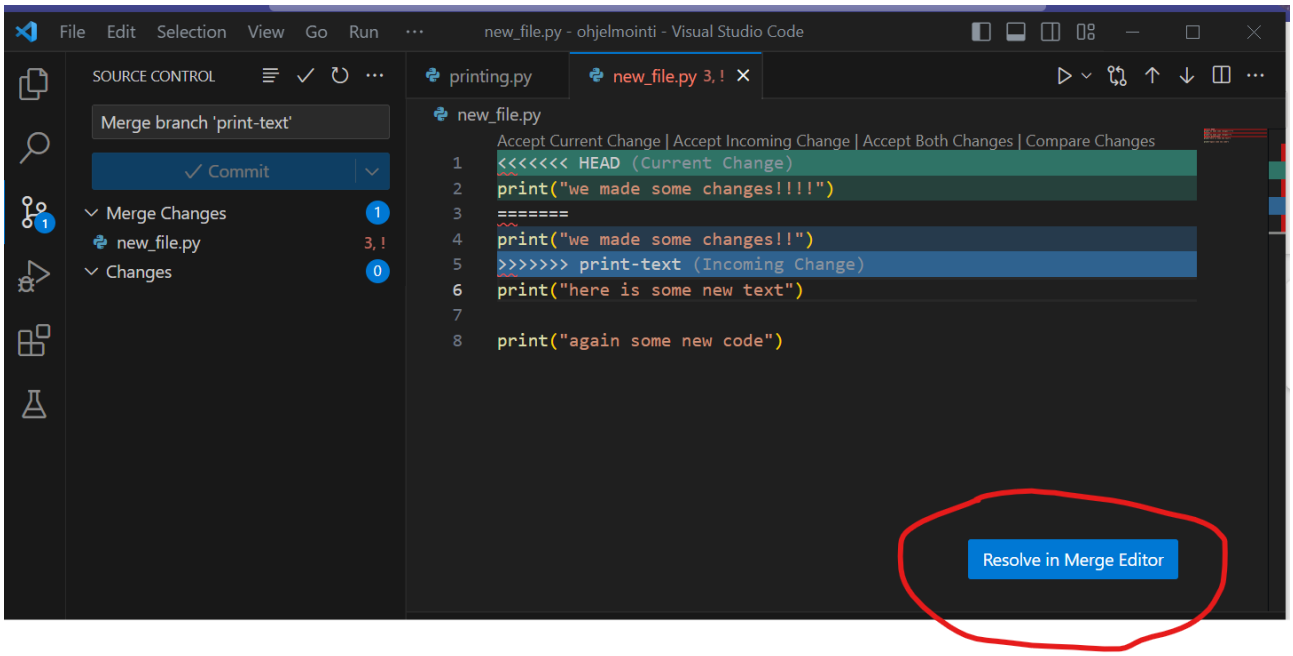
Capture 42 – Causing Conflict – Edit the Working Branch



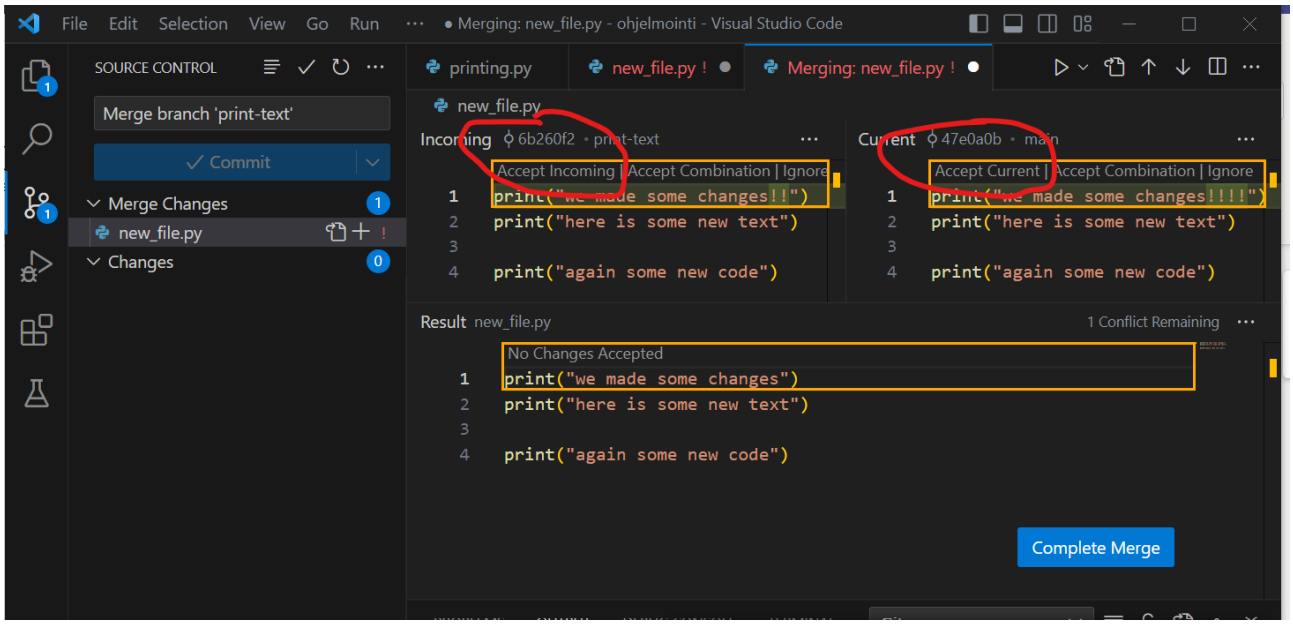
Capture 43 – We Have a Merge Conflict!



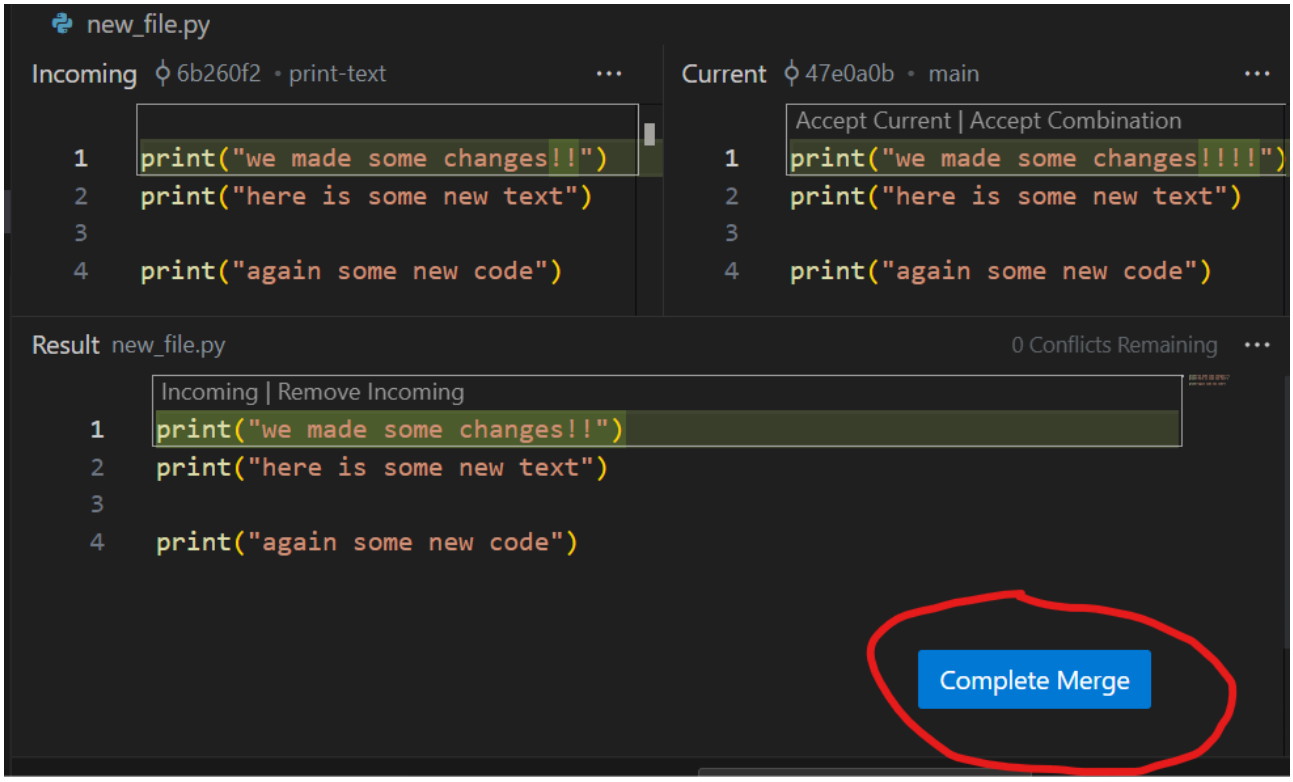
Capture 44 – Resolve in Merge Editor



Capture 45 – Select What You Want to Keep



Capture 46 – Complete Merge



Capture 47 – Example of Remote Repository After Solving Conflicts

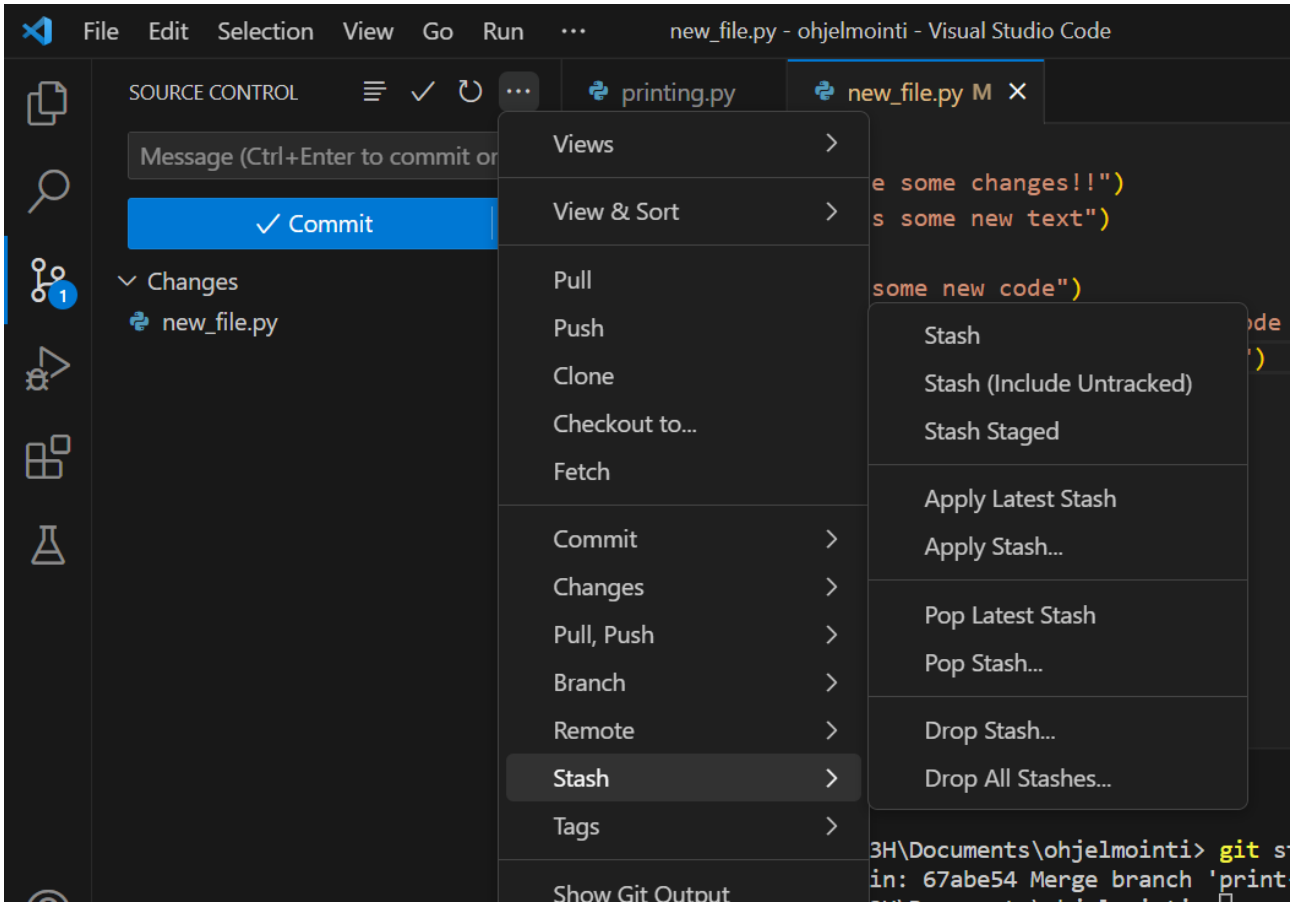
ohjelmointi / new_file.py

Ankinen fixed the first print

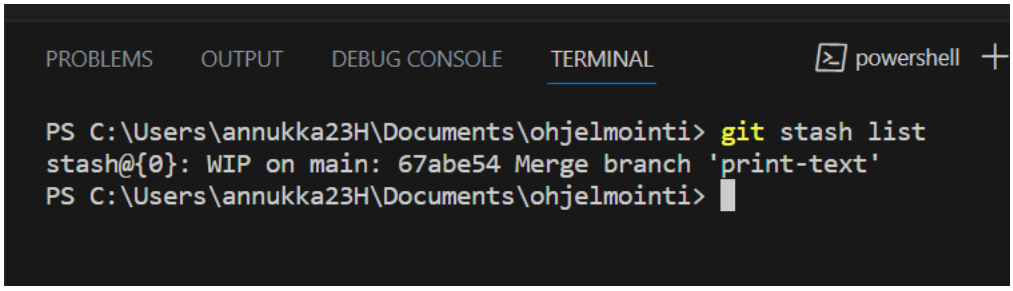
Code Blame 4 lines (3 loc) · 92 Bytes Code 55% faster with GitHub Copilot

```
1  print("we made some changes!!")
2  print("here is some new text")
3
4  print("again some new code")
```

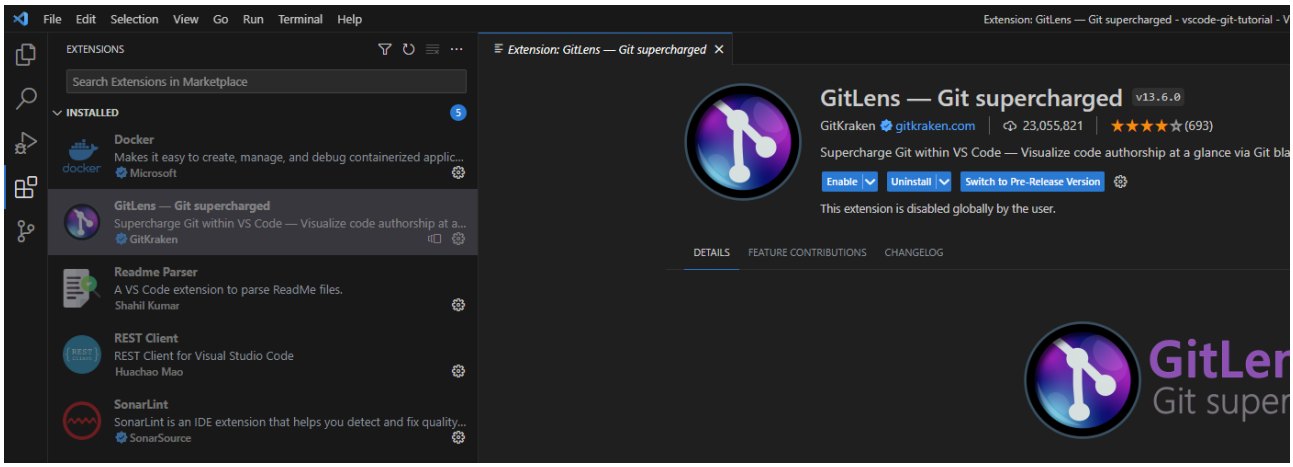
Capture 48 – Stash



Capture 49 – In Terminal Type: "git stash list"



Capture 50 – Install GitLens Extension



Capture 51 – How the Stash Looks Like with GitLens