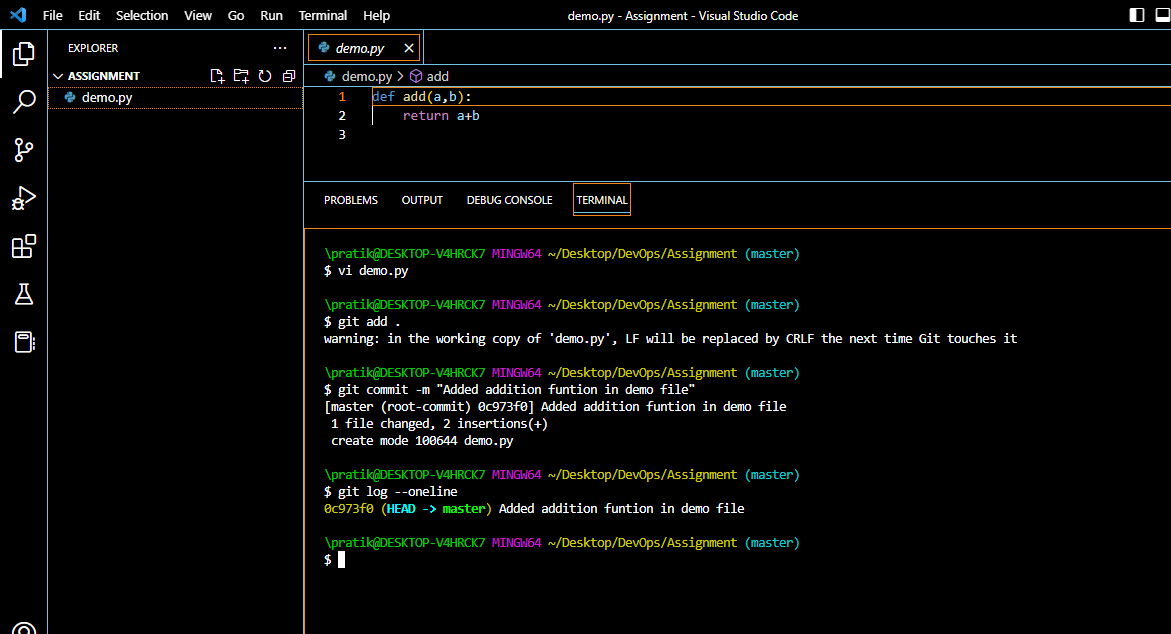
**Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.**

When working on a project, there may be situations in which we think of a feature to be added, but we don’t want to commit it to our project right away. If we are unsure about that addition, we can use git stash to store it.

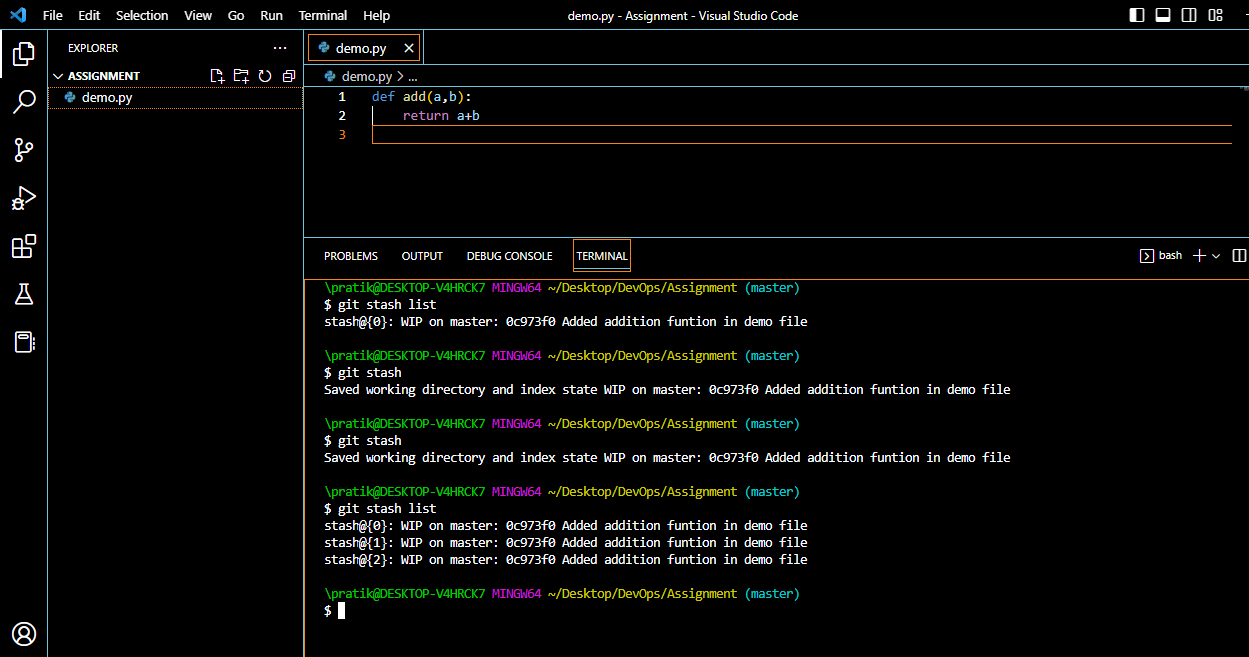
The git stash command stores modifications that we make to a file, and reverts the file back to the HEAD commit. These modifications are stored in a stack structure.

Use command “git stash” after making a modification. It creates a stash.

First we make a commit after adding an addition function to demo file.

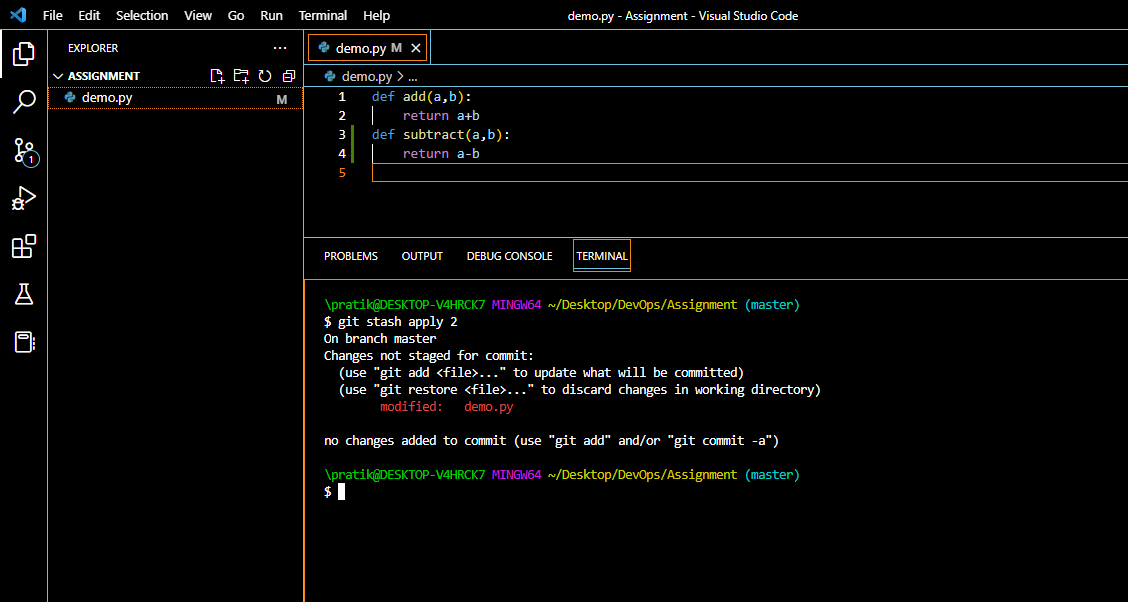


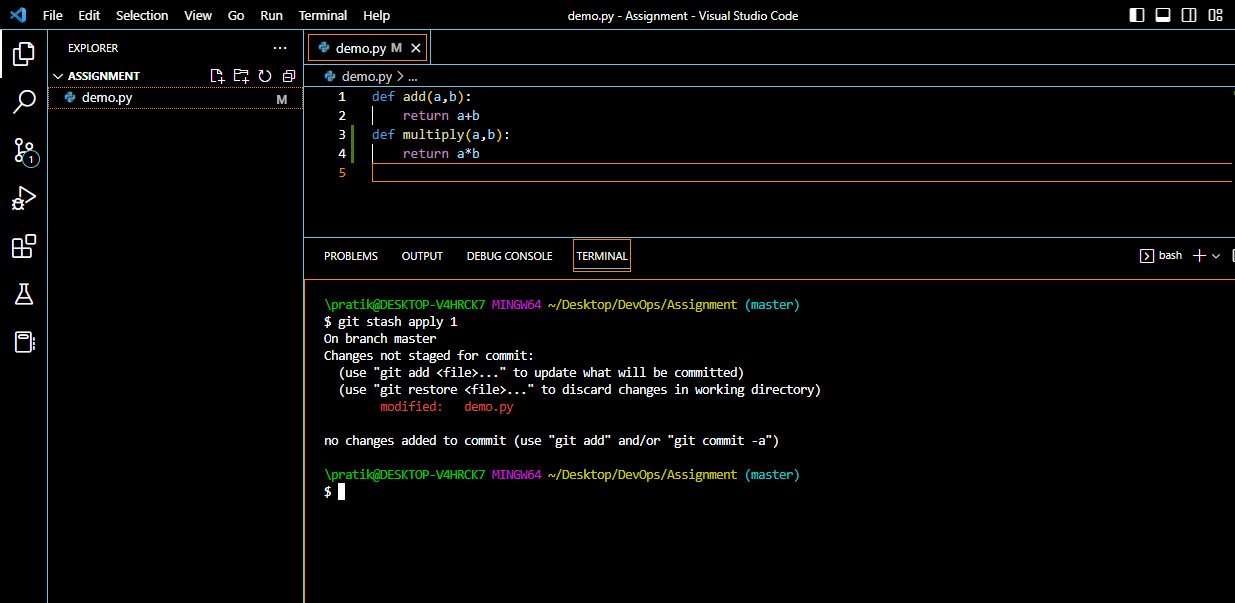
After this, we add a subtraction function and stash it. We also create multiplication and exponent functions and stash them respectively.

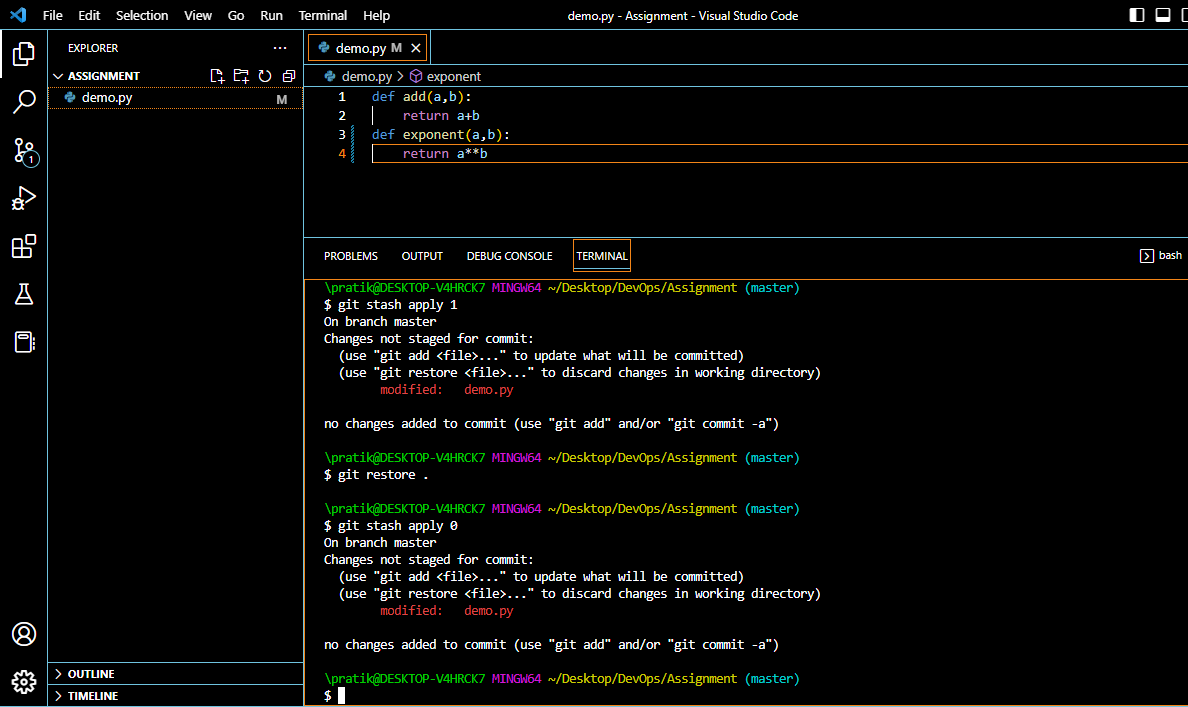


We can use the command “git stash list” to see our stashes with their indices.

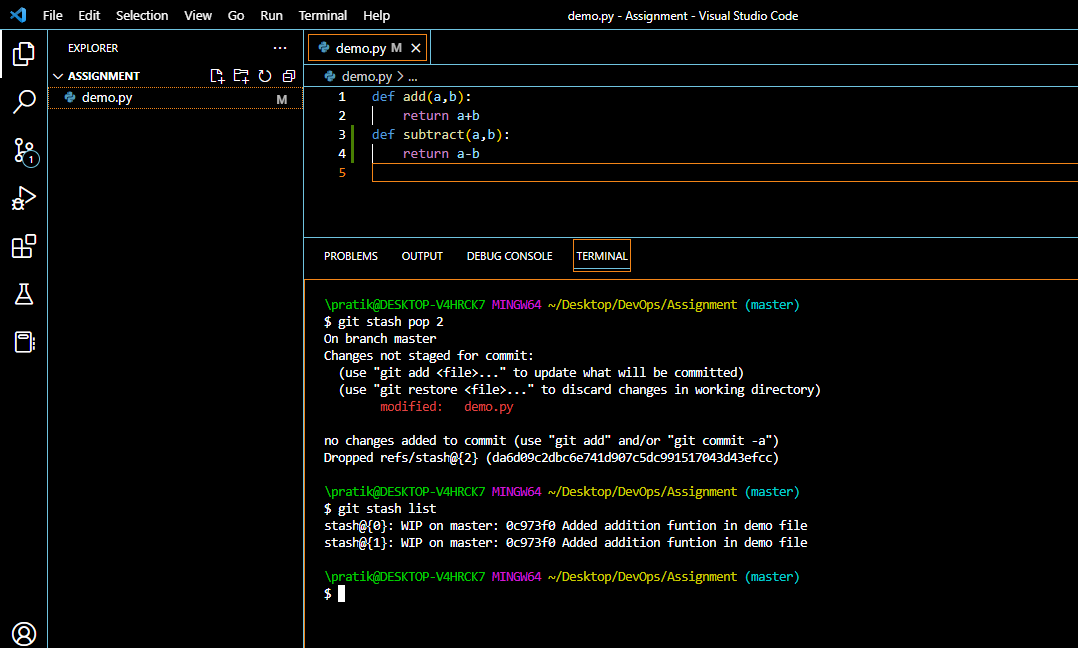
Now, we can use the command “git stash apply [index]” to apply the modifications of a stash to our working directory. From there, we can commit the changes if we like.





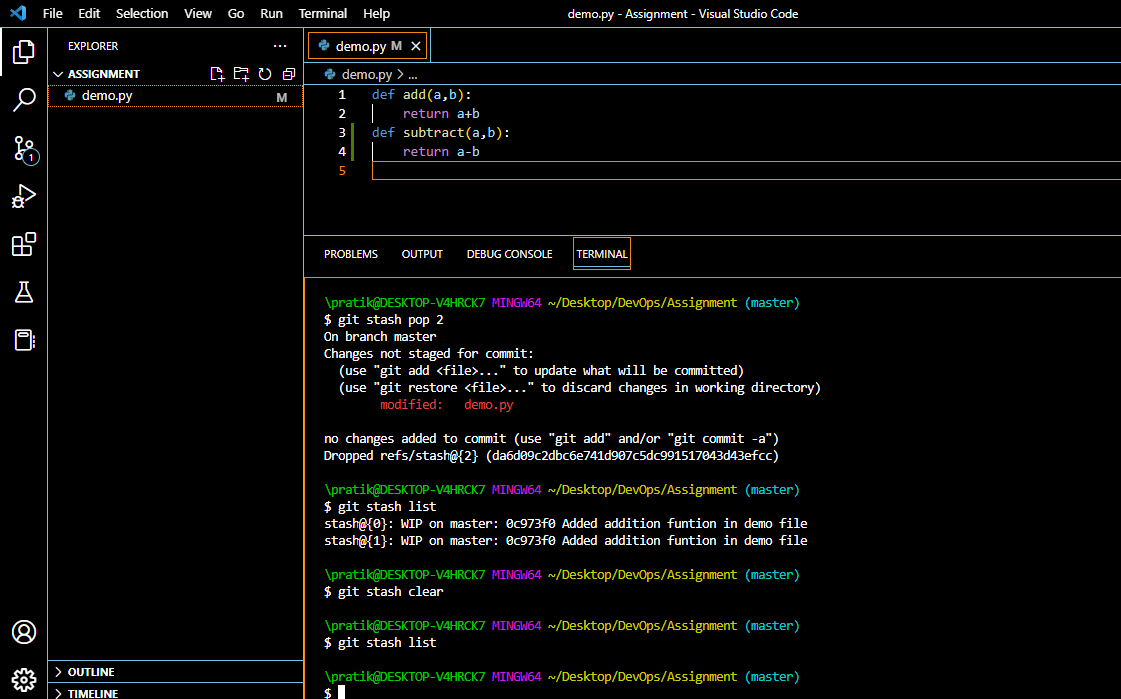


We can use the “git pop [index]” command to apply the changes of a stash while at the same time, removing that stash from the stack we have created.



As you can see, one stash was removed and only two remain.

We can use the command “git stash clear” to drop all the stashes we have created.

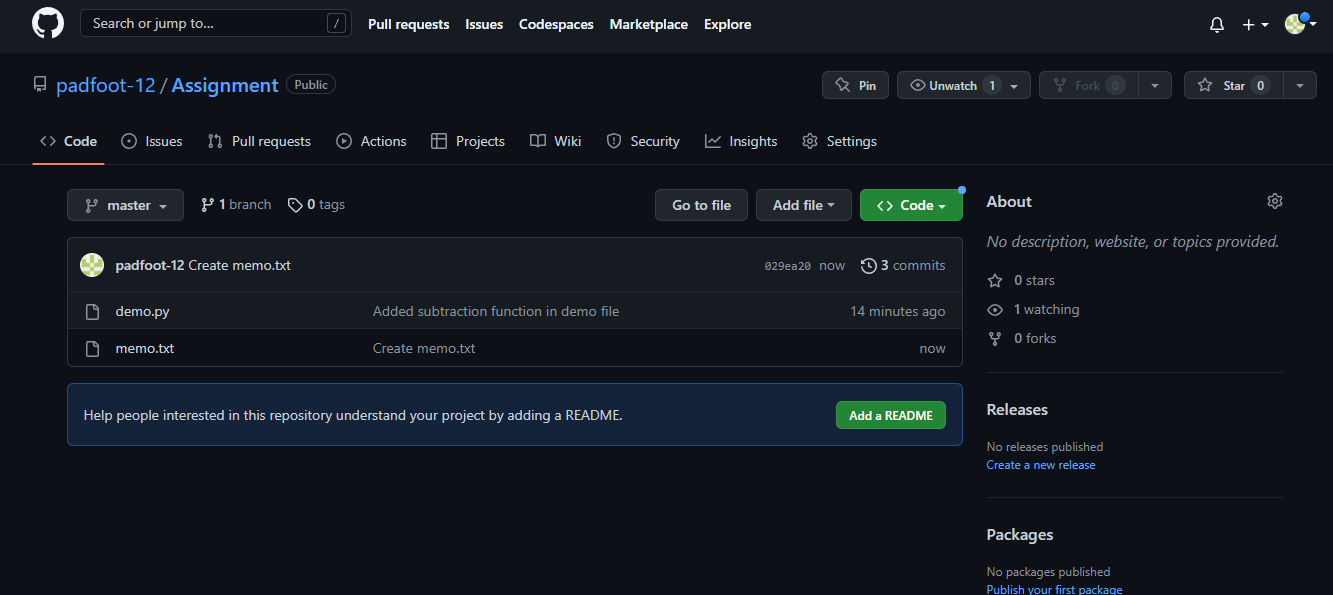


We have no stashes left.

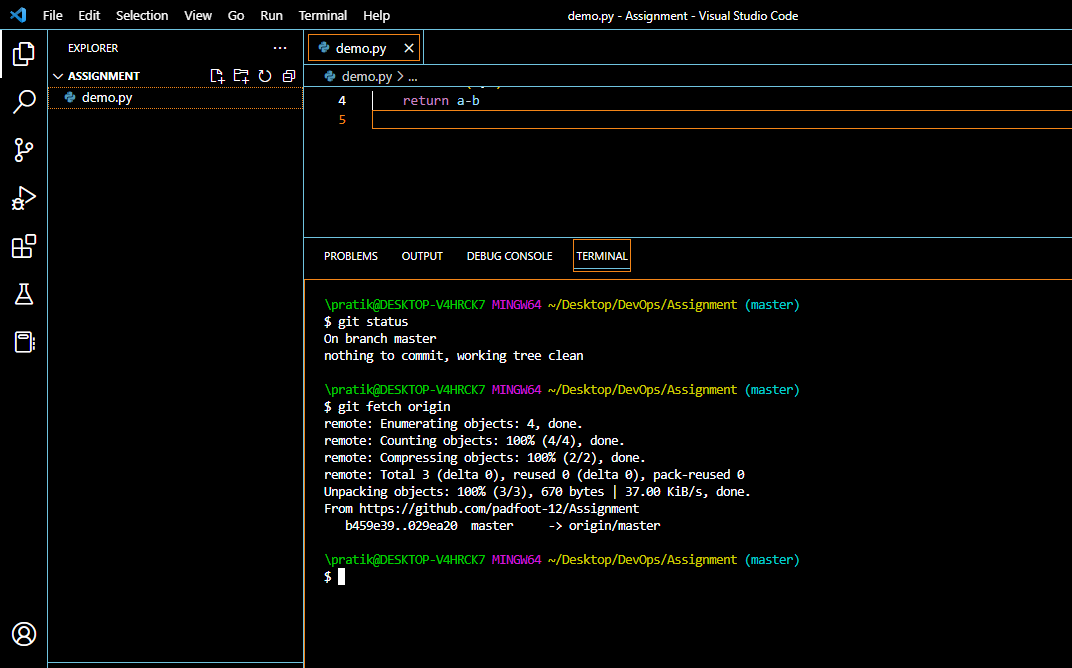
**Q2. By using a sample example of your choice, use the git fetch command and also use the git merge command and describe the whole process through a screenshot with all the commands and their output in git bash.**

The git fetch command is used to retrieve the changes made in our remote repository to our local repository. What this means, basically, is downloading additions made to the remote repository to our local repository. It does not apply those changes to our local repo directly.

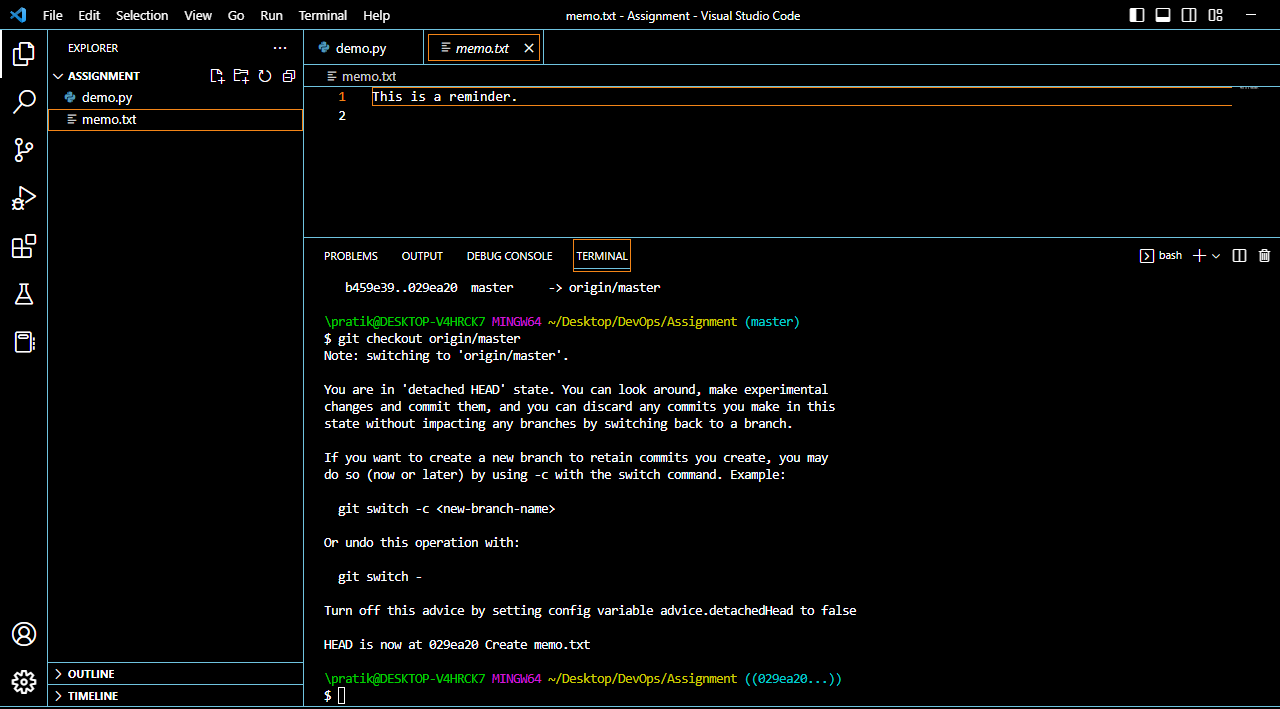
Here, we are creating a next text file called memo in our remote repository.



Now in our local repository, we go ahead and perform the “git fetch origin” command. This will fetch all changes through our remote “origin”.



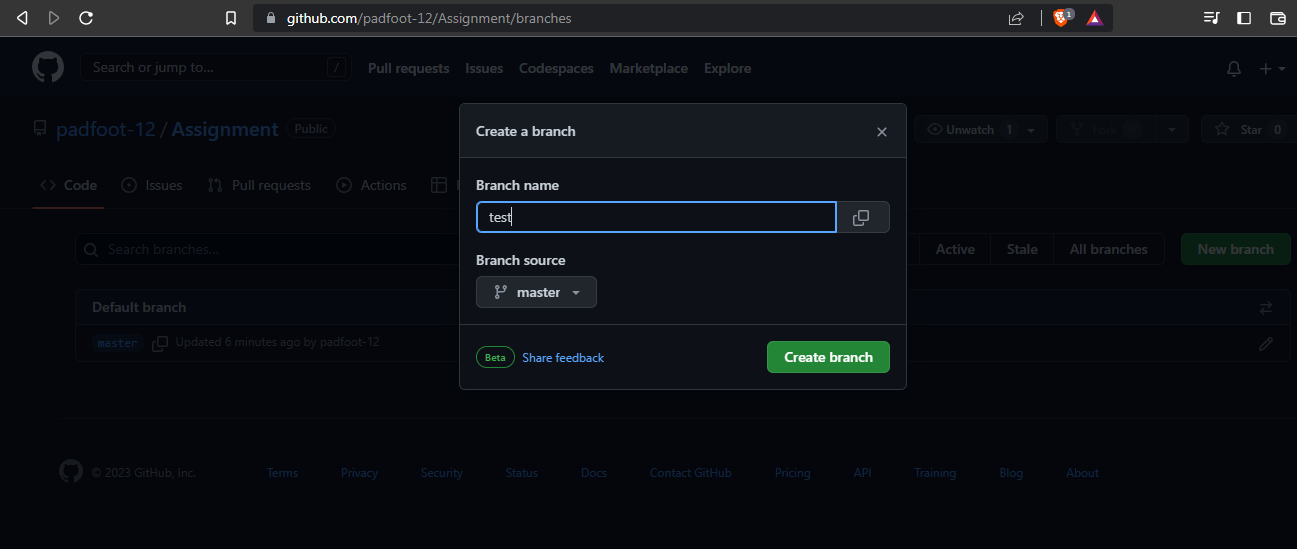
To see the changes, we can checkout the remote branch using command “git checkout origin/master”.



As you can see, now we can see the memo file in our local repository.

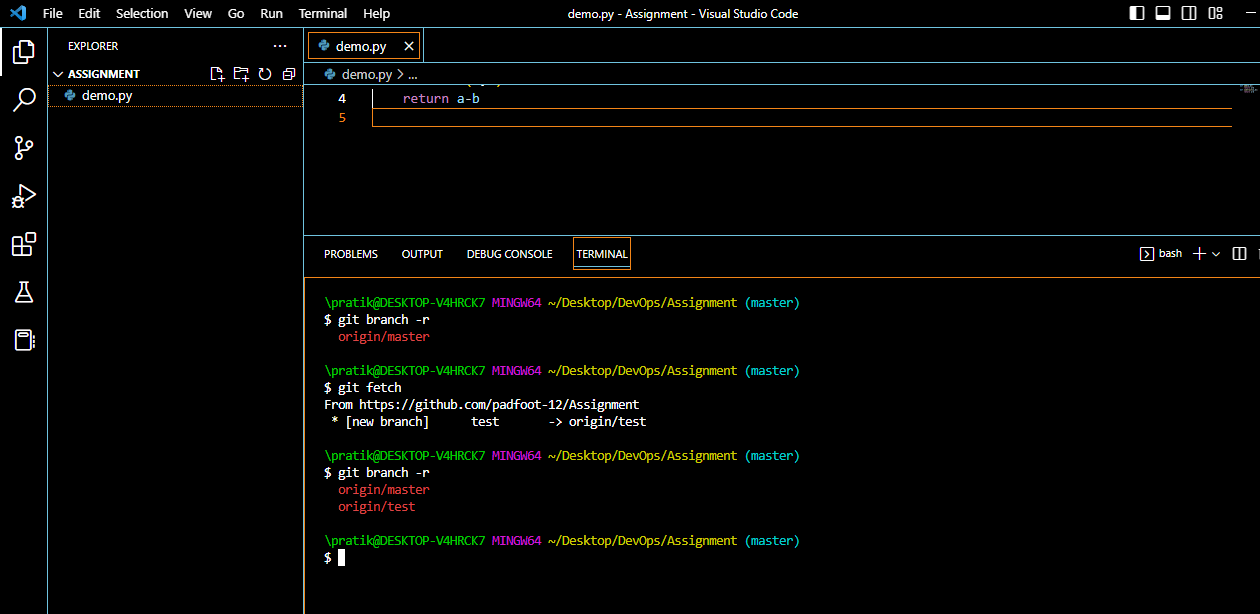
Not only can fetch command fetch changes in a branch, it can fetch a whole new branch in its entirety.

Suppose we create a new branch called “test” in our local repository. Now, this branch will not be available in our local repository.



When we perform the command “git branch –r” it won’t show us the new branch.

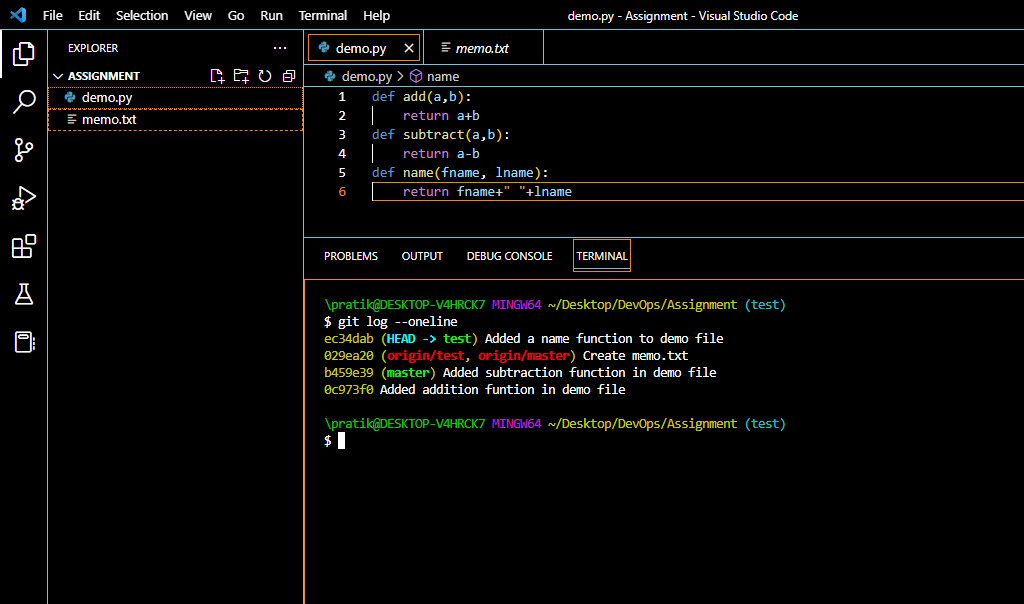
Use the command “git fetch” to fetch the new branch.



As you can see, the new branch is now visible in our local repository.

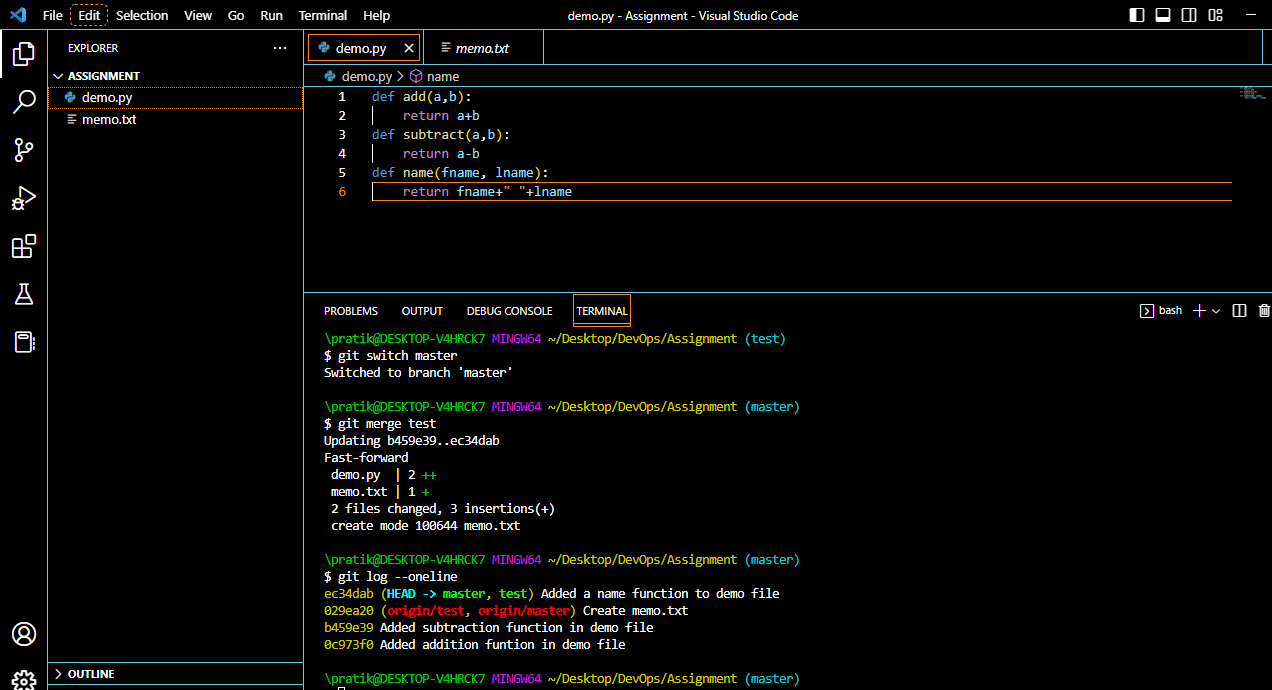
Git “merge” command is used to bring one branch up to date with the commits made on another branch.

Suppose we add a name function in our new branch test.



As we can see in the logs, the branch master is behind the test branch. If we want to include this new function in our master branch, we need to bring it up to date with our test branch, where the HEAD is pointing.

Switch to master branch using “git switch master”. Use command “git merge test” to bring our master branch up to date with the test branch.



As you can see, now our master branch is up to date with the test branch, and has the name function too.

**Q3. State the difference between git fetch and git pull by doing a practical example in your git bash and attach a screenshot of all the processes.**

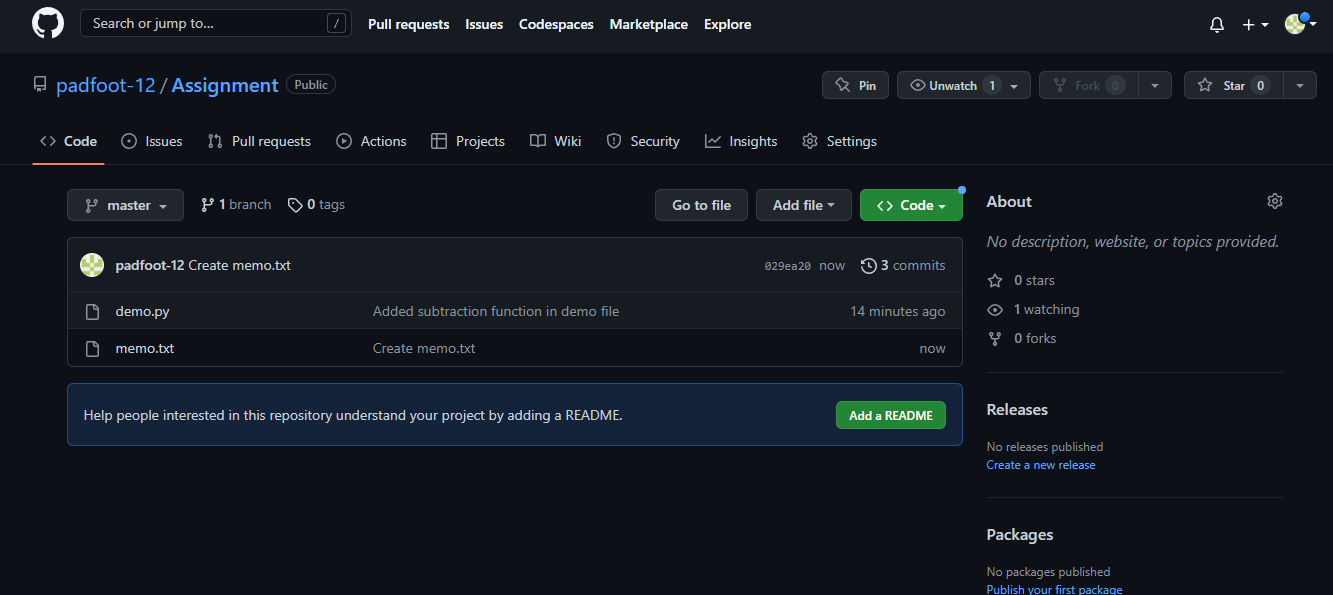
The git fetch command is used to retrieve the changes made in our remote repository to our local repository. What this means, basically, is downloading additions made to the remote repository to our local repository. It does not apply those changes to our local repo directly.

The git pull command is used to apply the changes fetched from a remote repository to our local repository. The pull command downloads and implements the changes as opposed to just downloading them, like the fetch command.

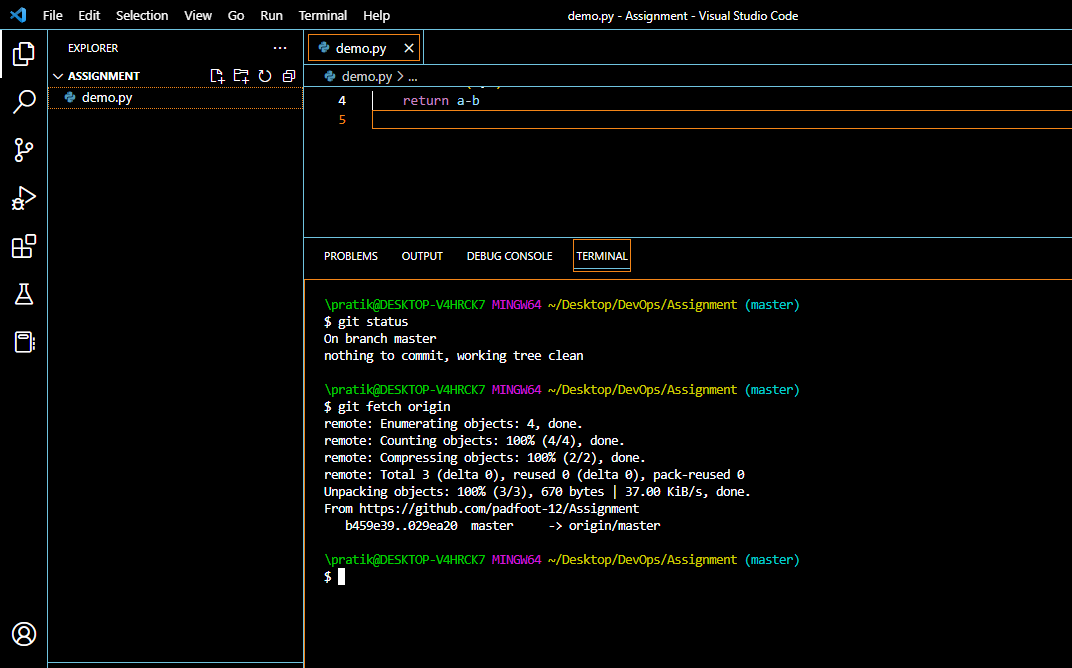
Pull =fetch + merge

**GIT fetch**

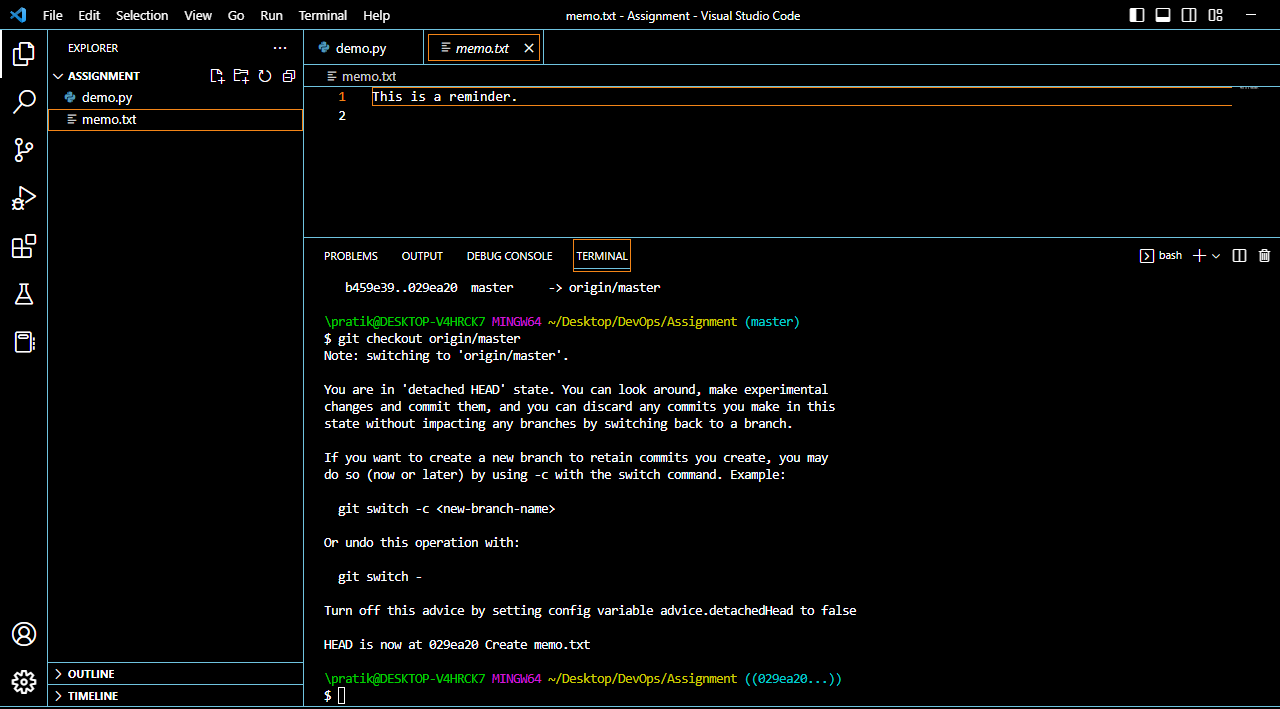
Suppose we create a new file memo in our remote repository.



This will not be visible in our local repository. We can fetch it using the command “git fetch origin”.

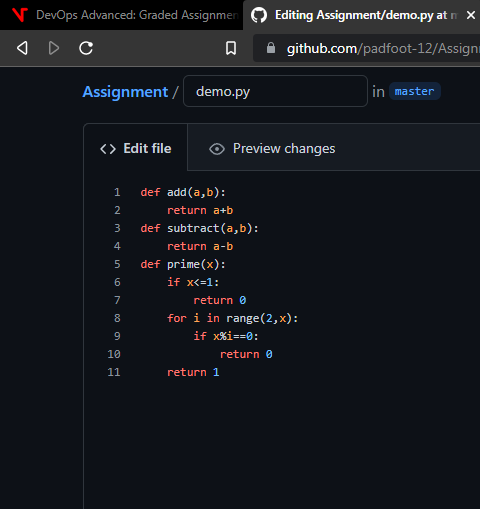


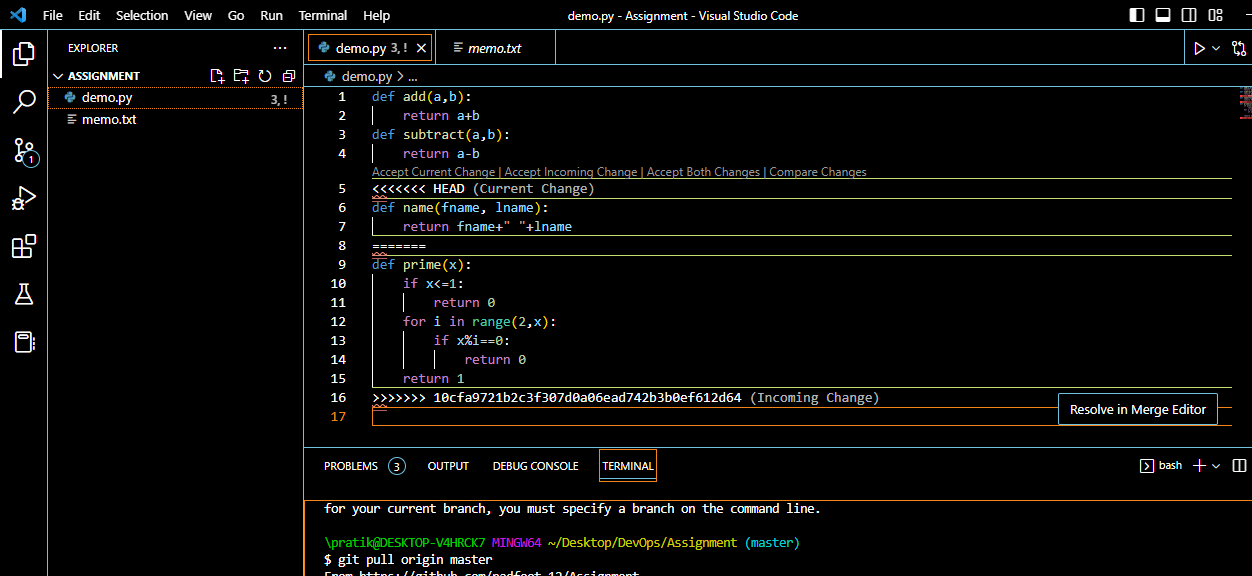
We can see the fetched changes by using “git checkout origin/master” command. They haven’t been directly applied to our local repository.



**GIT pull**

Suppose we add a prime number function in the demo file in our remote repository, and want to apply those changes to our local repository.



We can use the command “git pull origin master” to apply the changes made in our remote branch origin/master to our local branch master.

After resolving the merge conflicts, we get the following message.

