

Git Tutorial

ckeauunui

April 2022

Your tutorial must cover the use of stashes and branches, and one of either rebasing or merging.

Step 1: The first step the user should do is make sure Git is installed on their computer. If it is not installed, run pip install git --version to specify which version of Github to install. I am running Git 2.30.1 for this tutorial.

Step 2: From terminal, run cd/'path' to navigate through your directory until reaching where you want your new program to be saved. Then create an empty folder called GitTutorial with the commands: mkdir GitTutorial Then: cd GitTutorial

Step 3: Put your project into the GitTutorial folder, or create a new document called 'TestFile.txt' with the command: touch TestFile.txt. Then initialize a repository in the folder with the command: git init.

To add a file to this folder use the Git add command :git add 'filename.txt' Check its status with: git status

Step 4: After your files are accumulated in the GitTutorial folder, it is time to create a commit. This is essentially a copy of the current staging area the files are saved in. Do so by using the command: git commit -m "Message". The message is the editors note about the file(s) being committed and what the commit changes were. This could include any new/changed features, bug fixes, or optimization. All commits can be viewed in archive so keeping them labeled helps with overall project organization. Now the files have been added to the Github repo.

Step 5: To edit files that already exist in Github, we will use branching. This creates a temporary copy of the files in a new workshop area which allows us to edit the files without making changes to the original. Multiple branches can also exist at once which allows us to store multiple different versions of a project. Create and open a new branch called MyGitBranch by running the command: git checkout -b MyGitBranch Switch to a new branch using git checkout 'filename'.

Step 6: Once edits to the branch has been finalized you can merge it with the git main branch. This will add the edits to the saved Github project. Use the commands: git checkout master; and git merge MyGitBranch;

Step 7: Staching is similar to branches but on a lower scale. Staching is used to save states of your code, allowing 'dirty' or unfinished work to be saved

or restored to an older state. To create a stache, use: `git stache`. Use the code `git stache pop` to delete the stache, or use `git stache apply` to keep them.

Step 8: