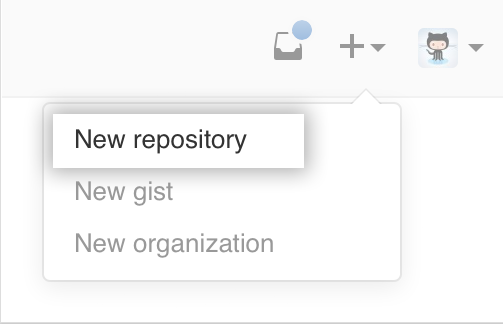
Adding an existing project to GitHub using the Git command line

**Step1 :**

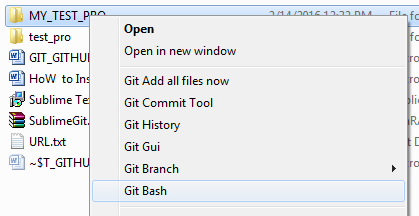
[**Create a new repository**](https://help.github.com/articles/creating-a-new-repository) on GitHub. To avoid errors, do not initialize the new repository with *README*, license, or gitignore files. You can add these files after your project has been pushed to GitHub.



**Step2 :**

Open Terminal (for Mac and Linux users) or the command prompt (for Windows users).

Right click and select the option Git Bash.



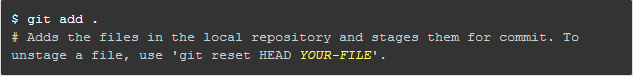
Step 3 :

Initialize the local directory as a Git repository.



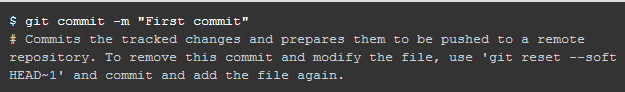
**Step 4 :**

1. Add the files in your new local repository or leave it if you have file already in this folder.
2. Next do This stages them for the first commit.



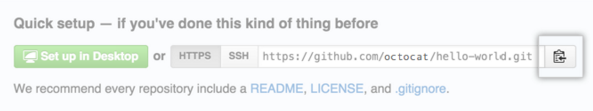
**Step 5**

Commit the files that you've staged in your local repository.



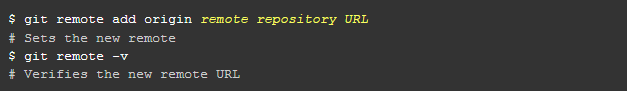
**Step 6:**

At the top of your GitHub repository's Quick Setup page, click  to copy the remote repository URL.



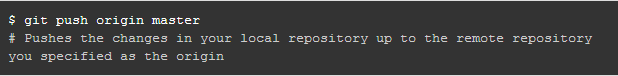
**Step 7:**

In the Command prompt, [add the URL for the remote repository](https://help.github.com/articles/adding-a-remote) where your local repository will be pushed.



**Step 8:**

[Push the changes](https://help.github.com/articles/pushing-to-a-remote) in your local repository to GitHub.

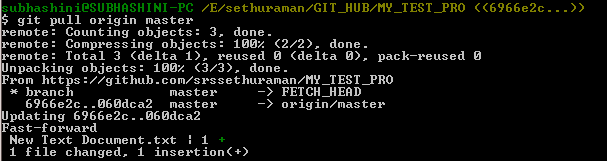


### Step 9 :  Get changes on GitHub back to your computer

Pull the changes from remote Github to your local repository

Right now, the repo on GitHub looks a little different than what you have on your local machine. For example, the commit you made in your branch and merged into the master branch doesn't exist in the master branch on your local machine.

In order to get the most recent changes that you or others have merged on GitHub, use the **git pull origin master**command (when working on the master branch).

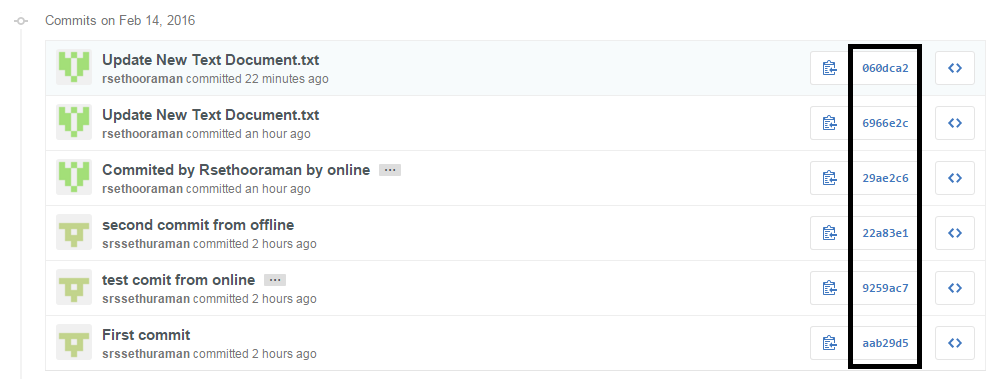


|  |
| --- |
| Difference Git Pull vs. Pull Request  ======================================  If you use git pull, you pull the changes from the remote repository into yours.  If you send a pull request to another repository, you ask their maintainers to pull your changes into theirs (you more or less ask them to use a git pull from your repository).  If you are the maintainer of that repository, it seems you're making it a bit more difficult by pretending you're playing two roles in that workflow. You might as well merge locally your development branch into your master branch and push that master branch into your GitHub repository directly.  (As a side note, if you're new to Git, I'd suggest using git fetch and then git merge instead of git pull. git pull is effectively git fetch followed by git merge, but doing them separately gives you better control over potential conflicts.) |

# Step 9 : [Checkout an older version of github repo](http://stackoverflow.com/questions/12256137/cloning-an-older-version-of-github-repo)

You can always checkout any state you like by using a commit hash.

For instance, by looking a the log, you identified that 233ab4ef was the state you were interested in, issue a **git checkout 233ab4ef**to checkout that state.



|  |
| --- |
| [What is the difference between git clone and checkout?](http://stackoverflow.com/questions/7298598/what-is-the-difference-between-git-clone-and-checkout) To sum it up, clone is for fetching repositories you don't have, checkout is for switching between branches in a repository you already have. |

# [Git Windows Command Prompt gets stuck during Git commands with (END)](http://stackoverflow.com/questions/3923596/git-windows-command-prompt-gets-stuck-during-git-commands-with-end)

* j move one line down
* k move one line up
* <space> move one page down
* b move one page up
* h show the help