



Product

HOME

JOBS

BLOG

OPEN SOURCE

HUBSPOT PRODUCT AND ENGINEERING BLOG

An Intro to Git and GitHub for Beginners (Tutorial)

OCT 1, 2015 / BY [MEGHAN NELSON](#)

Tweet

Share

Like 180

Share

Join our subscribers



Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)



In August, we hosted a [Women Who Code meetup](#) at HubSpot and led a workshop for beginners on using git and GitHub. I first walked through a [slide presentation](#) on the basics and background of git and then we broke out into groups to run through a tutorial I created to simulate working on a large, collaborative project. We got feedback after the event that it was a helpful, hands-on introduction. So if you're new to git, too, follow the steps below to get comfortable making changes to the code base, opening up a pull request (PR), and merging code into the master branch. Any important git and GitHub terms are in bold with links to the official git reference materials.

Step 0: Install git and create a GitHub account

The first two things you'll want to do are install git and create a free GitHub account.

Follow the instructions [here](#) to install git (if it's not already installed). Note that for this tutorial we will be using git on the command line only. While there are some great git GUIs (graphical user interfaces), I think it's easier to learn git using git-specific commands first and then to try out a git GUI once you're more comfortable with the command.

Once you've done that, create a GitHub account and public repositories, but there's a charge for private

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

Step 1: Create a local git repository

When creating a new project on your local machine using git, you'll first create a new **repository** (or often, '**repo**', for short).

To use git we'll be using the terminal. If you don't have much experience with the terminal and basic commands, check out [this tutorial](#) (especially the 'Navigating the Filesystem' and 'Moving Around' sections).

To begin, open up a terminal and move to where you want to place the project on your local machine using the `cd` (change directory) command. For example, if you have a 'projects' folder on your desktop, you'd do something like:

```
mnelson:Desktop mnelson$ cd ~/Desktop
mnelson:Desktop mnelson$ mkdir myproject
mnelson:Desktop mnelson$ cd myproject/
```

terminalcd.md hosted with  by GitHub

[view raw](#)

To initialize a git repository in the root of the folder, run the `git init` command:

```
mnelson:myproject mnelson$ git init
Initialized empty Git repository in /Users/mnelson/Desktop/myproject/.git/
```

gitinit.md hosted with  by GitHub

[view raw](#)

Step 2: Add a new file to the repo

Go ahead and add a new file to the project using the `touch` command.

Once you've added or modified files in a repository, you can commit those changes have been made inside the repository using the `git commit` command. We'll talk more about how to put it in a commit - we'll talk more about that in the next step.

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

```
mnelson:myproject mnelson$ touch mnelson.txt
mnelson:myproject mnelson$ ls
mnelson.txt
```

addfile.md hosted with  by GitHub

[view raw](#)

After creating the new file, you can use the `git status` command to see which files git knows exist.

```
mnelson:myproject mnelson$ git status
On branch master

Initial commit

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

        mnelson.txt

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

gitstatus.md hosted with  by GitHub

[view raw](#)

What this basically says is, "Hey, we noticed you created a new file called `mnelson.txt`, but unless you use the `'git add'` command we aren't going to do anything with it."

An interlude: The staging environment, the commit, and you

One of the most confusing parts when you're first learning git is the concept of the staging environment and how it relates to a commit.

A **commit** is a record of what files you have changed since the last time you made a commit. Essentially, you make changes to your repo (for example, adding a file or modifying one) and then tell git to put those files into the staging environment.

Commits make up the essence of your project at any point.

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

So, how do you tell git which files to put into a commit? This is where the **staging environment** or **index** come in. As seen in Step 2, when you make changes to your repo, git notices that a file has changed but won't do anything with it (like adding it in a commit).

To add a file to a commit, you first need to add it to the staging environment. To do this, you can use the `git add <filename>` command (see Step 3 below).

Once you've used the git add command to add all the files you want to the staging environment, you can then tell git to package them into a commit using the `git commit` command.

Note: The staging environment, also called 'staging', is the new preferred term for this, but you can also see it referred to as the 'index'.

Step 3: Add a file to the staging environment

Add a file to the staging environment using the `git add` command.

If you rerun the git status command, you'll see that git has added the file to the staging environment (notice the "Changes to be committed" line).

```
mnelson:myproject mnelson$ git status
On branch master

Initial commit

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

    new file:   mnelson.txt
```

addtostaging.md hosted with ❤ by GitHub

[view raw](#)

To reiterate, the file has **not** yet been added to a commit. but it's about to be.

Step 4: Create a commit

Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

It's time to create your first commit!

Run the command `git commit -m "Your message about the commit"`

```
mnelson:myproject mnelson$ git commit -m "This is my first commit!"
[master (root-commit) b345d9a] This is my first commit!
1 file changed, 1 insertion(+)
create mode 100644 mnelson.txt
```

commit.md hosted with  by GitHub

[view raw](#)

The message at the end of the commit should be something related to what the commit contains - maybe it's a new feature, maybe it's a bug fix, maybe it's just fixing a typo. Don't put a message like "asdfadsf" or "foobar". That makes the other people who see your commit sad. Very, very, sad.

Step 5: Create a new branch

Now that you've made a new commit, let's try something a little more advanced.

Say you want to make a new feature but are worried about making changes to the main project while developing the feature. This is where **git branches** come in.

Branches allow you to move back and forth between 'states' of a project. For instance, if you want to add a new page to your website you can create a new branch just for that page without affecting the main part of the project. Once you're done with the page, you can **merge** your changes from your branch into the master branch. When you create a new branch, Git keeps track of which commit your branch 'branched' off of, so it knows the history behind all the files.

Let's say you are on the master branch and want to create a new branch to develop your web page. Here's what you'll do: Run **git ch** automatically create a new branch and th that branch, off of the master branch.

After running the above command, you c branch was created:

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

```
mnelson:myproject mnelson$ git branch
  master
* my-new-branch
```

gitbranch.md hosted with  by GitHub

[view raw](#)

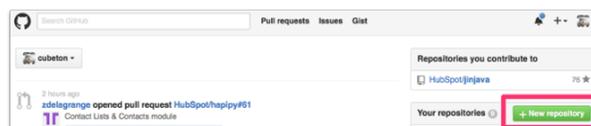
The branch name with the asterisk next to it indicates which branch you're pointed to at that given time.

Now, if you switch back to the master branch and make some more commits, your new branch won't see any of those changes until you **merge** those changes onto your new branch.

Step 6: Create a new repository on GitHub

If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify the project's code.

To create a new repo on GitHub, log in and go to the GitHub home page. You should see a green '+ New repository' button:



Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

After clicking the button, GitHub will ask


```
mnelson:myproject mnelson$ git remote add origin https://github.com/cubeton/mynewrepository.git
mnelson:myproject mnelson$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 263 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/cubeton/mynewrepository.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

addgithub.md hosted with  by GitHub

[view raw](#)

(You'll want to change the URL in the first command line to what GitHub lists in this section since your GitHub username and repo name are different.)

Step 7: Push a branch to GitHub

Now we'll **push** the commit in your branch to your new GitHub repo. This allows other people to see the changes you've made. If they're approved by the repository's owner, the changes can then be merged into the master branch.

To push changes onto a new branch on GitHub, you'll want to run **git push origin yourbranchname**. GitHub will automatically create the branch for you on the remote repository:

```
mnelson:myproject mnelson$ git push origin my-new-branch
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/cubeton/mynewrepository.git
 * [new branch]      my-new-branch -> my-new-branch
```

addnewbranchgithub.md hosted with  by GitHub

[view raw](#)

You might be wondering what that "origin" is that when you clone a remote repository. In nearly all cases this alias is called "origin" and it's the repository's URL. So, to push your changes

Join our subscribers

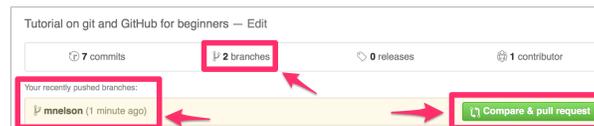
Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

the command: `git push git@github.com:git/git.git yourbranchname` or `git push origin yourbranchname`

(If this is your first time using GitHub locally, it might prompt you to log in with your GitHub username and password.)

If you refresh the GitHub page, you'll see a note saying a branch with your name has just been pushed into the repository. You can also click the 'branches' link to see your branch listed there.

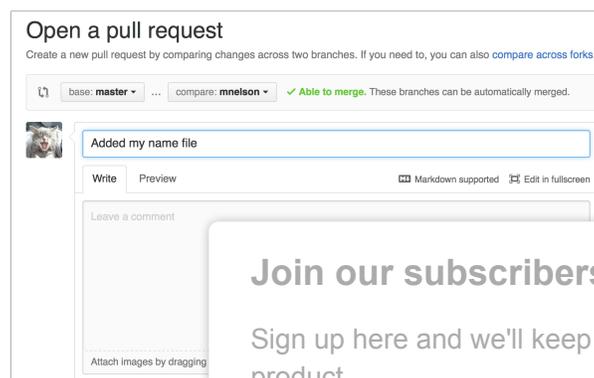


Now click the green button in the screenshot above. We're going to make a **pull request!**

Step 8: Create a Pull Request (PR)

A pull request (or PR) is a way to alert a repo's owners that you want to make some changes to their code. It allows them to review the code and make sure it looks good before putting your changes on the master branch.

This is what the PR page looks like before you've submitted it:

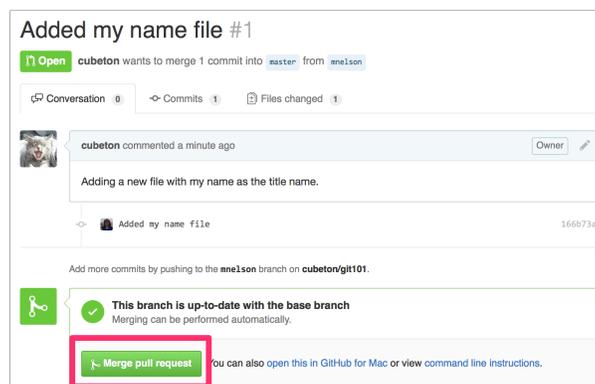


Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

And this is what it looks like once you've submitted the PR request:



You might see a big green button at the bottom that says 'Merge pull request'. Clicking this means you'll merge your changes into the master branch.

Note that this button won't always be green. In some cases it'll be grey, which means you're faced with a **merge conflict**. This is when there is a change in one file that conflicts with a change in another file and git can't figure out which version to use. You'll have to manually go in and tell git which version to use.

Sometimes you'll be a co-owner or the sole owner of a repo, in which case you may not need to create a PR to merge your changes. However, it's still a good idea to make one so you can keep a more complete history of your updates and to make sure you always create a new branch when making changes.

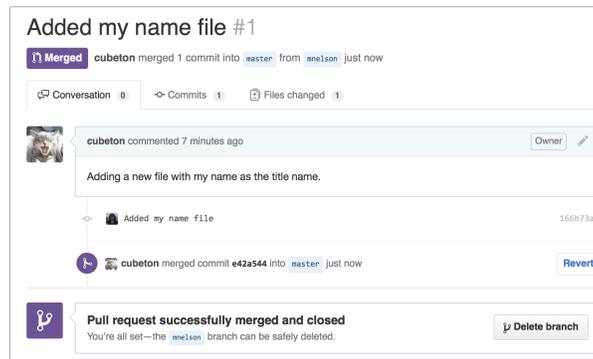
Step 9: Merge a PR

Go ahead and click the green 'Merge pull request' button. This will merge your changes into the master branch.

Join our subscribers ✕

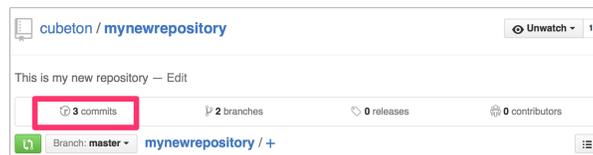
Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

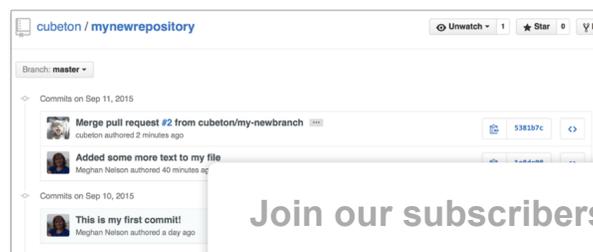


When you're done, I recommend deleting your branch (too many branches can become messy), so hit that grey 'Delete branch' button as well.

You can double check that your commits were merged by clicking on the 'Commits' link on the first page of your new repo.



This will show you a list of all the commits in that branch. You can see the one I just merged right up top (Merge pull request #2).



You can also see the **hash code** of the commit identifier for that specific commit. It's use

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

changes (use the `git revert <hash code number>` command to backtrack).

Step 10: Get changes on GitHub back to your computer

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).

```
mnelson:myproject mnelson$ git pull origin master
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From https://github.com/cubeton/mynewrepository
 * branch          master      -> FETCH_HEAD
  b345d9a..5381b7c  master      -> origin/master
Merge made by the 'recursive' strategy.
 mnelson.txt | 1 +
 1 file changed, 1 insertion(+)
```

pulloriginmaster.md hosted with ❤ by GitHub

[view raw](#)

This shows you all the files that have changed and how they've changed.

Now we can use the `git log` command again to see all new commits.

(You may need to switch branches back to the master branch. You can do that using the `git checkout master` command.)

```
mnelson:myproject mnelson$ git log
commit 3e270876db0e5fffd3e9bfc5edede89b64b83812c
Merge: 4f1cb17 5381b7c
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Fri Sep 11 17:48:11 2015 -0400

    Merge branch 'master' of https://github.com/cubeton/mynewrepository

commit 4f1cb1798b6e6890da797f98383e6337df577c2a
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Fri Sep 11 17:48:00 2015 -0400

    added a new file

commit 5381b7c53212ca92151c743b4ed7dde07d9be3ce
Merge: b345d9a 1e8dc08
```

Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

```
Author: Meghan Nelson <meghan@meghan.net>
Date:   Fri Sep 11 17:43:22 2015 -0400

Merge pull request #2 from cubeton/my-newbranch

Added some more text to my file

commit 1e8dc0830b4db8c93efd80479ea886264768520c
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Fri Sep 11 17:06:05 2015 -0400

Added some more text to my file

commit b345d9a25353037afdeaa9fcaf9f330effd157f1
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Thu Sep 10 17:42:15 2015 -0400

This is my first commit!
```

gitlogaftermerge.md hosted with  by GitHub

[view raw](#)

Step 11: Bask in your git glory

You've successfully made a PR and merged your code to the master branch. Congratulations! If you'd like to dive a little deeper, check out the files in [this Git101 folder](#) for even more tips and tricks on using git and GitHub.

I also recommend finding some time to work with your team on simulating a smaller group project like we did here. Have your team make a new folder with your team name, and add some files with text to it. Then, try pushing those changes to this remote repo. That way, your team can start making changes to files they didn't originally create and practice using the PR feature. And, use the git blame and git history tools on GitHub to get familiar with tracking which changes have been made in a file and who made those changes.

The more you use git, the more comfortable you'll... git with it. (I couldn't resist.)

Join our subscribers

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)



Written by [Meghan Nelson](#)

Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

SUBSCRIBE FOR UPDATES

[Subscribe me to the blog](#)

FIRST NAME

LAST NAME

EMAIL

Subscribing here provides HubSpot with contact information that we will use to email you HubSpot Product Blog content from time to time. We won't share your information with anyone outside of HubSpot. We may keep the information you submitted for up to three years (don't worry though, spam isn't our thing). If you'd like to know more about how we use your personal data please visit <https://legal.hubspot.com/privacy-policy>

SUBSCRIBE

TOPICS

[PRODUCT >>](#)

[ENGINEERING >>](#)

[DESIGN >>](#)

[PRODUCT MANAGEMENT >>](#)

[UX >>](#)

[CULTURE >>](#)

[CAREER GROWTH >>](#)

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)



Sam 10/2/2015, 9:34:59 AM

Very informative post! Thanks!

[REPLY TO SAM](#)

Deepika Sharma 2/9/2020, 11:07:40 PM

Thanks...it is very helpful .

[REPLY TO DEEPIKA SHARMA](#)

Vikram Ketkar 3/22/2020, 11:15:21 AM

Nice. it was a good overview with so

Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

[REPLY TO VIKRAM KETKAR](#)

June 12/13/2015, 9:14:56 PM

Thanks for the tutorial, but I have one question. I've done to step 8 (Step 8: Create a Pull Request (PR), but the github page could not find any changes, it says "master and my-branch-name are identical." I tried several times but still couldn't find the reason. I don't know where did I do wrong, anyone has the same problem?

[REPLY TO JUNE](#)

Meghan Nelson 12/14/2015, 11:04:26 PM

Hi June! I'm glad you're going through the tutorial. It sounds like what's happening is you've created your new branch on GitHub, but you haven't made changes (commits) on that branch or pushed your local changes (commits) onto GitHub.

What you can do is:

1. Make sure you're working on the correct branch (use the 'git branch' command). There should be an asterisk next to your branch name you're currently working on.
2. Make sure you've made commits on your branch. You can use the 'git status' command to see if there are any unsaved changes (You want it to say 'On branch my-branch-name nothing to commit, working directory clean'). You can use the 'git log' command to make sure that you have made commits on that branch. You'll need to make at least one commit for there to be differences to appear.
3. Finally once you're sure you've made a commit on that branch, try redoing the 'git push origin my-branch-name' command and refresh the page on GitHub. You should then your changes listed in your branch on GitHub.

I hope this helps, let me know if you run into any problems!

[REPLY TO MEGHAN NELSON](#)

June 12/15/2015, 6:24:09 AM

Hi Meghan,

Thanks for the answer. Yesterday I used the 'git push' command!

I'm completely new in coding, ha

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

something really easy to follow like your tutorial is such a great happiness.

Thanks so much!

[REPLY TO JUNE](#)

[Michael Owen](#) 12/30/2015, 5:50:48 PM

Thanks for the answer. Yesterday I tried it again and finally could make the git push command! I'm completely new in coding, have just studied it for a few months, and finding something really easy to follow like your tutorial is such a great happiness.

Thanks so much!

Thanks for the answer. Yesterday I tried it again and finally could make the git push command! I'm completely new in coding, have just studied it for a few months, and finding something really easy to follow like your tutorial is such a great happiness.

Thanks so much!

[REPLY TO MICHAEL OWEN](#)

[Kristin Day](#) 5/31/2016, 6:05:20 PM

This was so helpful! Thank you so much for posting!!

[REPLY TO KRISTIN DAY](#)

[Virendra More](#) 8/14/2016, 6:19:51 PM

Can I configure remote repository with corporate domain user accounts?

[REPLY TO VIRENDRA MORE](#)

[Jesse Lind](#) 11/30/2019, 2:21:52 AM

Hello. Great post--SO HELPFUL!!! :) I was floundering until I read it. You have the gift of teaching! Anyways...I followed the above tutorial and merged the branches on gitHub; however, once I ran `git pull origin master` from the command line, the extra branch (`test-branch`) that was merged into master on gitHub is still present on my local machine. Is that expected behavior? I figured the `git pull` command would cause my local machine to mirror what's in my gitHub repo.

[REPLY TO JESSE LIND](#)

[Pankaj Sharma](#) 12/3/2019, 1:55:59 PM

Simply awesome . very simple and beautiful

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

[REPLY TO PANKAJ SHARMA](#)

[Mn Mm](#), 12/14/2019, 10:54:44 PM

m

[REPLY TO MN MM](#),

Amalendu Kar 12/20/2019, 12:17:06 AM

Thanks, awesome tutorial

[REPLY TO AMALENDU KAR](#)

Carter 12/30/2019, 10:22:36 AM

Thank you so much! Guides like this are a lifesaver for absolute beginners like me.

[REPLY TO CARTER](#)

Joseph McCarron 1/6/2020, 6:38:32 AM

The reason why people are having the problem with the identical file thing is that after creating the branch the file was not changed so it was still identical to the master branch

[REPLY TO JOSEPH MCCARRON](#)

New Togit 2/8/2020, 1:07:14 PM

Joseph, yeah I had to redo this exercise 3 times because the part mentioning about modifying the working branch before sending it to GitHub was missing.

Otherwise this was a great tutorial!

[REPLY TO NEW TOGIT](#)

Ndip Des 1/7/2020, 12:57:00 PM

how can I view the side I just push in the browser

[REPLY TO NDIP DES](#)

Nayan Maheshwari 1/9/2020, 12:17:49 PM

Thank you so much for this simple yet effective tutorial. It really helped me and it will save me time for every time uploading my code.

[REPLY TO NAYAN MAHESHWARI](#)

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

Pranjal Saxena 1/20/2020, 2:55:49 PM

It helps for gate

[REPLY TO PRANJAL SAXENA](#)

Jake 1/22/2020, 12:56:03 AM

Awesome tutorial! I will definitely show this to my group to teach them the basics of git and github!

[REPLY TO JAKE](#)

Jenitha Rashmi 1/24/2020, 3:02:24 PM

Very informative....Nicely explained...Hats off to you Meghan :):)

[REPLY TO JENITHA RASHMI](#)

Reeja Rejin 1/28/2020, 10:10:45 PM

Would you also please post - " How to deploy a website using GitHub" for beginners. Thank You so much !

[REPLY TO REEJA REJIN](#)

New To Git 2/8/2020, 1:03:59 PM

Otherwise a great tutorial but as mentioned above you could add a paragraph about first modifying the mnelson.txt in working branch before sending it to GitHub. This would eliminate the general confusion why pull request could not be done (at there were not changes in the updated branch as done per your instructions)

[REPLY TO NEW TO GIT](#)

Deepika Sharma 2/9/2020, 11:15:47 PM

Thanks ..! it is very helpful.

[REPLY TO DEEPIKA SHARMA](#)

Damnd Belger 2/10/2020, 5:38:08 AM

i stalled at step 6. permission denied (pu

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

i'm pretty new to this. i found online to fix this error it's necessary to generate an ssh-key. but the follow up was unclear. what does it mean and how do i go about this.

REPLY TO DAMND BELGER

Alex 3/24/2020, 2:47:15 PM

<https://www.youtube.com/watch?v=ouO4BpGvEnE>

REPLY TO ALEX

Guru Sharan 2/12/2020, 12:12:22 AM

This is the best , cleanest and straightforward explanation on GIT. Reading once I can not forget for ever in my life. Good Job for creating and sharing this page.

REPLY TO GURU SHARAN

Andy 2/14/2020, 8:00:31 AM

Can you please check the link to the tutorial on "basic commands" in Step 1? I clicked through and was unable to find the tutorial at the link provided: <http://mac.appstorm.net/how-to/utilities-how-to/how-to-use-terminal-the-basics/>

Thank you.

REPLY TO ANDY

Muhammad Zawawi Manja 2/14/2020, 2:18:19 PM

This article is on the first page and the first row in the google.

REPLY TO MUHAMMAD ZAWAWI MANJA

Van Hudson Galvoso 2/21/2020, 8:04:12 AM

Thank you very much! This helps a lot.

REPLY TO VAN HUDSON GALVOSO

Karthik Kumar 2/26/2020, 5:11:20 AM

Thank you very much for taking time in r

REPLY TO KARTHIK KUMAR

Jan H. Kila 3/3/2020, 4:18:52 PM

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

It seems a good tutorial, however I stuck on: "check out this tutorial (especially the 'Navigating the Filesystem' and 'Moving Around' sections).".

This link seems to be broken.

[REPLY TO JAN H. KILA](#)

Nayana Chavda 3/6/2020, 1:47:02 AM

Hi,

It is really nice and helpful document.

I want to copy remote repository from github into new repository in github.

How Can I do it using command line?

Thanks,

Nayana

[REPLY TO NAYANA CHAVDA](#)

[Dlight Daily](#) 3/7/2020, 11:21:51 AM

GitHub is a code hosting platform for collaboration and version control.

GitHub lets you (and others) work together on projects.

GitHub essentials are:

Repositories

Branches

Commits

Pull Requests

Git (the version control software GitHub is built on)

Note:A a great way to learn GitHub, befo
in your own repository and merge them y

After merging you can delete the branch

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

more:<https://dlightdaily.blogspot.com/p/what-is-github-explained.html>

[REPLY TO DLIGHT DAILY](#)

[Fei Yao](#) 3/18/2020, 3:09:34 PM

Following Step 10, it might also be recommended to delete local branch by using ``git branch -d yourbranchname`` to keep local repo clear as well.

[REPLY TO FEI YAO](#)

Chandler Dev 3/22/2020, 2:13:15 AM

<https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>

[REPLY TO CHANDLER DEV](#)

First Name*

Last Name

Email*

Website URL

Comment*

We're committed to your privacy. HubSpot uses the information you provide to us to contact you about our relevant content, products, and services. You may unsubscribe from these communications at any time. For more information, check out our [Privacy Policy](#).

protected by reCAPTCHA

[Privacy](#) - [Terms](#)

SUBMIT COMMENT

Join our subscribers ×

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)

[What is HubSpot](#) | [Our Story](#)

[Facebook](#) |

Join our subscribers ✕

Sign up here and we'll keep you in the loop on all things product

[Subscribe me to the blog](#)