

# Forking And Branching in GitHub

Learn how to "fork" a GitHub repository and make changes to it.

## We'll cover the following



- GitHub pull requests
- Fork the repository
- Branch on the forked repository
- Make a change on the branch
- Understand the relationships
- Specify remote branch
- Delete a remote branch
- Make a pull request
- Process to create a pull request
- Pull requests in practice: rebasing

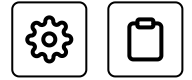
## GitHub pull requests #

For this section, you will be looking specifically at the GitHub model of pull requests. Not all pull requests and relationships between repositories are managed in exactly the same way among different Git servers.

The standard GitHub model is to:

- Fork the repository
- Make a branch on the forked repository
- Make changes on this branch
- Make a pull request to the original repository to merge this branch's changes

Your task now is to do this on the GitHub repository!



There is a file called `records/trained_users.txt` in this repository. You're going to add your name to it and raise that change as a pull request.

Remember that this is just one model of pull request! This lesson won't cover details on the others, but just be aware that they can differ so you are not surprised in the future.

If you haven't created a GitHub account, please do so now. It's free. Go to <https://github.com> (<https://github.com>) and follow the instructions to sign up.

## Fork the repository #

Next, you need to fork the repository. To do this, go to the `learngitthehardway` repository on GitHub:

<https://github.com/ianmiell/learngitthehardway>  
(<https://github.com/ianmiell/learngitthehardway>)

Click on the `Fork` button near the top.

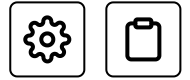
You will now have created a fork of the repository in your own account. Replace `YOURUSERNAME` with your username in the below URL and you should see the same repository homepage:

<https://github.com/YOURUSERNAME/learngitthehardway>  
(<https://github.com/YOURUSERNAME/learngitthehardway>)

## Branch on the forked repository #

To make a branch on your forked repository, type in these commands:

```
1  git clone https://github.com/YOURUSERNAME/learngitthehardway
2  cd learngitthehardway
3  git checkout -b myfirstbranch
4  git status
```



You just cloned your forked version of the repository and created a branch called `myfirstbranch`. As ever, running `git status` gives you a quick view of which branch you're on.

Terminal 1



Terminal



## Make a change on the branch #

Type in these commands to make a change and push it to GitHub:

```
5 echo 'my change to the README' >> README.md
6 git commit -am 'my change to the README'
7 git push -u origin myfirstbranch
```

The first command adds a line to the `README.md` file. The second commits the change you made to this new branch.

## Understand the relationships #

Make sure these relationships are clear in your mind!

- The `ianmiell/shutit` repository was forked to `YOURUSERNAME/shutit`.
- You cloned the repository `YOURUSERNAME/shutit` to a local repository `/path/to/shutit`.
- Within this local repository, you created a branch `myfirstbranch`.
- On this branch, you commit a change with the comment `my change to the README`.

Draw the above in your own diagram to help you.

## Specify remote branch #

Another way to push your branch (and one that may make the relationship clearer) is the following:



```
8 echo 'another change to the README' >> README.md
9 git commit -am 'another change to the README'
10 git push -u origin myfirstbranch:myfirstbranch
```

**Note:** Kindly enter your GitHub username to automate some previous commands that are dependent on it.

This code requires the following environment variables to execute:

github\_username      Not Specified...

Edit

Terminal 1



Terminal

What's changed here is that you have added

`myfirstgitbranch:myfirstgitbranch` to the branch argument of the command.

This indicates that the local branch `myfirstgitbranch` should be pushed to the remote branch `myfirstgitbranch`. The colon separates the two branch names. The first is the `local` one and the second is the `remote` one.

In this case, the branch names are the same (`myfirstgitbranch`) but this need not be the case. By default, Git assumes you want to match the names on the local and the remote repository, but it's useful to get into the habit of typing the full specification with the colon because there are times when it's useful to know that this mapping is possible.

To take advantage of the defaults, the last command could have been

To take advantage of the defaults, the last command could have been written like this:



```
git push -u origin myfirstbranch:
```

## Delete a remote branch #

You can use a similar trick to delete a remote branch.

To practice this, create a `tmpbranch` on your local and remote repository.

```
11 git branch tmpbranch
12 git checkout tmpbranch
13 echo 'a temp change on tmpbranch' >> README.md
14 git commit -am 'a temp change on tmpbranch'
15 git push origin tmpbranch:tmpbranch
```

This code requires the following environment variables to execute:



github\_username Not Specified...

Edit

Terminal 1



Terminal

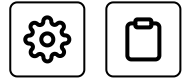


Now that you've created the `tmpbranch` on the remote repository, you might decide you've been too hasty and that "tmpbranch" is not needed on the remote.

To delete it on the remote, specify nothing before the colon like this:

```
16 git push origin :tmpbranch
```

This has the effect of removing the branch on the remote repository. If you look, it's still there on your local repository, so nothing has been lost.



Quite often, projects on GitHub can accumulate a lot of branches, and this method can be a handy and quick way to tidy up these branches.

## Make a pull request #

Now that you have a branch on the forked repository on GitHub, you want to get that branch's changes into the maintainer's repository. This is where you raise the pull request.

Go to GitHub in a browser and view your repository:

<https://github.com/YOURUSERNAME/learngitthehardway>

(<https://github.com/YOURUSERNAME/learngitthehardway>)

The instructions for creating a pull request are here:

<https://help.github.com/articles/creating-a-pull-request/>

(<https://help.github.com/articles/creating-a-pull-request/>)

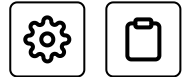
## Process to create a pull request #

I won't repeat it here because the workflow can change. But in essence, the general process is to:

- Go to your branch.
- Generate a new pull request.
- Fill out the form.
- Wait.
- Celebrate your PR's acceptance into the code, or chase the maintainer (nicely!) for an update.

You can create a pull request *across forks* (a request to the upstream (original) repository) or against another branch in a GitHub repository. “*Across forks*” is what's most commonly meant by a public GitHub PR, which is a request to accept a change made to a repository under your control to a repository under someone else's (usually more *senior* to the

project).



## Pull requests in practice: rebasing #

Maintainers will often ask that you rebase your branch to the principal branch (usually `master`) before making a pull request. You will remember the `git rebase`. If you don't remember, you might want to go back and read over it again.

Maintainers will want you to rebase so that the work of merging any changes made, since you forked from the origin, is done by you who is the submitter rather than them. This also makes the history of the main line easier.

If you didn't understand the above paragraph, then definitely work through the rebase chapter

(<https://www.educative.io/collection/page/10370001/6075350136651776/6228325110906880>) again!

The goal is that all the messy work is done on your secondary branch (which in Git is a more disposable thing) and the good stuff makes its way into the main line. Many projects will delete branches once they have served their purpose, and Git supports this.

```
17 git branch -d mybranch
```

It will even warn you if the branch has not been merged into the branch you are currently on!



```
18 git branch -d abranch
```


### Output:

```
error: The branch 'abranh' is not fully merged.  
If you are sure you want to delete it, run 'git branch -D abranch'  
.
```

 **Back**


What Are Pull Requests?




**Next** 

Make a Pull Request (delete)

☒ Mark as Completed

 Report an Issue

 Ask a Question  
([https://discuss.educative.io/tag/forking-and-branching-in-github\\_\\_pull-requests\\_\\_learn-git-the-hard-way](https://discuss.educative.io/tag/forking-and-branching-in-github__pull-requests__learn-git-the-hard-way))