Configure Git

Associate local changes with your name + email

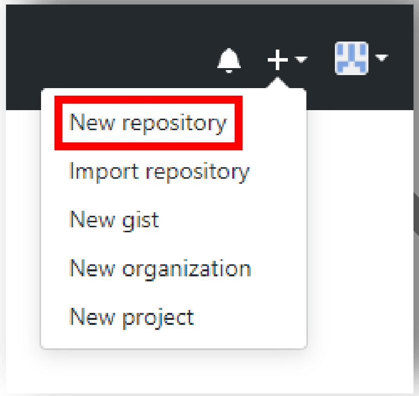
```
$ git config --global user.name “<first_name> <last_name>”
$ git config --global user.email “<email_address>”
```

=

Ryan Petersburg
ryan.petersburg@yale.edu

Clone a Repository

Download a Github repository to your current working directory

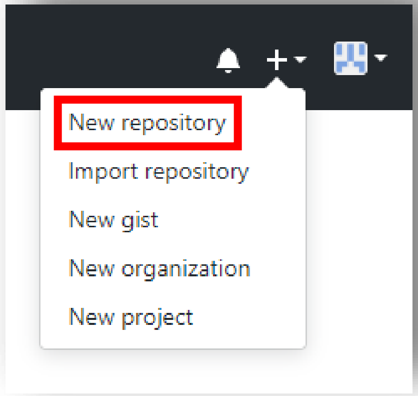


```
$ git clone https://github.com/<username>/<project_name>.git
```

```
$ cd new-repo
$ ls -a
.     ..    .git  README.md
$
```

Initialize a Repository

Create a new repository in your local directory.



You will need to create a Github repository separately if you want to push changes to it later!

```
$ mkdir new-repo
$ cd new-repo
$ git init
$ ls -a
.     ..    .git
$ git remote add origin https://github.com/<etc...>
```

Basic Git Workflow

Edit your code and save changes along the way!

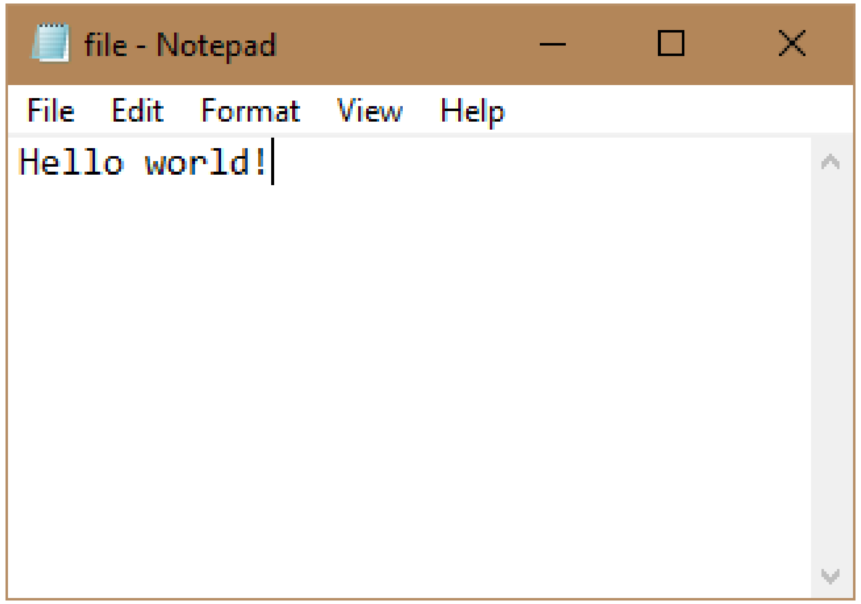
local repository

```
$ git commit -m “first commit”
```

staging area

```
$ git add file.txt
```

workspace



```
Hello world!
```

rinse, wash, repeat!!!

After adding a file for the first time, you can combine these two steps using

```
$ git commit -am “all
other commits”
```

Useful Local Commands

If you're interested

```
$ git status
```

See which files have been changed, staged, removed, etc…

```
$ git diff --staged
```

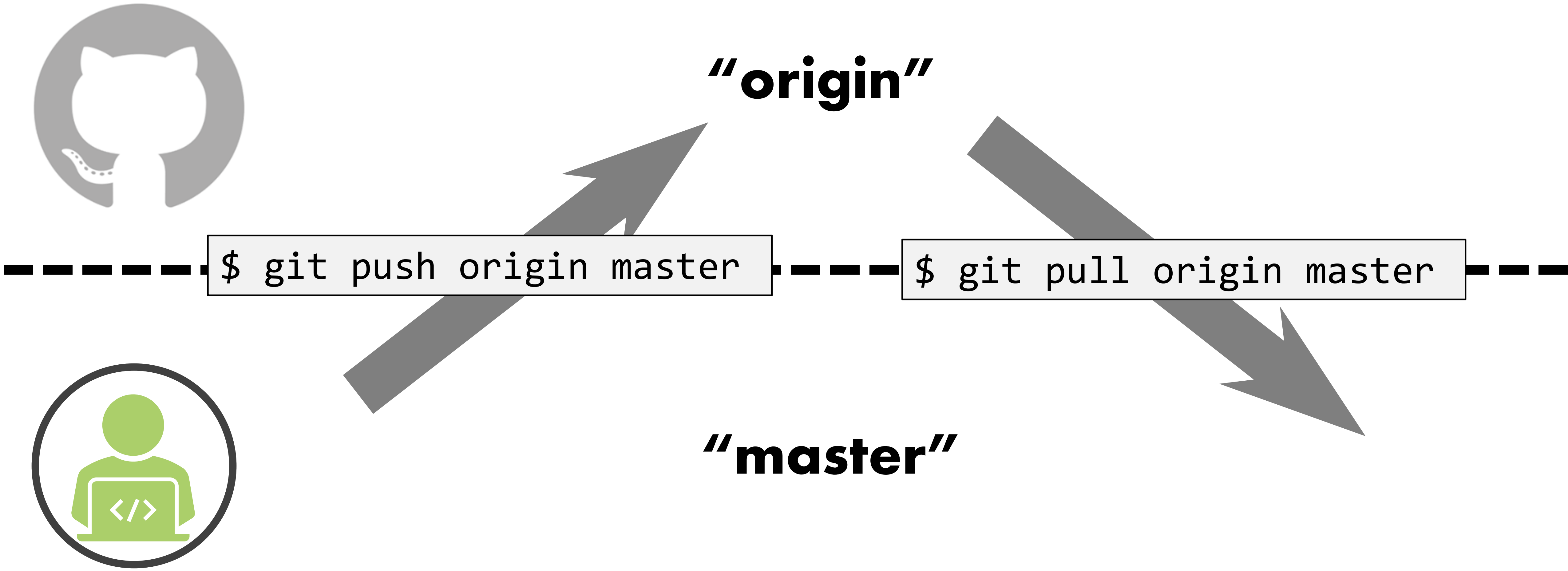
Check how staged files have been modified compared to previous commit

```
$ git log
```

See a history of all recent commits

Interfacing with Github

Push changes from your local repository to Github and pull changes from Github locally

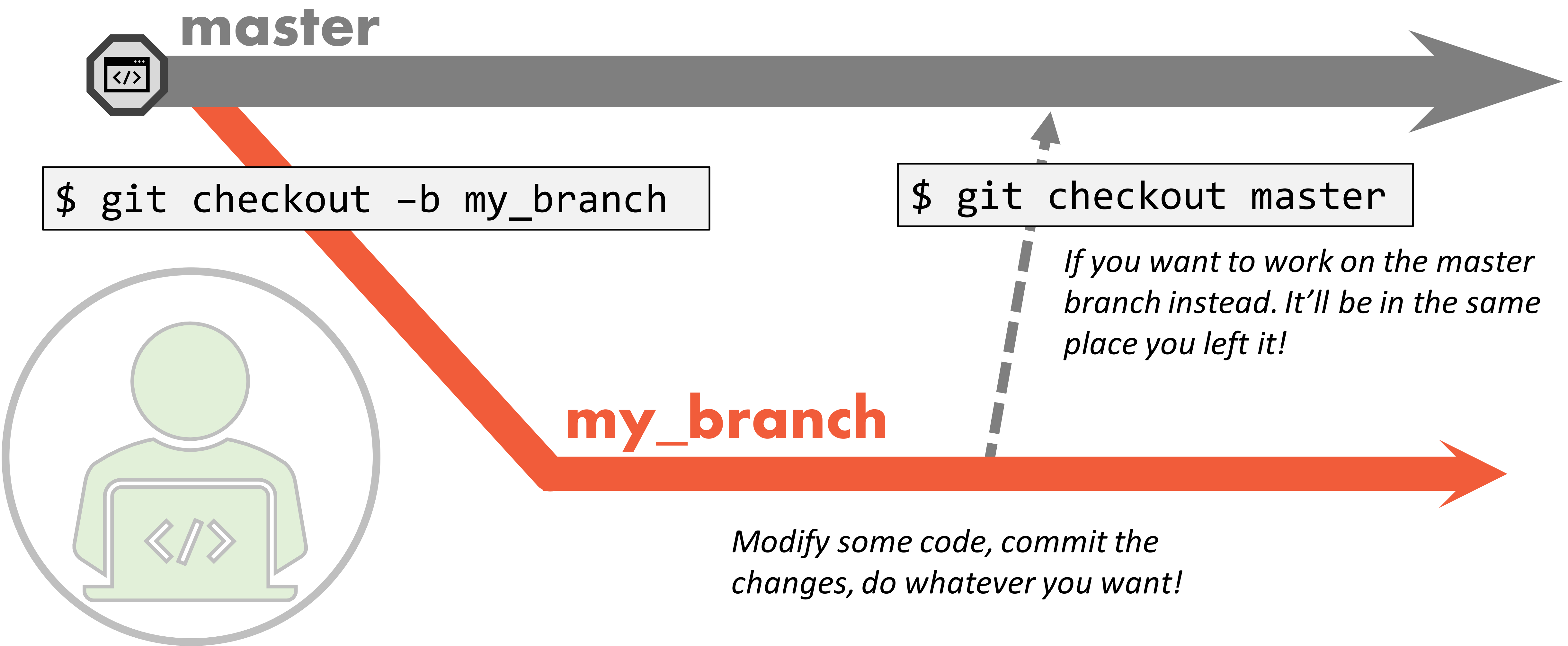


```
$ git push origin master
```

```
$ git pull origin master
```

Branching

Change and test your code without modifying a working version (or collaborate with others!)



```
$ git checkout -b my_branch
```

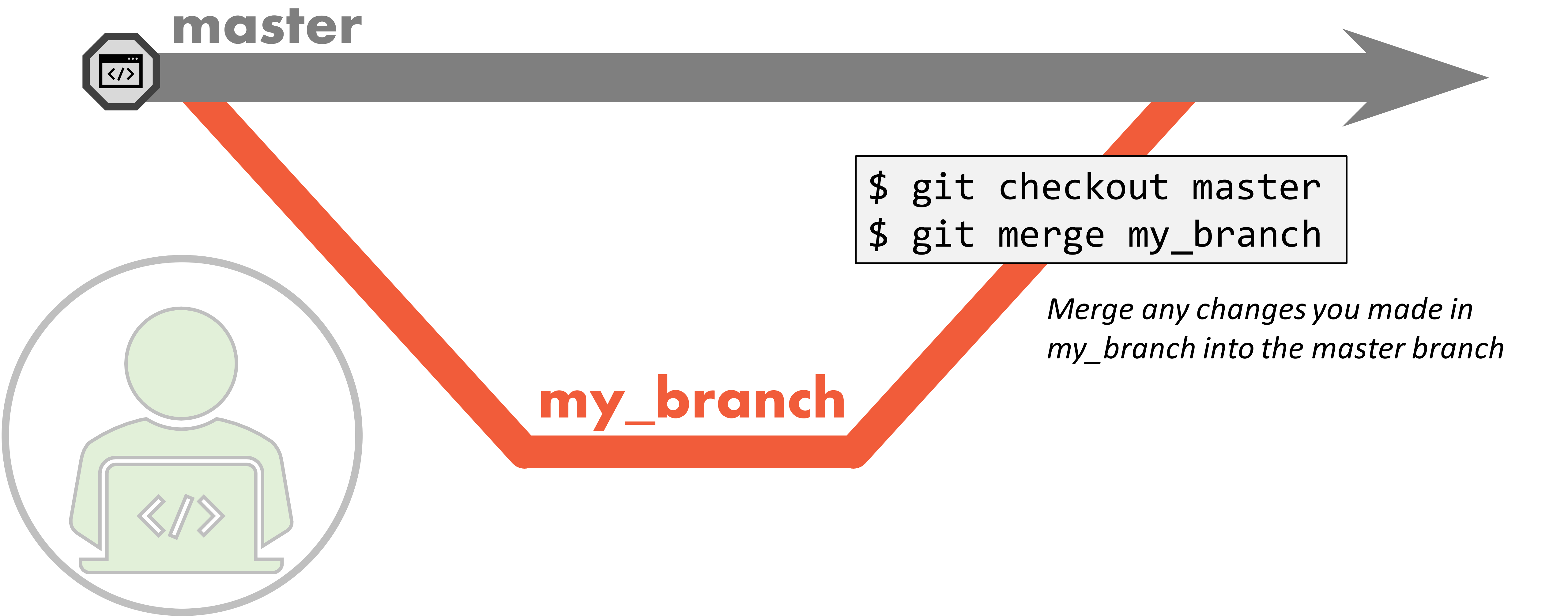
```
$ git checkout master
```

If you want to work on the master branch instead. It'll be in the same place you left it!

Modify some code, commit the changes, do whatever you want!

Branching

Change and test your code without modifying a working version (or collaborate with others!)

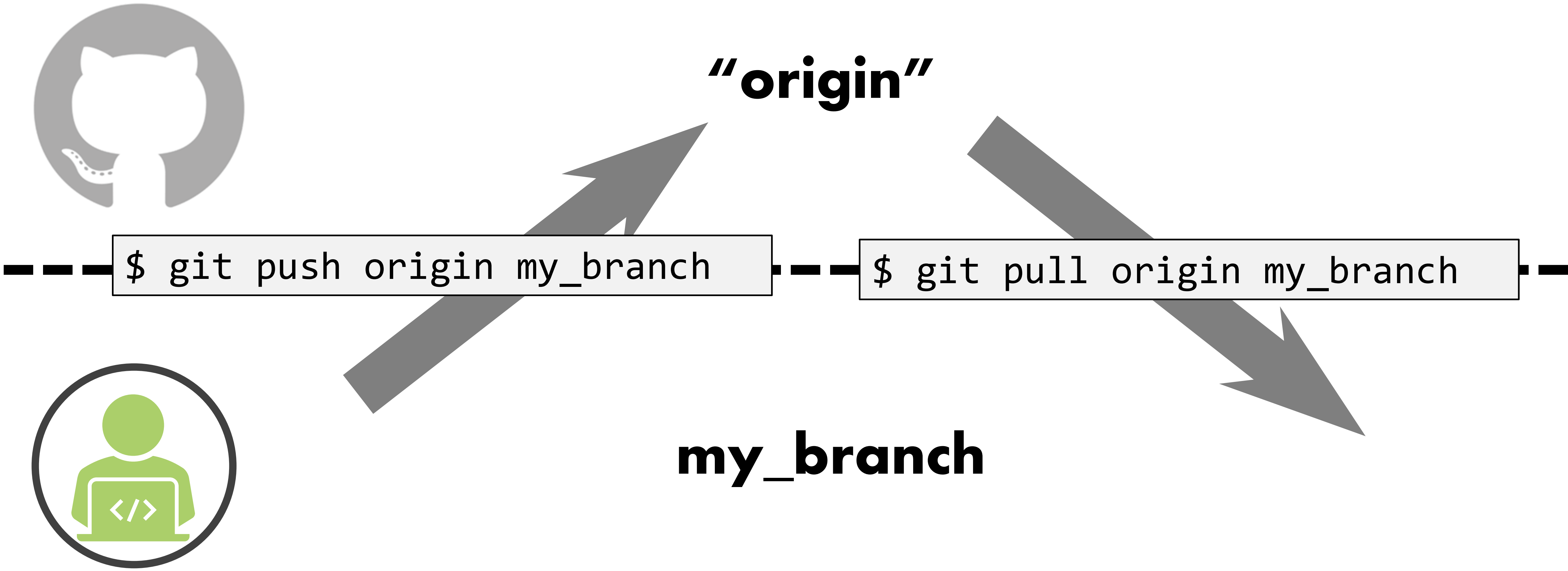


```
$ git checkout master
$ git merge my_branch
```

Merge any changes you made in my_branch into the master branch

Interfacing with Github

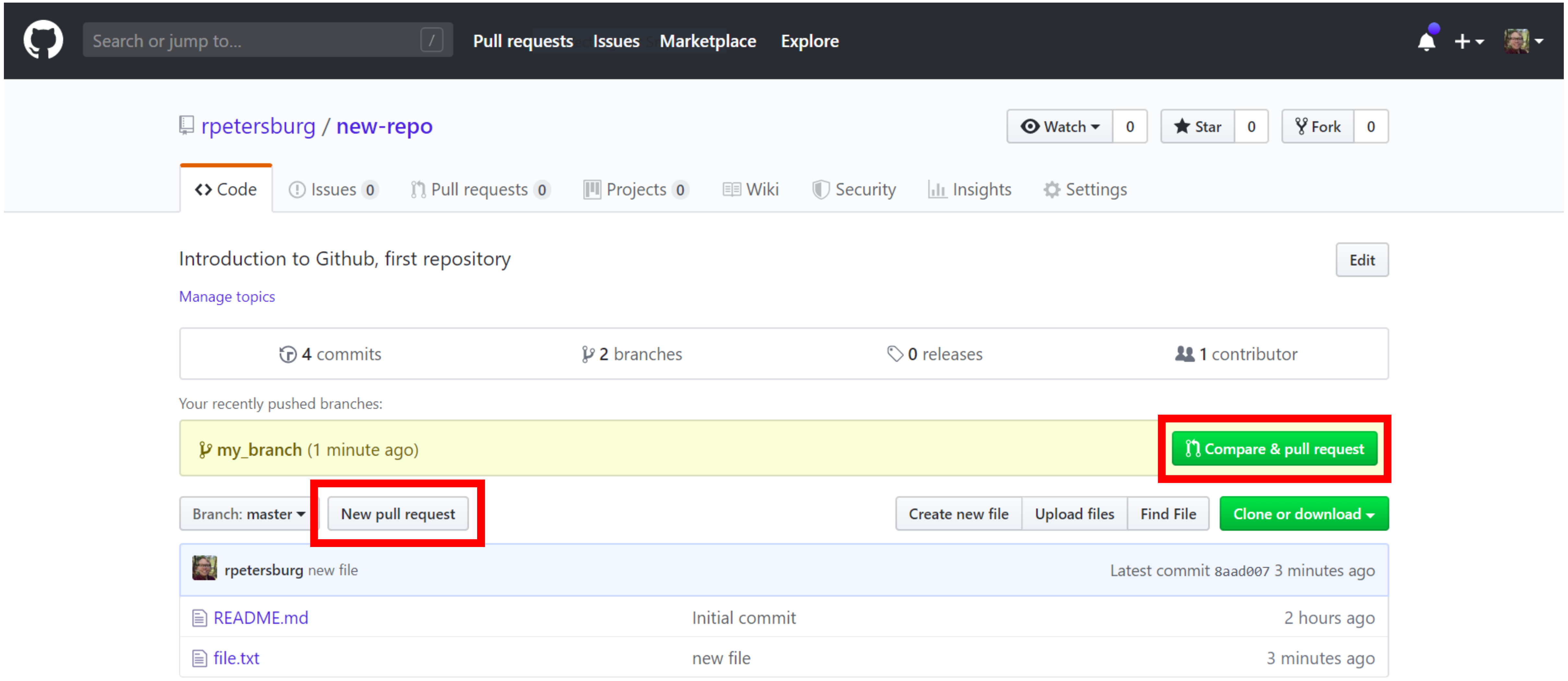
Push changes from your local repository to Github and pull changes from Github locally



```
$ git push origin my_branch
```

```
$ git pull origin my_branch
```

Pull Requests



Introduction to Github, first repository

Your recently pushed branches:

my_branch (1 minute ago)

rpetersburg new file — Latest commit 8aad007 3 minutes ago

README.md	Initial commit	2 hours ago
file.txt	new file	3 minutes ago

Summary

Download the Github repository to you local directory

```
$ git clone https://github.com/<username>/<project_name>.git
```

Edit files, stage them, and then commit (save) the changes

```
$ git add <file_name>
$ git commit –m “<commit_description>”
```

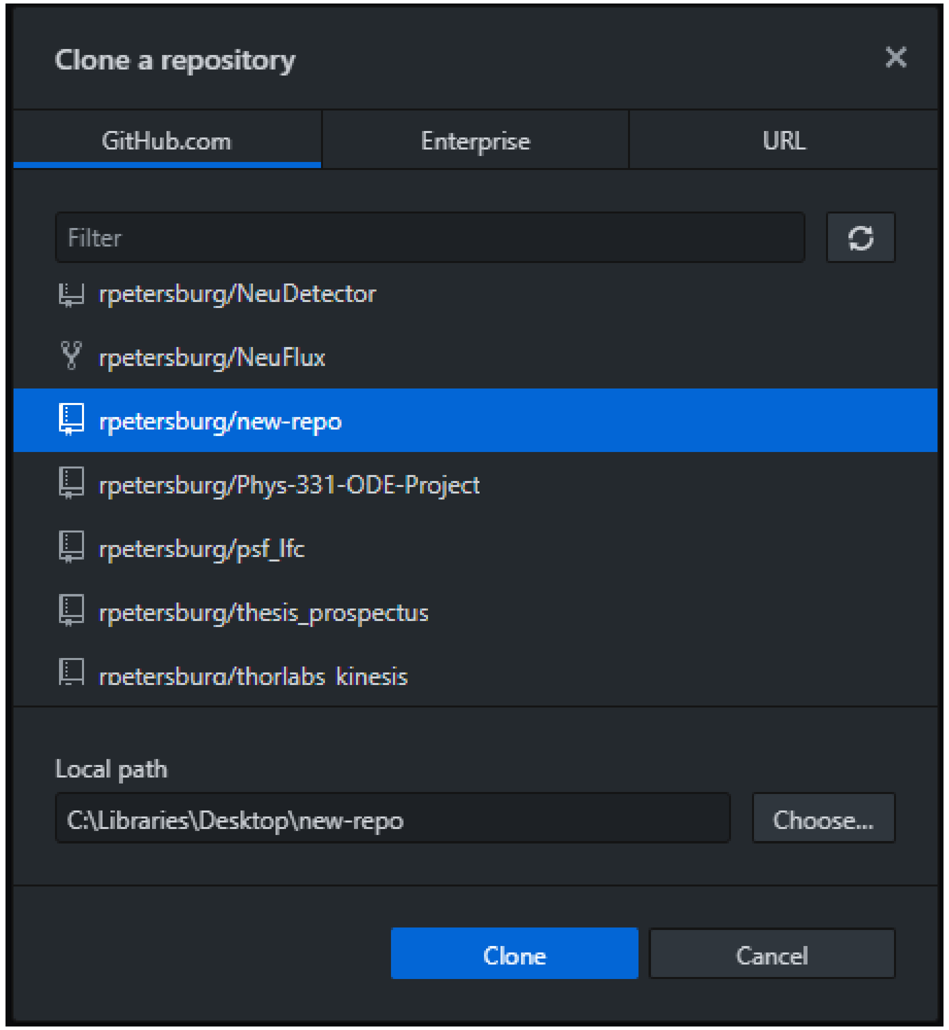
Create a new branch, edit some files, then merge the changes back into the master branch

```
$ git checkout –b <branch_name>
$ git checkout master
$ git merge <branch_name>
```

Push/pull changes to/from the Github repository

```
$ git push origin <branch_name>
$ git pull origin <branch_name>
```

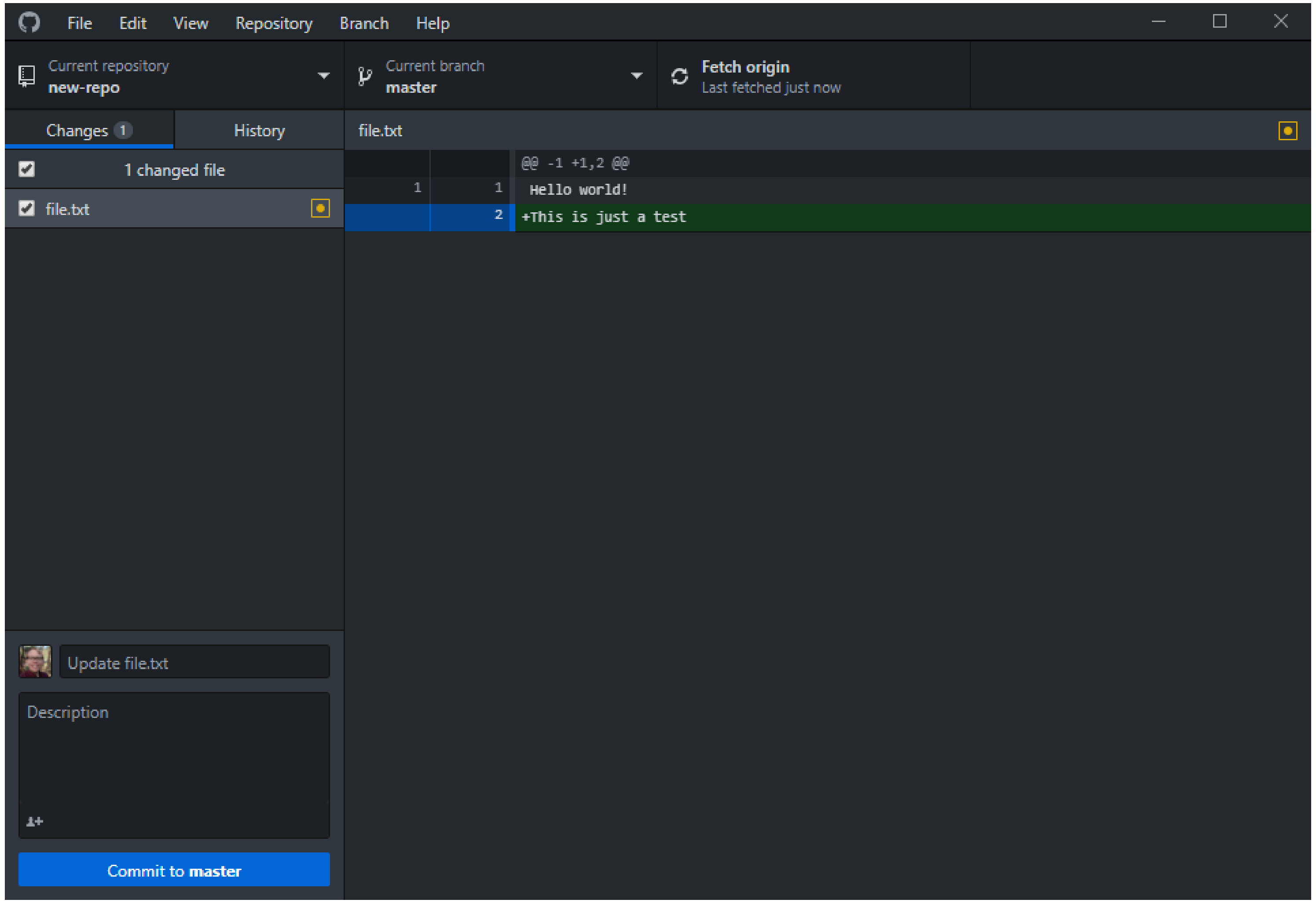
Github Desktop



Clone a repository

rpetersburg/NeuDetector
rpetersburg/NeuFlux
rpetersburg/new-repo
rpetersburg/Phys-331-ODE-Project
rpetersburg/psf_lfc
rpetersburg/thesis_prospectus
rpetersburg/thorlabs_kinesis

Local path
C:\Libraries\Desktop\new-repo



Current repository
new-repo
Current branch
master
Fetch origin
Last fetched just now

1 changed file
file.txt

```
@@ -1 +1,2 @@
1   1   Hello world!
    2  +This is just a test
```

Update file.txt
Description

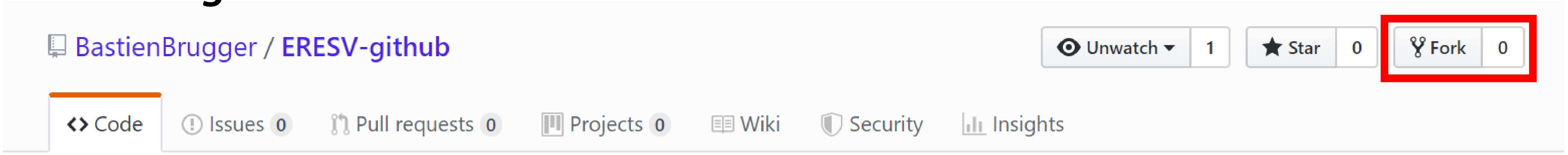
Extra Topics

Rebasing

```
$ git rebase master
```

Add the commits from master that occurred since you created your branch. This is DISTINCT from "merging", but performs a similar function.

Forking



Copies a public repository into a personal repository. The original repository is called the "upstream" while your forked repository is still called "origin". Any modifications you make to "origin" will not affect "upstream", but you can submit a "pull request" if you would like to contribute your updates!