Configure Git

Associate local changes with your name + email

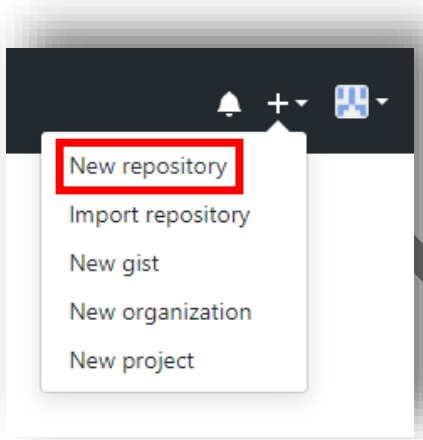
```
$ git config --global user.name “<first_name> <last_name>”
$ git config --global user.email “<email_address>”
```

=

Ryan Petersburg
ryan.petersburg@yale.edu

Clone a Repository

Download a Github repository to your current working directory

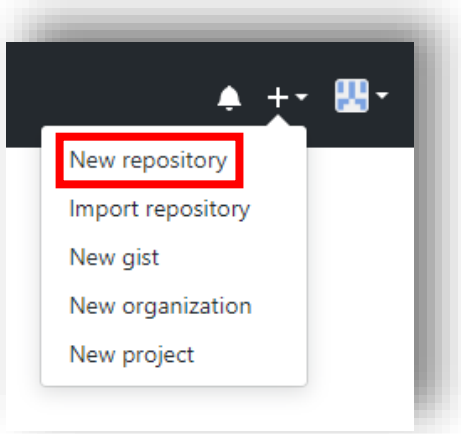



```
$ git clone https://github.com/<username>/<project_name>.git
```

```
$ 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!

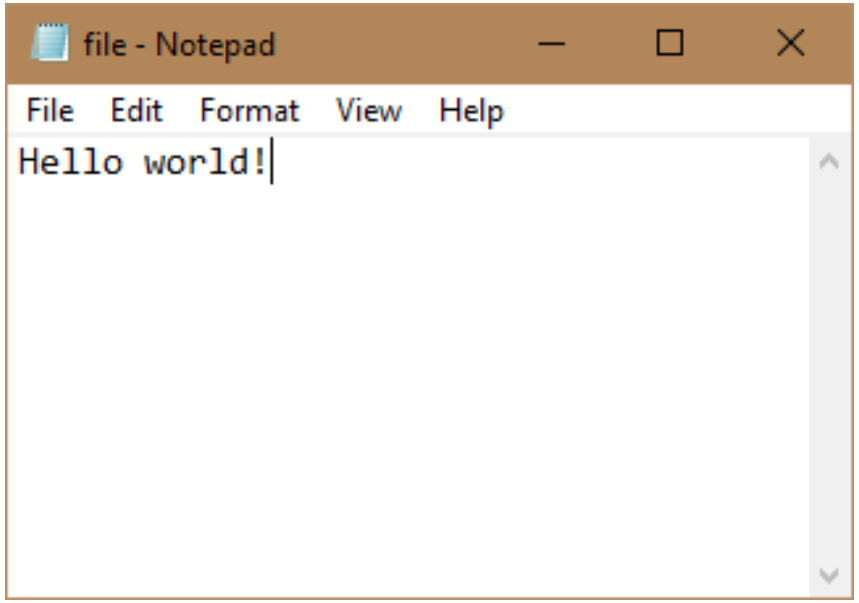
local repository

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

staging area

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

workspace



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
```

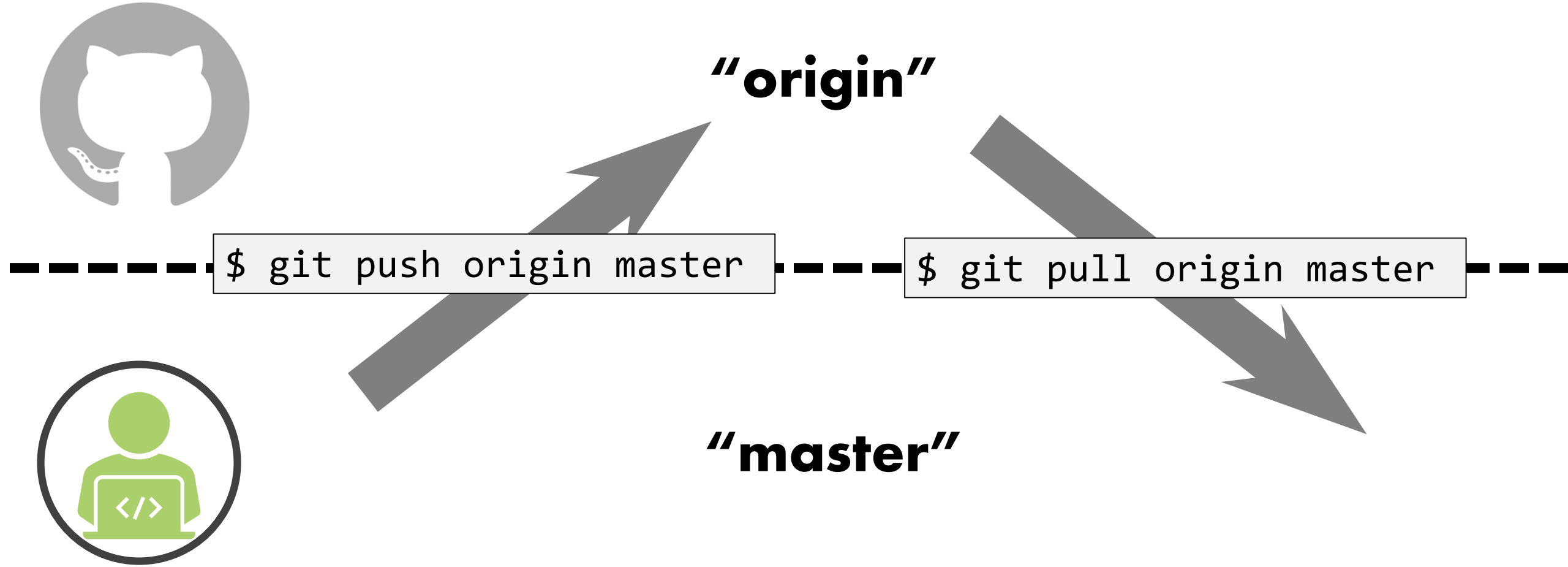
Check how staged files have been modified compared to previous commit

```
$ git log
```

See a history of all recent commits

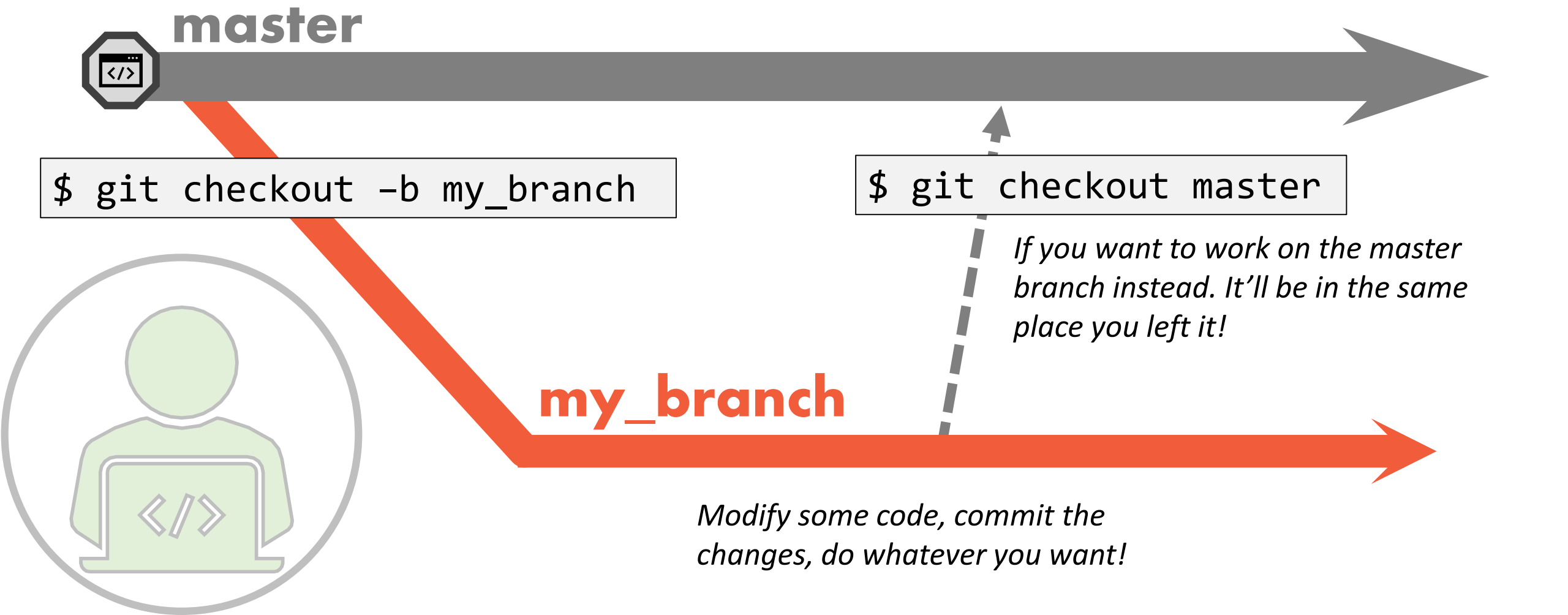
Interfacing with Github

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



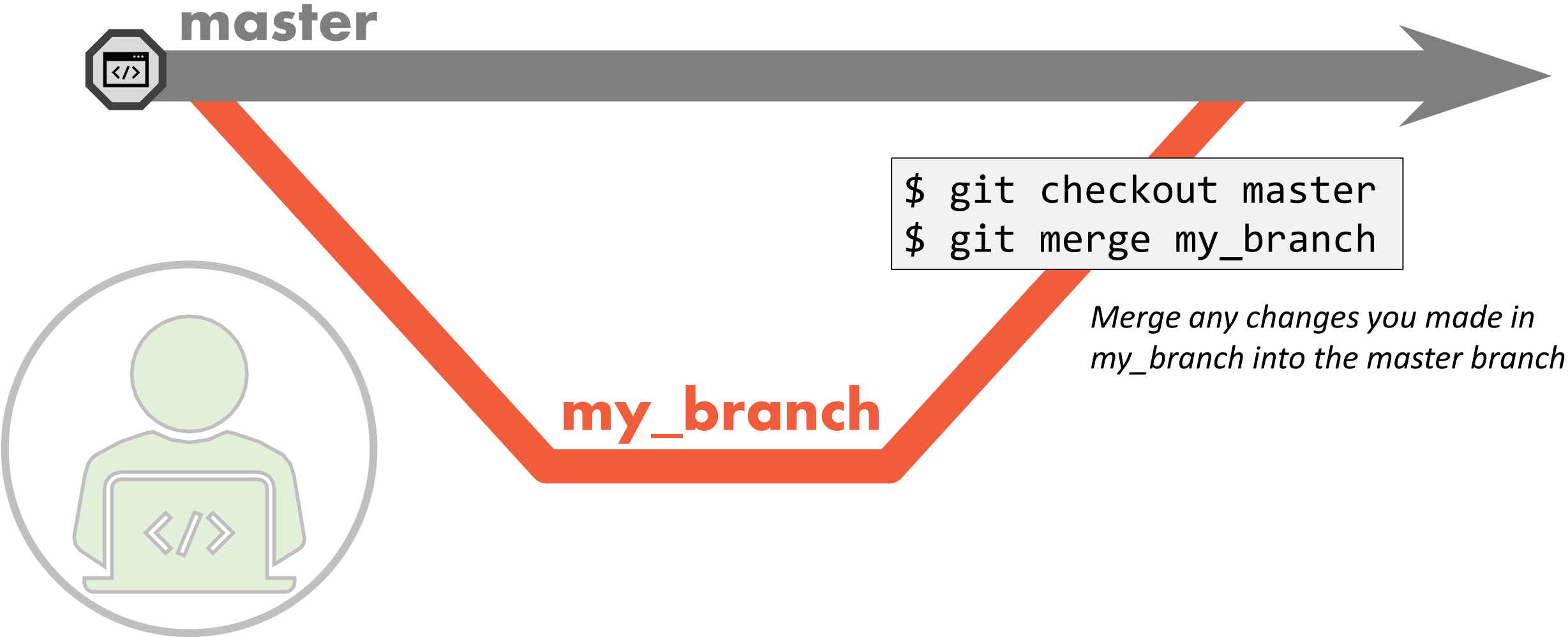
Branching

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



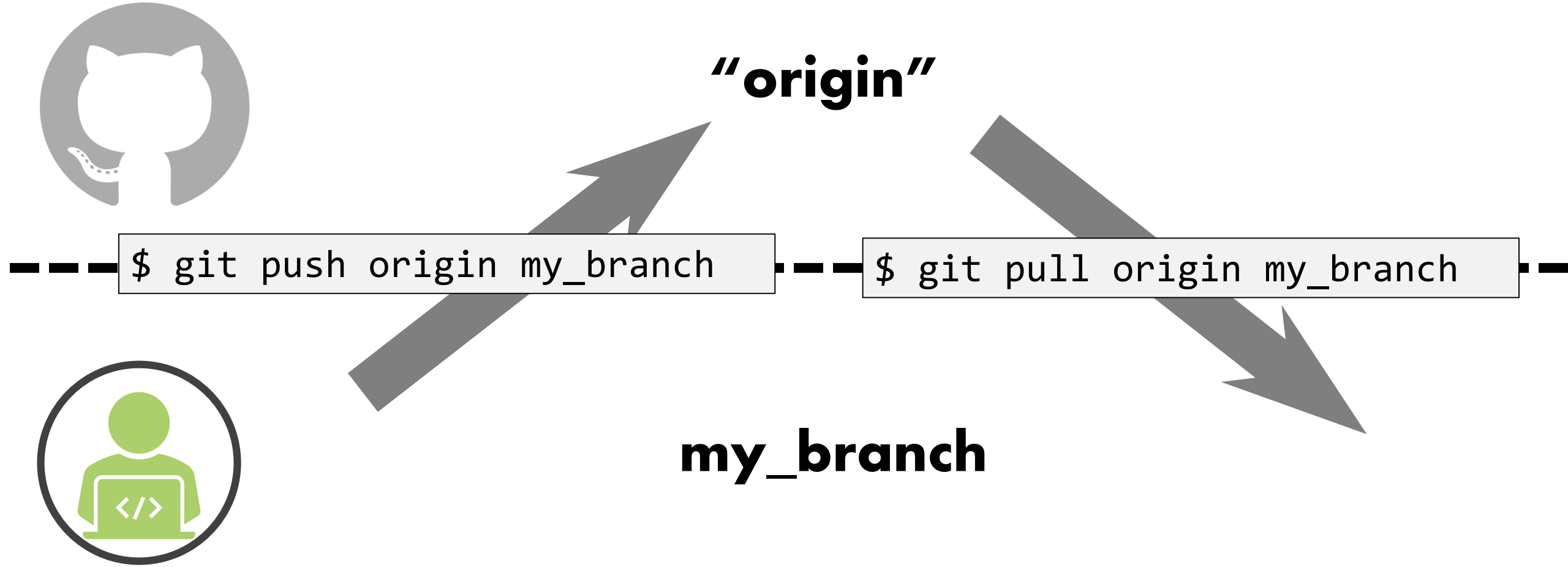
Branching

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

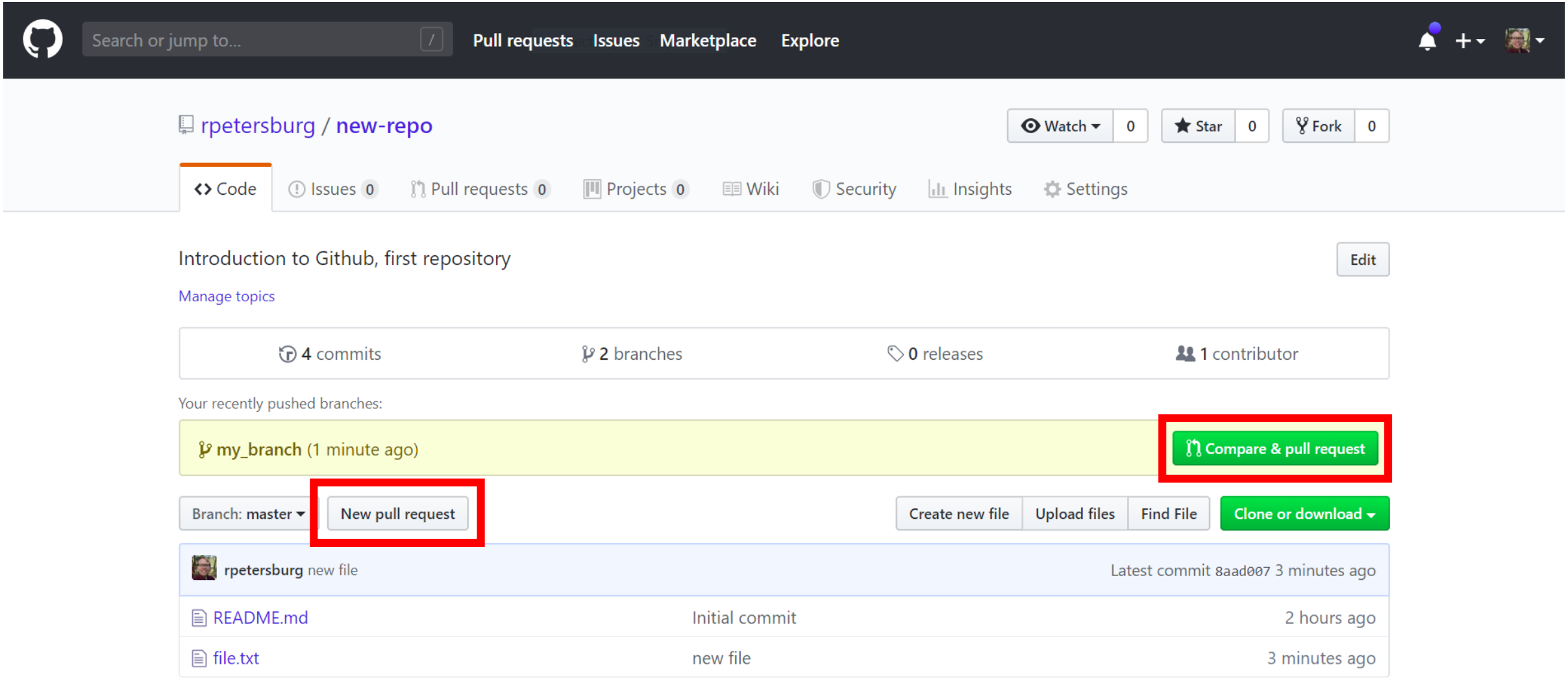


Interfacing with Github

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



Pull Requests



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