

Getting Started with GitHub

1. Configuring Git

- 1.1. Tell git who you are (replace “Mona Lisa” with your own name):

```
git config --global user.name "Mona Lisa"
```

- 1.2. Tell git what your e-mail address is (replace e-mail address with your own):

```
git config --global user.email "mlisa@berkeley.edu"
```

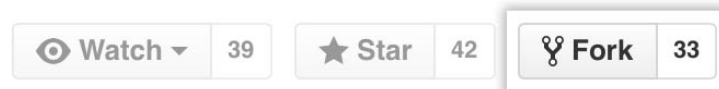
- 1.3. Double check that git got all that right

```
git config --list
```

2. Forking the class repo

- 2.1. Go to <https://github.com/mxndrwgrdnr/CYPLAN255>

- 2.2. On the top right of the page, click the button labeled “Fork”



- 2.3. Now, instead of seeing this at the top left of your screen

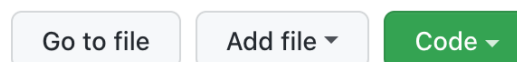


you should see something like this:

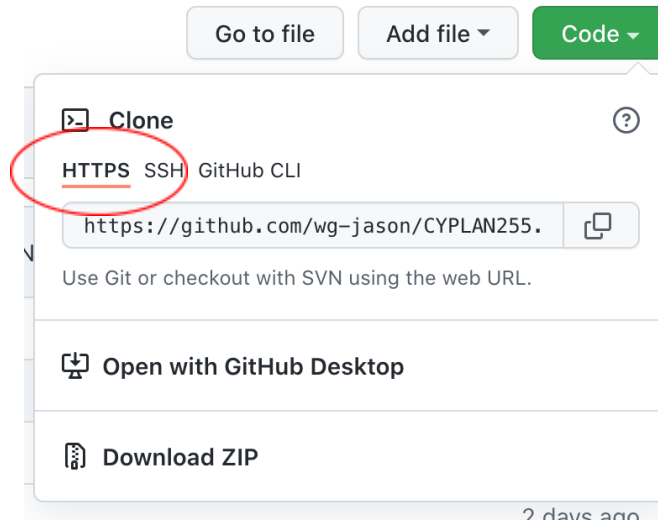


3. Cloning your fork

- 3.1. From your fork, click the green button labeled “Code”



- 3.2. In the drop-down menu, make sure you have HTTPS selected



- 3.3. Click the copy button to copy the URL to your clipboard:



- 3.4. Open up a terminal/bash shell/command prompt and type:

```
pwd
```

This is the location on your local computer to which you are about to copy the repo. Take note. If this is not where you want the repo to be stored on your computer, either `cd` to the right location before continuing, or you can simply move the project (repo) directory afterwards.

- 3.5. Now in the same terminal type

```
git clone <paste the URL you copied here>
```

and hit `<Enter>`.

- 3.6. To check that you did everything correctly, now do

```
git remote -v
```

and you should see something that looks like this:

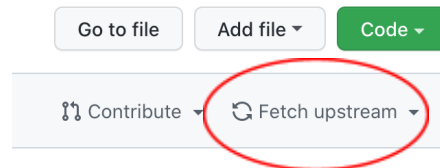
```
origin    <URL of your fork.git> (fetch)
origin    <URL of your fork.git> (push)
```

If instead of the URL of *your* fork, you see the URL of the main repo ("https://github.com/mxndrwgrdnr/CYPLAN255.git"), then you have cloned the

main repo rather than your fork. Delete the folder you're in, return to step 3.1, and try again with the right URL.

4. Syncing your fork

- 4.1. Navigate to your GitHub fork in a browser and click the button labeled “Fetch upstream” under the green “Code” button



and then click “fetch and merge” when prompted. Do this as often as you like.

- 4.2. Now that your *remote* is synced, you need to copy those changes to your *local* copy. To do that, all you need to do is

```
git pull
```

5. Submitting assignments

- 5.1. As mentioned in the assignment README, assignments should be submitted on the “assignments” branch. To change branches do

```
git checkout assignments
```

- 5.1.1. If you get an error telling you that no such file or branch exists, that's because you created and cloned your fork before the assignments branch was created on the *upstream* repo. That means your fork is behind or “stale”, and needs to be updated. Go back to Step 4 and try fetching from upstream again.
- 5.1.1.1. If you do that and still see the same error, then your best bet is to fetch the upstream changes from the command line. This is the shortcut I mentioned in lecture and illustrated on slide 18.

Do the following steps in order, copying the commands exactly:

- ```
git remote add upstream
https://github.com/mxndrwgrdnr/CYPLAN255.git
```
- ```
git fetch upstream
```
- ```
git checkout assignments
```

Continue on to step 5.1.2

- 5.1.2. If you get an error that says:

```
fatal: 'assignments' could be both a local file and a tracking branch.
Please use -- (and optionally --no-guess) to disambiguate
```

then try doing exactly what Git suggests:

```
git checkout assignments --
```

I probably shouldn't have created a branch with the same name as a directory in the repo 🙄.

To confirm that you are now on the right branch, do

```
git branch
```

- 5.2. **Remember, do not do your work in a file with the same name as a file that you find in the repo.** You may use existing files as a template by creating a copy, renaming that copy with a unique identifier (first and last name, or last 4 digits of your student ID). For example:

```
cp assignment_0.txt assignment_0_max_gardner.txt
```

Or just create a brand new file and do your work there.

- 5.3. When you are ready to push your changes from local to remote, you must first tell Git which files contain the changes you want to push:

```
git add assignments_0_max_gardner.txt
```

- 5.4. Then you are ready to *commit* those changes. No going back now!

```
git commit -m "mandatory descriptive message goes here"
```

- 5.5. OK so we're about to push the changes, but we have a few options for how to do that. The easiest, foolproof way to do it is to be as explicit as possible by telling Git exactly where you want to push to. by specifying the name of the remote and the branch on the remote where you want your commits to go:

```
git push origin assignments
```

This is useful because sometimes your local copy might be tracking multiple remotes. In fact, if you followed step 5.1.1.1 that's exactly what you did (try `git remote -v` and you'll see what I mean). But if your Git project has multiple remotes to choose from, it might not know which one you want to push to, and it might actually push to the wrong one by mistake if you're not careful. The other way to avoid pushing to the wrong remote is to *set the default*:

```
git branch --set-upstream-to origin/assignments
```

and now you can simply do

```
git push
```

and Git will know exactly where to push to.

- 5.6. At this point, if you cloned your repo with HTTPS, Git should prompt you to enter your github username and password in the terminal. Go ahead and do that.

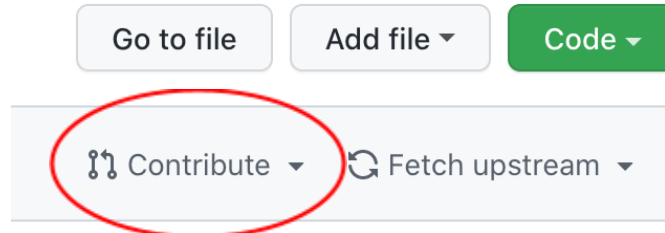
- 5.6.1. If you find it cumbersome to have to enter your username and password every time you want to make a commit, there are other authentication options that you can try, although they can be a bit trickier to set up. See here for more details:

<https://docs.github.com/en/get-started/getting-started-with-git/why-is-git-asking-for-my-password>.

- 5.6.2. If instead of asking for your username and password, Git asks you for a token or a key or something else, you probably did not clone your fork via HTTPS. Your safest bet would be to delete your project directory on your computer, and go back to step 3.1. If you've done that a few times and are still having the same issue, please reach out to Max or Irene.

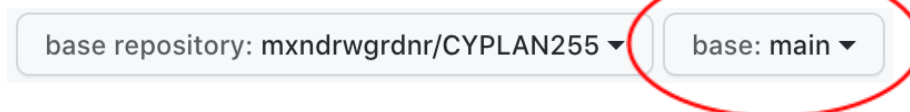
- 5.7. Once you've pushed, go back to the browser and look at your fork. You should see your commit message at the top of the page along with a note that says changes were pushed "a few seconds ago".

- 5.8. When you're ready, click the button labeled "Contribute", just to the left of the "Fetch upstream" button you used to sync your fork:



and click "Open Pull Request" when prompted.

- 5.9. From the Pull Request page, click on the "base: main" button



and change it to "base: assignments".

- 5.10. Then do the same thing for the "head" repository (your fork), by clicking on the "compare: main" button, and selecting the "assignments" branch. You should now see something like this:



- 5.11. When you're ready, click the green "Create pull request" button:



Create pull request

- 5.12. Lastly, give your pull request a descriptive title, and a helpful comment or two. The title does NOT need to be unique. Then click the green button once more and give yourself a pat on the back :)