Get Your GitHub Classroom Repo

Stat 133, Fall 2019

This document contains the instructions to get your **private GitHub Classroom repository**. This is the repo that you will use to submit your Workout assignments. Submissions uploaded to a different repository will be ignored (no exceptions).

GitHub Classroom

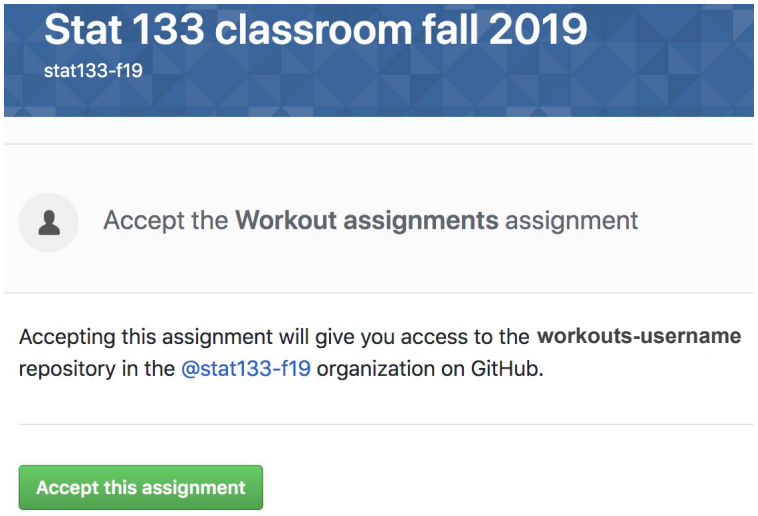
0) We are assuming that you already have a github account.

1) Sign-in to your github account.

2) Use the following invitation link to get your **GitHub Classroom** repository

https://classroom.github.com/a/_EvQwpkt

3) You should get redirected to `https://classroom.github.com`, more specifically, to the "Stat 133 classroom fall 2019" organization. Here, you will see a message *"Accept the Workout assignments"* assignment.

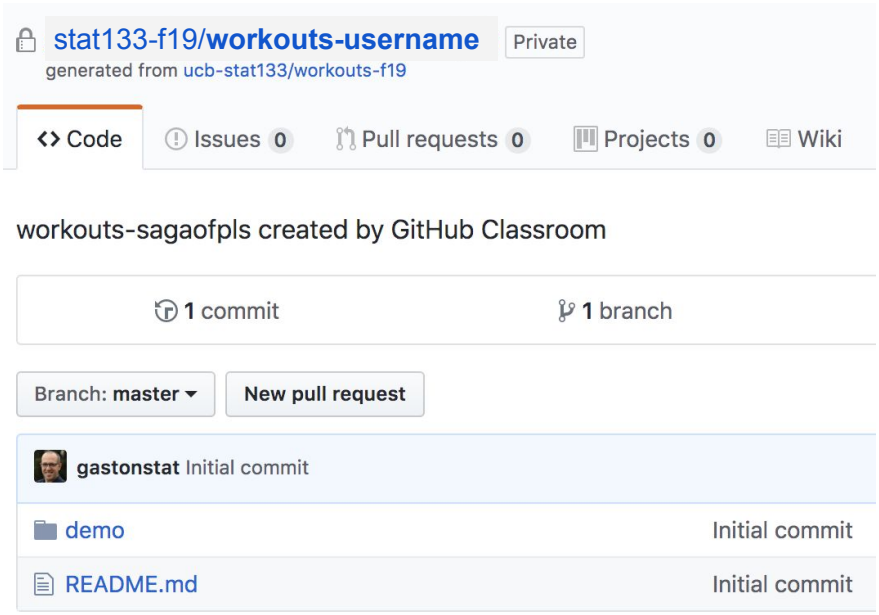


4) After clicking on the **Accept this assignment** button, you should be able to see a new message indicating that your assignment repository is being setup: this involves 1) creating the repository, and 2) importing starter code. When the setup is finished, you should see the message *You are ready to go!*

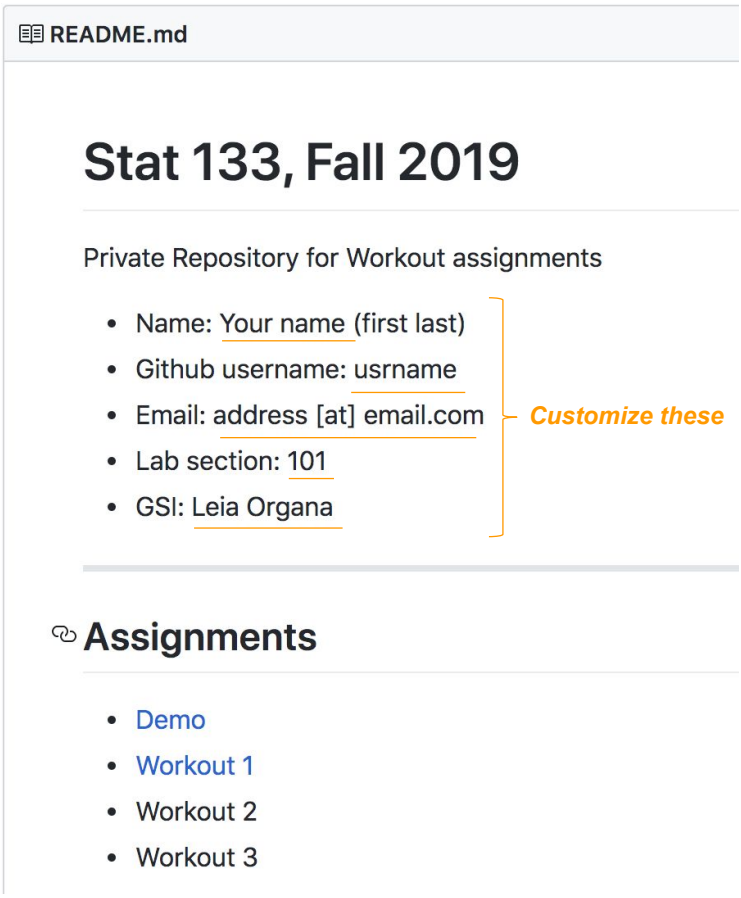
5) At this moment, you will probably receive an email with an invitation to join `@stat133-f19`. In theory, there is no need to accept the invitation since your repository has already been created.

Your Private GitHub Classroom Repository

What you need to do next is to click on the assignment link created for your github account. You should be able to see your new repository (with your own username).



As you can tell, your repo contains a `README.md` file with some default content:



Edit the `README.md` file

Click on the `README.md` file, and then look for the icon of a pencil so that you can edit the contents of the file.

Customize the information about `Name`, `Github username`, your `Email` address (the one linked to your github account), the `Lab section` you are registered in, and the name of your `GSI`

Look for the button **Commit changes** so you can "save" the modifications done to the `README.md` file.

Your Local Repo and the Remote Repo

So far you have your GitHub Classroom repository. This repository will be your **remote** repository. However, you also need to to create a **local** repository (i.e. in your computer), inside the directory that you use for Stat 133.

- Open a bash terminal

- Change directory to your dedicated directory for Stat 133
- In theory, your Stat 133 directory should look more or less like this (assume the directory is named `stat133`)

```
stat133/
   labs/
      lab01/
      lab02/
      lab03/
      lab04/
   warmups/
      warmup01/
      warmup02/
      warmup03/
      warmup04/
```

Create a new subdirectory `workouts/` inside `stat133`. Your filestructure should now look like this:

```
# assuming your working directory is stat133/
mkdir workouts
```

```
stat133/
   labs/
      lab01/
      lab02/
      lab03/
      lab04/
   warmups/
      warmup01/
      warmup02/
      warmup03/
      warmup04/
   workouts/
```

Move inside the recently created `workouts/`:

```
cd workouts
```

- Initialize the directory as a Git repository

```
git init
```

- Locate the name of your github repo (with **your own username!!**)

```
https://github.com/stat133-f19/workouts-username
```

- To add a remote repository use the command below with **your own username**:

```
git remote add origin https://github.com/stat133-f19/workouts-username
```

- Pull down the content in the remote repo (`origin`) to your local repo (`master`)

```
git pull origin master
```

Pushing changes to the remote repo

Now that you have linked your local repo with your remote repo, you can start pushing (i.e. uploading) commits to GitHub. Try the following modifications.

In your computer (your local repo), use a text editor (e.g. the editor in RStudio) to open the `README.md` file that is inside the `demo/` folder. Under the title **Demo**, you should be able to see a first line of text:

```
This is just a demo folder with some dummy content.
```

Delete the starting paragraph and replace it with the text below (or any other text that you want to add):

```
This is just a demo folder for testing purposes.
```

Now follow these steps:

- Save the changes made in `README.md`.
- Go to the your bash terminal.
- Change directory to the `demo/` folder:

```
cd demo
```

- Check the status

```
git status
```

- Add the changes to git:

```
git add README.md
```

- Commit the changes with a descriptive message:

```
git commit -m "demo: update readme file"
cd ..
```

- Push the changes to the remote repo (`origin`) from your local repo (`master`)

```
git push origin master
```

Go to your Github repository and refresh the browser. If everything went fine, you should be able to see the updated contents of the `README.md` file inside `demo/`.

Push your Workout1 assignment

In this part, we are assuming that you've already made some progress in your workout-1 assignment.

So let's add some of the workout-1 files to the `workout1/` folder.

- Copy the global `README.md` file inside the folder `workout1/` (of your local repo).
- Check the status of your repo with `git status`.

```
git status
```

- Use `git add` to add the current modifications.

```
git add README.md
```

- And then commit the changes, including a meaningful commit message:

```
git commit -m "readme: add readme"
```

- Push the commits to the remote repo: `git push origin master`

```
git push origin master
```

- Go to your Github repository and refresh the browser.