

Setting up your Development Environment

Ling 144: Computational Methods

Winter 2022

Instructor: Kelsey Kraus

1. Create a Github account

If you already have a Github account, great! You're done with this step. Move on to the next step.

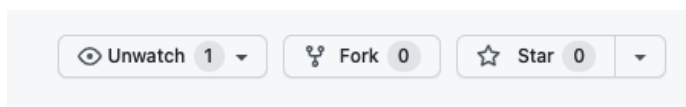
If you need a Github account:

1. Go to <https://github.com>
2. Click on **Sign up**
3. Use your UCSC email address to sign in
4. When prompted, enter a username. It does not matter what your username is, as long as you have signed up with your UCSC email. But be aware that your collaborators in the course will see this
5. Type 'n' to unsubscribe from product updates
6. You'll be asked to 2-factor verify your email address
7. Make sure you are signed up for the Free subscription

2. Fork the class repository

There's not much there currently, but you'll need access to a repository for the next steps.

1. Navigate to our class repository: <https://github.com/berlinguist/Ling144-Winter2022>
2. In the top right corner, click "Fork"



3. Click on your profile avatar in the top right corner and go to "Your Repositories." This should bring you to a page where you can click on our course repository, and see that you now have a copy of it associated with your account.
4. Click again on your avatar and navigate to **Settings > Developer Settings > Personal Access Tokens**.
5. Click **Generate New Token** and fill out the following fields:
 - a. **Note:** Repl.it access for Github
 - b. **Expiration:** 90 Days
 - c. **Select Scopes:**
 - i. *check the **repo** checkbox*
 - ii. *check the **admin:repo_hook** checkbox*

- iii. *check the **admin:public_key** checkbox*
- d. Scroll to the bottom of the page and click the **Generate token** button
- e. Copy the personal authentication token that appears in the green box and paste somewhere accessible. You will need this later.

3. Create a Repl.it account

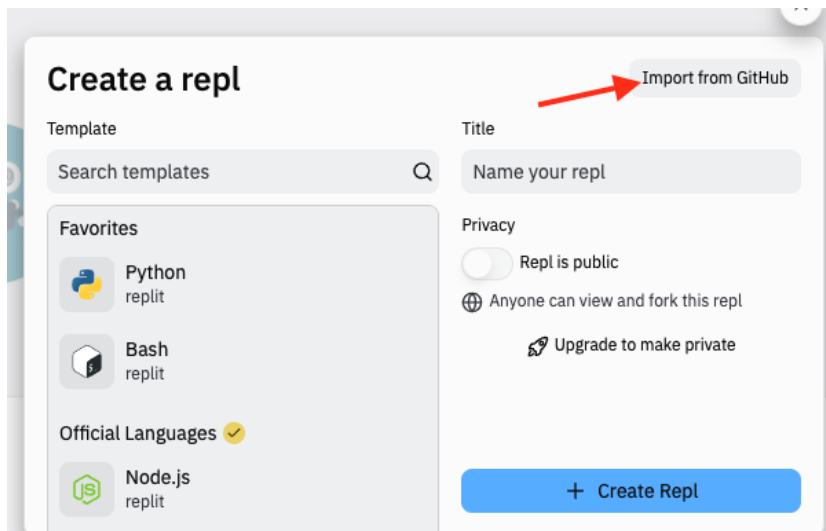
Repl.it is an online development environment. Often users will decide on their own app or local development environment that suits them best, and will run Python as installed on their own computer. One downside of this approach is that everyone's installations will be slightly different, and it makes debugging difficult (especially in a remote format!). With Repl.it, we'll sidestep a lot of those kinds of issues. You can, of course, choose to set up your own environment and Python version, but be warned that debugging issues will be up to you!

1. Go to <https://repl.it>
2. Click **sign up**
3. Create a username (can be the same or different as in Github, up to you)
4. Use your UCSC email address

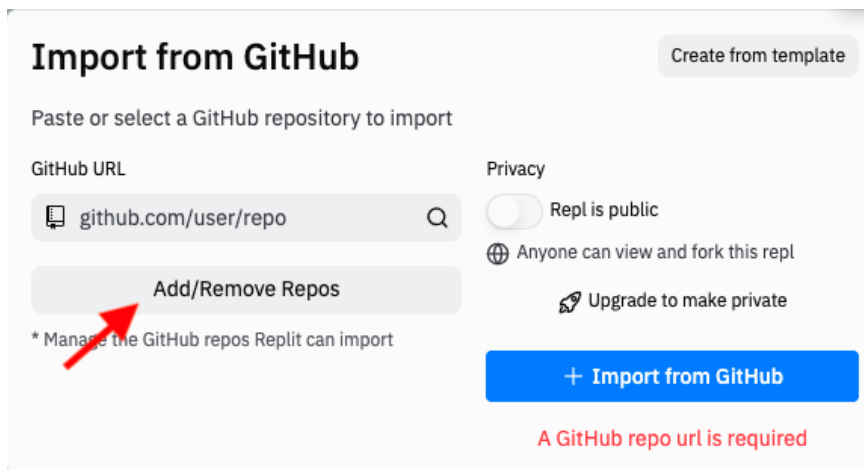
4. Link your Github and Repl.it accounts

Part of our course will be learning about version control and project management for code- and data-based projects, so you'll want to both have access to branches of your github repository, as well as be able to use Repl.it to stage changes the that you'll be making to your github repository.

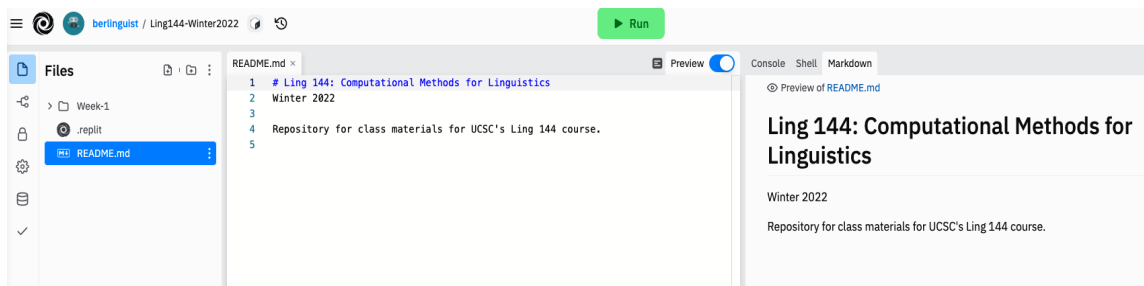
1. Logged in to your Repl.it account, click the button "+ Create Repl"
2. In the upper right of the dialogue box that appears, select "Import from Github"



3. On the dialogue box that shows up, select “Add/Remove Repos”



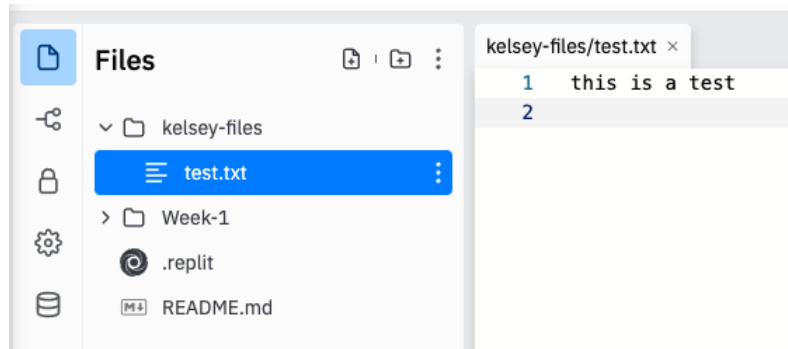
4. If you're still logged in to Github, you'll be redirected to a page with all of your active Repos. If not, it will take you to a login/authorization page, and then redirect you to your list of repos. Select the course repo.
5. Install & authorize
6. You should be redirected back to your Repl.it page, where you can now select the repo that you want to import, and click the “Import from Github” button (you may have to retrace steps 1-3)
7. You should now have a view like this with 3 panes:



You'll now need to make a minimal change to your repository to set up the last authentication part of the puzzle.

8. Click on the “Files” icon on the left, and then on “Add folder.”
 - a. Name it something like “<your-name>-files” – mine’s called **kelsey-files**

- b. In that folder, create a test file called test.txt. Type something minimal in there:



9. On the rightmost pane, click the **Shell** tab
10. We'll now make a quick commit (we'll go over the details of this later). In the console, type:
- git add <your-name>-files/test.txt**
 - Use the command **git status** to make sure that your changes are staged:

```
~/Ling144-Winter2022$ git add kelsey-files/test.txt
~/Ling144-Winter2022$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   kelsey-files/test.txt
```

- git commit -m "first commit"
 - PAUSE HERE** – You should receive a prompt to log in to Github with your username and password. You **WILL NOT USE YOUR REGULAR PASSWORD!**
 - Go to wherever you stored the personal authentication token you generated on Github.
 - Type `git push`
 - Type in your username
 - PASTE IN YOUR PERSONAL ACCESS TOKEN AS YOUR PASSWORD**
(note that you will not see the password being pasted in. This is ok! Just press enter anyway. And as we saw in class, you might have to do this twice)
11. You should be good to go! If all has gone correctly, you should now be able to navigate to your Github repo and see your new file and test folder :)