GitHub Cheat-Sheet

Overview

1. Install the `git` program on your PC
2. Create an account on the GitHub website
3. Generate an access token for your GitHub account and save it
4. Create a repo
5. Clone the repo to your PC
6. Set the upstream branch to your GitHub using `git branch -u origin main`
7. Place a copy of all of your Python files inside the repo
8. Add all of your files to the project using the `git add` command
9. Commit all of your changes using the `git commit -am "message"` command
10. Push your changes to your GitHub using `git push` – will prompt for your GitHub username and password (access token)

GitHub vs. git

GitHub and `git` are two different things.

- `git` is a program, which helps you manage software projects on your computer by providing *version control*
- GitHub is a website where you can share your `git` projects and browse other users' `git` projects

To share your code on GitHub, you will need both.

Installing git

Mac and Linux come with `git` already installed.

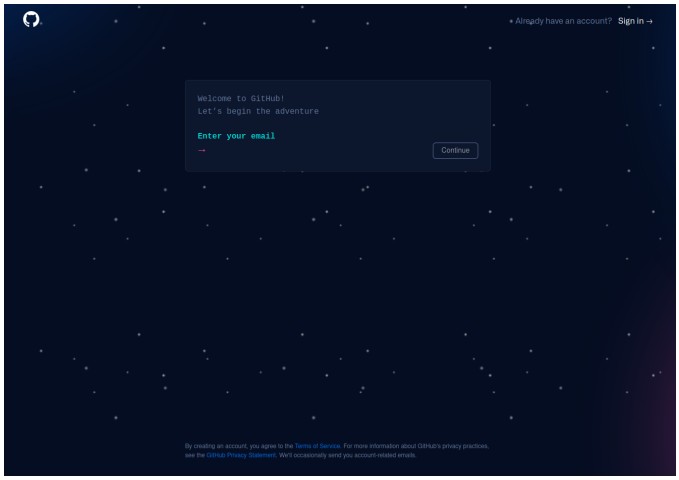
Install `git` for Windows using:
`https://gitforwindows.org/`

Verify your installation at the command line:

```
$ git --version
```

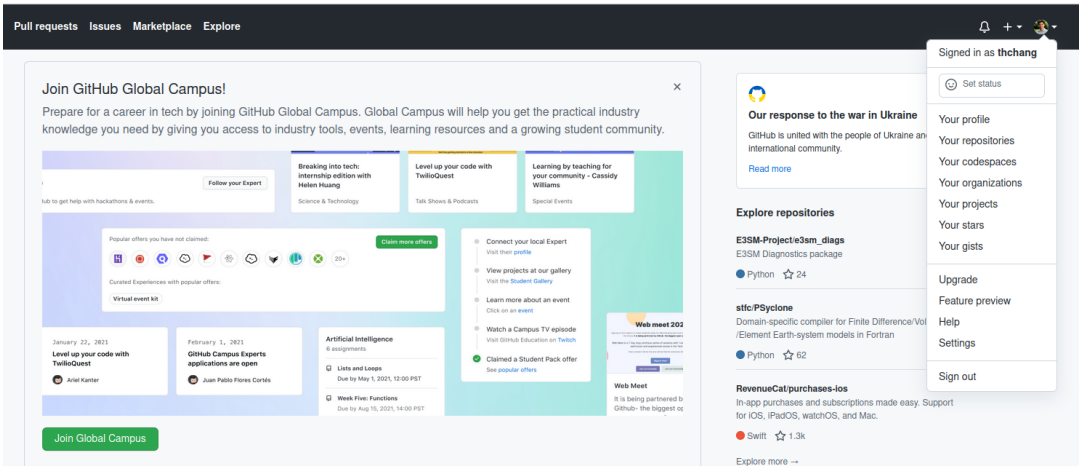
Signing Up for GitHub

Create an account on GitHub using a permanent email address (one that you will have access to indefinitely). If your school will delete your email when you graduate, you might not want to use your school email.



Remember the username, password, and email address that you signed up with. You'll need them regularly to use GitHub.

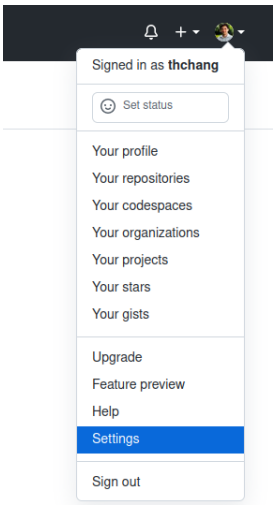
To finish setting up your profile, click your profile picture in the top right corner and go to your profile to finish setting up your account.



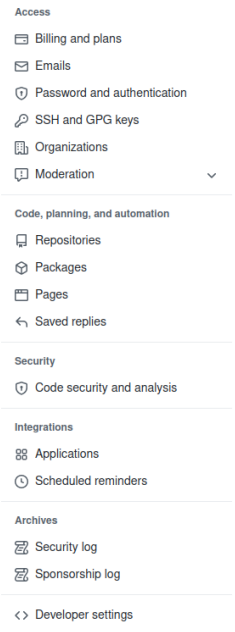
Generating a Personal Access Token

To push code to your GitHub page using `git`, you will need a *personal access token*.

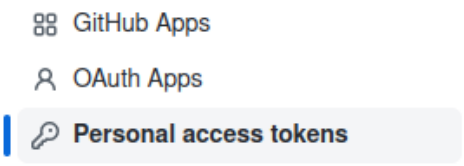
To create one, go to → settings:



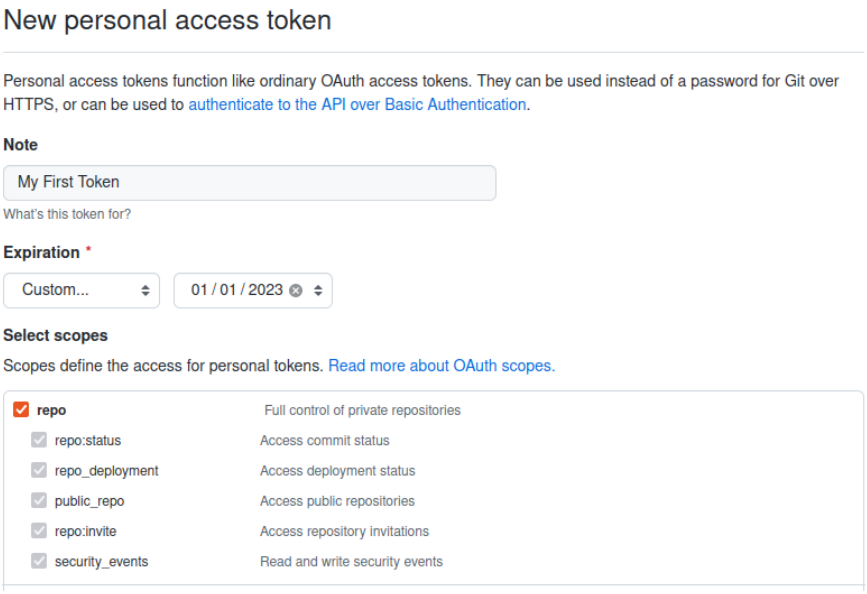
Then go to → developer settings:



Then go to the → token tab:

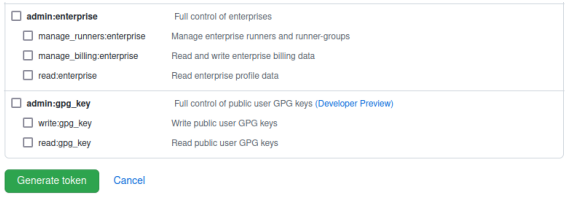


You need to generate a personal access token to push changes from your computer (your password won't work).

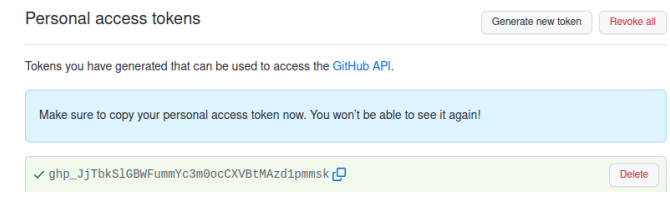


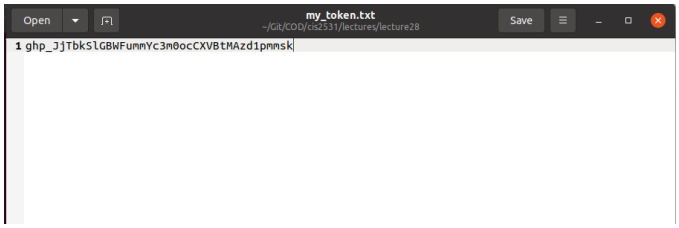
Give yourself full `repo` access. You don't need anything else for now. You will need to set an expiration date. After your expiration date, the token will expire and you'll have to create a new one. Unless you are expecting attempted hackers (in which case I would go shorter), I recommend a maximum of 1 year for each token.

Your token is like your password, keep it secret. But if it gets out, you can always delete it and generate a new one at any time. I will have deleted this token by the time I post this, so you can't break into my account with this token.



After generating the token, GitHub lets you view the new token one time. You will need this token to push code to your GitHub account in the future. However, the token can never be viewed again, so make sure you write it down somewhere:

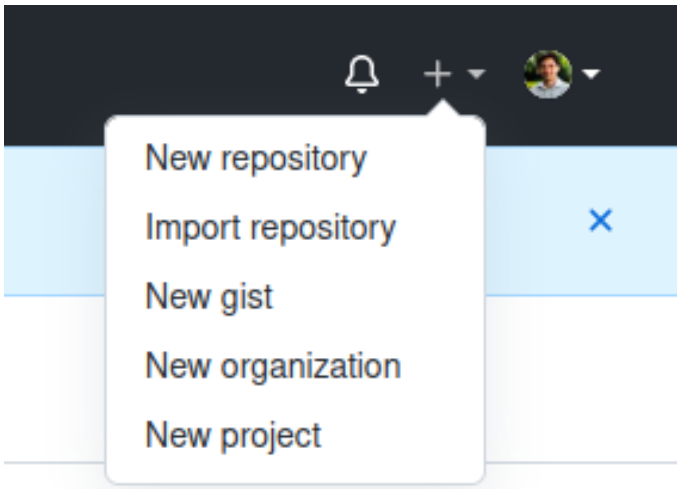




Creating a GitHub repo:

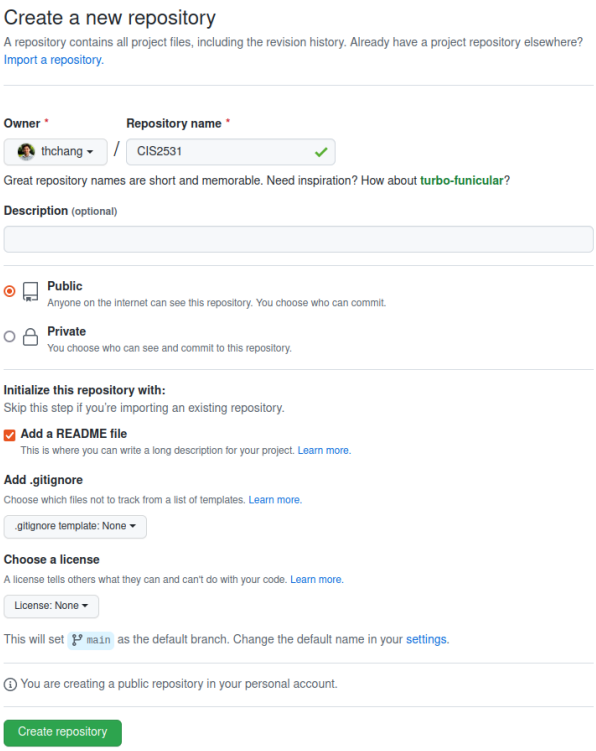
A *repo* (short for repository) is a remote webpage on GitHub, where all your code can be saved and shared with other people.

Now that you have `git` installed, you have an account, and you have a personal access token, you can create a new repo:



and give it a good name.

This is the name that people will use to search it on GitHub's website.



Once you've created a project, clone the online repo to your local machine by using the `git clone` command:

```
git clone https://github.com/user-name/repo-name.git
```

- Replace `user-name` with your user name
- Replace `repo-name` with your repo name

This create a new directory (`repo-name`) in your current working directory. Before you start, set the "upstream" to your remote GitHub repository. You only need to do this once after cloning the repository:

```
cd repo-name
git branch -u origin main
```

You should fill this new directory with the code for your project.

Create and/or copy all the Python (or other required) files for your project inside the new directory.

Then add all of these files to your project, using the `git add` command:

```
git add filename1 filename2 ...
```

Add a directory to add every file in that directory:

```
git add myDirectory
```

When you're done, commit your changes using the `git commit` command:

```
git commit -am "Just added 2 files and directory for the demo handout."
```

Finally, push your changes to your GitHub account:

```
git push
```

You will be asked for your user name and password. Type your GitHub username or email for the username. For the password, **do not** use your GitHub password. Instead, copy-paste your personal access token into the password field and press `return`.

In the future, if you would like to download changes to this GitHub project, made on a different machine, use:

```
git pull
```

To pull in changes from the remote URL.

Add a README file (optional)

A README file will be displayed on the front page of your project, and should tell the user what the project is and the basics of how to install and use it:

README.md:

```
# Welcome to my repo!

This is my first GitHub Uplaod!
Type:  git clone ...  to clone it!
```

Reference Card

git Command	Description
`git clone https://github.com/uname/repo`	Download repo from the given URL
`git branch -u origin main`	Set remote using cloned URL, main branch
`git add filename`	Add `filename` to the `git` project
`git add dirname`	Add contents of `dirname` to project
`git commit -am "Description of changes"`	Commit all changes to `add`ed files in project
`git push`	Send all changes to the remote Repo
`git pull`	Pull any changes from the remote Repo