What's Next?

What's Next?

Showing off on GitHub and Future Studies

Showing Off Your Work

Showing Off Your Work

Want to show off what you've done?

Showing Off Your Work

Want to show off what you've done?

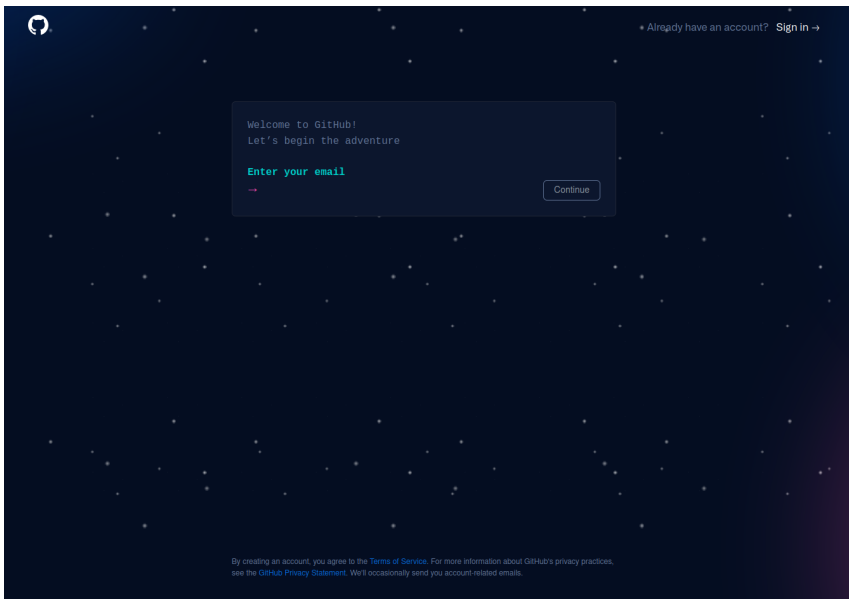
Showing Off Your Work

Want to show off what you've done?

`https://github.com/signup`

Sign-Up Page

Create an account using a permanent email address (one that you will have access to indefinitely):



Install git on your Computer

Install git on your Computer

Mac and Linux come with `git` already installed

Install git on your Computer

Mac and Linux come with `git` already installed

For Windows:
`https://gitforwindows.org/`

Install git on your Computer

Mac and Linux come with `git` already installed

For Windows:
`https://gitforwindows.org/`

Verify installation:

Install git on your Computer

Mac and Linux come with `git` already installed

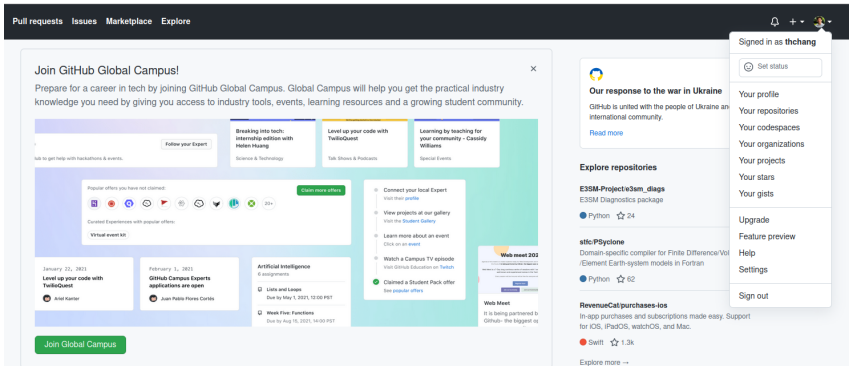
For Windows:
`https://gitforwindows.org/`

Verify installation:

```
>>> git --version
```

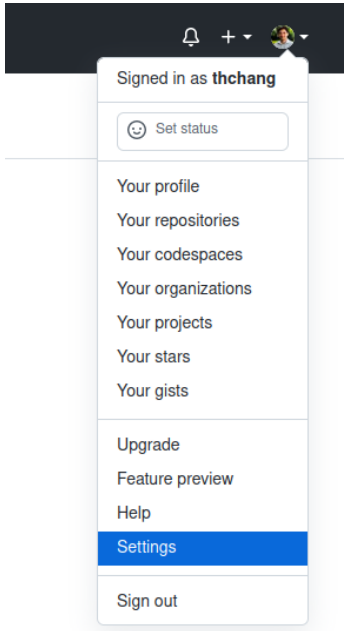
Create Your Profile

Click your profile picture in the top right corner and go to your profile to finish setting up your account:

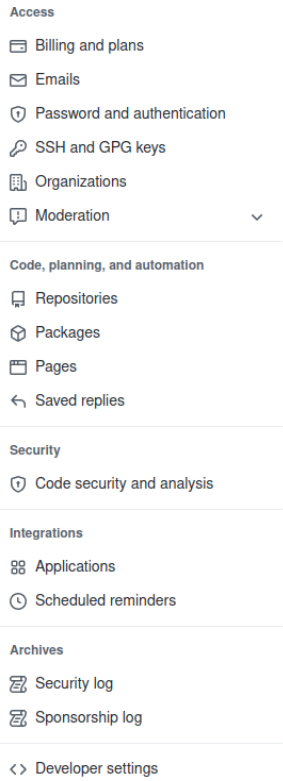


Go to Settings

When you're done, go to settings:

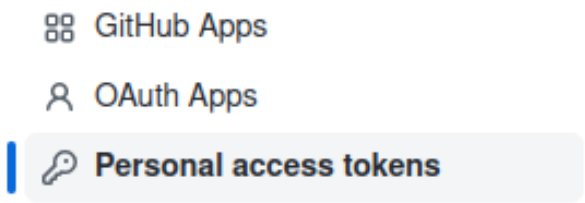


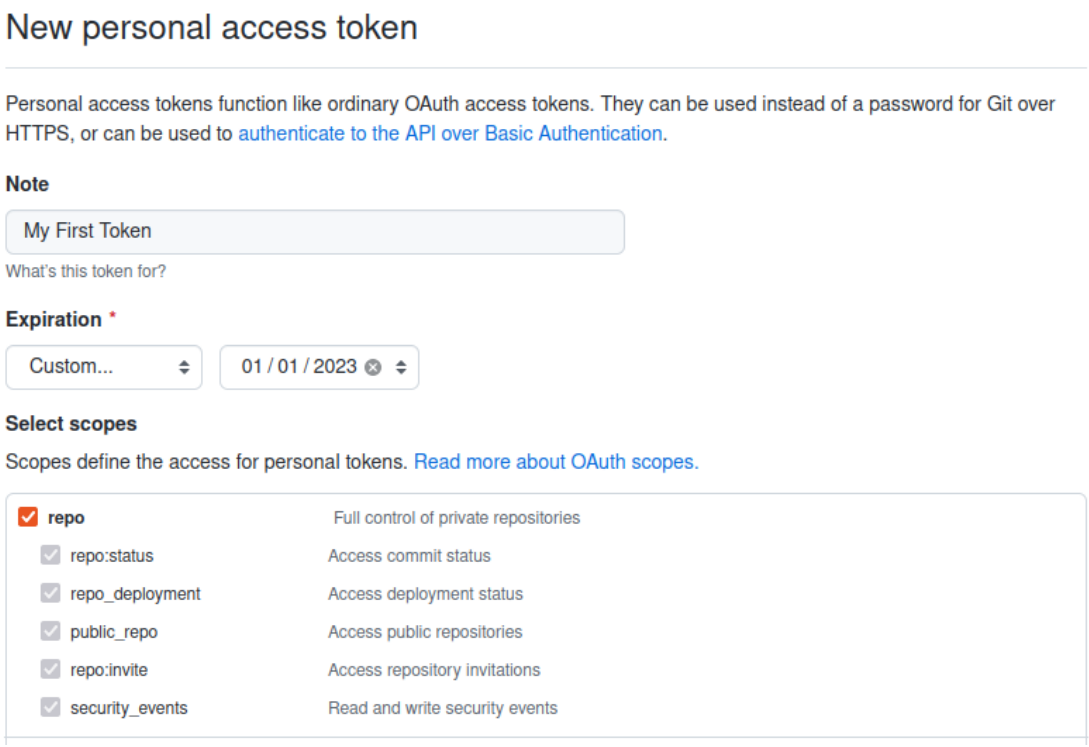
Developer settings



Token Tab

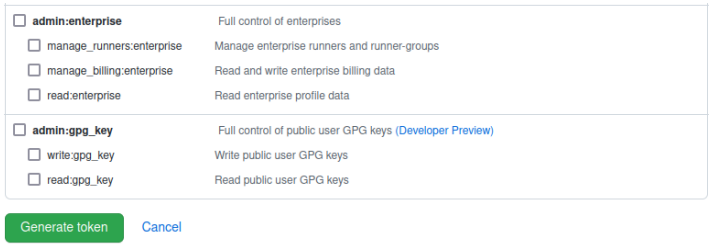
You need to generate a token to push changes from your computer (your password won't work):

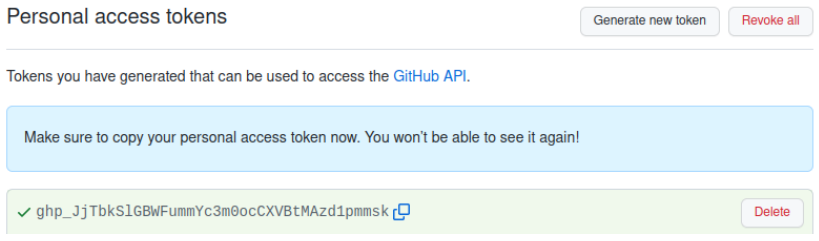


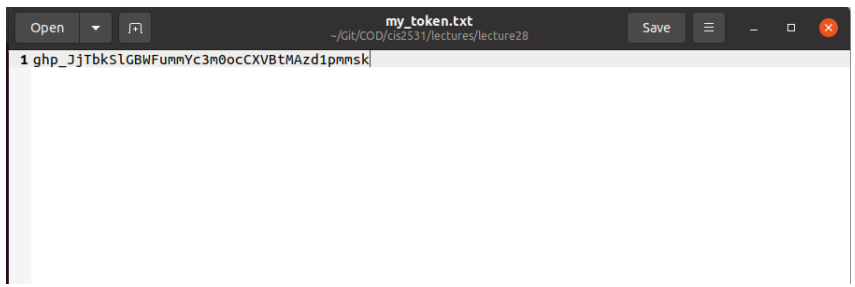


Generate Token

Save your token in a file

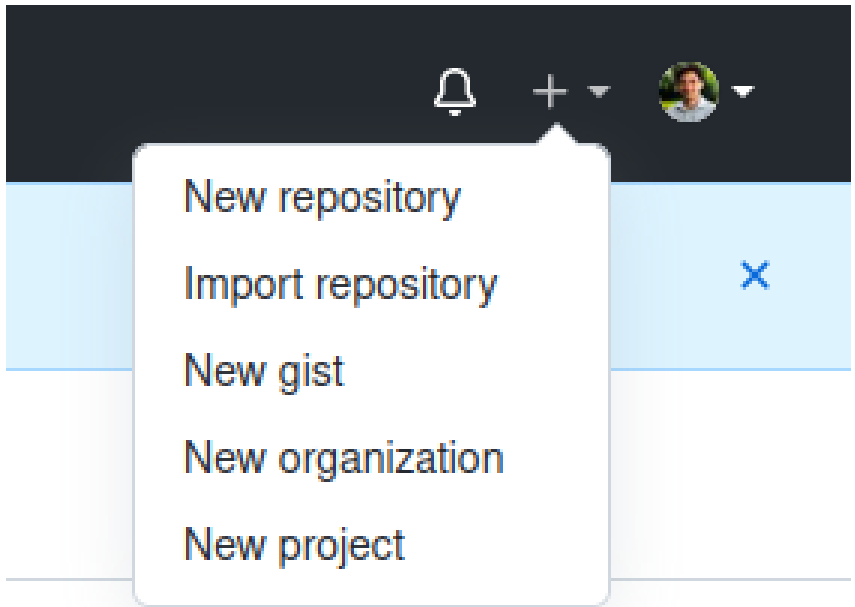




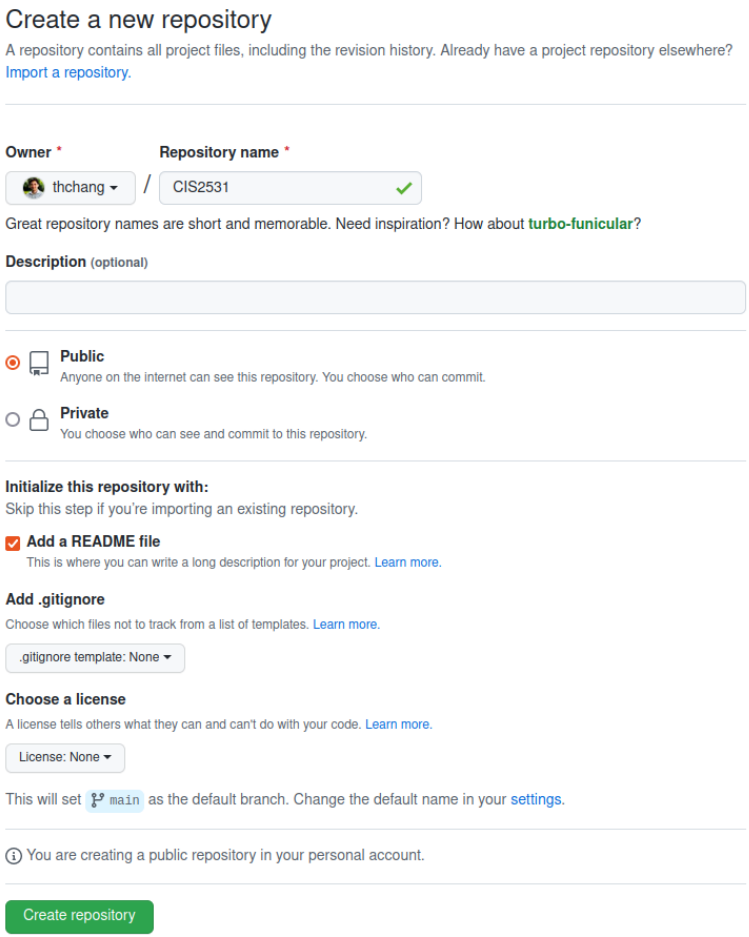


Create a Repo

Now create a new repo



Give Your Repo a Name



Clone your Repo on your local machine

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

Clone your Repo on your local machine

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

Replace `user-name` with your user name

Clone your Repo on your local machine

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

Replace `user-name` with your user name

Replace `repo-name` with your repo name

Clone your Repo on your local machine

```
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
Create/copy all the files for your project inside that directory

Clone your Repo on your local machine

Clone your Repo on your local machine

Add files to your project:

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

```
git push
```

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

```
git push     (--setupstream origin master)
```

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

```
git push     (--setupstream origin master)
```

Will ask for your user name and password. Type your github username and copy-paste your access token for the password

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

```
git push     (--setupstream origin master)
```

Will ask for your user name and password. Type your github username and copy-paste your access token for the password

```
git pull
```

Clone your Repo on your local machine

Add files to your project:

```
git add filename
```

Commit changes:

```
git commit -am "message about what you changed"
```

Push changes:

`git push` (`--setupstream origin master`)

Will ask for your user name and password. Type your github username and copy-paste your access token for the password

`git pull` Pull changes from remote

Add a README

Add a README

README.md:

Add a README

README.md:

```
# Welcome to my repo!

This is my first GitHub Uplaod!
```

Clone Other People's Projects

Clone Other People's Projects

You can clone other people's projects also using `git clone`

Clone Other People's Projects

You can clone other people's projects also using `git clone`

Of course, you won't be able to push changes

Clone Other People's Projects

You can clone other people's projects also using `git clone`

Of course, you won't be able to push changes

If you want to make changes, fork their project first to your own repo, then clone your fork

Link your GitHub

Once you have a couple projects, you should link your GitHub on your Resume:

Link your GitHub

Once you have a couple projects, you should link your GitHub on your Resume:

Tyler H. Chang
City: xxxxxx GitHub: `https://github.com/thchang`
email: xxx@xxx.xxx Phone: xxx-xxx-xxxx

Experience:
Professional Googler *at Google* (10 years)
· Googles stuff ... (on my personal laptop)
· Didn't actually work at Google
· May never have had a job at all?

Next steps

Next steps

What to study next?

What to study next?

What part of this course did you enjoy most?

Front-end Developer

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

- `UX/design`

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

- `UX/design`
- `html/css`

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

- `UX/design`
- `html/css`
- Web scripting languages: `javascript`, `php`

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

- `UX/design`
- `html/css`
- Web scripting languages: `javascript`, `php`
- Python web-scripts: `pyscript`

Front-end Developer

Builds pretty websites and GUIs (like tkinter)

- `UX/design`
- `html/css`
- Web scripting languages: `javascript`, `php`
- Python web-scripts: `pyscript`
- Python frameworks: `django`, `flask`

Back-end Developer

Back-end Developer

Builds complex programs and database systems to make the website work

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`
- Common database systems (such as `SQL`)

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`
- Common database systems (such as `SQL`)
- Python frameworks: `django`, `flask`

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`
- Common database systems (such as `SQL`)
- Python frameworks: `django`, `flask`
- common data structures & algorithms

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`
- Common database systems (such as `SQL`)
- Python frameworks: `django`, `flask`
- common data structures & algorithms
- standard Python libraries

Back-end Developer

Builds complex programs and database systems to make the website work

- Continue to improve your `Python`
- Common database systems (such as `SQL`)
- Python frameworks: `django`, `flask`
- common data structures & algorithms
- standard Python libraries
- operating systems

Software Engineer

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms
- Keep improving your `Python`

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms
- Keep improving your `Python`
- Learn about standard Python libraries (like `pandas` and `numpy`)

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms
- Keep improving your `Python`
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms
- Keep improving your `Python`
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)
- Programming styles and workflows (test-driven development, agile methodology, etc.)

Software Engineer

Designs and builds entire applications from conception to completion based on company specs

- Procedural, Object-Oriented, and Event-Driven Programming
- Advanced Data Structures and Algorithms
- Keep improving your `Python`
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)
- Programming styles and workflows (test-driven development, agile methodology, etc.)
- Tools to automate this process

Dev-Ops

Dev-Ops

Supports software engineers with tools for automating their workflow

Dev-Ops

Supports software engineers with tools for automating their workflow

- Advanced understanding developer styles and best practices

Dev-Ops

Supports software engineers with tools for automating their workflow

- Advanced understanding developer styles and best practices
- Aware of all common developer workflows and techniques (such as agile methodology, etc.)

Dev-Ops

Supports software engineers with tools for automating their workflow

- Advanced understanding developer styles and best practices
- Aware of all common developer workflows and techniques (such as agile methodology, etc.)
- Tools to automate these processes

Dev-Ops

Supports software engineers with tools for automating their workflow

- Advanced understanding developer styles and best practices
- Aware of all common developer workflows and techniques (such as agile methodology, etc.)
- Tools to automate these processes
- Pros and cons of these tools

Data Analyst

Data Analyst

Maintains company data and uses data to generate actionable business strategies

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms
- Learn about standard Python libraries (like `pandas` and `numpy`)

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)
- Basic plotting and machine learning tools (like `scikit-learn` and `matplotlib`)

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)
- Basic plotting and machine learning tools (like `scikit-learn` and `matplotlib`)
- Statistics

Data Analyst

Maintains company data and uses data to generate actionable business strategies

- Common Data Structures and Algorithms
- Learn about standard Python libraries (like `pandas` and `numpy`)
- Database systems (such as `SQL`)
- Basic plotting and machine learning tools (like `scikit-learn` and `matplotlib`)
- Statistics
- Strong communication skills