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?

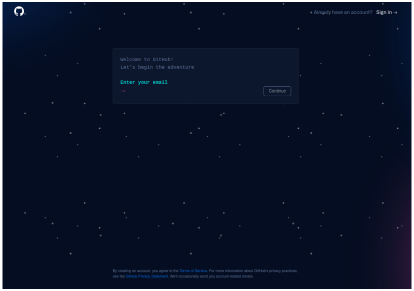
Showing Off Your Work

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):



Welcome to GitHub!
Let’s begin the adventure

Enter your email
→

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

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

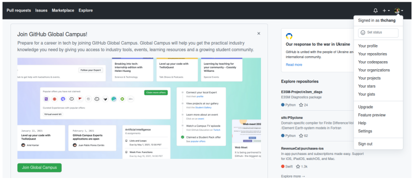
For Windows:
`https://gitforwindows.org/`

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:



Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

Our response to the war in Ukraine

GitHub is united with the people of Ukraine an international community.

Explore repositories

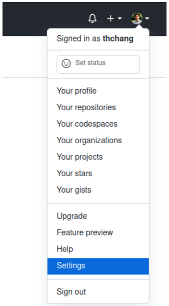
E3SM-Project/e3sm_diags
E3SM Diagnostics package
Python 24

stfc/PSyclone
Domain-specific compiler for Finite Difference/Vol /Element Earth-system models in Fortran
Python 62

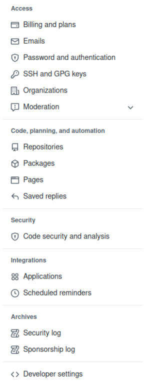
RevenueCat/purchases-ios
In-app purchases and subscriptions made easy. Support for iOS, iPadOS, watchOS, and Mac.
Swift 1.3k

Go to Settings

When you're done, go to settings:

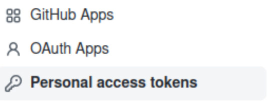


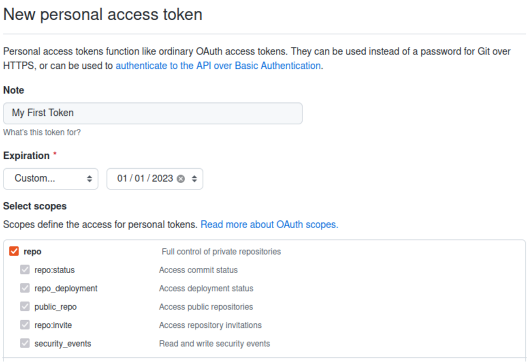
Developer settings



Token Tab

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




New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

My First Token

What's this token for?

Expiration *

Custom... 01 / 01 / 2023

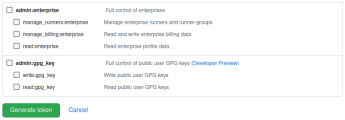
Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

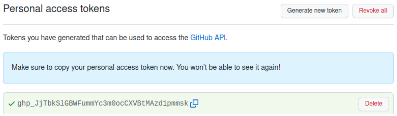
✓ **repo**	Full control of private repositories
✓ repo:status	Access commit status
✓ repo_deployment	Access deployment status
✓ public_repo	Access public repositories
✓ repo:invite	Access repository invitations
✓ security_events	Read and write security events

Generate Token

Save your token in a file



admin:enterprise	Full control of enterprises
manage_runners:enterprise	Manage enterprise runners and runner-groups
manage_billing:enterprise	Read and write enterprise billing data
read:enterprise	Read enterprise profile data
admin:gpg_key	Full control of public user GPG keys (Developer Preview)
write:gpg_key	Write public user GPG keys
read:gpg_key	Read public user GPG keys

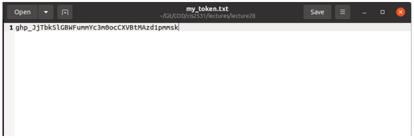


Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

```
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
```

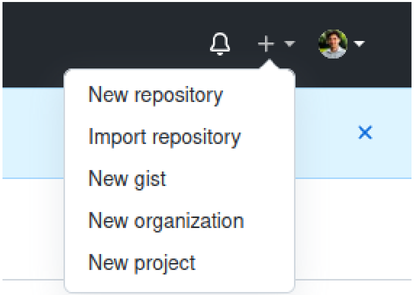


my_token.txt
~/Git/COD/cis2531/lectures/lecture28

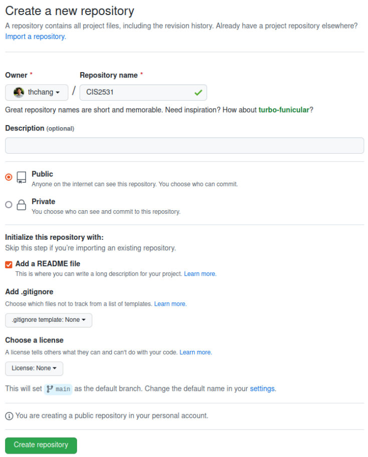
```
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
```

Create a Repo

Now create a new repo



Give Your Repo a Name


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * thchang / Repository name * CIS2531 ✓

Great repository names are short and memorable. Need inspiration? How about turbo-funicular?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

✓ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

This will set `main` as the default branch. Change the default name in your settings.

ⓘ You are creating a public repository in your personal account.

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"
```

Set upstream to your remote repository (only first time):

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

Push changes:

```
git push
```

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

Push changes:

```
git push
```

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"
```

Set upstream to your remote repository (only first time):

```
git branch -u origin main
```

Push changes:

```
git push
```

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?

Next steps

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