



# Introduction to Git

Presented to CSUSB Data Analytics Working Group

**Youngsu Kim**

High Performance Computing Faculty Fellow  
Assistant Professor of Mathematics

2/29/2024



# Git



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

-- [www.git-scm.com](http://www.git-scm.com)



# How much do we know about git?

❏ Have you heard about git?

❏ Ans: Yes (2), No (1)

❏ What about GitHub?

❏ Ans: Yes (1), No (1)

❏ Do you use it? If so, what are your use cases?

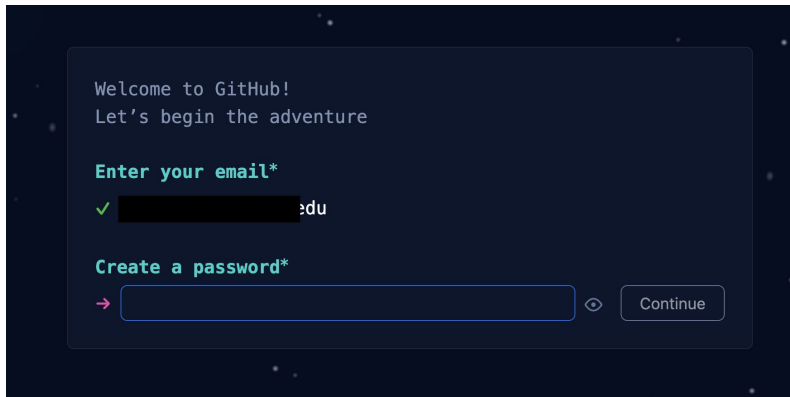
❏ Ans:



# GitHub (or any git) Account

Helpful to follow along interactive session later

<https://github.com/signup>


A screenshot of the GitHub signup form on a dark background. The form is a light gray rectangle with rounded corners. It contains the following text: "Welcome to GitHub!" and "Let's begin the adventure" in a light gray font. Below this, "Enter your email\*" is written in a teal font. A green checkmark is to the left of a black input field containing "example@du". Below that, "Create a password\*" is written in a teal font. To the left of a white input field is a red arrow. To the right of the input field is a small eye icon. Further right is a rounded button labeled "Continue" in a light gray font.

Welcome to GitHub!  
Let's begin the adventure

Enter your email\*

✓ example@du

Create a password\*

→  



# Git vs GitHub

Feature	Git	GitHub
Definition	Distributed Version Control System (DVCS)	Web-based platform for collaborative software development
Purpose	Tracks changes, enables version control, local work	Hosts Git repositories, facilitates collaboration, project management
Key Features	- Local repository with full history	- Hosting Git repositories in the cloud
	- Branching and merging	- Issue tracking
	- Version control capabilities	- Pull requests
Usage	Version control, tracking changes, managing history	Hosting repositories, collaborating, managing issues
Dependency	Standalone, does not require internet	Web-based, requires internet and GitHub account
Examples	CLI tools like Git Bash, Git CLI	Web interface, desktop applications like GitHub Desktop

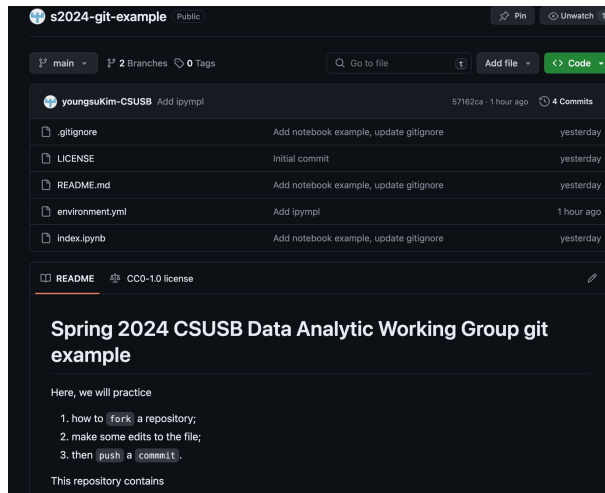
# Git vs GitHub (cont'd)

```
s2024-git-example > ls -l
total 656
-rw-r--r--  1 006501270  staff   7048 Feb 28 19:34 LICENSE
-rw-r--r--  1 006501270  staff    204 Feb 28 19:35 README.md
-rw-r--r--  1 006501270  staff    116 Feb 28 19:57 environment.yml
-rw-r--r--  1 006501270  staff 316947 Feb 28 19:34 index.ipynb
s2024-git-example > cat README.md
# Spring 2024 CSUSB Data Analytic Working Group git example

Here, we will practice

1. how to `fork` a repository;
1. make some edits to the file;
1. then `push` a `commit`.

This repository contains
```



The screenshot shows the GitHub interface for a public repository named 's2024-git-example'. The repository is owned by 'youngsuKim-CSUSB'. It has 2 branches and 0 tags. The file list shows: .gitignore, LICENSE, README.md, environment.yml, and index.ipynb. The README.md file is selected, showing its content: '# Spring 2024 CSUSB Data Analytic Working Group git example', 'Here, we will practice', a list of three steps (1. how to `fork` a repository; 1. make some edits to the file; 1. then `push` a `commit`.), and 'This repository contains'. The repository was created 57162ca 1 hour ago and has 4 commits.



# What?

One “may” think of Git and GitHub as

- Python and Jupyter Lab/Notebook
- R and RStudio

\*git-cli offers more for certain purposes



## Three Popular Git Services

 **Bitbucket**

 **GitLab**

 **GitHub**





# Examples

- ❏ <https://youngsukim-csusb.github.io/> my webpage
- ❏ <https://github.com/youngsuKim-CSUSB/presentations.git> this presentation
- ❏ <https://github.com/youngsuKim-CSUSB/s2024-git-example>
- ❏ <https://github.com/tensorflow/tensorflow>
- ❏ [https://gitlab.nrp-nautilus.io/youngsu\\_kim/sphinx-test;](https://gitlab.nrp-nautilus.io/youngsu_kim/sphinx-test;)  
[https://youngsu\\_kim.pages.nrp-nautilus.io/sphinx-test/](https://youngsu_kim.pages.nrp-nautilus.io/sphinx-test/)
- ❏ <https://pypi.org/project/pandas/>
- ❏ <https://github.com/jupyterhub/binderhub>
- ❏ <https://github.com/suthakaranr/my-first-binder>



# Key Terms in Git

Loosely

- ❑ repository (repo): a project
- ❑ branch of a repo: often main and branches of main
- ❑ pull & push: download & upload
- ❑ commit: snapshot
- ❑ fork: copy a repo

We create a **repository** and add files to the main **branch**. When done, we **commit** and **push** to the **repo**. Today, we **fork** a **repo**, make changes, **commit**, and **push**.



# Git Demo

Session 1: set up a **repo** with a Binder link, pull request

- ❑ Create a GitHub account
- ❑ **Fork** the example repo; <https://github.com/youngsuKim-CSUSB/s2024-git-example>
- ❑ Make some changes and **commit**
- ❑ Make a **pull request**

Session 2: initialize/create a repo and explore GitHub action, open-ended

- ❑ Create a **repo**
- ❑ Add README.md
- ❑ Create a **branch** and explore **merging**