HW 2 - Introduction to GitHub

ULAB - Physics and Astronomy Division

Due Sunday, September 29th, 2024 at 11:59pm

1 GitHub Repositories

During the lecture on Monday, September 29th, we covered how to clone the ULAB remote GitHub repository. **Cloning** a repository means creating a local copy of the public code on your personal laptop. Going forward, all ULAB homework assignments (including this one) will be accessed through the `ulab_2024` GitHub repository!

Important Reminder: You only need to clone the `ulab_2024` repository on your command line **ONCE**! Next lecture we will cover how to update the remote repository.

GitHub is a platform where programmers (like you!) can store, manage and share code. It uses **Git**, a version control system that tracks changes in the code. You can think of GitHub as a cloud-based storage system (or cloud-based filing cabinet) that not only holds code but also keeps track of every version and change made.

A **directory** (or folder) is simply a place on your computer where files and data are stored. A **repository** (often called a **repo**) is an advanced directory that can store files/data but also manage history, versions and collaborations.

1.1 Cloning a GitHub Repository

After cloning the `ulab_2024` repository (which is how you found this homework assignment), navigate to the repo by calling `cd ulab_2024`. Then, run `git status`. Take a screenshot of your terminal showing the output from `git status` inside of the `ulab_2024` repository.

Do some research (on the internet) on what running `git status` on the command line inside a repository does. Write a few sentences explaining its purpose. You might encounter the concept of **staging**—don't worry if it seems confusing; we'll cover topics like **pull/push/branch/stage/commit/add** in the next lecture!

1.2 Viewing Code History

A **commit** is like a snapshot of your project at a specific point in time. It records the changes you (or others) made to the files, allowing you to revert to that version if needed. Each commit includes:

- A unique ID
- A message (describing what change has been made)
- A record of what changed in the files

Inside the `ulab_2024` repository, run `git log --oneline`. What is the unique ID associated with the message 'Initial commit'?

Research three other flags/arguments that can be attached to `git log`. Run them on your command line inside of the `ulab_2024` repository. Describe what the new command is doing, what you learned about `ulab_2024` when you ran it, and what happened in your Terminal. Take a screenshot of one of them.

1.3 Making a GitHub account

Follow the instructions given during lecture (or look at the slides) and make a GitHub account with your **@berkeley.edu** email address. Take a screenshot of your profile.

2 Anaconda

On bCourses, under the **python** folder, there is a file called **Installation Guide**. Go to that document and scroll down to the section called **Anaconda (Jupyter Notebooks)**. Follow the directions and install Anaconda on your computer. Once you are done installing run `jupyter notebook` on your command line and take a screenshot of the output.