HW 2 - Introduction to GitHub

ULAB - Physics and Astronomy Division

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

1 GitHub Repositories

During lecture on Monday, September 29th, we covered how to clone the ULAB remote GitHub repository. **Cloning** a repository means you're 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 a GitHub repository to your local machine **ONCE**! Next lecture we will cover how to update your remote repositories.

Important Reminder: You only need to clone a repository to your local machine once. Afterward, you'll simply update it by pulling changes as needed.

GitHub is a platform where programmers (like you!) can store, manage and share code. It uses **Git**, a version control system, to keep track of changes in the code. You can think of GitHub as a cloud-based storage system (or cloud-based filing cabinet) for your code that tracks every version and change.

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

1.1 Cloning a GitHub Repository

Once you have cloned the `ulab_2024` repository (which is how you found this homework assignment), navigate into 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.

Research (on the internet) what calling `git status` on the command line inside a repository is doing. Write a couple of sentences explaining `git status`. You

will come across the concept of **staging**, don't worry if this doesn't make sense. We will cover **pull/push/branch/stage/commit/add** all next lecture!

1.2 Viewing a Code's History

A **commit** is like taking a snapshot of your coding project at a specific point in time. It saves the changes you've (or other people) have made to the files, so you can go back to that version later if needed. Each commit contains:

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

Inside of the `ulab_2024` repository, on your command line call `git log --oneline`. What is the unique ID attached to the message 'Initial commit'?

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.