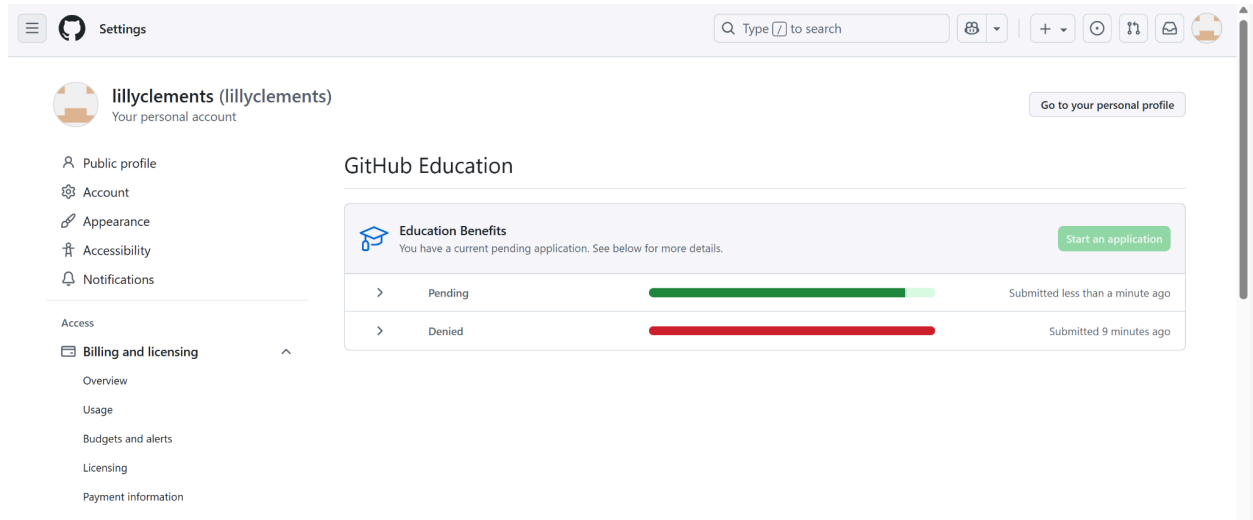
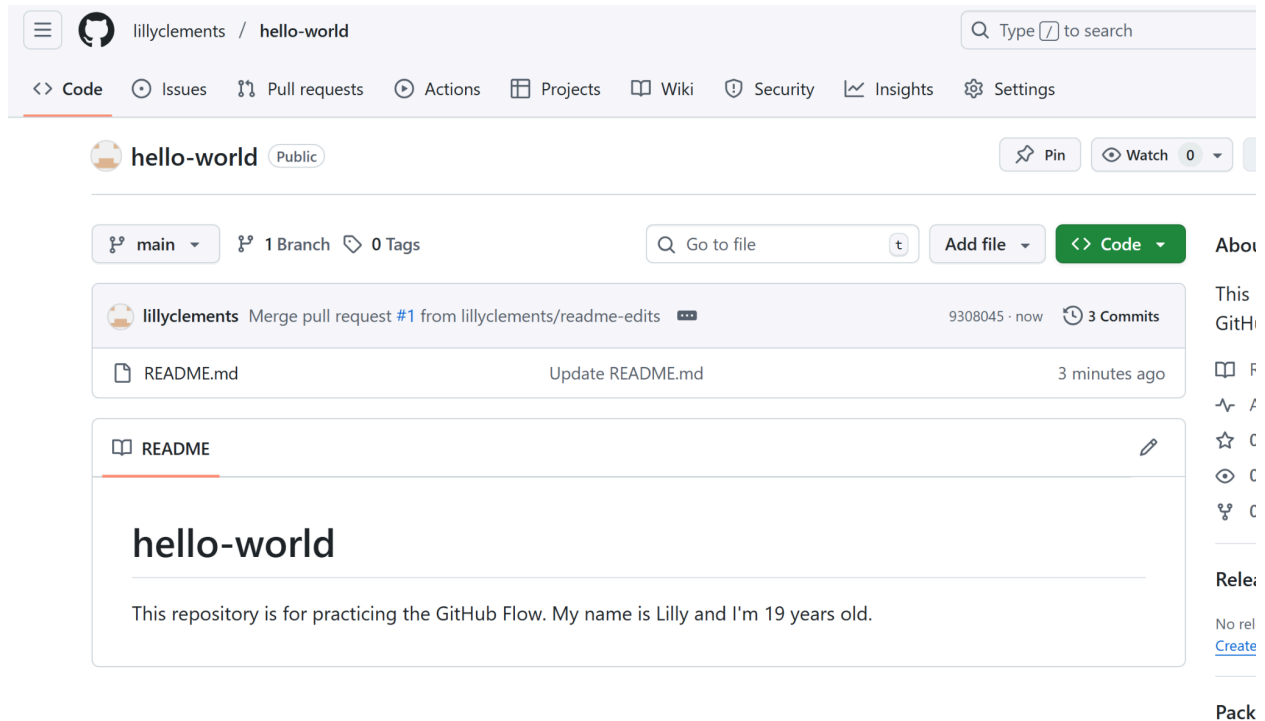


## Homework 4 Log

I am still awaiting my GitHub Student Developer Pack application to get approved, but I have included a screenshot below of my pending application submission.



My result from completing the Hello-world tutorial is shown below:



## VERSION CONTROL:

**Version control** is an essential tool for engineers, even if they are not full-time software developers, because it allows for tracking and managing changes to code. It helps organize a project's development process and provides a safety net: if a mistake is made, developers can revert to earlier versions of the code, compare changes, and fix issues with minimal disruption to the team. Good version control software supports a developer's preferred workflow without forcing a specific one.

**Source Code Management (SCM)** is a related concept that involves tracking modifications to a source code repository. SCM tools are critical for managing the complexity and costs of growing software projects. They allow multiple people to collaborate on the same project without overwriting each other's work or causing conflicting edits.

One of the most widely used version control systems today is Git. Git is popular because it is flexible, efficient, and scalable for both small and large projects. It supports various nonlinear workflows, integrates easily with other systems, and allows developers to create and manage branches for different features or experiments. However, a common criticism of Git is that it can be difficult to learn. Some of its terminology and workflows are unique and may be confusing to newbies or those familiar with other systems.

Looking back, I think version control would have been extremely helpful during my end-of-year ASTE-101 project, which involved writing a Python program to map the terrain of a planet. My team worked simultaneously in Visual Studio Code, and I think using version control could have helped us coordinate our changes, reduce errors, and improve overall efficiency.

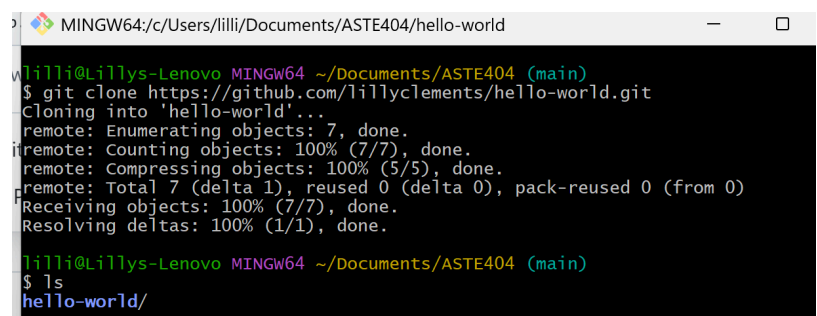
---

### Git from the Command Line

I used Git Bash as my terminal and followed the prompts to set up git, set my username, and cache my GitHub credentials.

### Cloning the Tutorial Repo Locally

Next, I cloned my GitHub tutorial repository to my local computer using Git. I created a folder for the assignment, navigated to it in the terminal, and ran the `git clone` command with my repository's HTTPS URL to download the project files locally so I could edit and track changes directly on my computer. My terminal is shown on the right.



```
MINGW64:/c/Users/lilli/Documents/ASTE404/hello-world
lilli@Lillys-Lenovo MINGW64 ~/Documents/ASTE404 (main)
$ git clone https://github.com/lilliclements/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
lilli@Lillys-Lenovo MINGW64 ~/Documents/ASTE404 (main)
$ ls
hello-world/
```

