

Intro to Jupyter and GitHub

Jiri Musil

Department of Science, Technology, Engineering & Math,  
Houston Community College

Natural Language Processing (NLP) - ITAI 2373

Patricia McManus

June 13<sup>th</sup>, 2025

## What I Did

In this introductory lab, I worked through two primary workflows: GitHub for code version control and Google Colab for Jupyter notebooks. I was intentionally curious about its features and how to potentially use them in the future.

Here are the steps I took:

1. **Created a GitHub account:** I registered for a free account at GitHub.com, verified my email, selected a concise username, and activated two-factor authentication to keep future repositories secure.
2. **Explored the GitHub interface:** On the dashboard, I navigated through the Repositories, Issues, and Pull Requests tabs to find the code, discuss bugs or feature requests, and review how contributions are proposed and merged. By hovering over the bell icon, I accessed the notifications feed, which is useful for keeping track of team activity.
3. **Created a repository named jupyter-exploration:** I utilized the “+ / New repository” shortcut, entered the repository name, set it to public as required for submission, selected “Add a README”, and clicked Create. The README was instantly visible in the main branch.
4. **Made my first commit:** I opened README.md, chose Edit, and incorporated a short overview of the lab. After entering README update in the commit message box, I committed directly to main. So far, it has been a very smooth and intuitive process.
5. **Accessed Jupyter via Google Colab:** Instead of installing software locally, I launched Colab, logged in with my Google account, and chose File → New notebook. Within moments, a new .ipynb file with a Python 3 kernel was created.
6. **Launched and explored the notebook environment:** I observed the recognizable menu bar, the variable inspector sidebar, and the RAM/disk usage meter. The option to toggle GPU/TPU hardware was also apparent, though not needed for this straightforward task.
7. **Created my first notebook:** I pressed Ctrl + M + M to change the first cell to Markdown, then typed “Here is my first Markdown cell in Jupyter, as I embark on my exploration journey.” Underneath this, I added the code print to introduce myself briefly and pressed Shift + Enter, which produced the expected output right below the cell.
8. **Saved and downloaded the notebook:** I saved the notebook as Intro\_Jiri\_Musil.ipynb by selecting File → Save a copy in Drive. Next, I opted for File → Download → Download .ipynb to create a local copy for uploading to GitHub.
9. **Uploaded the notebook with GitHub Desktop:** Once I installed GitHub Desktop, I cloned the jupyter-exploration repository to a local folder. I then added Intro\_Jiri\_Musil.ipynb, wrote “Add my first Jupyter Notebook” as the commit message, selected Commit to main, and clicked Push origin. Afterward, I checked GitHub.com to ensure the notebook displayed correctly in the repository viewer.

## **What I Learned**

This concise exercise solidified two essential skills: distributed version control and interactive computing.

While I grasped the concept of commits and branches, actually entering a commit message and observing the history update solidified my understanding. GitHub's web-based editor made it easier to create a first commit, while GitHub Desktop demonstrated how local clones synchronize with the remote repository. I also understood how a README acts as living documentation that evolves alongside the code. Observing the Issues and Pull Requests tabs suggested future developments teamwork. Understanding that every change, whether it's a typo fix or a substantial code addition, can be tracked, discussed, and rolled back provides psychological safety when experimenting.

Google Colab demonstrated that powerful notebook technology is available without installation hassles. Combining Markdown for exposition with executable Python for evidence creates a powerful approach to teaching and data analysis pattern. Storing the notebook in Drive, downloading it, and subsequently uploading it to GitHub showcased the portability of the .ipynb file. GitHub's built-in notebook renderer is especially helpful for demonstrating results to reviewers who remain in the browser.

GitHub Desktop initially could not authenticate due to my 2FA being enabled. The solution involved creating a personal access token and using it as the password, serving as a useful lesson in secure credential management. When I made my first commit, GitHub Desktop alerted me to mixed line endings. I chose to accept the suggested normalization, realizing that even unseen characters play a crucial role in version control.

This lab delivered an impressive amount of practical knowledge in less than an hour. I left with a functional GitHub repository, a working Jupyter notebook, and, most importantly, the assurance that I can establish a reproducible data science workspace anytime and anywhere.

### **Link to my GitHub repository:**

<https://github.com/JiriCZTX/jupyter-exploration.git>