<https://github.com/Osakhra/jupyter-exploration>

**a. What I Did**

1. **GitHub Account Setup (This was already completed)**
   * Navigated to github.com and signed up for a free account using my email address.
   * Verified my email and configured my profile settings.
2. **Repository Creation**
   * On the GitHub dashboard, clicked the “+” icon and selected **New repository**.
   * Named the repository **jupyter-exploration**, added a short description (“Lab 01: Jupyter & GitHub exploration”), and initialized it with a README file.
   * I made the repository private without realizing, I would later go back and learn how to switch it to public through the Danger Zone setting.
3. **First Commit**
   * Opened the README.md in the GitHub web editor.
   * Replaced the placeholder text with a one-sentence description of the lab session.
   * Committed directly to the **main** branch.
4. **Choosing a Jupyter Environment**
   * Decided to use **Visual Studios Code** for ease of setup and my personal ease of use.
5. **Notebook Creation and Exploration**
   * Added a **Markdown** cell with the text:
   * # My first markdown cell in Jupyter
   * Added a **Code** cell with:
   * print("Hello, World!")
   * Executed both cells (Shift + Enter) and confirmed the Markdown rendered properly and the code printed “Hello, World!”.
6. **Saving and Downloading**
   * Saved the notebook in Colab.
   * Downloaded the .ipynb file to my computer via **File -> Download -> .ipynb**.
7. **Local Repository Management**
   * Installed **GitHub Desktop** and signed in.
   * Cloned the **jupyter-exploration** repository to  
     C:\Users\jcast\Documents\GitHub\jupyter-exploration.
   * Copied the downloaded My\_First\_Notebook.ipynb into that folder.

‘ PS C:\Users\jcast\Documents\GitHub\jupyter-exploration> dir ‘  
‘ PS C:\Users\jcast\Documents\GitHub\jupyter-exploration> # Adjust the source filename if yours is different ‘  
‘ PS C:\Users\jcast\Documents\GitHub\jupyter-exploration> copy-item ‘   
‘ "C:\Users\jcast\Downloads\My\_First\_Notebook.ipynb" ‘

1. **Staging, Committing, and Pushing**
   * Opened PowerShell in the repo directory.
   * Ran:
   * git add My\_First\_Notebook.ipynb
   * git commit -m "Add initial Jupyter notebook"
   * git push origin main
   * Verified on GitHub that both the README and notebook appeared in the repository.

**b. What I Learned**

This lab introduced two foundational tools for data science and collaborative development: **GitHub** and **Jupyter Notebooks**.

* **Version Control with GitHub**  
  Learning to create a repository and make commits helped me understand how changes are tracked over time. Initializing with a README and pushing files via both the web interface and GitHub Desktop clarified the workflow between local and remote repositories. Encountering and resolving the authentication prompt reinforced my understanding of Git credential management.
* **Interactive Computing with Jupyter**  
  Working in Google Colab required no local installation, which demonstrated the accessibility of cloud-based notebooks. I experienced firsthand how Markdown cells allow clear documentation alongside executable code cells. Executing Python code inline illustrated how notebooks serve as both development environments and interactive reports.
* **Challenges and Solutions**
  + **PATH and Command Issues**: Initially, the jupyter notebook command wasn’t recognized because the Scripts folder was not on my system PATH. I learned to verify the Python installation path and adjust environment variables accordingly.
  + **File Placement**: I accidentally downloaded the notebook to a different folder, which caused git add to fail. Discovering how to locate files with dir C:\Users\jcast\Downloads\\*.ipynb and copying them into the repo improved my file-management skills.
  + **Authentication Pop-Up**: A blank browser window appeared when pushing via PowerShell. Installing and using GitHub Desktop bypassed this issue and showed an alternative Git workflow tool.

Overall, this lab strengthened my confidence in seamlessly moving between browser-based and local development, and it highlighted best practices for documenting work, tracking changes, and collaborating on code.