Andrew Badzioch
Kolby Boyd
Florentin Degbo
Natalia Solorzano
Jeralynn Sparks

Prof. Patricia McManus
ITAI 1378
29 January 2024

The Tools of the Trade: GitHub, Python & Jupyter Notebooks

I. What is GitHub?

A. **Definition:** GitHub is a platform for hosting and collaborating on coding projects. It uses Git, a version control system, to manage and track file changes. Developers may use GitHub to store their **code repositories**, track issues, and help collaboration among team members. GitHub is widely used in software development for managing and sharing code, coordinating the contributions of multiple developers, and tracking the progress of a project.

B. GitHub plays a significant role in collaborative software development by providing a platform where developers can share, store, and collaborate on coding projects:

1. **Code Hosting:** GitHub serves as a central **repository** for the storage of code, which makes it accessible to all team members.
2. **Version Control:** With Git integrated, GitHub allows developers to track **changes to their code over time**. This allows multiple contributors to work on the same codebase at the same time without interference from other developers.
3. **Branching and Merging:** GitHub allows developers to create **branches** to work on specific features or fixes independently. These branches can be **merged** back into the main codebase, which helps **parallel development**.
4. **Issue Tracking:** GitHub supplies tools for tracking **bugs**, feature requests, and other issues. Developers can create, assign, and prioritize issues, which encourages communication and coordination among members.
5. **Pull Requests:** Developers can propose **changes to the codebase** through pull requests. This allows others to review the proposed changes, supply feedback, and suggest modifications before integrating them into the main codebase.
6. **Collaboration:** GitHub offers features such as wikis, project boards, and discussions to further help collaboration and communication among team members.

II. What is a Jupyter Notebook?

A. **Definition:** Jupyter Notebooks are **interactive computing environments** that allow for the creation and sharing of documents holding live code, equations, visualizations, and narrative text. They support multiple programming languages, such as Python, R, and Julia. The notebook interface consists of **cells** where developers can write and execute code, as well as **markdown cells** that are used for documentation. Jupyter Notebooks are widely used in data science, research, education, and various other industries for prototyping, data analysis, and collaborative work. They encourage an exploratory approach to programming and data analysis.

B. Jupyter Notebooks play a crucial role in supporting interactive computing and documentation by supplying a platform where code, visualizations, and explanation text can seamlessly coexist.

1. **Interactive Computing:** Jupyter Notebooks allows users to execute code interactively within individual cells. This means they can write a code snippet, run it, and immediately see the output. This interactive nature encourages experimentation, debugging, and interactive development. Users can modify code on the fly, test ideas, and visualize results without leaving the notebook environment.

2. Documentation: Jupyter Notebooks combine code with **markdown cells**, allowing users to add **rich text** explanations, equations, images, and hyperlinks. This capability makes it easy to document the thought process, **methodology**, and interpretation of results alongside the code. Documentation within Jupyter Notebooks helps to convey insights, share knowledge, and provide context to the analysis or project.

III. Real World Applications (into a single workflow for efficient project development)

A. GitHub

Let's Imagine our group decides to build a football video game together. We use GitHub to help us work on the game more quickly.

- First, we created a place on GitHub to store all their game code. It is like a giant virtual box where they keep everything.
- Each member copies the game code from GitHub to their computer. It is like having a personal workspace.
- They use separate spaces called branches to work on various parts of the game, like improving how the ball moves. For example, one friend creates a branch called "ball-physics" to make the ball in the game move more realistically.
- After improving their own space, they share their work on GitHub.
- Members review each other's work on GitHub. We talk about what is good and what could be better. They also use automated tests to ensure the changes will not break anything in the game.
- When everyone agrees that the improvements are reasonable, we combine all the changes back into the main game. It is like returning all the upgraded parts to the big virtual box.
- We use GitHub to keep track of tasks, like adding cool player animations or improving the game's goals. It is like having a to-do list.
- To stay organized, they use a Project Board on GitHub. It helps them move tasks from "To Do" to "In Progress" to "Done" as they finish them.
- We write down valuable information, like how to set up the game on a computer, in a file called README.md. It is like a guidebook for anyone who wants to join and help.
- To discuss extensive ideas, we use GitHub Discussions. It is like a giant chat room where they can talk about what features would make the game even more exciting.
- With tools in GitHub, we ensure the game is always playable and has no problems.
- As we progress, we celebrate by creating unique versions of the game called releases on GitHub.

Finally, GitHub became the primary tool for working together on the football game. It helps them collaborate, keep track of changes, and make sure the game is always getting better.

B. Python

Python is the programming language for coding football games.

- First, we Create a virtual environment using Python to isolate the game's code.
- Virtual environment creation.
- Activation.
- Utilize a requirements.txt file to list and install necessary Python packages.
- Write the football game's logic and functionalities using Python, focusing on player movements, ball physics, and scoring mechanisms.

- Use Jupyter Notebooks to experiment with and test various Python functionalities related to the game.
- Commit changes made to the Python code to the GitHub repository, just like with the game code.
- When working on specific Python functionalities, create branches, make changes, and open pull requests on GitHub.
- Integrate continuous integration (CI) tools, such as GitHub Actions, to automatically check if the Python code works well with the game.
- Update the README.md file with instructions on using Python for certain aspects of the game or changing specific features.

C. Jupyter notebooks

Using Jupyter Notebooks for Experimentation:

- Integrate Jupyter Notebooks into the workflow for experimenting and testing various aspects of the football game.
- Install Jupyter using the following command.
- Develop Jupyter Notebooks for specific experiments and testing related to the football game.
- Use Jupyter Notebooks as a friendly interface to explore and run Python code snippets related to the game's logic and functionalities.
- Document and explain code experiments, ideas, and results directly within Jupyter Notebooks.
- Commit Jupyter Notebooks to the GitHub repository along with the rest of the project code.
- Collaborate with team members by sharing Jupyter Notebooks, allowing them to run and experiment with the code.
- Use GitHub for version control of Jupyter Notebooks, keeping track of changes and improvements.
- Leverage Jupyter Notebooks to enhance understanding and communication among team members, especially for complex algorithms or experiments.
- Integrate Jupyter Notebooks into the continuous integration (CI) process to ensure that changes work seamlessly within the project.

IV. Citations & Images

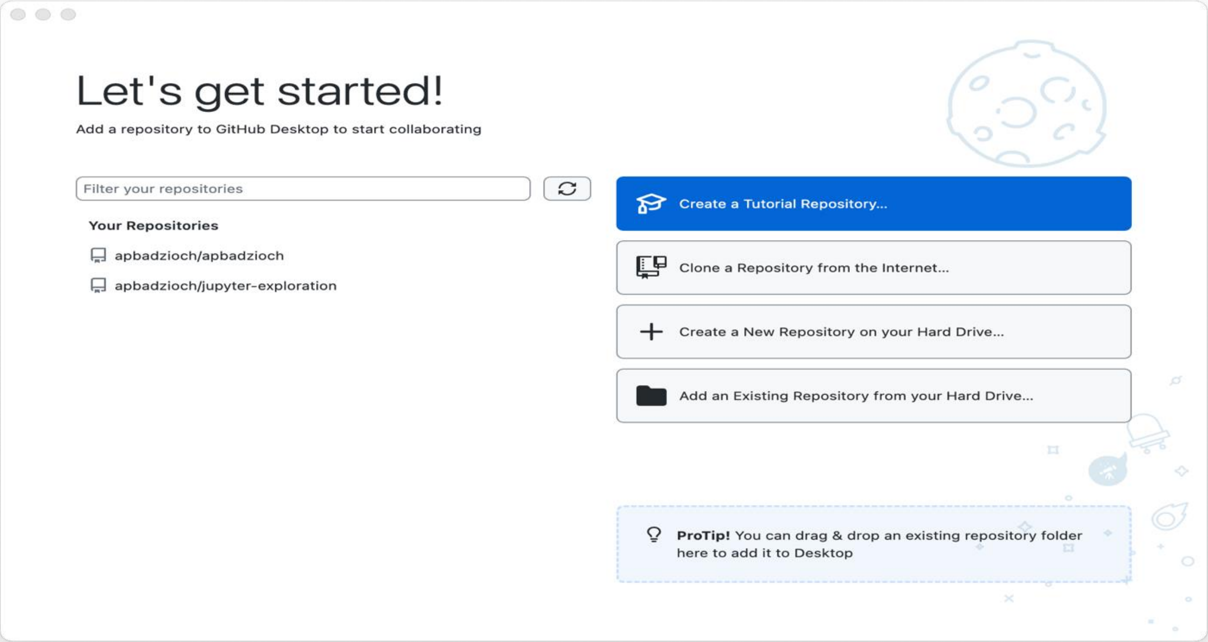
A. Sources

Integrating Jupyter notebooks, version control, and github classroom. YouTube. (2020, July 7). https://youtu.be/VscM6WEIrNE?si=Q4hFoaDocv1HFfrU

Jupyter. (n.d.). *How to run git clone HTTPS://.... in Jupyter Notebook · issue #3395 · Jupyter/notebook*. GitHub. https://github.com/jupyter/notebook/issues/3395

Let's build from here. GitHub. (n.d.). https://github.com/

B. GitHub



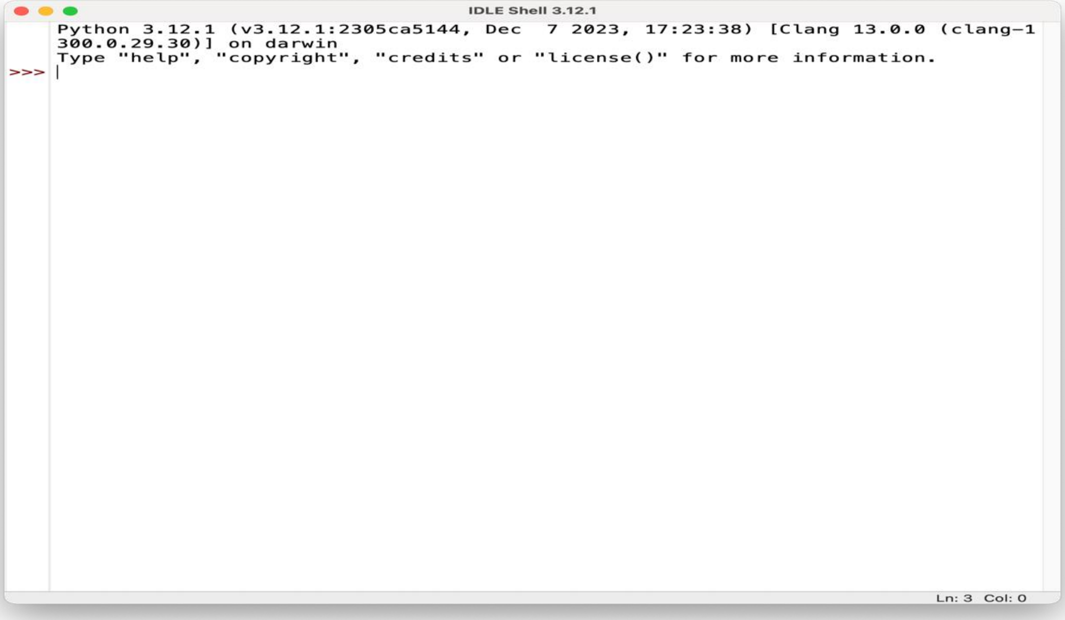
Let's get started!

Add a repository to GitHub Desktop to start collaborating

Filter your repositories
Your Repositories
apbadzioch/apbadzioch
apbadzioch/jupyter-exploration
Create a Tutorial Repository...
Clone a Repository from the Internet...
Create a New Repository on your Hard Drive...
Add an Existing Repository from your Hard Drive...

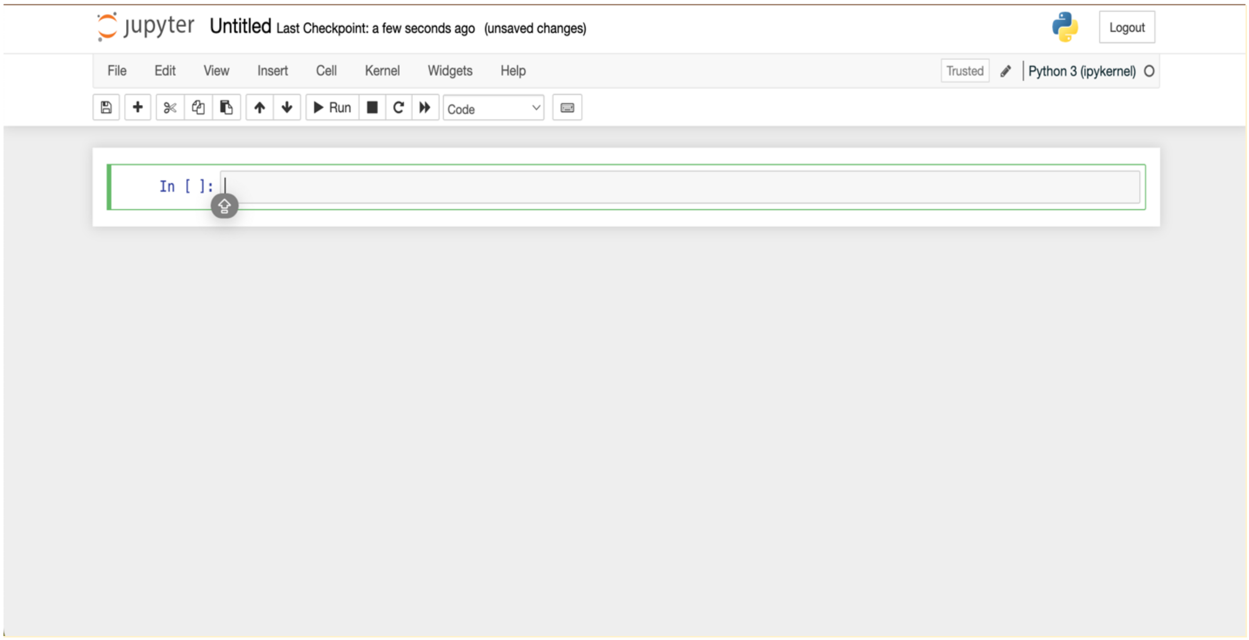
ProTip! You can drag & drop an existing repository folder here to add it to Desktop

C. Python



```
Python 3.12.1 (v3.12.1:2305ca5144, Dec  7 2023, 17:23:38) [Clang 13.0.0 (clang-1
300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

D. Jupyter Notebooks


jupyter
Untitled Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
In [ ]: