TOOL

Upload Jupyter Notebooks to GitHub Repository

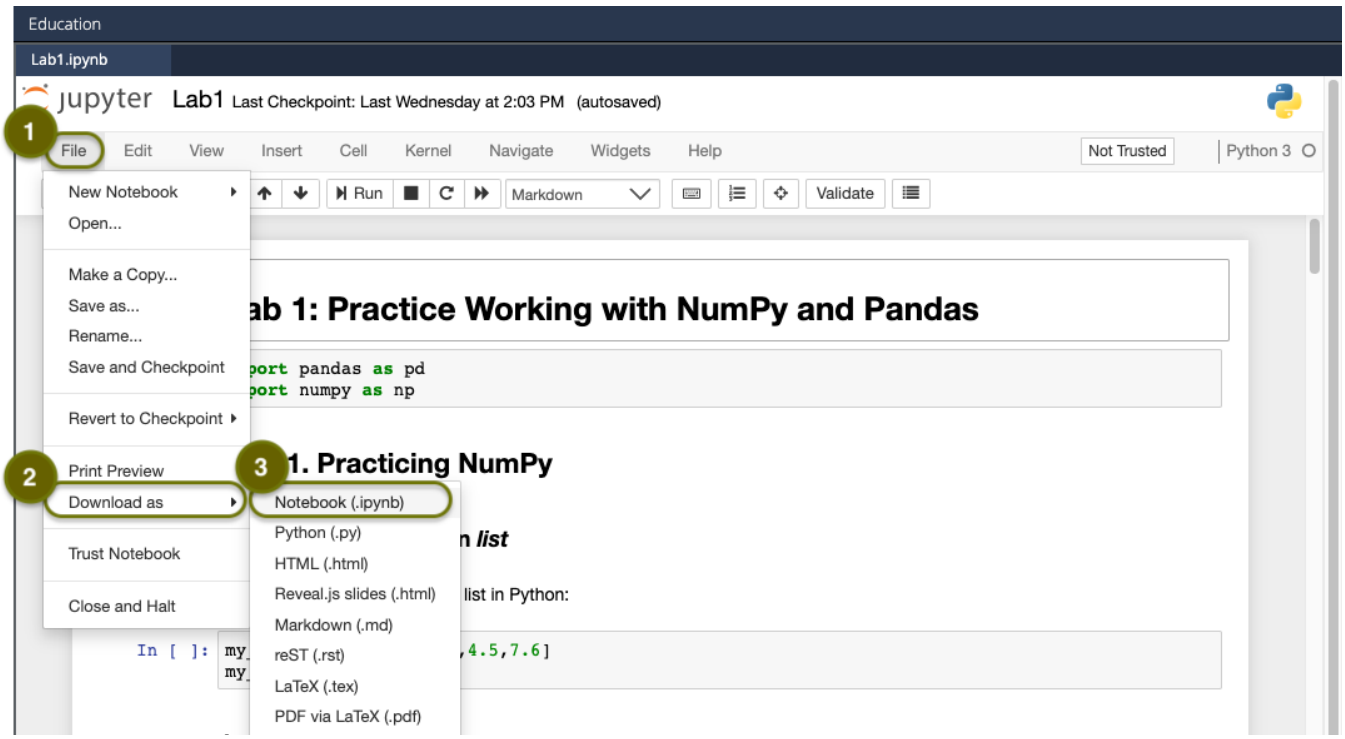
Overview

You may have reason to store and view your Jupyter Notebook assignment files (.ipynb) outside the eCornell Codio environment. Perhaps you want to create a portfolio or revisit assignments after course completion. This can be accomplished using GitHub, a code hosting platform for version control and collaboration. GitHub is a free tool that provides code hosting in a private online repository (A directory or storage space where your projects can live.)

Use the following steps to learn how to download assignment files and then upload them to your own private code repository.

Download the Codio Assignment File

1. In Canvas, in the Jupyter Notebook assignment toolbar, Click File > Download as > Notebook(.ipynb)



2 **Optional:** Once downloaded, rename the file on your PC if you want a specific filename prior to uploading to GitHub.

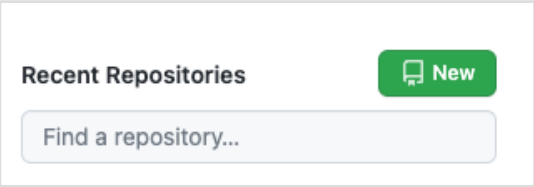
3 **Note:** If you want to capture the notebook's output, you should run it, save it after the process finishes, and download the file.

Create a New Repository on GitHub

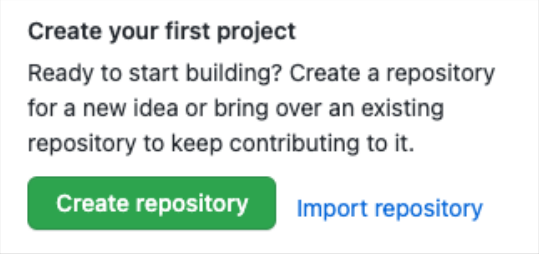
1 If you already have a personal GitHub account, sign in to github.com.

   a. If needed, you can sign up for a free personal GitHub account at github.com. Click **Sign Up** in the upper right corner of the GitHub home page.

2 On the GitHub home page, click **New.**



3 Or click **Create Repository** if you're creating your first repository with a new GitHub account.



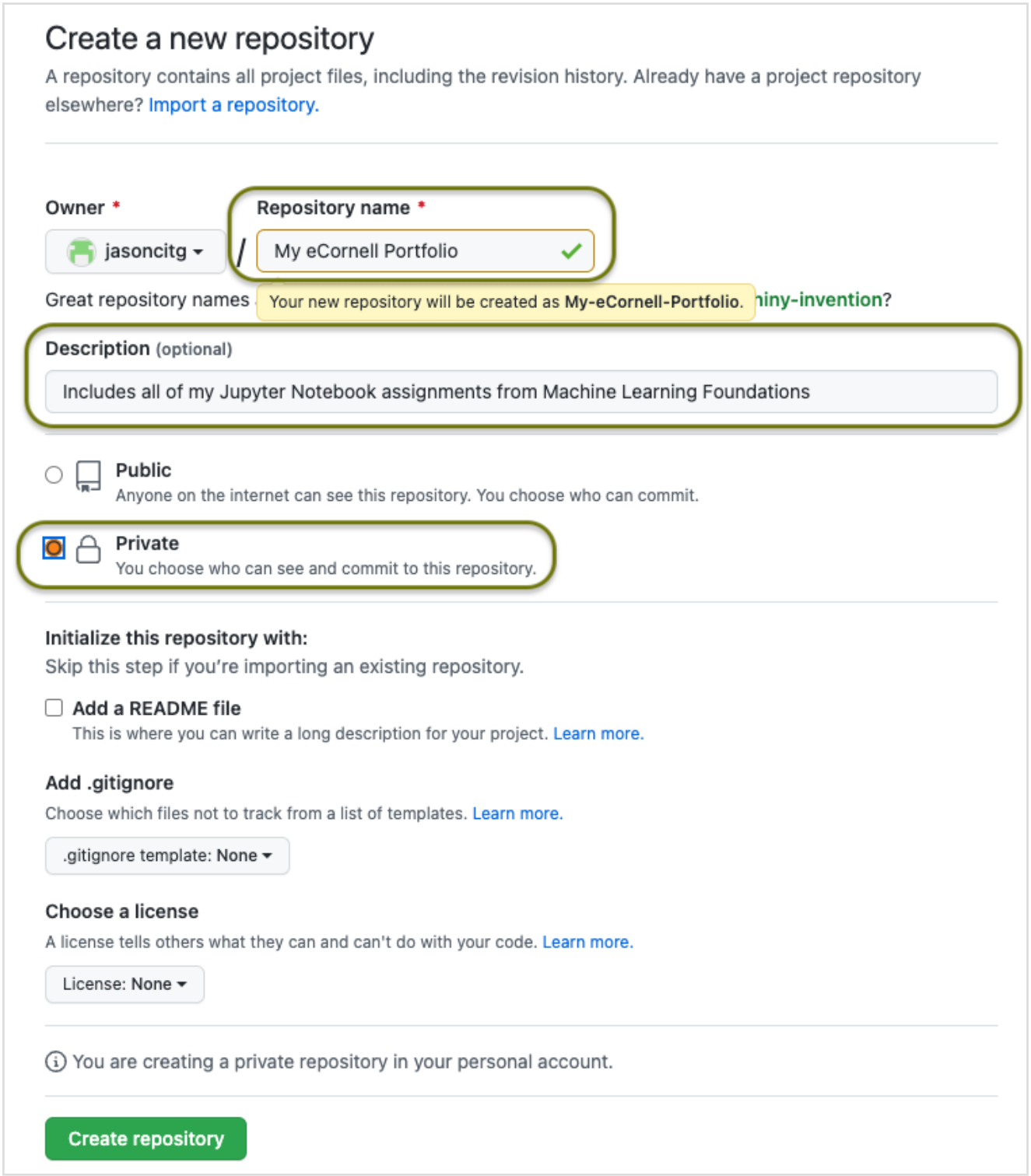
4 Enter a name for the new repository in the **Repository name** field. (ex. My Cornell Portfolio)

5 **Optional:** Enter a description for the portfolio in the description field.

Important: You must set the repository to Private. Not setting it to private could violate Cornell University's Code of Academic Integrity. Before posting, please review Cornell's Code of Academic Integrity and eCornell's policy regarding plagiarism (the presentation of someone else's work as your own without source credit). Failure to set repositories to private could violate this Code and subject you to proceedings under the Code.

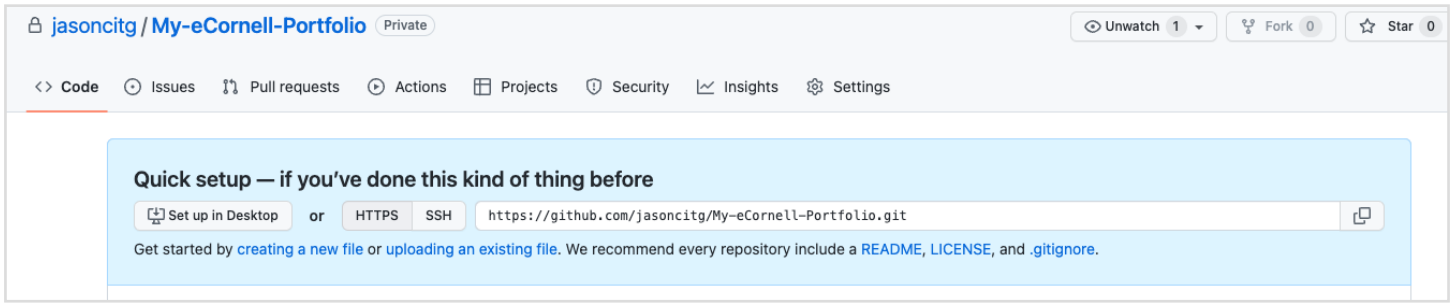
6 **Optional:** Check the **Add a README** file box if you wish to write a longer description or include file notes.

7 Click **Create repository**.



Upload the Jupyter Notebook File (.ipynb)

GitHub will open the repository page immediately after the repository is created. It's time to upload your first Jupyter Notebook file.



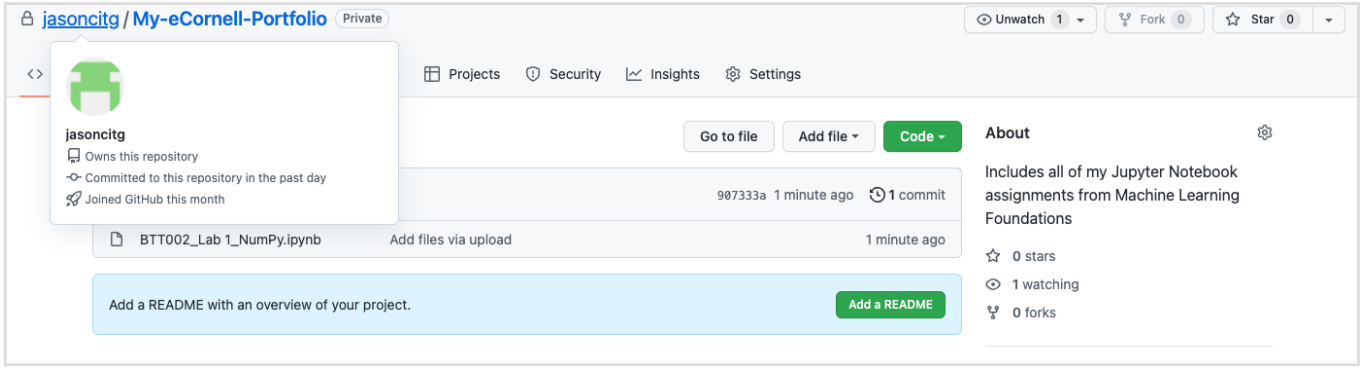
Note: These instructions are for users of the GitHub file upload GUI interface. Please note there are multiple ways to upload a file to GitHub depending on which version (ex., desktop app vs web version.) Visit the GitHub support site to learn about additional file upload methods.

If uploading to a new repository:

1. Click **uploading an existing file** in the top blue section of the upload page.
2. Browse through your directory, click the file to select it (example filename: Lab1.ipynb), and click Open.

If uploading to an existing repository:

1. Open the desired repository, Click **Add file** -> **Upload files**.
2. Click Commit changes. You should receive an upload confirmation on the next page. Please see the example below.



View the Uploaded Jupyter Notebook File

Click the uploaded file link in the main files section of the repository. The code contained in the uploaded file can be viewed on Github in both source and rendered blobs.

- Click on the desired filename from the Code tab.
- Toggle between source and rendered blobs with the small icons in the GitHub file toolbar.

Note: To execute the file, you must download it to a local machine to run via a terminal window or server-client application (ex., Jupyter Notebook app.)

Additional Resources

If you're new to GitHub, it may be helpful to learn more about Git and how it differs from GitHub.

Git	GitHub
Git is a version control system that must be installed on your computer. It is software.	Git is a version control system that must be installed on your computer. It is software.
Ideal for solo users.	Ideal for solo users.
Git is a command-line tool.	Git is a command-line tool.

The following common commands will help you get started in Git:

git init	Turns a directory into an empty Git repository.
git add	Adds files into a staging area for Git.
git commit	Record the changes made to a file to a local repository.
git status	Returns the current state of the selected repository.
git config	Allows the user to assign settings and configurations.
git branch	Determine what branch the local repository is on, add a new branch, or delete a branch.
get checkout	Switch branches
git merge	Integrate branches
git remote	Connect a local repository with a remote repository.
git clone	Create a local working copy of an existing remote repository
git pull	Get the latest version of a repository.
git push	Sends local commits to the remote repository.
git stash	Save changes made when they're not in a state to commit them to a repository.
git log	Show the chronological commit history for a repository.

Refer to this Git Cheat Sheet for additional terminal commands.