

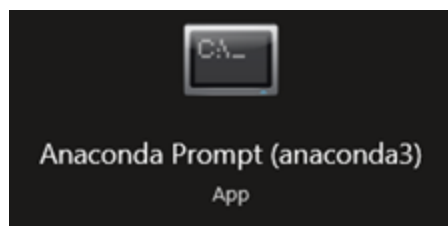
In this guide, we will set up Python 3 along with Jupyter graphical script editing environment on your local computer (assumed to be Windows OS)

Step 1: Install Anaconda with Python 3

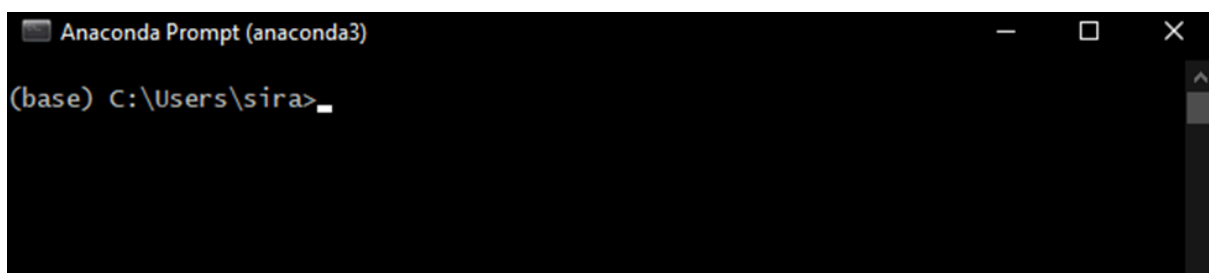
- Go to <https://www.anaconda.com/> and download the installer for Windows (current version should come with Python 3.9)
- This will set up a **virtual environment** with its own Python so that you won't accidentally change Python that came with the Windows OS or other software

Step 2: Launch Anaconda Prompt on your computer

- Search for Anaconda in your search bar or look in your program list (the bottom left button with Windows logo).



- Once you launched Anaconda Prompt, you should see a command line with **(base)** at the front. This signifies that you are in a new virtual environment.



- To check that you are using Anaconda's Python, and not the OS's, type **where python**. If your computer has multiple Python installed, all will be shown. But the top one should be located in your Anaconda3 folder.

```
(base) C:\Users\Sira>where python
C:\Users\Sira\anaconda3\python.exe
C:\Python38\python.exe
C:\Users\Sira\AppData\Local\Microsoft\WindowsApps\python.exe
C:\Python27\python.exe
```

Step 3: Managing Python library

- To install/uninstall/upgrade Python library, we use the **pip** command.
- **pip list** will show the list of installed libraries

```
(base) C:\Users\Sira>pip list
Package           Version
-----
aiohttp           3.8.1
aiosignal         1.2.0
alabaster          0.7.12
```


- **pip install** followed by library name will install that library, if it is not already installed
 - **pip install --upgrade** followed by library name will update the library, as well as other dependencies, to the latest version
 - To install the Jupyter library for editing Python code, run **pip install jupyter**
 - Let's pre-install other libraries that we will be using in this course

pip install numpy scipy pandas scikit-learn umap-learn matplotlib seaborn statsmodels

- Specific library version can be specified with == (if you ran into compatibility issue) as follows: **pip install jupyter==1.0**
- **pip uninstall** followed by library name will uninstall the library

Step 4: Setting up Jupyter environment

- From Anaconda prompt, you can launch Jupyter graphical editor with **jupyter notebook**.
- However, you may want to set up the folder where Jupyter will keep the script files (this process will allow you to set up a remote server with Jupyter in the future).
 - First, run **jupyter notebook generate-config**
 - This will create a file name **jupyter_notebook_config.py** in your user home directory (usually **C:\Users*user name*\jupyter\jupyter_notebook_config.py**)
 - Open this file in text editor, like Notepad

 jupyter_notebook_config.py - Notepad

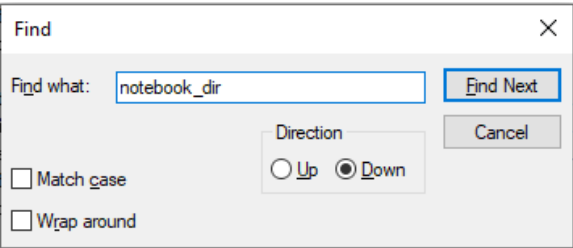
File Edit Format View Help

Configuration file for jupyter-notebook.

```
#-----
# Application(SingletonConfigurable) configuration
#-----
## This is an application.
```

```
## The date format used by logging formatters for %(asctime)s
# Default: '%Y-%m-%d %H:%M:%S'
# c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'
```

- Search for the pattern **notebook_dir**. Edit the path to where you want to keep your Python scripts.



```
# Default: ''
# c.NotebookApp.notebook_dir = ''

## Get the path to the notebook directory.
# This is the directory where the notebook will be
# opened.
# Default: ''
# c.NotebookApp.notebook_dir = ''

## Dict of Python modules to load as notebook server extensions
# be used to enable and disable the loading of the extensions
# will be loaded in alphabetical order.
# Default: {}
# c.NotebookApp.nbserver_extensions = {}

## The directory to use for notebooks and kernels.
# Default: ''
# c.NotebookApp.notebook_dir = 'C:\\Users\\Sira\\Dropbox'
```

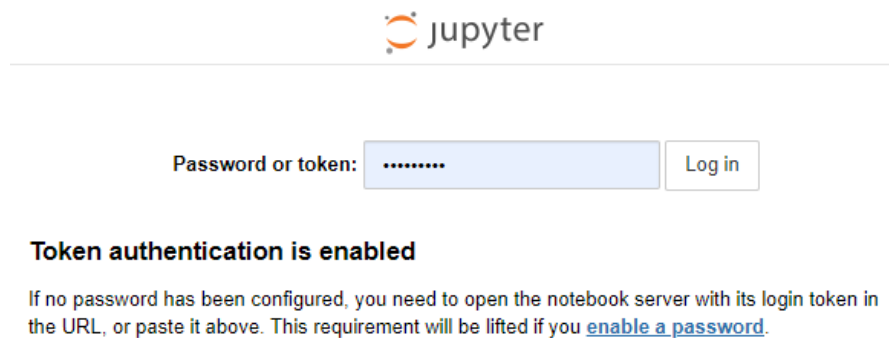
Step 5: Launching your first Python notebook

- Run **jupyter notebook**. You should see something like this on Anaconda Prompt

```
[I 10:17:58.828 NotebookApp] Serving notebooks from local directory: C:\Users\Sira\Dropbox
[I 10:17:58.828 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 10:17:58.829 NotebookApp] http://localhost:8888/?token=f7716b28a7d0d54d114109a27ccb32f24195eb04cf5949ba
[I 10:17:58.829 NotebookApp] or http://127.0.0.1:8888/?token=f7716b28a7d0d54d114109a27ccb32f24195eb04cf5949ba
[I 10:17:58.830 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 10:17:58.918 NotebookApp]

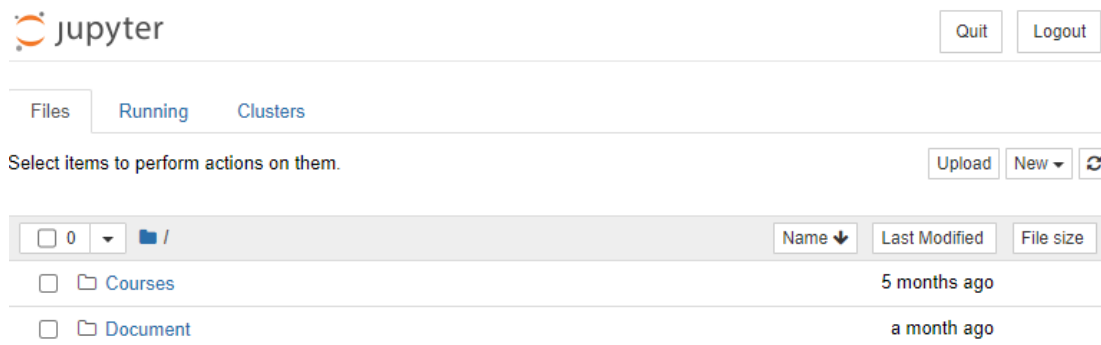
To access the notebook, open this file in a browser:
file:///C:/Users/Sira/AppData/Roaming/jupyter/runtime/nbserver-7752-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=f7716b28a7d0d54d114109a27ccb32f24195eb04cf5949ba
or http://127.0.0.1:8888/?token=f7716b28a7d0d54d114109a27ccb32f24195eb04cf5949ba
```

- A web browser should also be open with a page that looks like this



The image shows the Jupyter login page. At the top is the Jupyter logo. Below it is a form with a label "Password or token:" followed by a text input field containing seven dots. To the right of the input field is a "Log in" button. Below the form, the text "Token authentication is enabled" is displayed. Underneath this, a paragraph explains that if no password is configured, the user needs to use the token in the URL or paste it above, and that this requirement will be lifted if they enable a password.

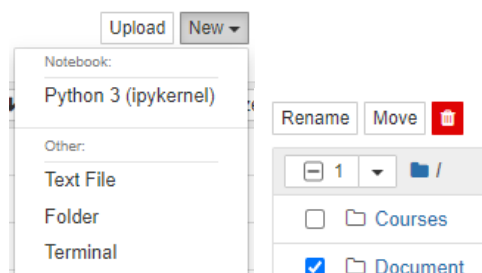
- If you follow the instruction to use token or password to log in correctly, you will be brought to a new page that looks like this. Note that the files and folders that show up will differ depending on the location you set in **notebook_dir**.



The image shows the Jupyter file browser interface. At the top is the Jupyter logo and "Quit" and "Logout" buttons. Below the logo are tabs for "Files", "Running", and "Clusters". A message says "Select items to perform actions on them." with "Upload", "New", and "Refresh" buttons. Below this is a table of files and folders. The table has columns for "Name", "Last Modified", and "File size". The first row shows a folder named "Courses" with a last modified time of "5 months ago". The second row shows a folder named "Document" with a last modified time of "a month ago".

	Name	Last Modified	File size
<input type="checkbox"/>	Courses	5 months ago	
<input type="checkbox"/>	Document	a month ago	

- You can use the **New** button to create new folder, text file, or Python notebook. You can rename or delete file and folder by checking the boxes as shown below.



The image shows the Jupyter file browser interface with the "New" button clicked, opening a dropdown menu. The menu has two sections: "Notebook:" and "Other:". Under "Notebook:", there is a button for "Python 3 (ipykernel)". Under "Other:", there are buttons for "Text File", "Folder", and "Terminal". To the right of the menu, there are buttons for "Rename", "Move", and a delete icon. Below these buttons is a table of files and folders, similar to the one in the previous image, but with the "Document" folder selected (checked).

	Name	Last Modified	File size
<input type="checkbox"/>	Courses	5 months ago	
<input checked="" type="checkbox"/>	Document	a month ago	

- Copy **3000788_Fall2022_L21_python_102722.ipynb** and **3000788_Fall2022_L21_python-more_102722.ipynb** to the folder you set in **notebook_dir**. Then open them in Jupyter. You should see something like this at the top of the page.

Welcome to Python programming via Jupyter interface

This is called a **Markdown** panel

```
In [ ]: x = 5
        print(x * 20 + x - 3)
        print('This is called a Code panel')
```

Markdown panel helps other understand your code

And can be very **fancy**

See more: <https://medium.com/ibm-data-science-experience/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed>

Demo 1: Gut microbiota

Step 6: Let's learn the language

- Please watch the video in which I explain the codes and follow through!