



# Introduction to Jupyter Notebook

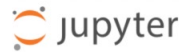
# Jupyter Notebook

Jupyter notebook is a document which contain both computer code (e.g. Python) and rich text elements (text, equations, visualizations, etc...).

Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.

# Opening Jupyter Notebook

Open the Jupyter notebook app.(Or on the terminal, type in “jupyter notebook”). This will start up Jupyter and your default browser should start (or open a new tab) to the following URL: <http://localhost:8888/tree>



Quit

Logout

Files

Running

Clusters

Upload

New ▾



Select items to perform actions on them.

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📁 Applications		8 days ago	
<input type="checkbox"/>	📁 Desktop		3 days ago	
<input type="checkbox"/>	📁 Documents		4 days ago	
<input type="checkbox"/>	📁 Downloads		7 hours ago	

# Server

In addition to the browser interface shown in the previous slide, another window should open that looks like:

```
134540 — jupyter-notebook > python — 110x26
Last login: Thu Jan 6 21:01:19 on ttys000
[(base) 134540@L-FVFGH23YQ6L3-134540 ~ % jupyter notebook ]
[I 2022-01-14 19:29:35.998 LabApp] JupyterLab extension loaded from /Users/134540/opt/anaconda3/lib/python3.9/
site-packages/jupyterlab
[I 2022-01-14 19:29:35.998 LabApp] JupyterLab application directory is /Users/134540/opt/anaconda3/share/jupyt
er/lab
[I 19:29:36.001 NotebookApp] Serving notebooks from local directory: /Users/134540
[I 19:29:36.001 NotebookApp] Jupyter Notebook 6.4.5 is running at:
[I 19:29:36.001 NotebookApp] http://localhost:8888/?token=d2aa76440ba4b7fd6bcb895d9c04d28fc4f1339e92fe82f3
[I 19:29:36.001 NotebookApp] or http://127.0.0.1:8888/?token=d2aa76440ba4b7fd6bcb895d9c04d28fc4f1339e92fe82f3
[I 19:29:36.001 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confir
mation).
[C 19:29:36.005 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/134540/Library/Jupyter/runtime/nbserver-31636-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=d2aa76440ba4b7fd6bcb895d9c04d28fc4f1339e92fe82f3
or http://127.0.0.1:8888/?token=d2aa76440ba4b7fd6bcb895d9c04d28fc4f1339e92fe82f3
[I 19:33:54.140 NotebookApp] Creating new notebook in
[I 19:33:54.159 NotebookApp] Writing notebook-signing key to /Users/134540/Library/Jupyter/notebook_secret
[I 19:33:55.063 NotebookApp] Kernel started: c46e37dc-2376-4de2-8af1-96a0a15eb34f, name: python3
[I 19:35:56.274 NotebookApp] Saving file at /Untitled.ipynb
```

This is a “notebook server” that is running on your machine - it basically handles all of the communication between your browser and your machine.

**DO NOT close this window! Closing it will close the server causing your notebook to no longer work!**

# Zip Folder

A Jupyter notebook file has the extension .ipynb. Double clicking on it **will not** open the notebook! You must open it from the browser interface.

**How to open a .ipynb notebook(next three slides):**

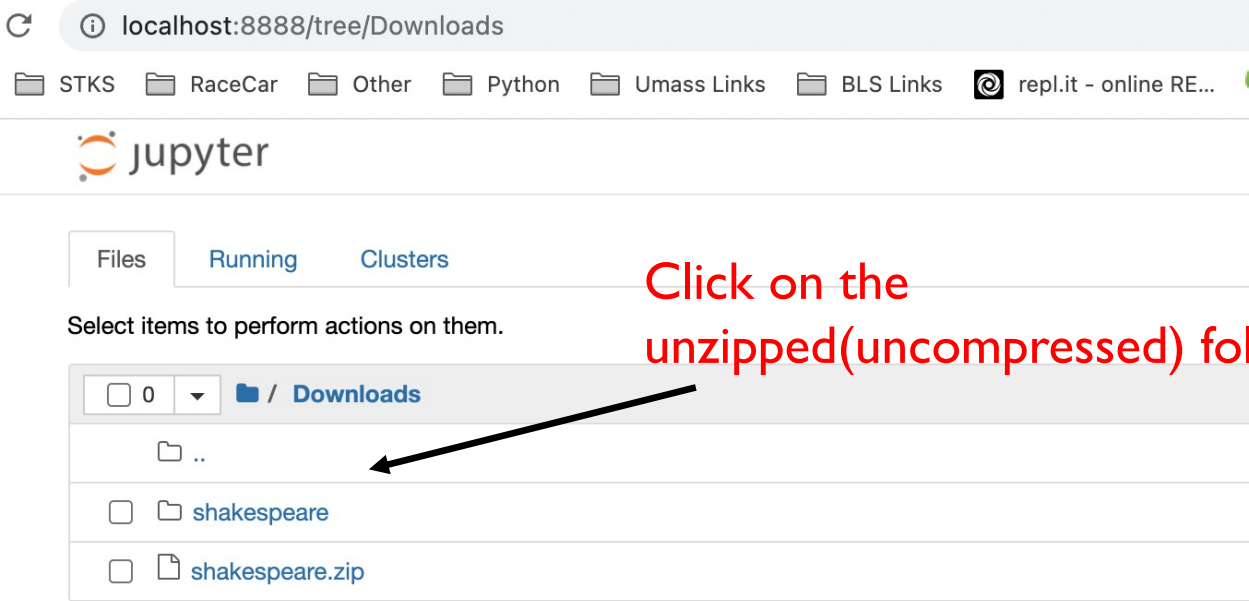
For our class, Jupyter notebook assignments will contain Jupyter notebook .ipynb file along with other files such as text files or images bundled in a zip(compressed) folder.

Download the zip folder. Uncompress it by either double click on it(on Mac) or right click and “extract all”(on Windows).



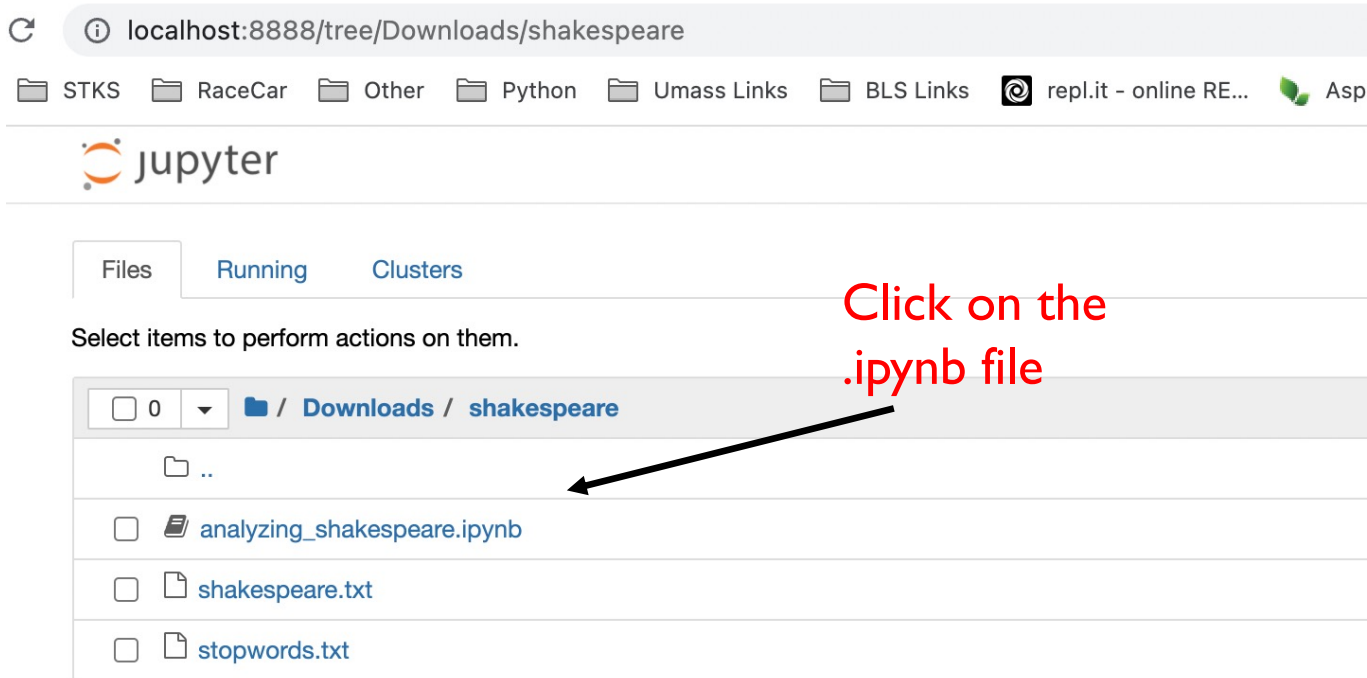
# Opening a .ipynb file

To open the Jupyter notebook, navigate using the browser interface to the folder. Then click on the unzipped folder:



# Opening a .ipynb file

Then click on the .ipynb file.



The screenshot shows a web browser window with the address bar displaying 'localhost:8888/tree/Downloads/shakespeare'. The browser's tab bar includes 'STKS', 'RaceCar', 'Other', 'Python', 'Umass Links', 'BLS Links', 'repl.it - online RE...', and 'Asp'. The JupyterLab interface features the 'jupyter' logo and three tabs: 'Files', 'Running', and 'Clusters'. Below the tabs, a message reads 'Select items to perform actions on them.' The file browser shows the path '/ Downloads / shakespeare' and a list of files: '..' (a folder), 'analyzing\_shakespeare.ipynb' (a file with a notebook icon), 'shakespeare.txt' (a text file), and 'stopwords.txt' (a text file). A red annotation 'Click on the .ipynb file' with a black arrow points to the 'analyzing\_shakespeare.ipynb' file.

# Cells

There are two kinds of cells in a Jupyter notebook.

**1) Markdown cells** contain text, images, equations, etc..

```
This is a markdown cell. It typically contains text that documents or explains your code.
```

**2) Code cells** contain code that can be executed.

```
In [ ]: x = 10|
```

To run a cell and execute the code in the cell, press: **Shift + Enter**.

```
In [1]: x = 10
```

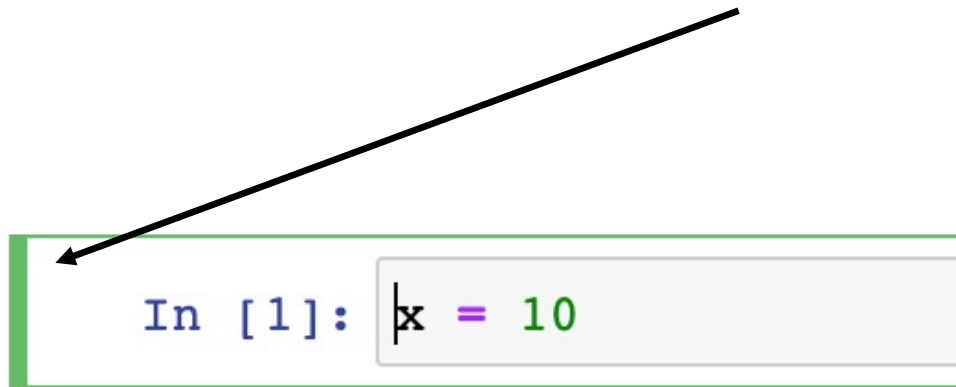


# Edit Mode

A notebook has two modes of operation: command and edit modes.

**Edit Mode (Click on textbox in the cell to begin editing the cell)**

Edit mode allows you to edit that cell by writing either code or markdown. Your selected cell will be surrounded by a **green** border when you are in edit mode.



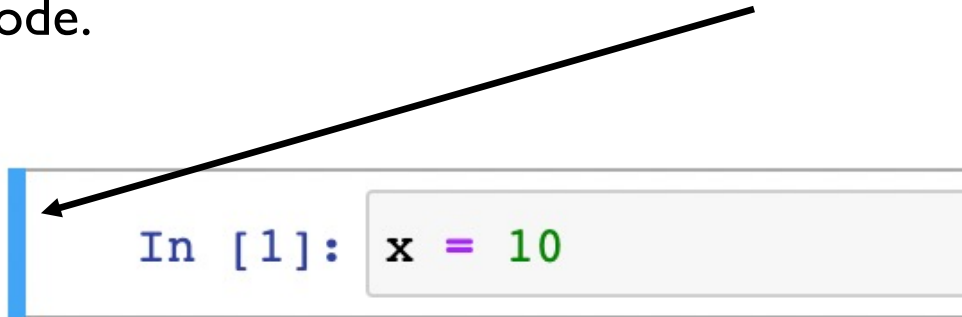
```
In [1]: x = 10
```

# Command Mode

## Command Mode (Press <ESC> to activate)

When in command mode, you can use keyboard shortcuts to create/delete/cut/paste notebook cells, and to change a cell's type between code and markdown modes.

Your selected cell will be surrounded by a **blue** border when you are in command mode.



```
In [1]: x = 10
```

# Command Mode

When in command mode, you can use keyboard shortcuts to create/delete/cut/paste notebook cells, and to change a cell's type between code and markdown modes.

- create a new cell above the current cell: `a`
- create a new cell below the current cell: `b`
- delete the current cell: `dd`
- restart the notebook kernel (kill all executions and erase all defined variables): `00`
- change the current cell's type to "Code": `y`
- change the current cell's type to "Markdown": `m`

# Cells

You can write code across multiple cells and execute them one at a time. (press **Shift + Enter**)

```
In [1]: x = 10
```

Notebook keeps track  
of order of executed cells.

```
In [2]: y = x + 2  
z = "hello"  
print(y, z)
```

```
12 hello
```

Printing output

```
In [3]: z = "hi"  
x + y
```

Last expression is returned  
in an output cell.

```
Out[3]: 22
```

Note: Notice that the notebook “knows” about its variables across its cells. This doesn’t just work from top to bottom. You can define  $z = 2$  in the third cell, and then execute code that references  $z$  in the first cell. What really matters is the **order** in which the cells are executed.

# Cells

```
In [4]: def twice(x):  
        return 2 * x
```

```
In [5]: print(twice(10))
```

20

← Instead of printing returned value as we did in Replit,  
we can get the returned value from the output cell.

```
In [6]: twice(10)
```

Out[6]: 20

# References

1) Python Like You Mean it. Retrieved from:

<http://www.pythonlikeyoumeanit.com/index.html>