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…).

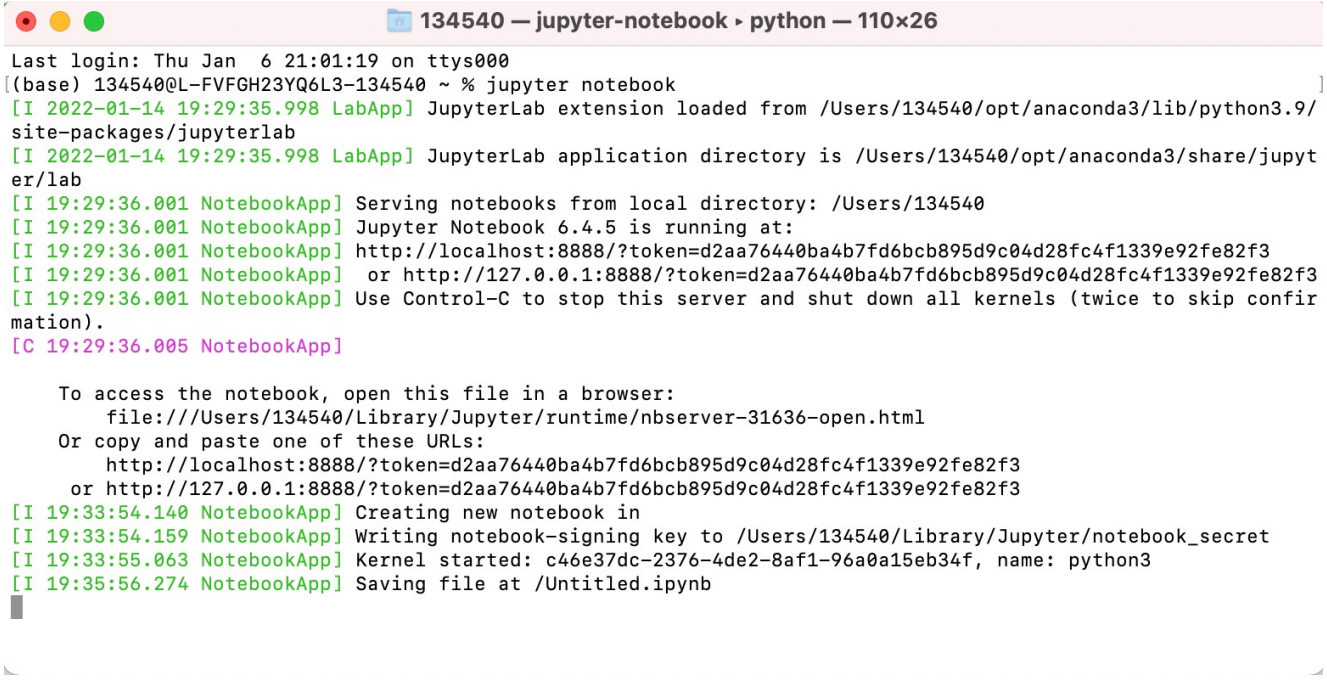
Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.

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

Server

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



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!

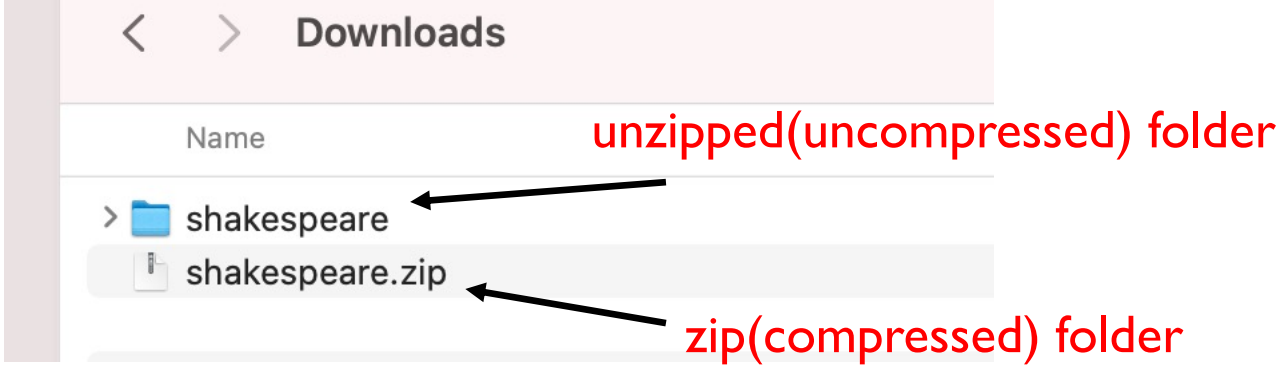
Zip Folder

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):

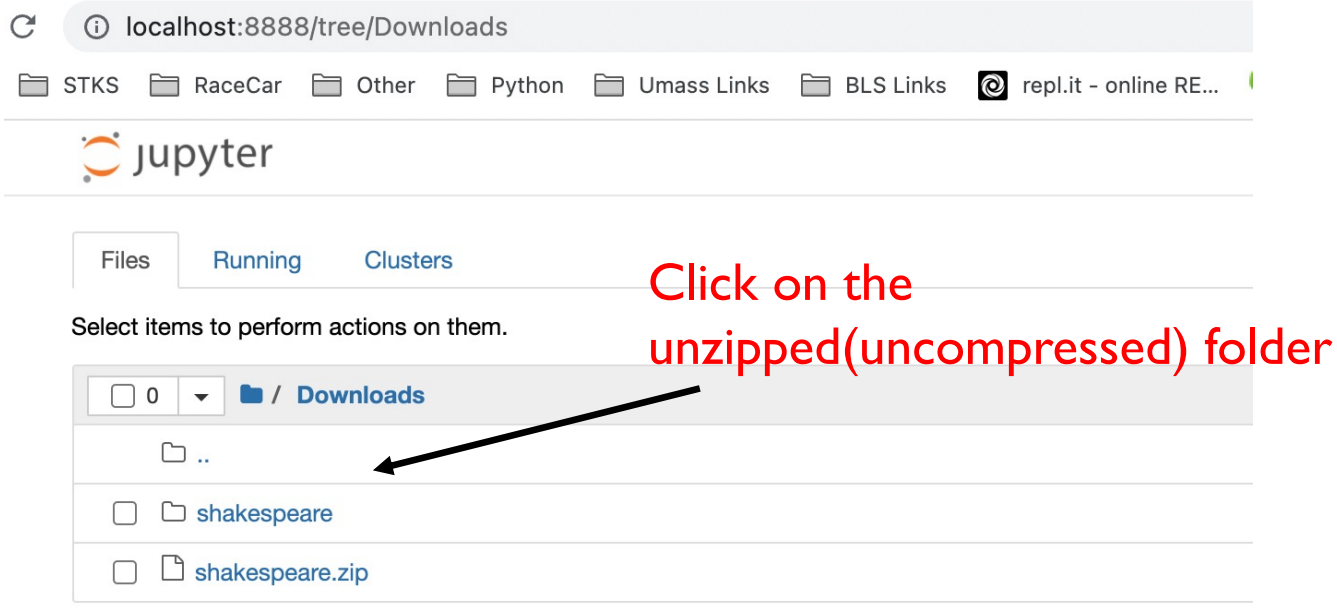
For our class, Jupyter notebook assignments will contain Jupyter notebook .ipynb file along with other files such as text files or images bundled in a zip(compressed) folder.

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



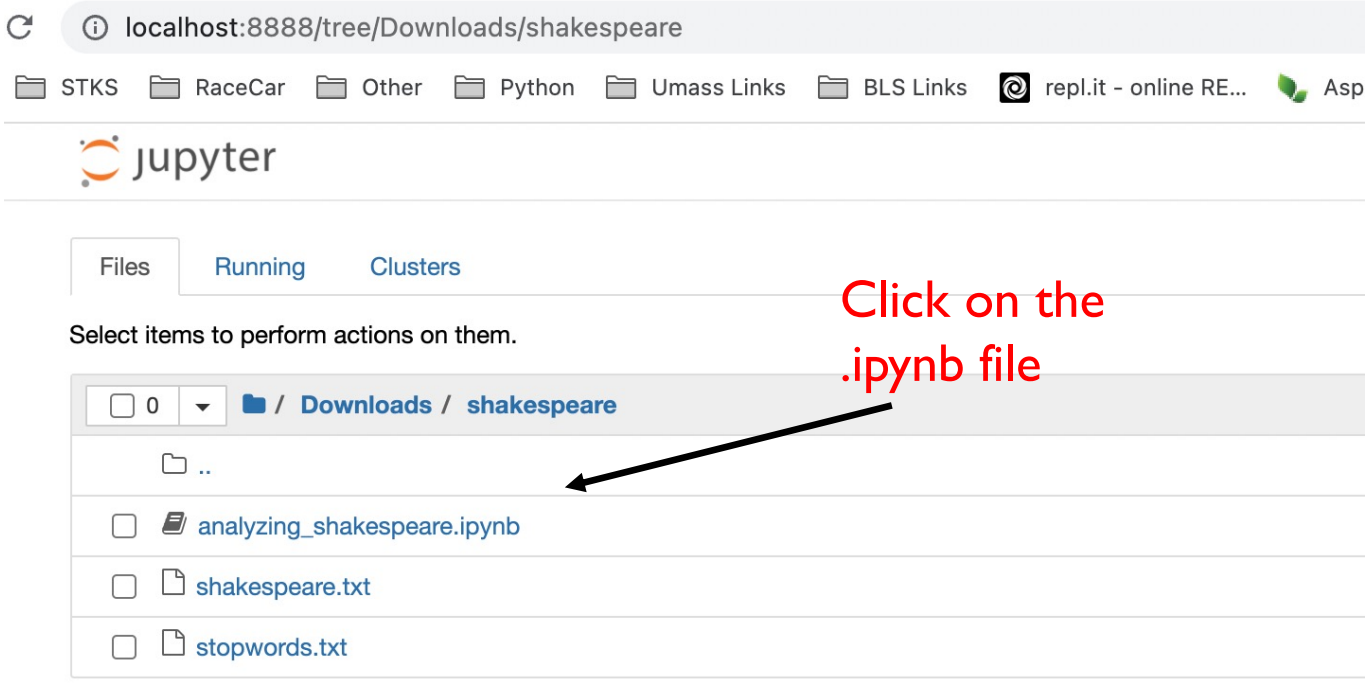
Opening a .ipynb file

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.



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.

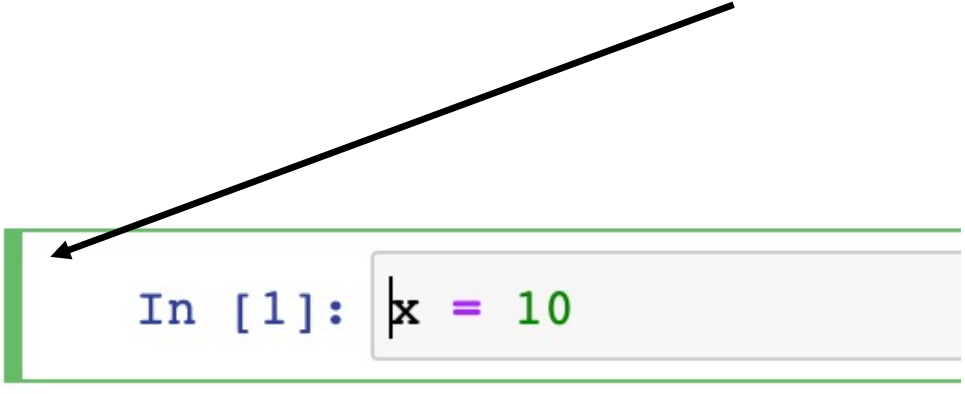
```
In [1]: x = 10
```

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.

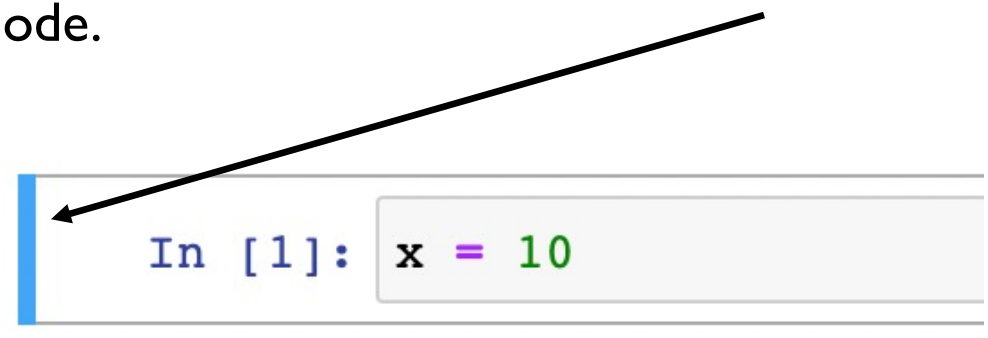


Command Mode

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.



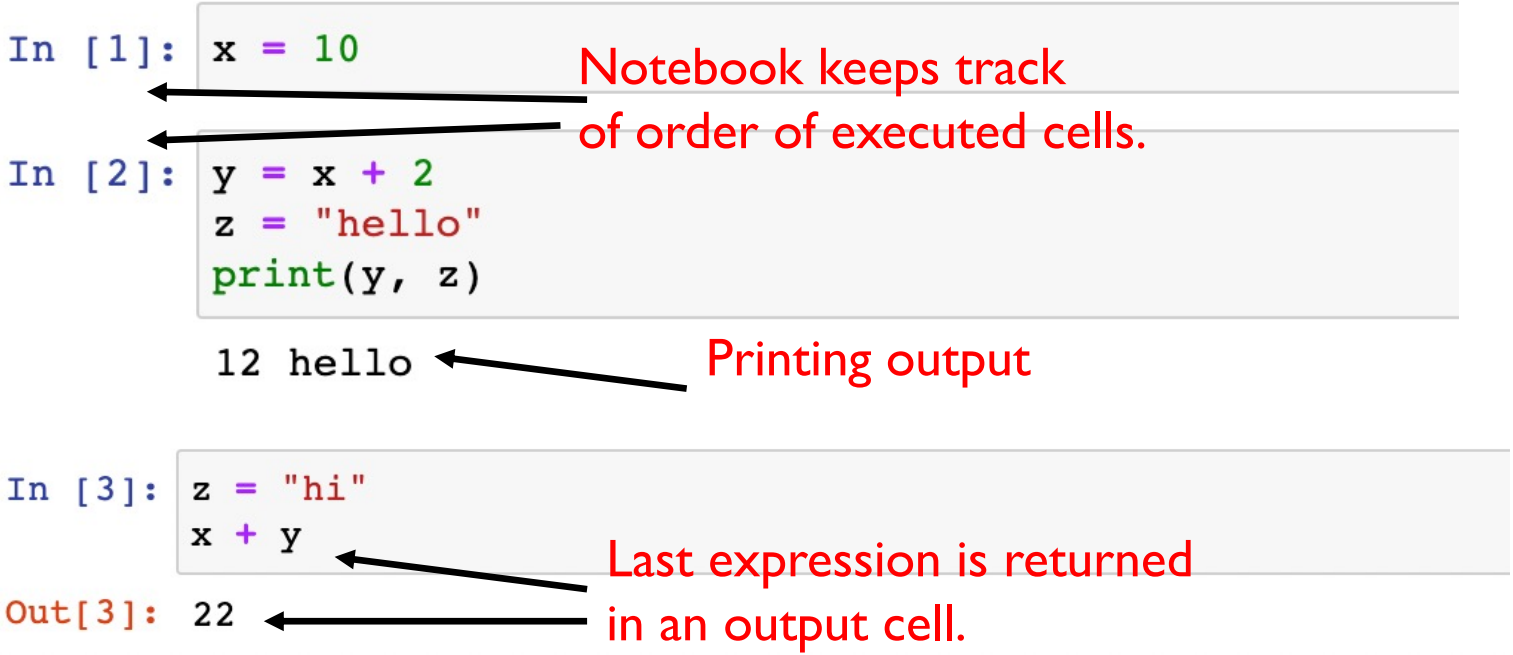
Command Mode

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)



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

In [6]: twice(10)
Out[6]: 20
```

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

References

1) Python Like You Mean it. Retrieved from: http://www.pythonlikeyoumeanit.com/index.html