Information, Computation, Communication

Learning Python

Logo downloaded from https://en.wikipedia.org/wiki/Project_Jupyter

Agenda

- What is Jupyter?
- Jupyter Notebook
  - Requirements and installation
  - Code cells
  - Dropdown menus
  - Closing notebooks
  - Text cells and Markdown
- Jupyter Notebook and VS Code

What is Jupyter?

Project Jupyter

- Link: jupyter.org
- Open-source project for interactive computing
- Used in data science, machine learning, scientific computing, etc.
- Three dimensions of Jupyter [1]:
  - Interactive computing
  - Computational narratives
  - The idea that Jupyter is more than software
- The core of Jupyter is the Jupyter Notebook

[1] Jupyter: Thinking and Storytelling with Code and Data. computer.org

1. Interactive Computing

Interactive computation is a persistent computer program that runs with a "human in the loop", where the primary mode of interaction is through the same human iteratively writing/running blocks of code and looking at the results. [1]

- Program has a working memory, which records the results of previous computations to make them available for subsequent ones
- User provides input to the program by writing code instead of using graphical or touch interfaces
- A single human is both the user and the author of the program

[1] Jupyter: Thinking and Storytelling with Code and Data. computer.org

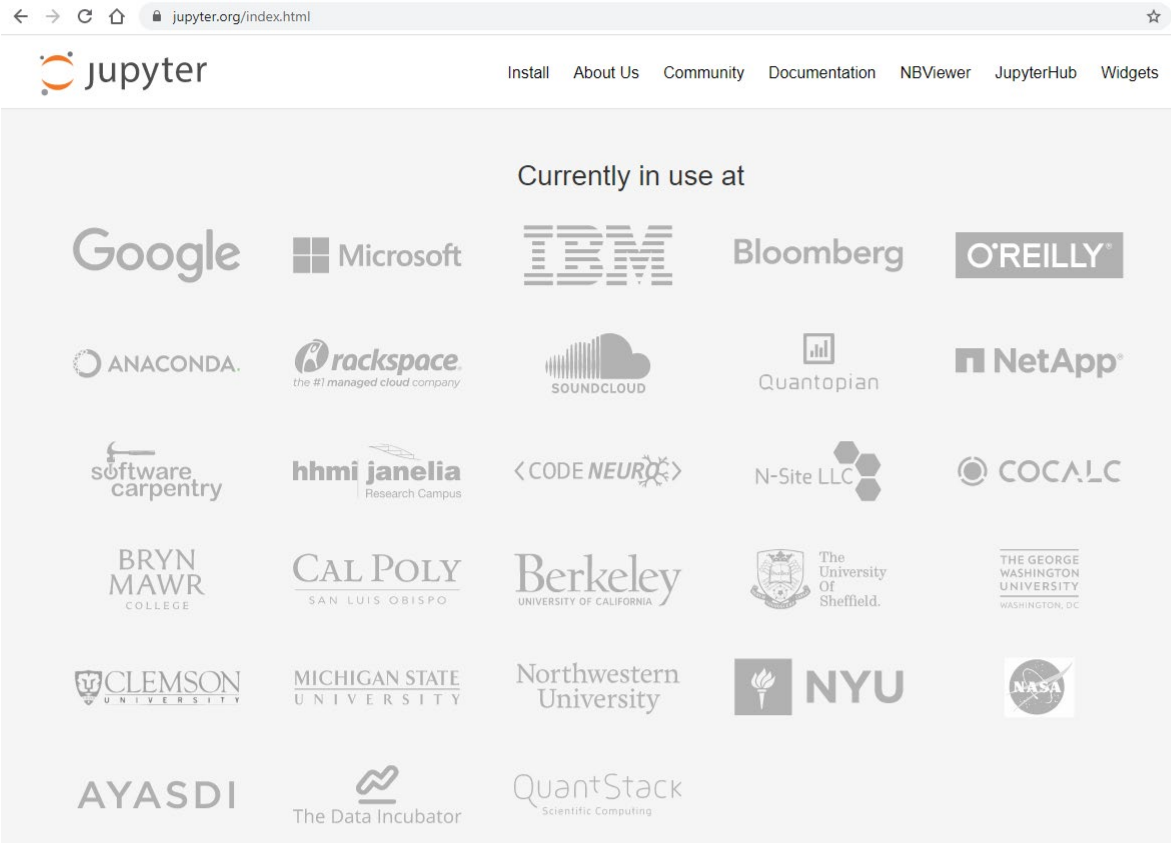
2. Computational Narratives

- Ability to create and share a story with code and data
- Making interactive computations reproducible (share code and data)
- Provide artifacts that can be shared with others in both the original as well as other forms (e.g., websites, books, online documentation, etc.)

[1] Jupyter: Thinking and Storytelling with Code and Data. computer.org

3. More Than Software

- Jupyter is more than software
- It builds and consists of services, open standards and protocols, and community (e.g., VS Code, GitHub, and many more)
- Millions of users



[1] Jupyter: Thinking and Storytelling with Code and Data. computer.org

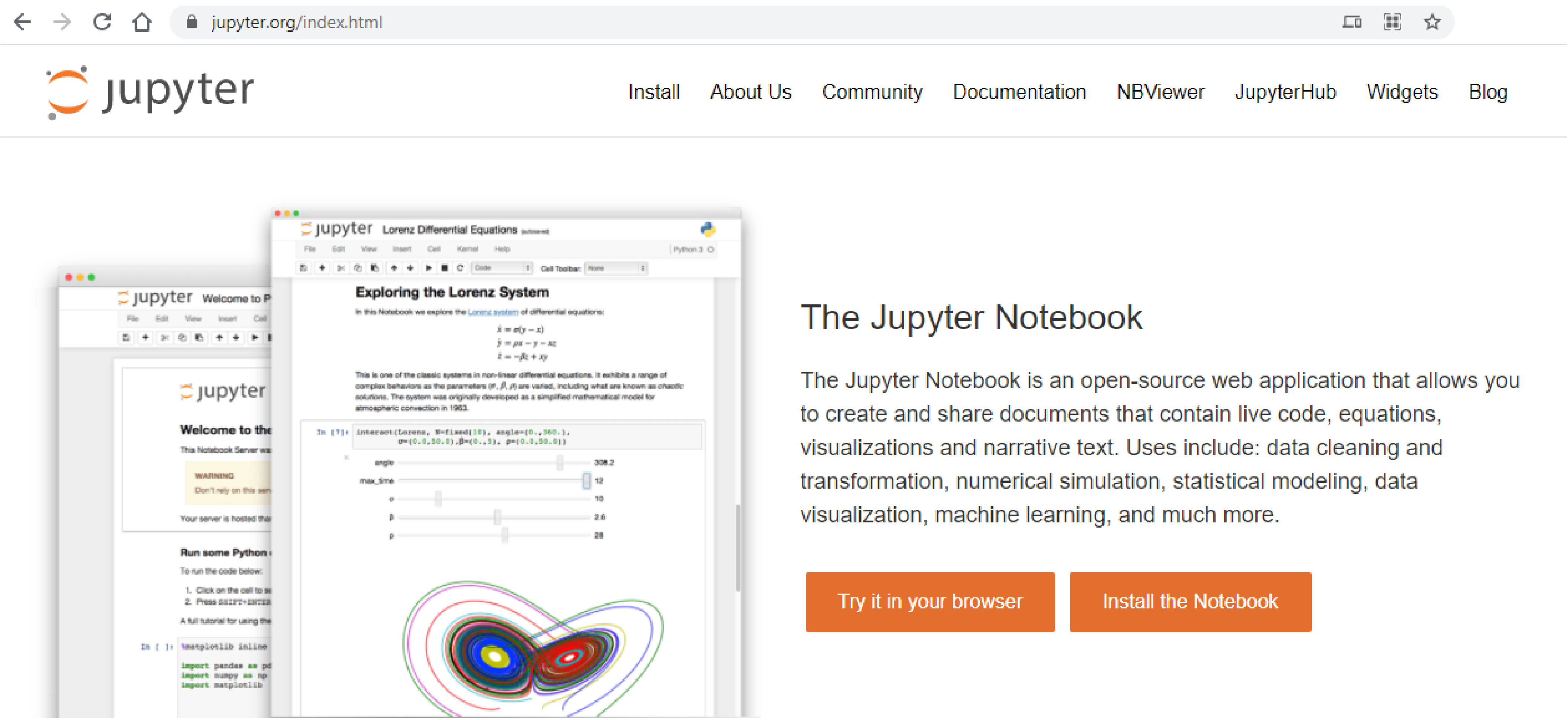
Jupyter Notebook

Jupyter Notebook

"Jupyter Notebook is an open document format and web application that enables users to compose and share interactive programs, which combine live code with narrative text, equations, interactive visualizations, images, and more." [1]

[1] Jupyter: Thinking and Storytelling with Code and Data. computer.org

Jupyter Notebook



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Requirements and Installation

Requirements: Python 3.3 or greater

Installation:

- If you do not have Python installed, the easiest is to install Anaconda distribution (contains both Python and >700 packages)
  - Link: Mac OS X
  - Link: Windows
  - Link: Linux
- To install only the Jupyter Notebook, type in the terminal:

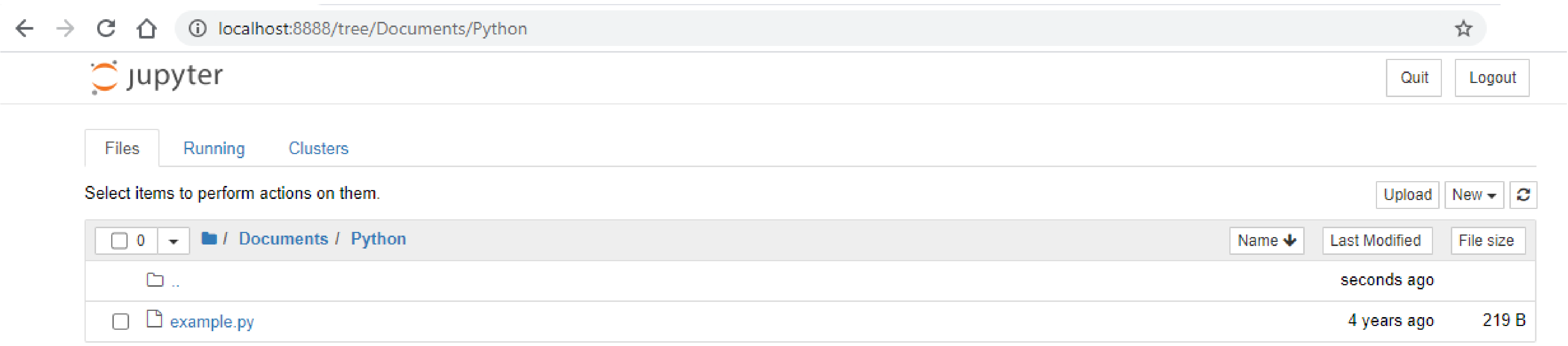
```
pip install jupyter
```

Creating a Jupyter Notebook

- Launch Jupyter Notebook by typing in the terminal

```
jupyter notebook
```

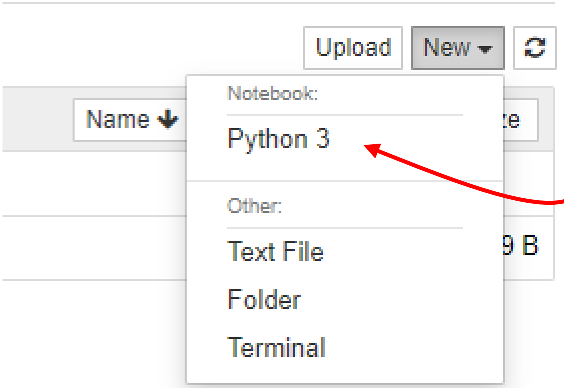
- After launching, your local web browser should be redirected to a default address http://localhost:8888 and a page that resembles the following should open:



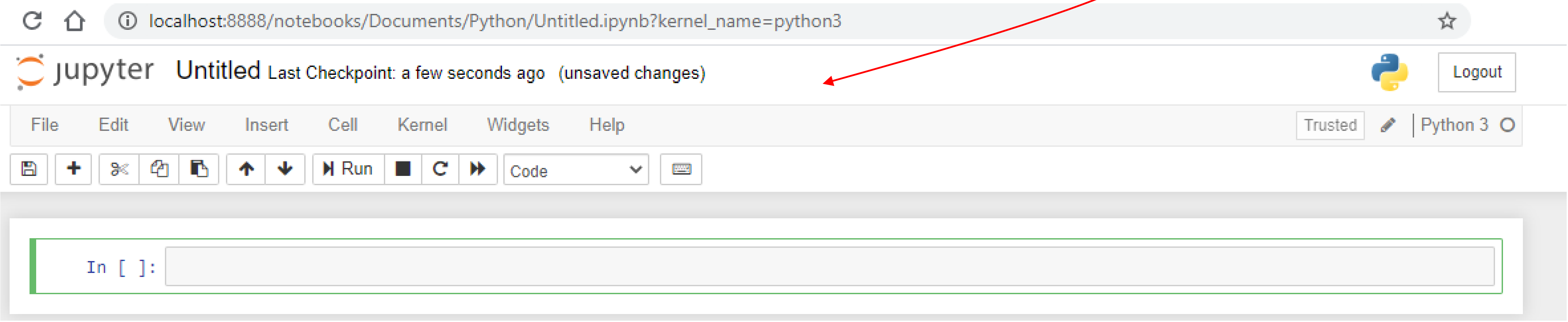
Select items to perform actions on them.

Name	Last Modified	File size
..	seconds ago	
example.py	4 years ago	219 B

Creating a Jupyter Notebook

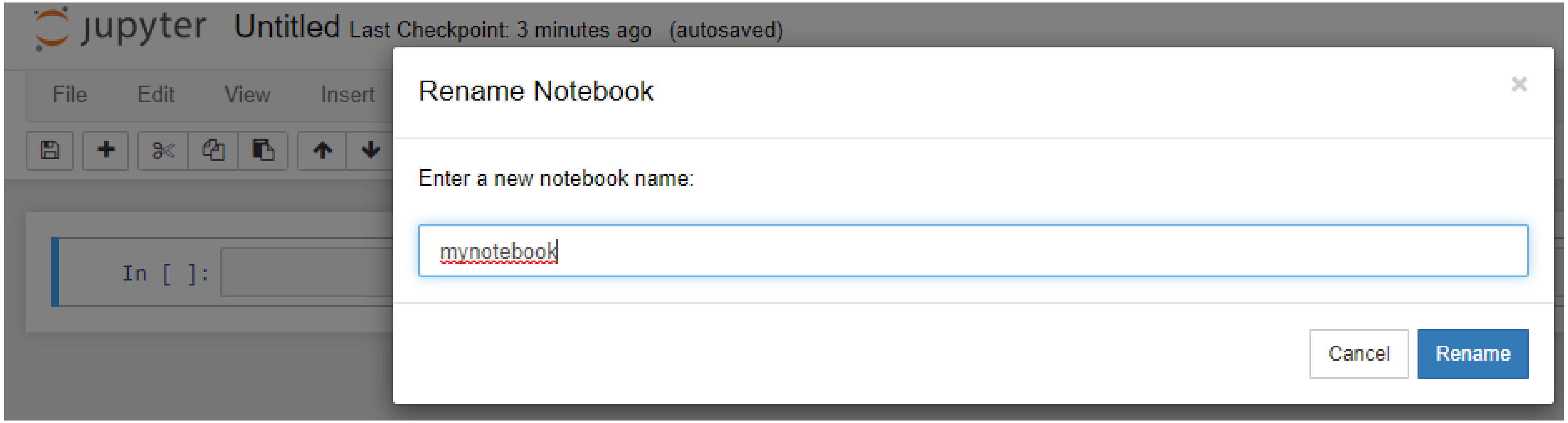


- By selecting Python 3 from the New dropdown menu, you will create a new blank Jupyter Notebook file


localhost:8888/notebooks/Documents/Python/Untitled.ipynb?kernel_name=python3
jupyter
Untitled Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

Creating a Jupyter Notebook

- Let us rename the file from Untitled to mynotebook, for example



Untitled Last Checkpoint: 3 minutes ago (autosaved)

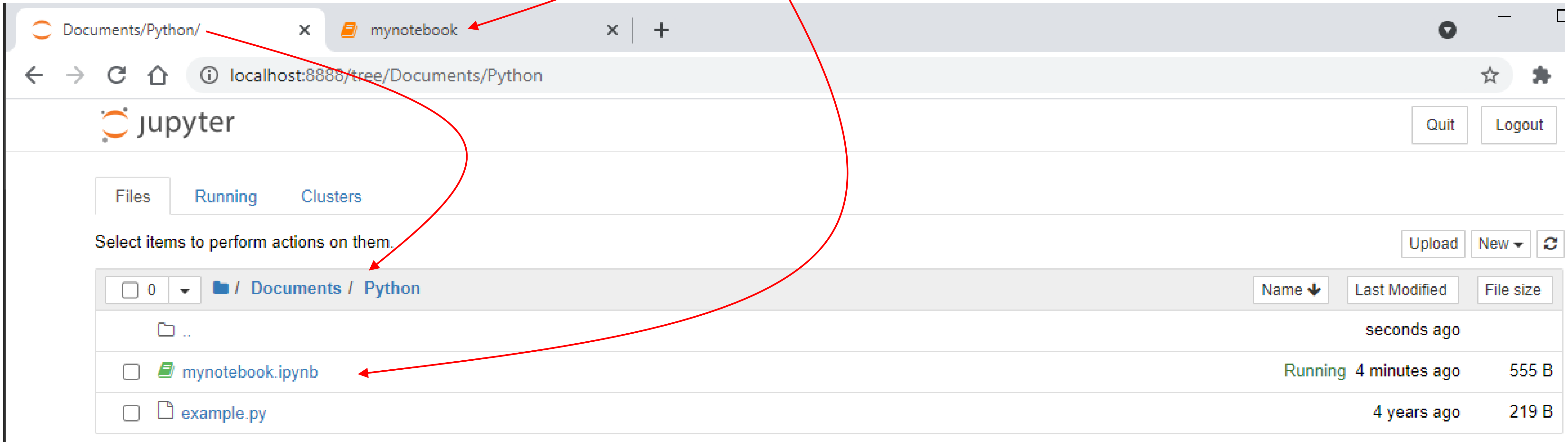
Rename Notebook

Enter a new notebook name:

mynotebook

Creating a Jupyter Notebook

- Once renamed, the new Notebook (a file with extension ipynb) appears in the corresponding directory



Select items to perform actions on them.

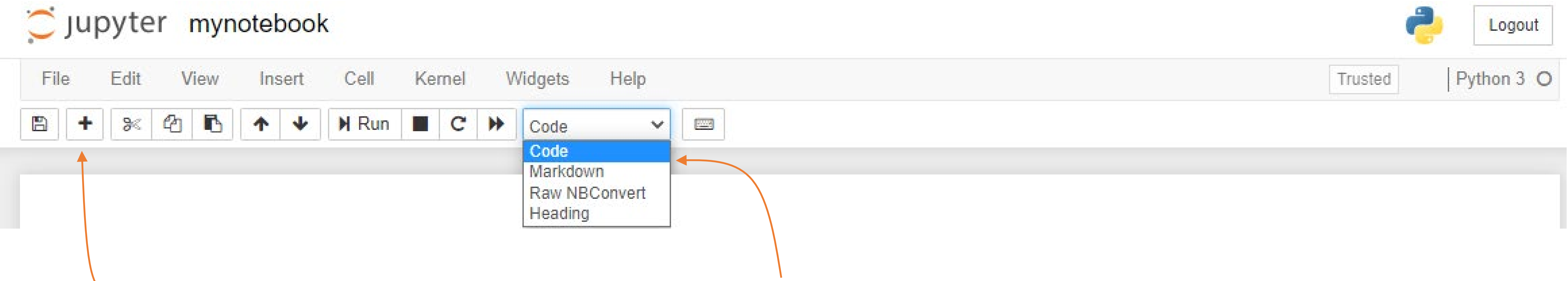
/ Documents / Python

	Name		Last Modified	File size
	..		seconds ago	
□	mynotebook.ipynb	Running	4 minutes ago	555 B
□	example.py		4 years ago	219 B

Jupyter Notebook Cells

Cells

- Notebook is composed of cells
- Two main types of cells
  - code cells
  - text cells (in a Markdown format)

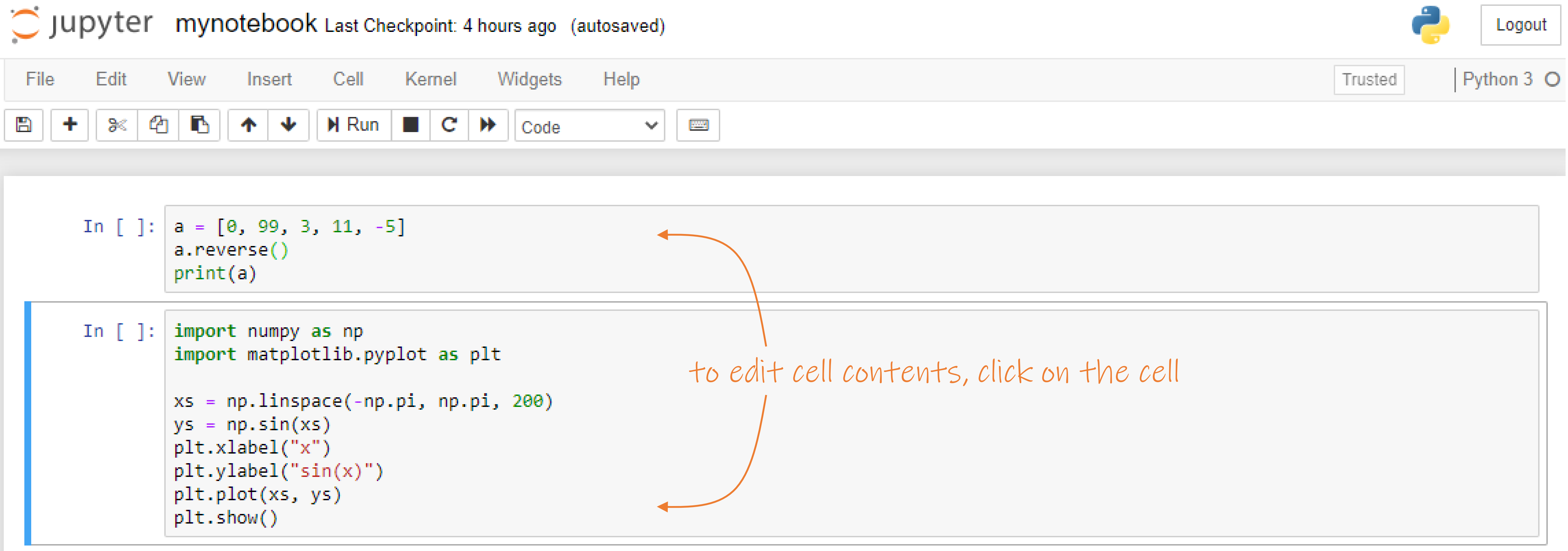


create or add a new cell

choose cell type

Running Code Cells

- Example, after adding two cells with some code:



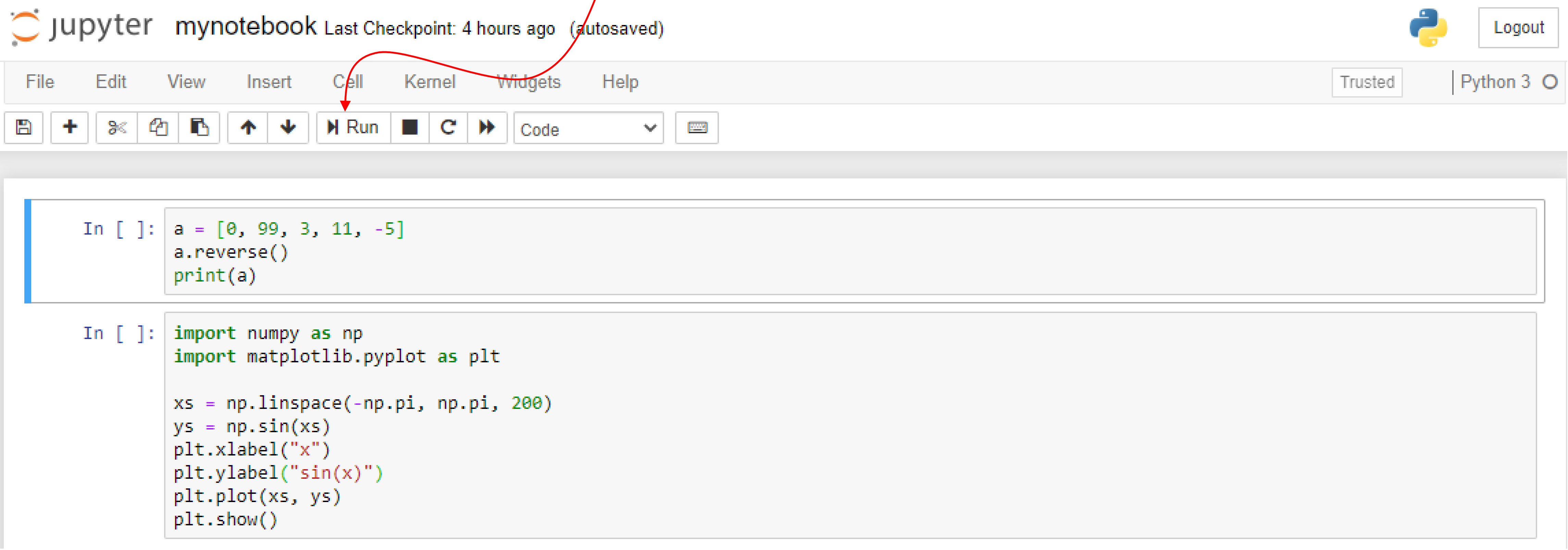
```
a = [0, 99, 3, 11, -5]
a.reverse()
print(a)
```

```
import numpy as np
import matplotlib.pyplot as plt

xs = np.linspace(-np.pi, np.pi, 200)
ys = np.sin(xs)
plt.xlabel("x")
plt.ylabel("sin(x)")
plt.plot(xs, ys)
plt.show()
```

to edit cell contents, click on the cell

Running Code Cells

- To run a single cell, we should select that cell (by clicking on it) and press CTRL+ENTER (or Run button)



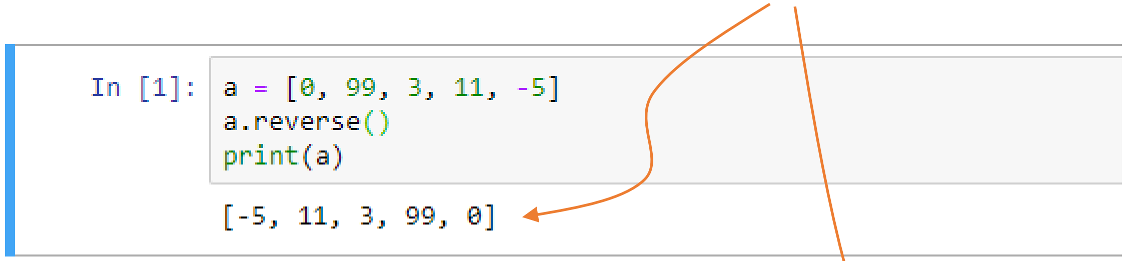
```
In [ ]: a = [0, 99, 3, 11, -5]
        a.reverse()
        print(a)
```

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt

        xs = np.linspace(-np.pi, np.pi, 200)
        ys = np.sin(xs)
        plt.xlabel("x")
        plt.ylabel("sin(x)")
        plt.plot(xs, ys)
        plt.show()
```

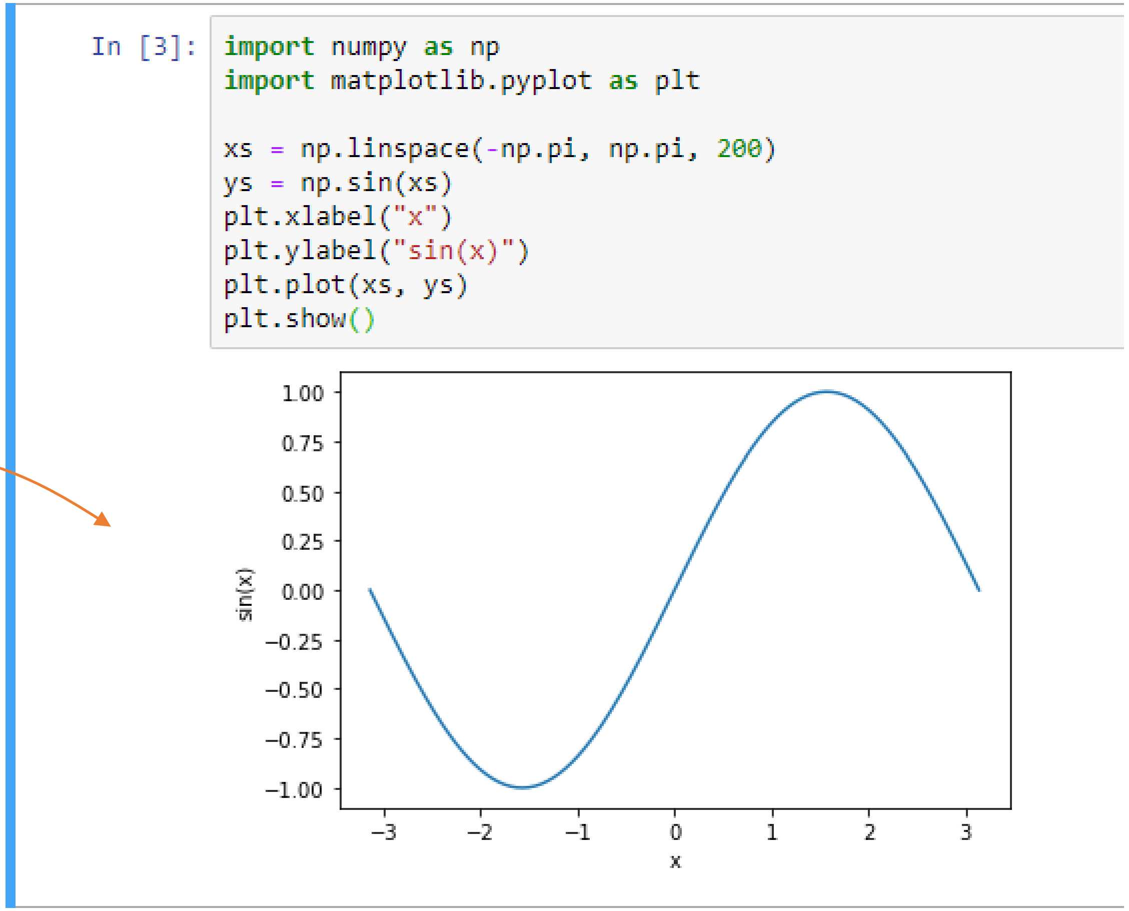
Running Code Cells

- To run a single cell, we should select that cell (by clicking on it) and press CTRL+ENTER (or Run button)

results



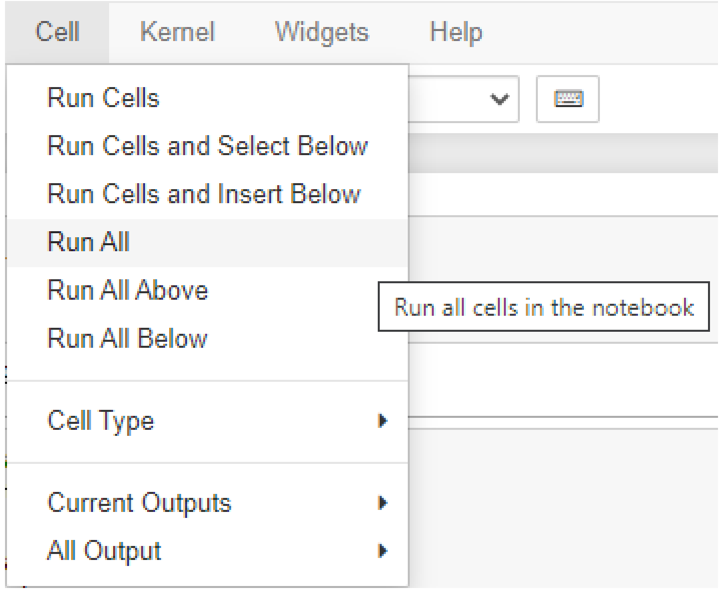
```
In [1]: a = [0, 99, 3, 11, -5]
        a.reverse()
        print(a)

        [-5, 11, 3, 99, 0]
```



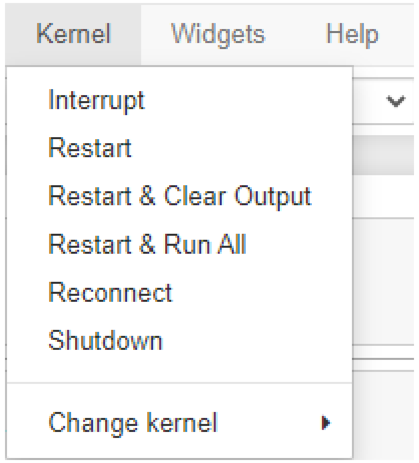
```
In [3]: import numpy as np
        import matplotlib.pyplot as plt

        xs = np.linspace(-np.pi, np.pi, 200)
        ys = np.sin(xs)
        plt.xlabel("x")
        plt.ylabel("sin(x)")
        plt.plot(xs, ys)
        plt.show()
```

Running Code Cells

- To run all cells, we choose Run All from Cell dropdown menu



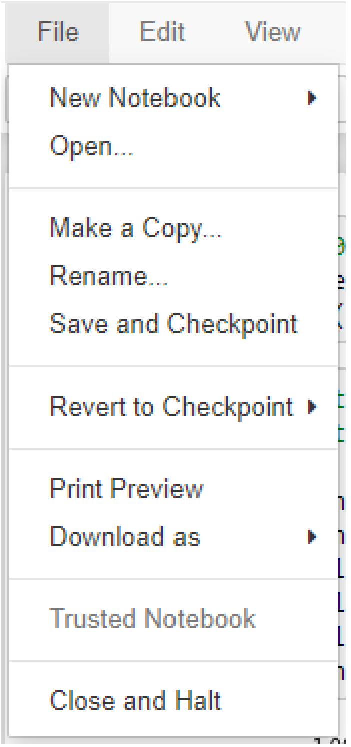
Kernel



- A notebook kernel is a computational engine that executes the code in the notebook document
- When the notebook is executed (either cell by cell or Cell → Run All), the kernel performs the computation and produces the results
- Commonly used options:
  - Interrupt: stop the execution of a particular cell
  - Restart: restart the kernel
  - Restart & Clear Output: restart the kernel and reset all previously run cells
  - Shutdown: terminates the session of that notebook

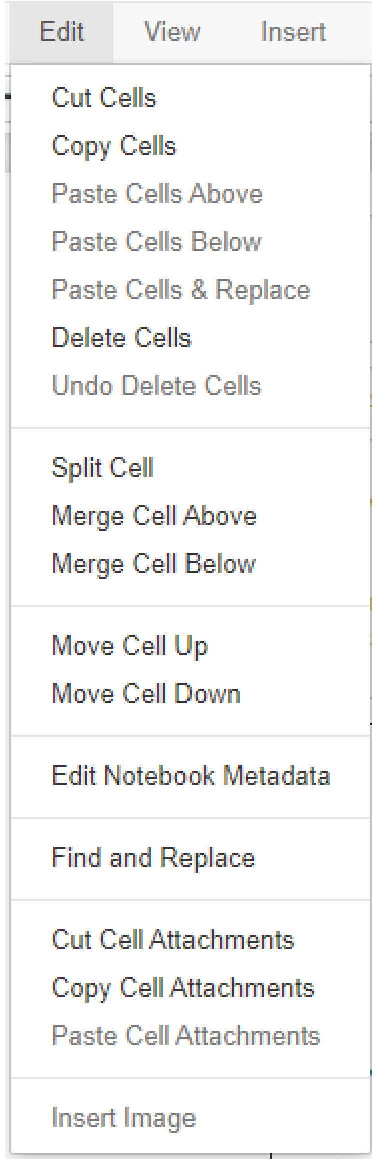
Dropdown menus

File



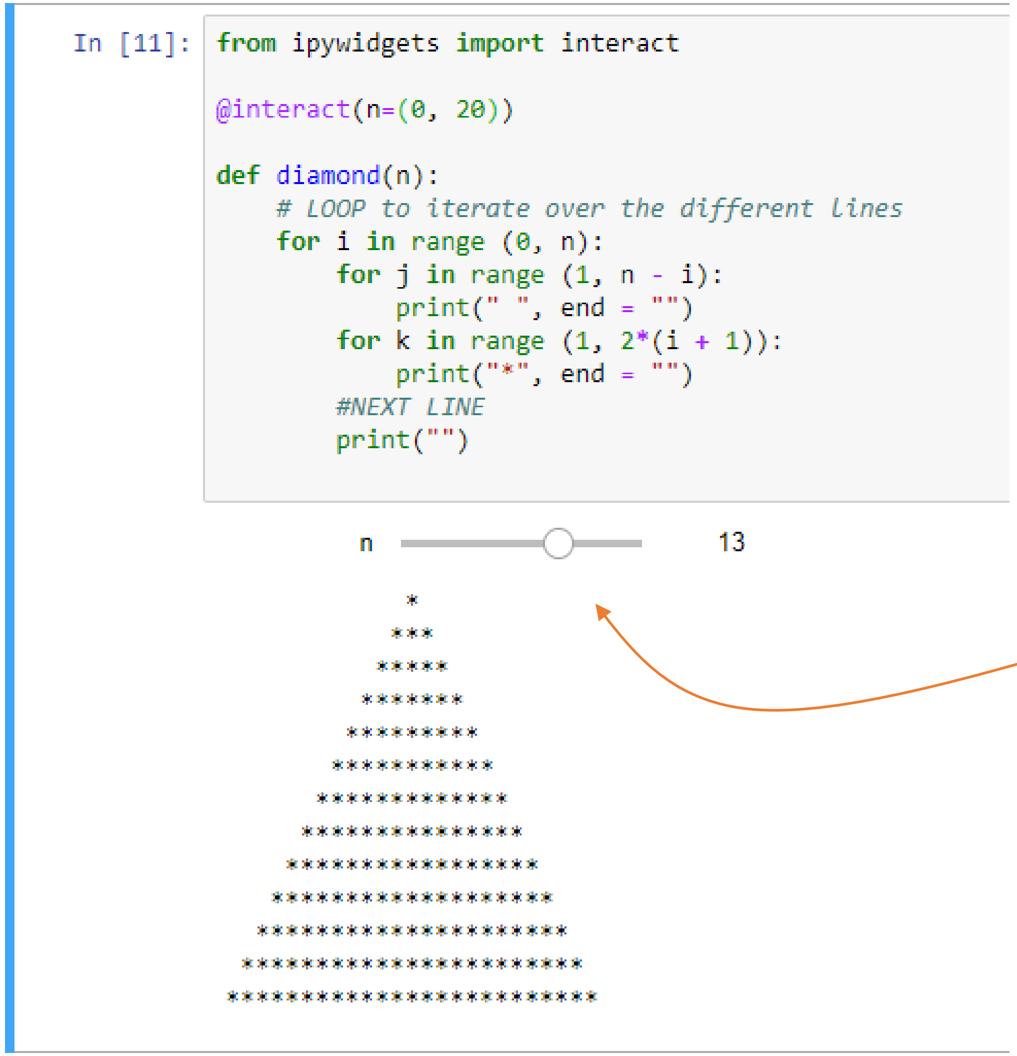
- File dropdown menu allows to
  - create new notebooks
  - open an existing notebook
  - make duplicates of the notebook
  - rename the notebook
  - save the notebook (shortcut: CTRL+S)
  - revert to checkpoint (revert to the last saved state)
  - print preview (gets you a clean view of the code)
  - download as HTML, Python script, Markdown, PDF, etc.
  - close and halt ends the currently running session and shuts down the kernel

Edit



- Edit dropdown menu allows to manipulate cells
  - cut
  - copy or paste or replace
  - delete or undo delete
  - split or merge cells
  - move cells (up or down)
  - find and replace
  - etc.

Widgets



```
In [11]: from ipywidgets import interact

         @interact(n=(0, 20))

         def diamond(n):
             # LOOP to iterate over the different lines
             for i in range (0, n):
                 for j in range (1, n - i):
                     print(" ", end = "")
                 for k in range (1, 2*(i + 1)):
                     print("*", end = "")
                 #NEXT LINE
                 print("")
```

n 13

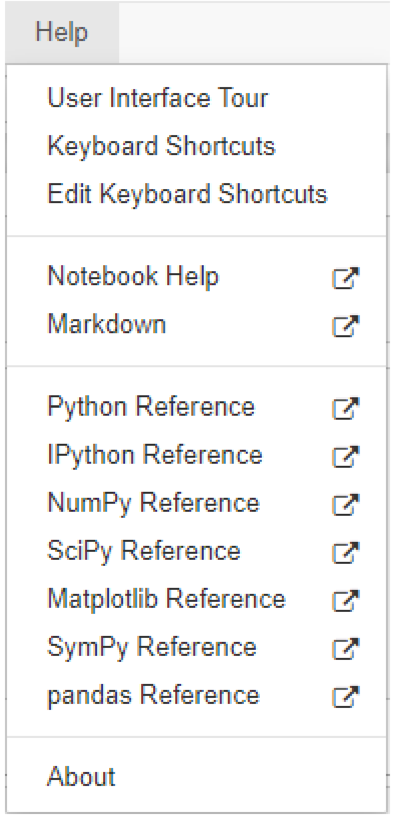
```
            *
           ***
          *****
         *******
        *********
       ***********
      *************
     ***************
    *****************
   *******************
  *********************
 ***********************
*************************
```

- ipywidgets package
- Widgets are eventful objects, such as sliders, progress bars, text boxes, radio buttons
- List of widgets: link

slider widget allowing dynamic change of variable n

- Widgets dropdown menu saves or clears widget state

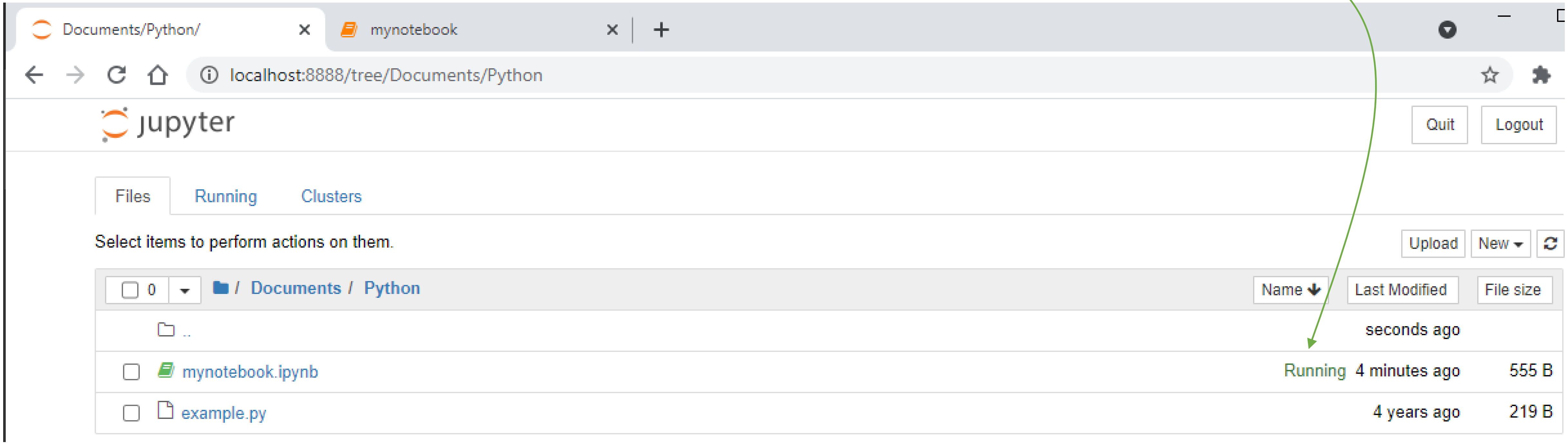
Help



- User interface tour
- Keyboard shortcuts (you can edit them)
- Notebook help
- Markdown help, etc.

Closing Notebooks

- Closing the browser (or the tab) will not close the Jupyter Notebook
- Steps to properly close a running notebook
  1. save it first (CTRL+S)
  2. shut down its running kernel (File → Close and Halt)
  3. quit the terminal from which you started the notebook



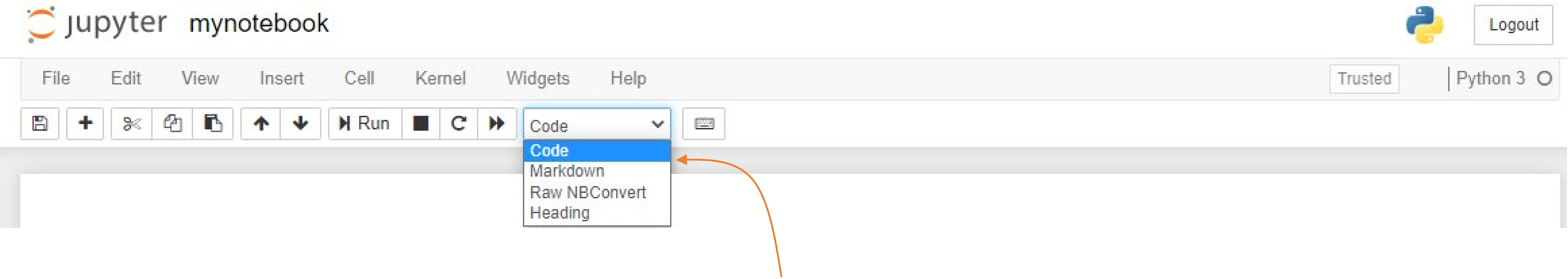
Select items to perform actions on them.

Name		Last Modified	File size
..		seconds ago	
mynotebook.ipynb	Running	4 minutes ago	555 B
example.py		4 years ago	219 B

Text Cells and Markdown

Text cells

- We can combine text with code
- For text, choose the text cell type



choose cell type

- The text should be entered in Markdown format/syntax
  - Link: [Wiki]
  - Link: [cheatsheet]

Markdown Syntax

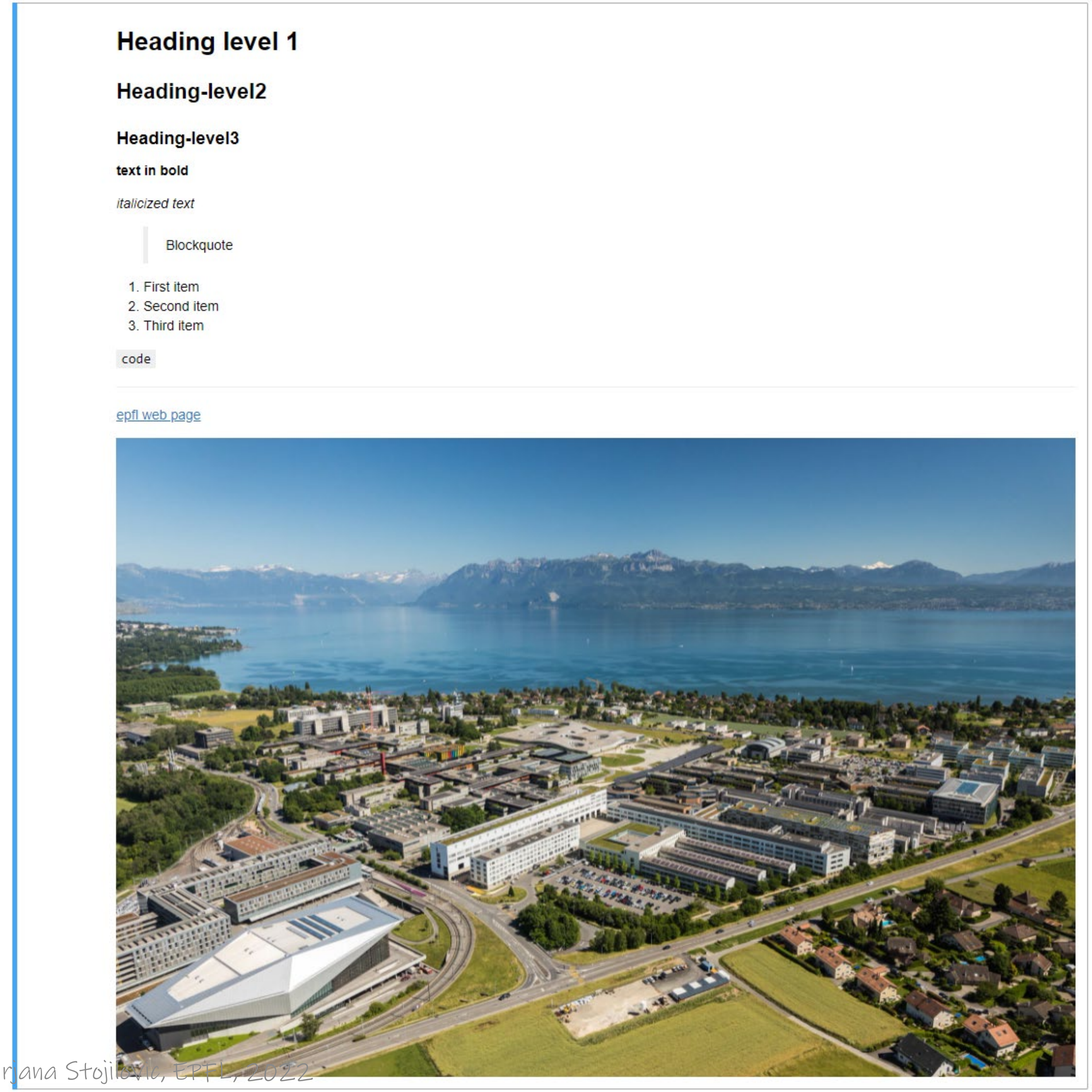
Element	Markdown Syntax
Heading	# Heading-level 1 ## Heading-level2 ### Heading-level3
Bold	**text in bold**
Italic	*italicized text*
Blockquote	> Blockquote
Ordered list	1. First item 2. Second item 3. Third item
Unordered list	- First item - Second item - Third item
Code	`code`
Horizontal rule	---
Link	[title](https://mylink.ch)
Image	![alt text](image.jpg)

Markdown Example

After running a cell with text in Markdown syntax:

Markdown Syntax
Heading-level 1 ## Heading-level2 ### Heading-level3
text in bold
italicized text
> Blockquote
1. First item 2. Second item 3. Third item
\`code\`

[epfl web page](https://epfl.ch)
![EPFL campus](image.jpg)



Heading level 1

Heading-level2

Heading-level3

text in bold

italicized text

> Blockquote

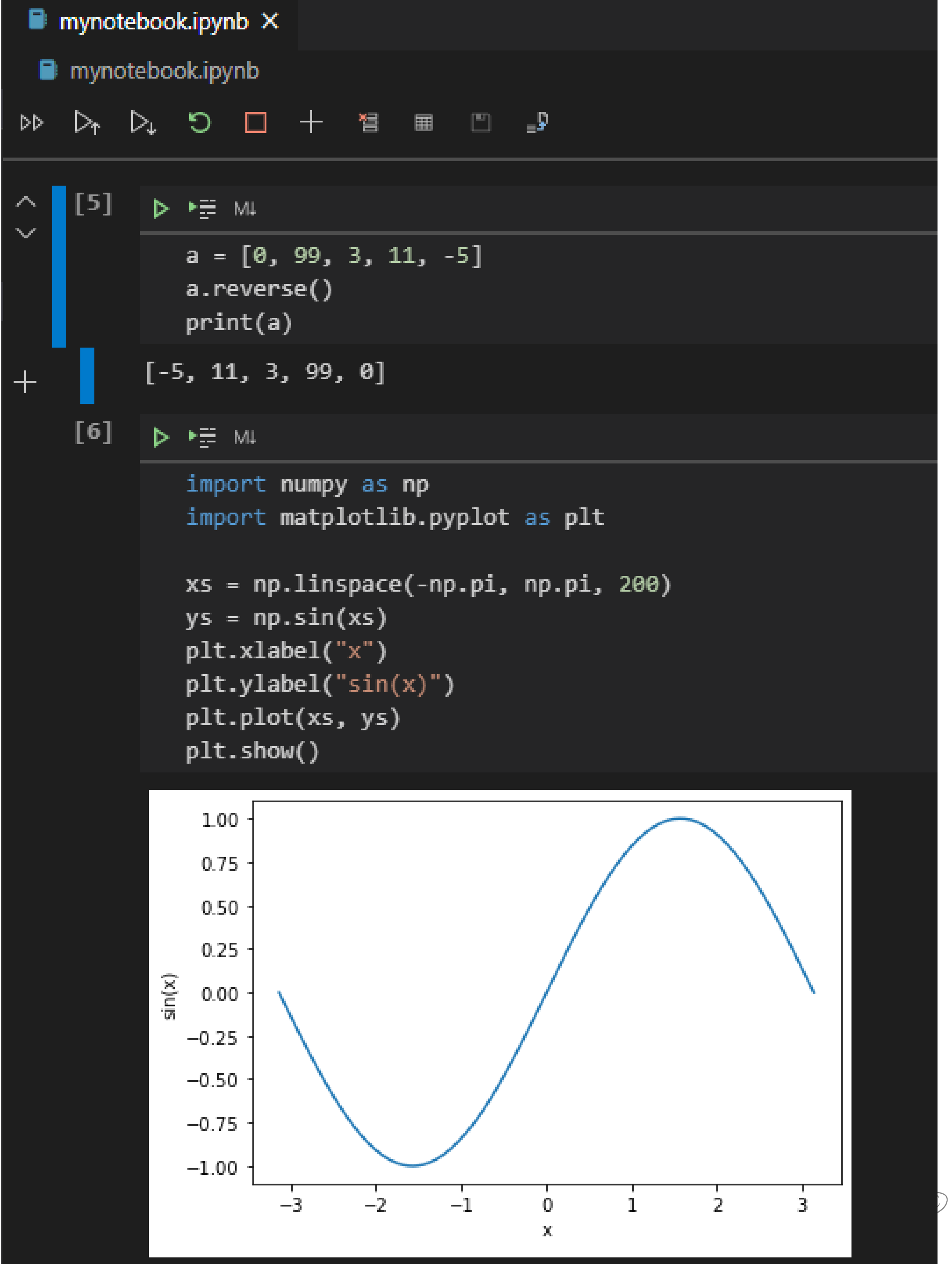
1. First item
2. Second item
3. Third item

`code`

---

epfl web page

Jupyter Notebook and VS Code



```
[5]
a = [0, 99, 3, 11, -5]
a.reverse()
print(a)
```

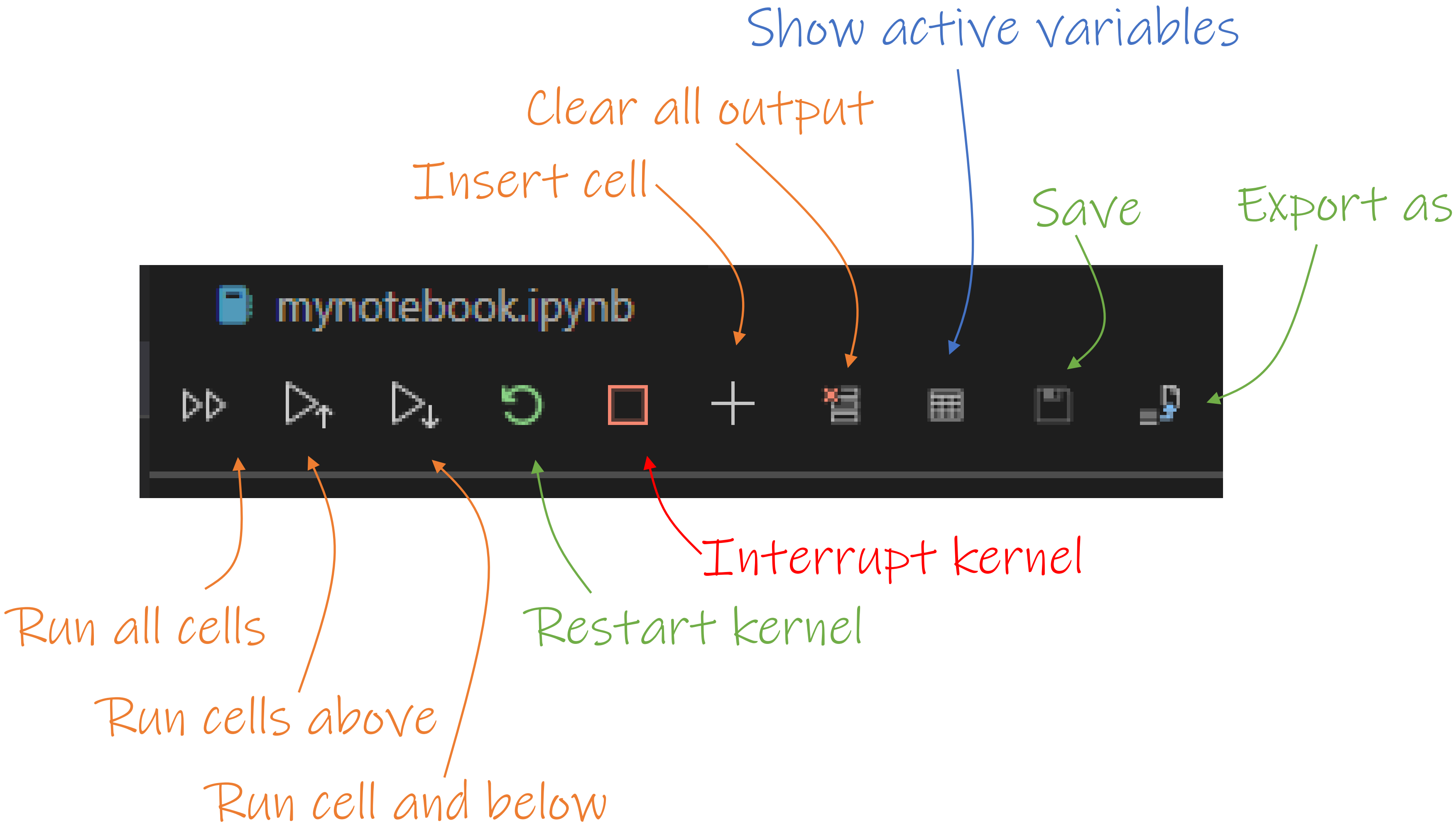
```
[-5, 11, 3, 99, 0]
```

```
[6]
import numpy as np
import matplotlib.pyplot as plt

xs = np.linspace(-np.pi, np.pi, 200)
ys = np.sin(xs)
plt.xlabel("x")
plt.ylabel("sin(x)")
plt.plot(xs, ys)
plt.show()
```

In VS Code

- Open the folder where your Notebook (file with extension ipynb) is located
  - File → Open Folder
- Open the document
  - click on it or
  - File → Open File

In VS Code



Show active variables

Clear all output

Insert cell

Save

Export as

Run all cells

Run cells above

Run cell and below

Restart kernel

Interrupt kernel

Youtube video tutorial for Jupyter Notebook: link

Markdown cheat sheet: link