EBA3400
Programming, Data Extraction and Visualization

Wei-Ting Yang

Department of Data Science and Analytics

Outline

- Course information
- What is programming
- Programming environment
- Introduction to Jupyter notebooks

The agenda for each week

- Synchronous
  - Monday 16:00-17:45

- Asynchronous
  - Exercises/quizzes/video/reading (It will not be counted as part of your grade)
  - The solution to the exercises will be uploaded to itslearning every Wednesday at 15:00

- TA session (Zacky Dhaffa Pratama)
  - Wednesday 16:00-17:00

Course content

Part 1: Basics of programming (w34-40)

- Programming languages
- Programming environment
- Python syntax
- Variables
- Strings and numbers
- Lists
- Conditional statements
- Loop statements
- Functions
- Dictionary, tuple, and set

Part 2: Data extraction and visualization (w41-47)

- Pandas Series and DataFrame
- Read csv files
- Basic statistics
- Data manipulation
- Missing data handling
- Data aggregation
- Data visualization – Pandas
- Data visualization - Matplotlib
- Data visualization - Seaborn
- Time series data

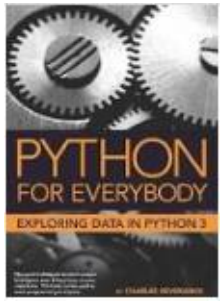
Assessment

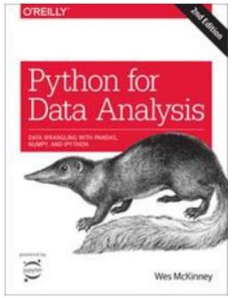
Exam	Weight	Date	Duration	Group/Individual	Exam type
Exam-1	40%	30/9/2022 9:00-12:00	3 hours	Individual	Written submission (Jupyter notebook)
Exam-2	30%	Hand-out: 11/11/2022 9:00 Hand-in:18/11/2022 12:00	1 week	Group(1~3)	Written submission (Jupyter notebook)
Exam-3	30%	25/11/2022 13:00-13:45	45 minutes	Individual	Structured test (Multiple-choice)

Other resources (optional)

- Online resources
  - https://docs.python.org/3.8/tutorial/index.html
  - https://www.geeksforgeeks.org/python-programming-language/?ref=shm
  - https://realpython.com/
  - https://www.w3schools.com/python/default.asp

- Books:
  - Python for Everybody
  - Python for Data Analysis





What is programming

What's programming

Programming:
A way to instruct the computer to perform some tasks.

- Calculate the average reported cases over the last 14 days.
- Calculate the quarterly revenue.
- Extract key words from webpages.
- Face recognition.

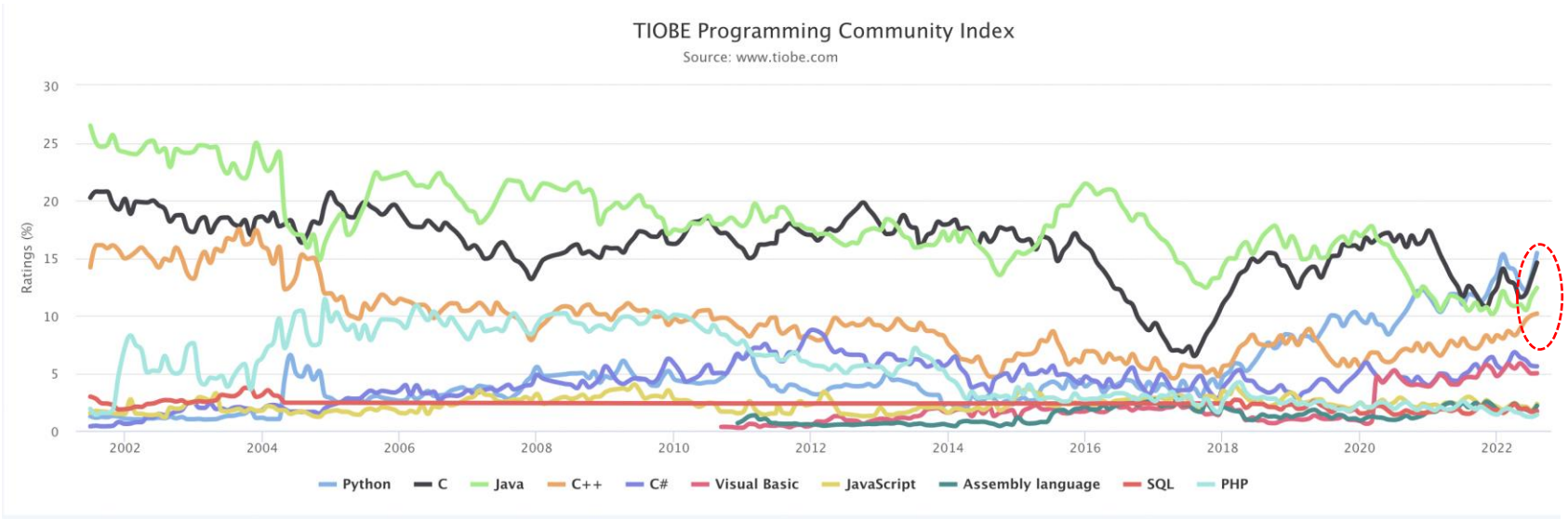
A programming language:
A computer language used to write a set of instructions that computers can understand.

Python	JavaScript	Perl
Java	Swift	Scala
C++	Go	C
Ruby	Rust	Kotlin
Julia	R	Matlab

Python

- Python is a high-level programming language
- Created by Guido van Rossum
- Clear, logical code for small and large projects

First released in 1991



2022 Top3

1. Python (15.42%)
2. C (14.59%)
3. Java (12.40%)

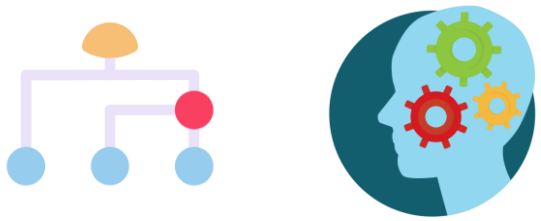
https://www.python.org/
https://www.tiobe.com/tiobe-index/

Why we should learn programming

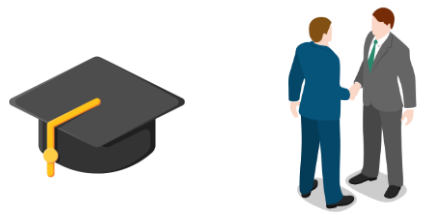
Understand the technology is our daily life

Make tasks easier for you

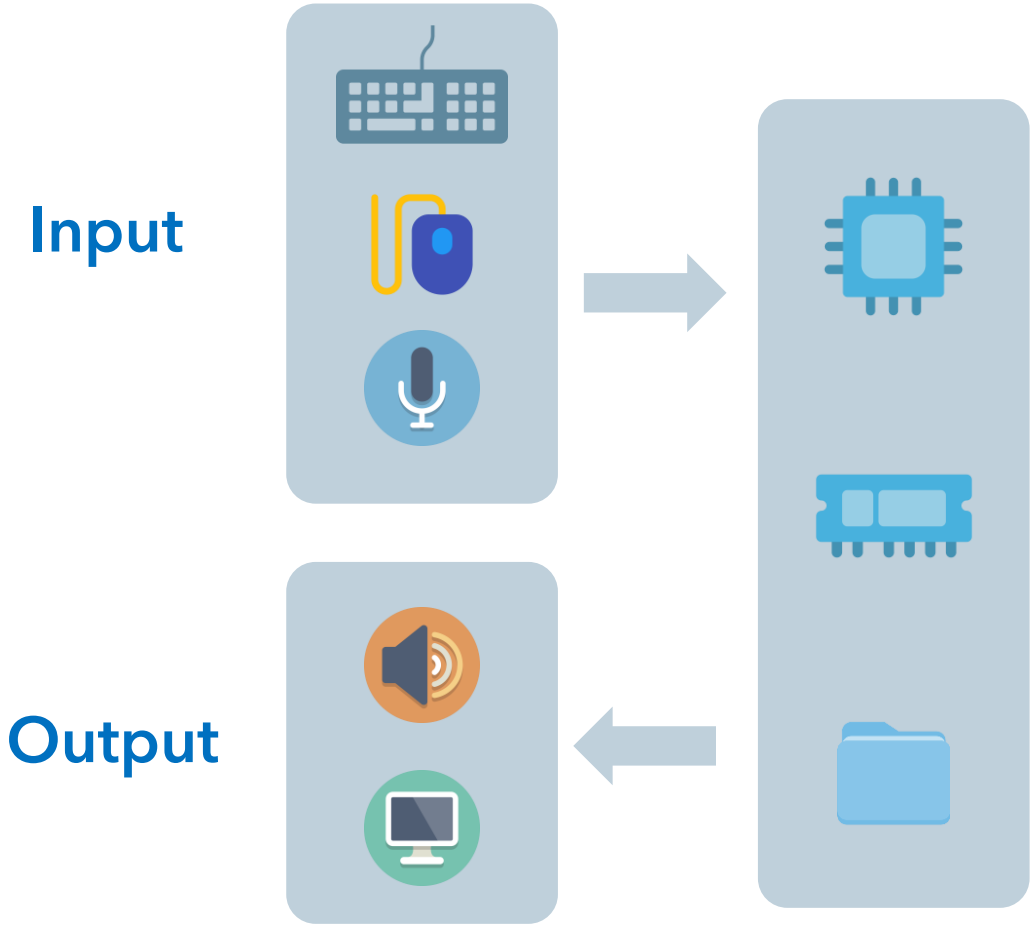
Develop structured thinking and logical skill



Make your job application stand out



How computers work



Processing & Storage

Central Processing Unit (CPU)

- Executes the program and performs the computations.

Main memory

- Stores program operations and data while a program is being executed.
- The main memory is fast but is lost when the computer is turned of. This is also known as RAM.

Secondary memory

- Stores programs and data long term.

BI

- A 3.0 GHz CPU means that the CPU will perform 3 billion operations per second.
- RAM is short for “random access memory”

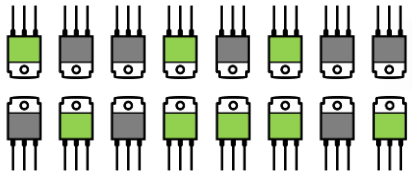
Computer hardware

- Computers have two main parts:
  - Hardware: physical parts of the computer.
  - Software: the code that runs on the computer.

chip

CPU, memory, flash chip

billions of transistors



ON/OFF

- The transistors (switches) is controlled based on the voltage.

bit (**binary digit**)

= bit

(1Byte = 8 bits)

- Computers only understand 1(ON) and 0 (OFF) → binary.
- Each transistor represents a digit of binary.

Binary system for number

Decimal system (also called base-ten system)

$10^3 = 1000$	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
5	**1**	**3**	**6**

$$(5 \times 1000) + (1 \times 100) + (3 \times 10) + (6 \times 1) = 5136$$

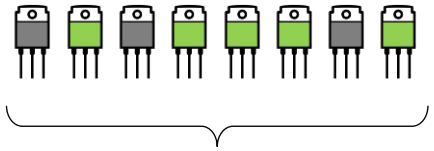
Binary system

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	**0**	**0**	**1**

Binary to Decimal

$$(1 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1) = 9$$

1Byte = 8 bits



256 possible values (0~255)

- 1 Byte = 8 Bits
- 1 Kilobyte (KB) = 1024 Bytes
- 1 Megabyte (MB) = 1024 Kilobytes
- 1 Gigabyte (GB) = 1024 Megabytes
- 1 Terabyte (TB) = 1024 Gigabytes
- 1 Petabyte (PB) = 1024 Terabytes

Binary system for character

- ASCII, stands for American Standard Code for Information Interchange. It is a character encoding standard for electronic communication.

Decimal	Binary	Symbol	Description
0	00000000	NUL	Null char
1	00000001	SOH	Start of Heading
…	…	…	…
97	01100001	a	Lowercase a
98	01100010	b	Lowercase b
99	01100011	c	Lowercase c
100	01100100	d	Lowercase d
101	01100101	e	Lowercase e
102	01100110	f	Lowercase f
103	01100111	g	Lowercase g
104	01101000	h	Lowercase h
105	01101001	i	Lowercase i
…	…	…	…
254	11111110	þ	Latin small letter thorn
255	11111111	ÿ	Latin small letter y with diaeresis

01101000
01100101
01101100
01101100
01101111
00100000
01110111
01101111
01110010
01101100
01100100

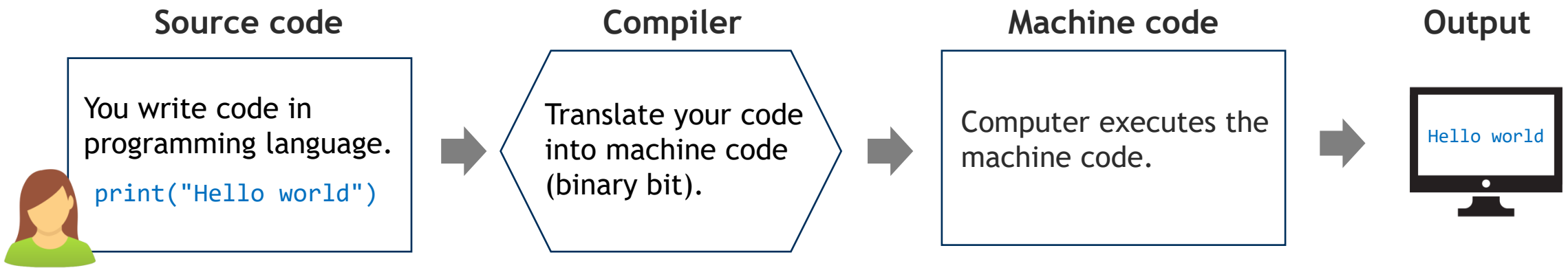
hello world

https://www.ascii-code.com/

Other encoding systems: UTF-8, UTF-16, ISCII

Programming language

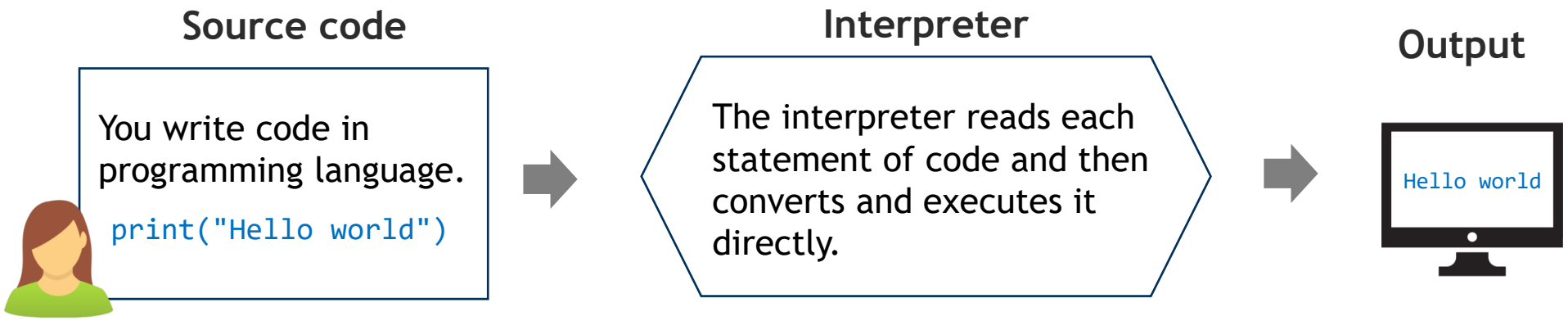
- Programming languages provide an interface between programmers and machine language (binary code).
- You can use syntax that is **English like** and **easier to understand** to express what you want the computer to do.



Programming language

- **A Compiled language**: A programming language whose programs are typically translated into machine language by a compiler before being executed. (e.g., C, C++)

- **An Interpreted language**: A programming language whose program is directly converted and executed by an interpreter. (e.g., python, perl, JavaScript)



Programming environment

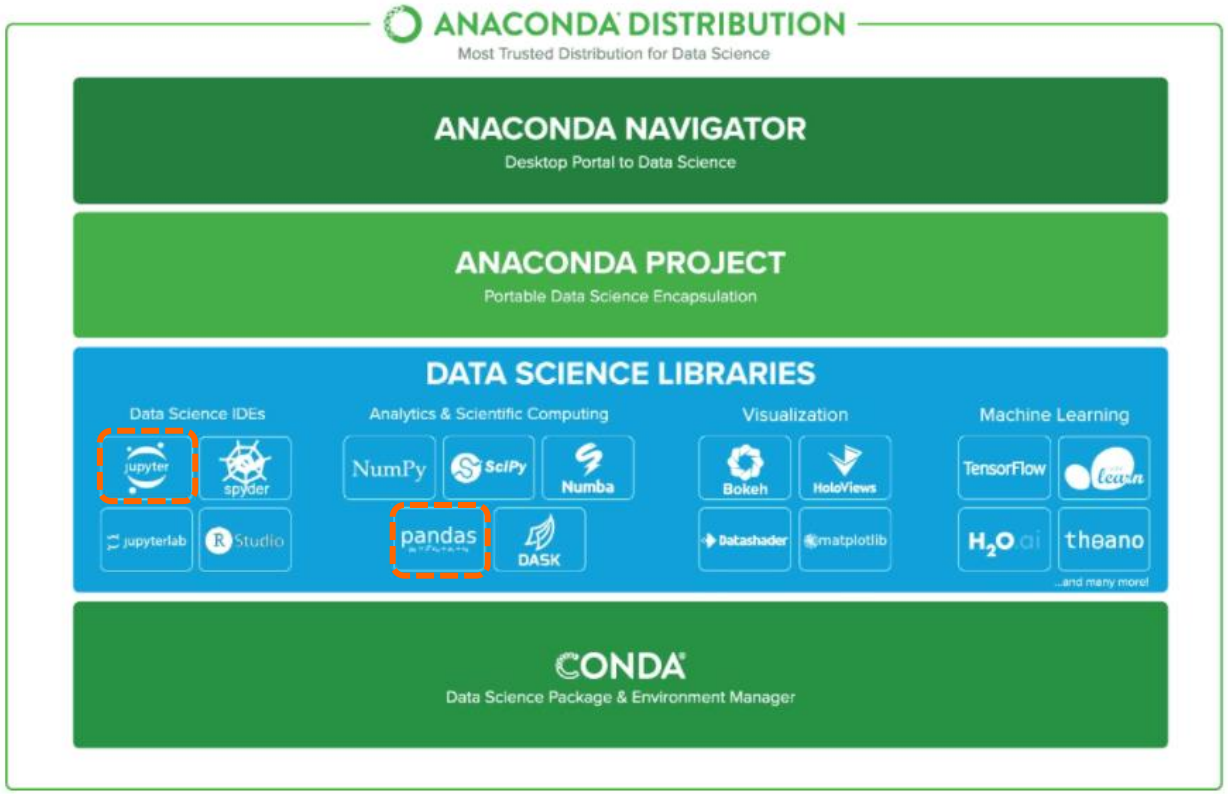
Programming environment

- **IDE (Integrated Development Environment)**
  - A software that provides programmers a set of tools for development.
    - Text editor
    - Build automation tools (compiling computer source code into binary code, packaging binary code, and running automated tests.)
    - Debugging
  - Popular python IDEs: IDLE, Spyder, PyCharm, Thonny, **Jupyter Notebook**.

- **Code editor**
  - A text editor with some added functionalities.
  - Popular code editors: Atom, VScode, Vim.

Programming environment

- **Anaconda** is a distribution of packages. It provides everything you need for data science, including conda, numpy, scipy, jupyter notebook, etc.



Conda is an environment manager and a package manager for installing multiple versions of software packages and their dependencies and switching easily between them.

Project Jupyter

- A project and community whose goal is to "develop **open-source** software, open-standards, and services for interactive computing across dozens of programming languages“
- “Jupyter” is a reference to the three core programming languages supported by Jupyter, which are **Julia, Python, R.** Nowadays, the Jupyter system supports **over 100 programming languages** (called “kernels” in the Jupyter ecosystem) .
- Products: Jupyter Notebook, JupyterHub, and JupyterLab.

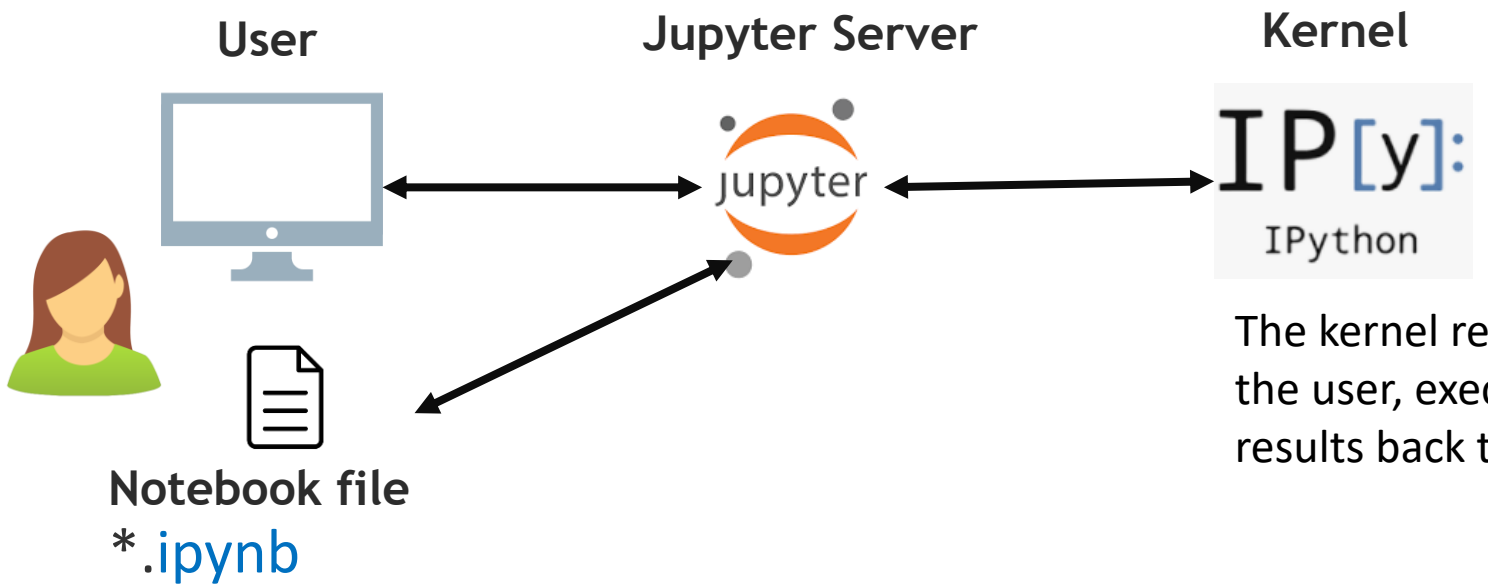
https://jupyter.org/

Open-source software (OSS) is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose.

Jupyter Notebook

- The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain code, equations, visualizations and text.

How do Jupyter Notebooks work?

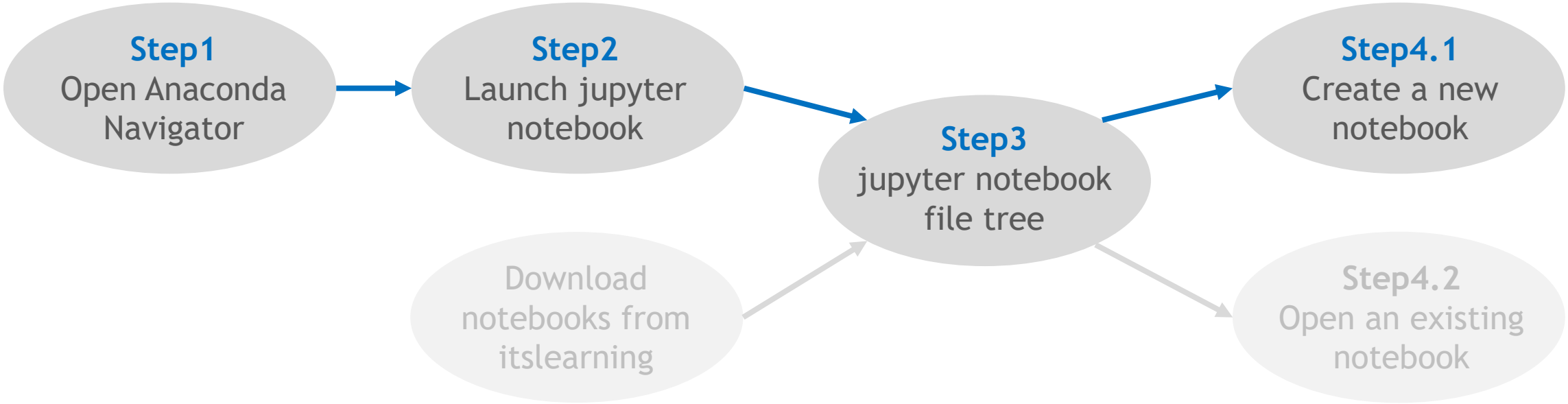


The kernel receives the code sent by the user, executes it, and returns the results back to the client for display.

Kernel https://jupyter.readthedocs.io/en/latest/projects/kernels.html

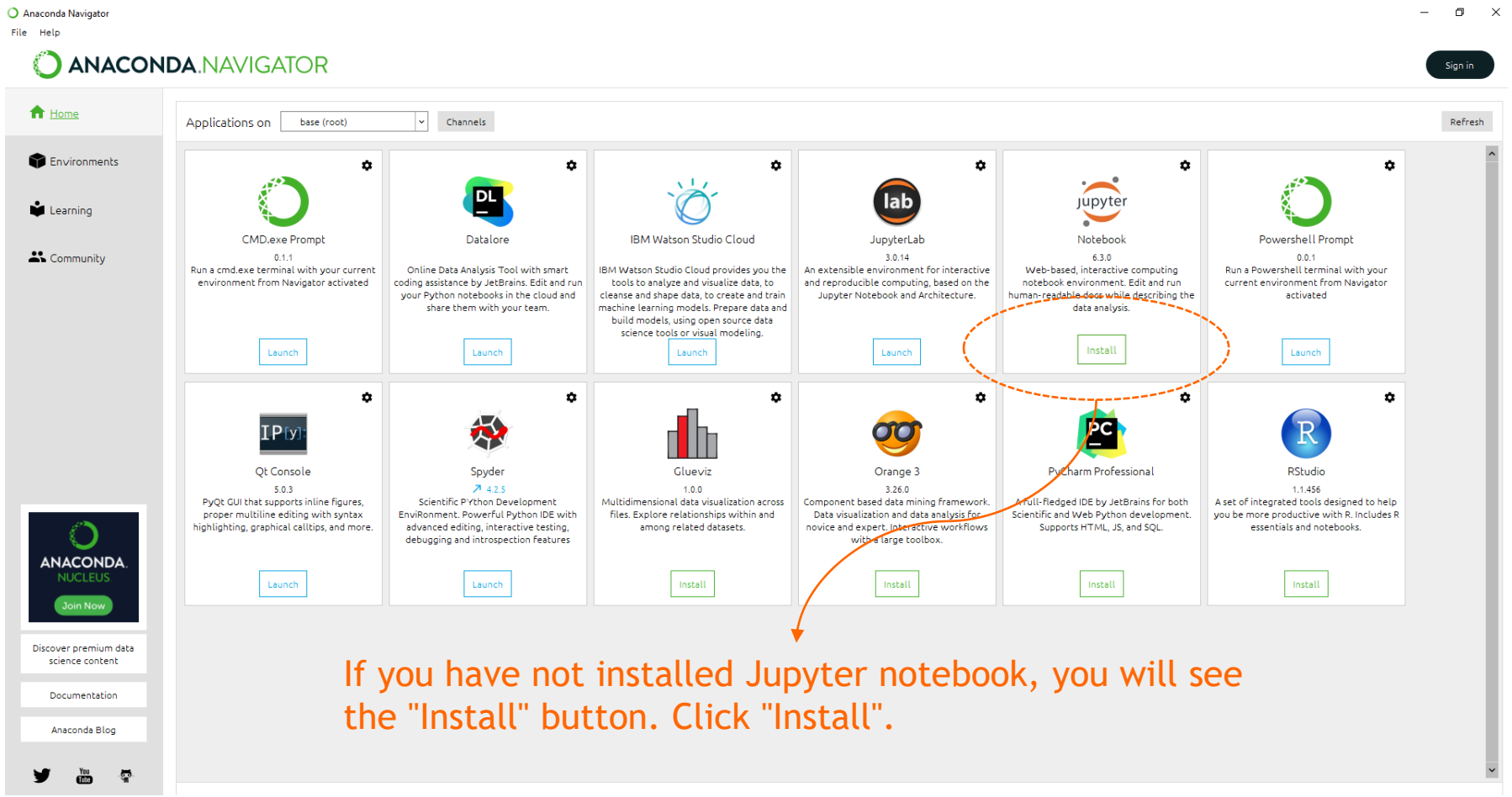
Jupyter notebooks are automatically saved and stored on disk in the open-source JavaScript Object Notation (JSON) format and with a .ipynb extension.

Open a Jupyter notebook



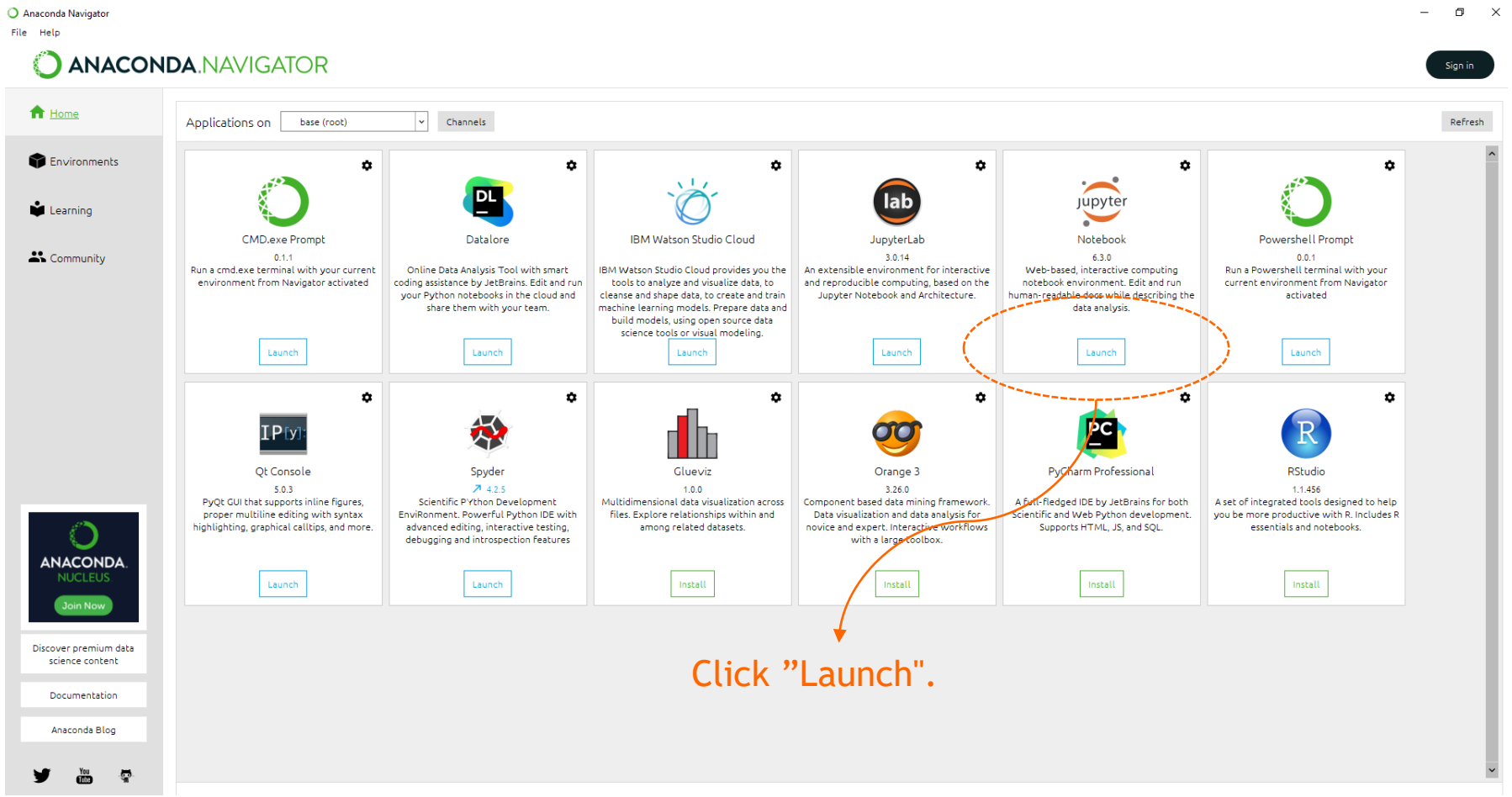
Open a Jupyter notebook – Step1

Step1
Open Anaconda Navigator



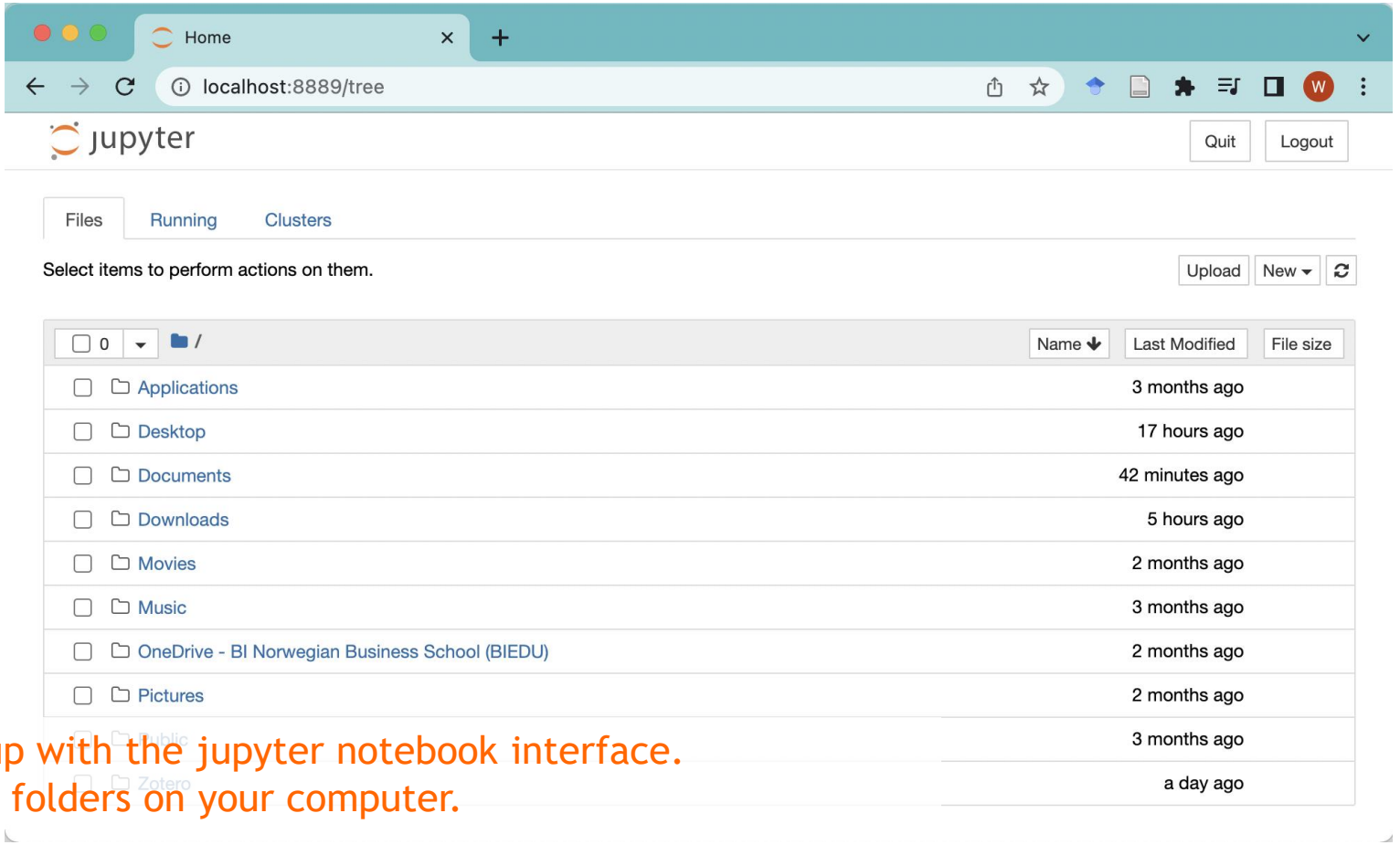
Open a Jupyter notebook – Step2

Step2
Launch jupyter notebook



Open a Jupyter notebook – Step3

Step3
jupyter notebook
file tree

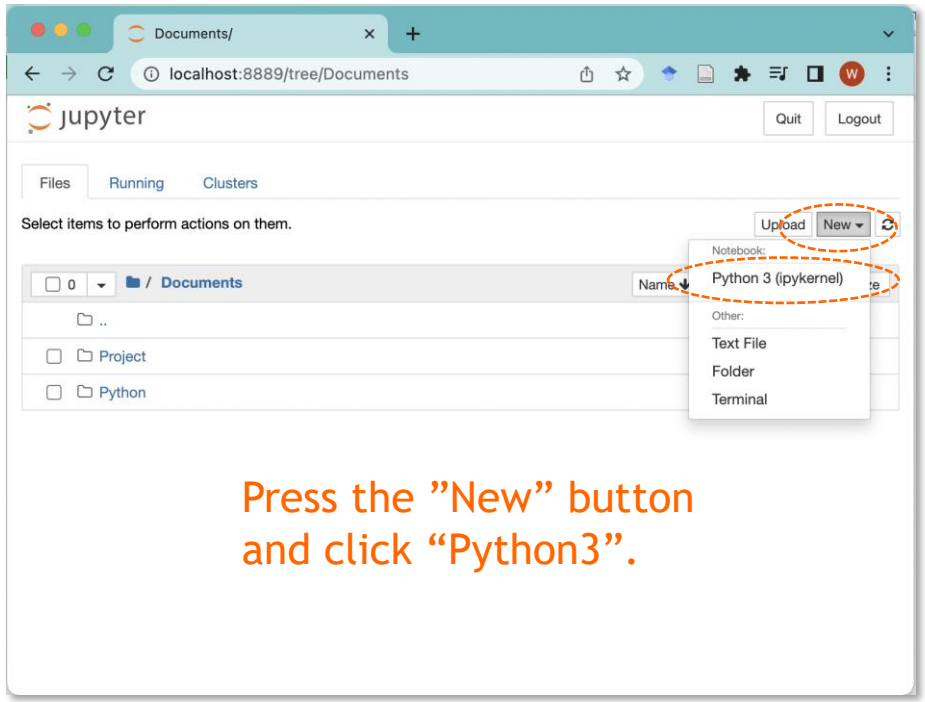


- A browser window pops up with the jupyter notebook interface.
- You can see a file tree of folders on your computer.

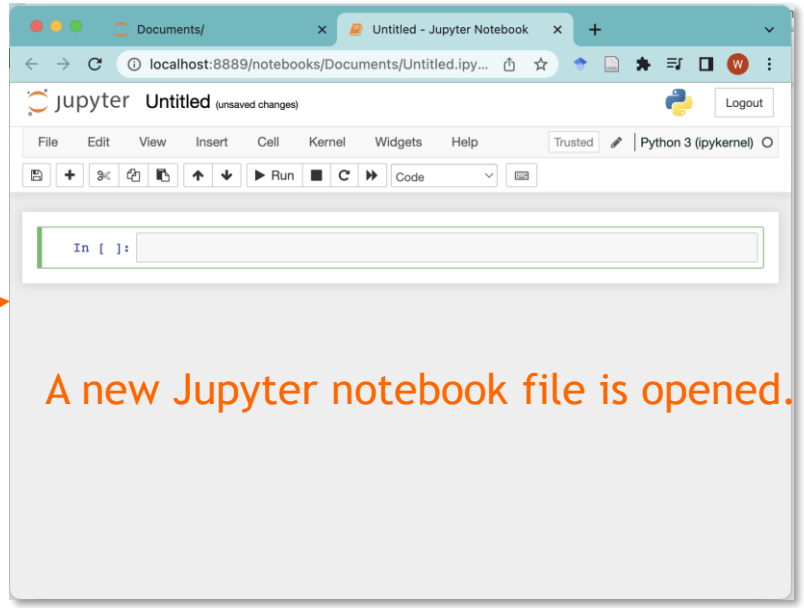
The localhost is the default name describing the local computer address.

Open a Jupyter notebook – Step4.1

Step4.1
Create a new notebook

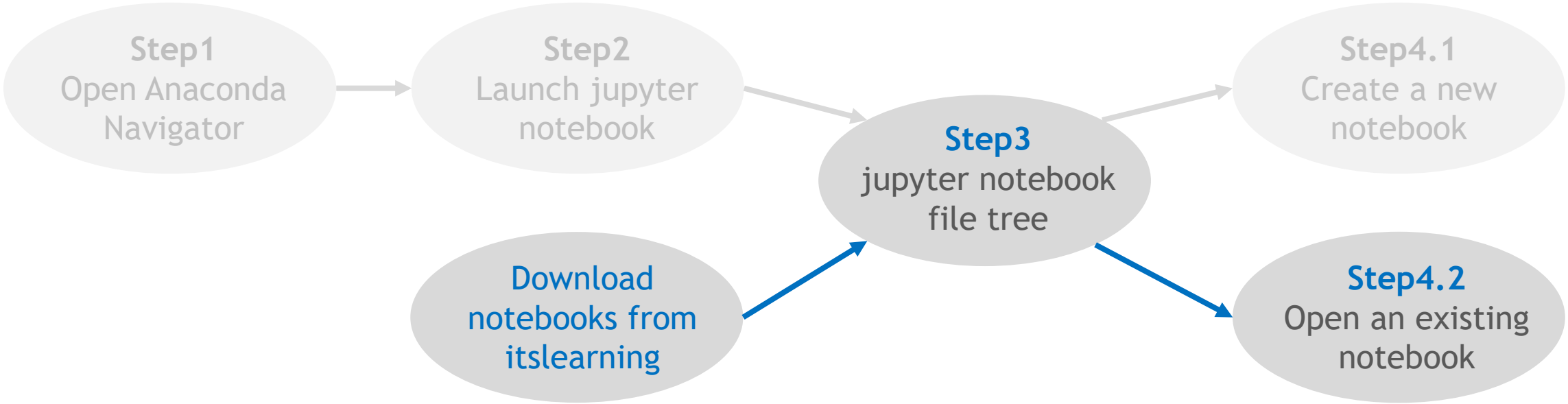


Press the ”New” button and click “Python3”.



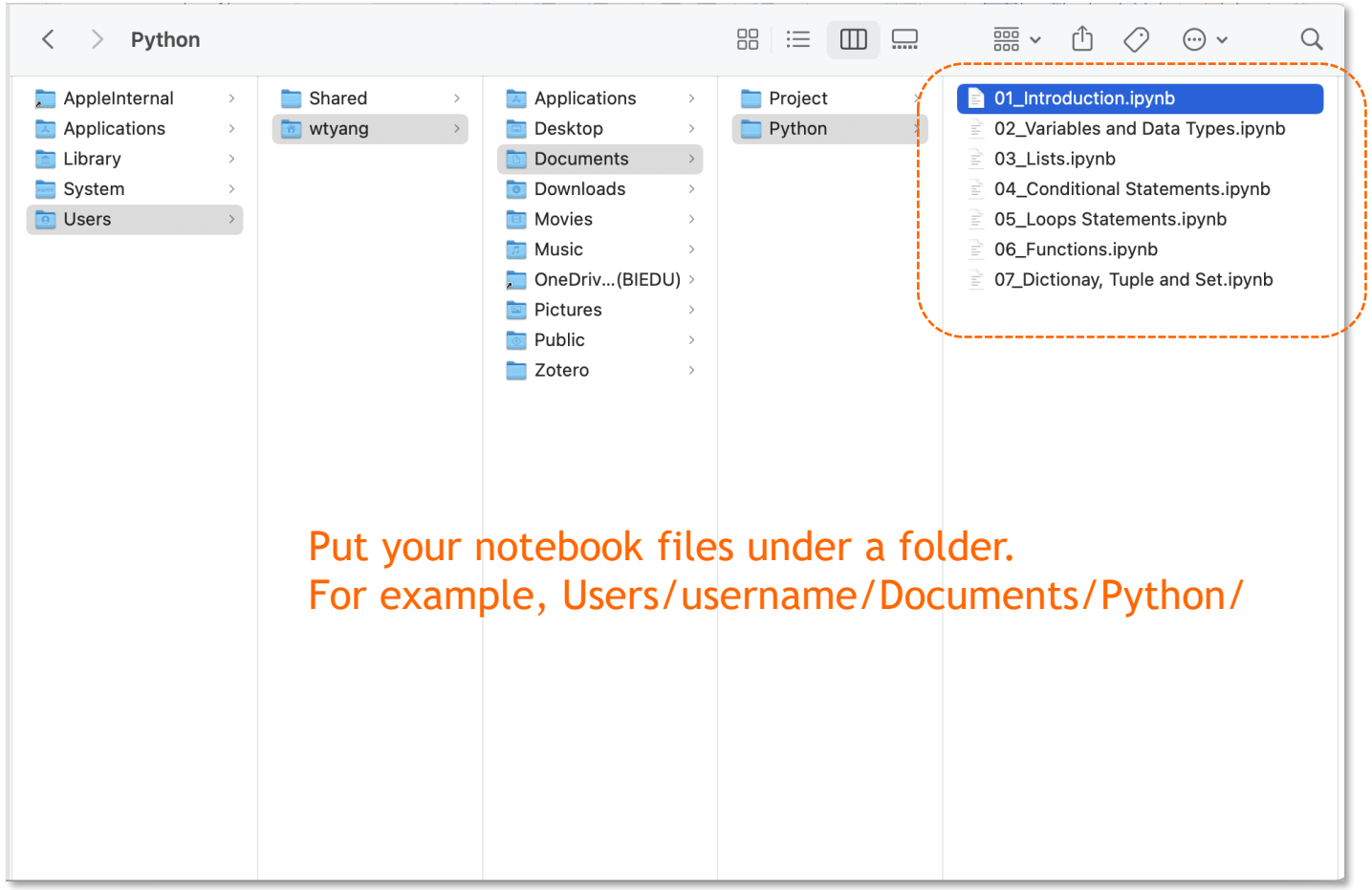
A new Jupyter notebook file is opened.

Open a Jupyter notebook



Open an existing Jupyter notebook

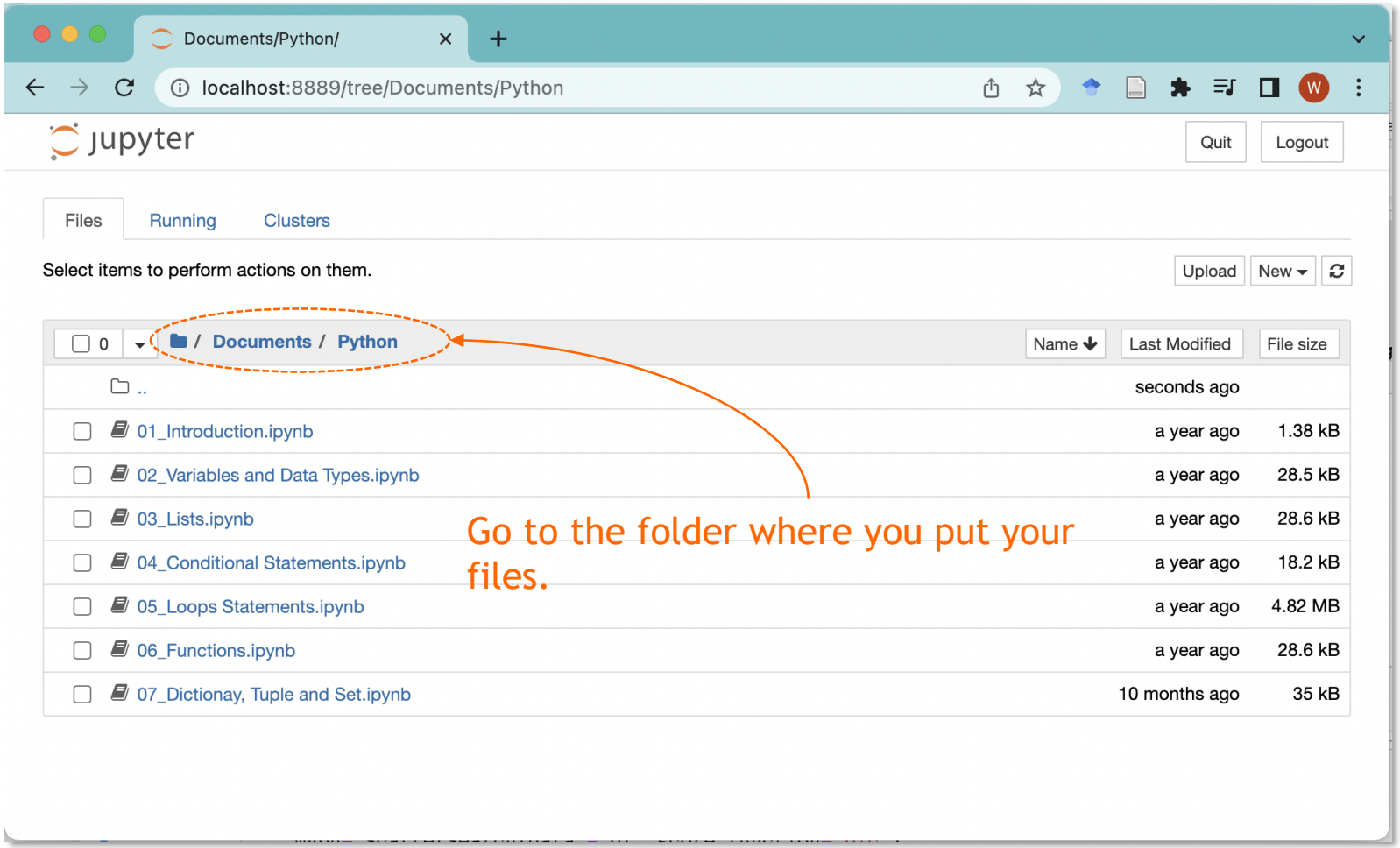
Download notebooks from itslearning



Put your notebook files under a folder.
For example, Users/username/Documents/Python/

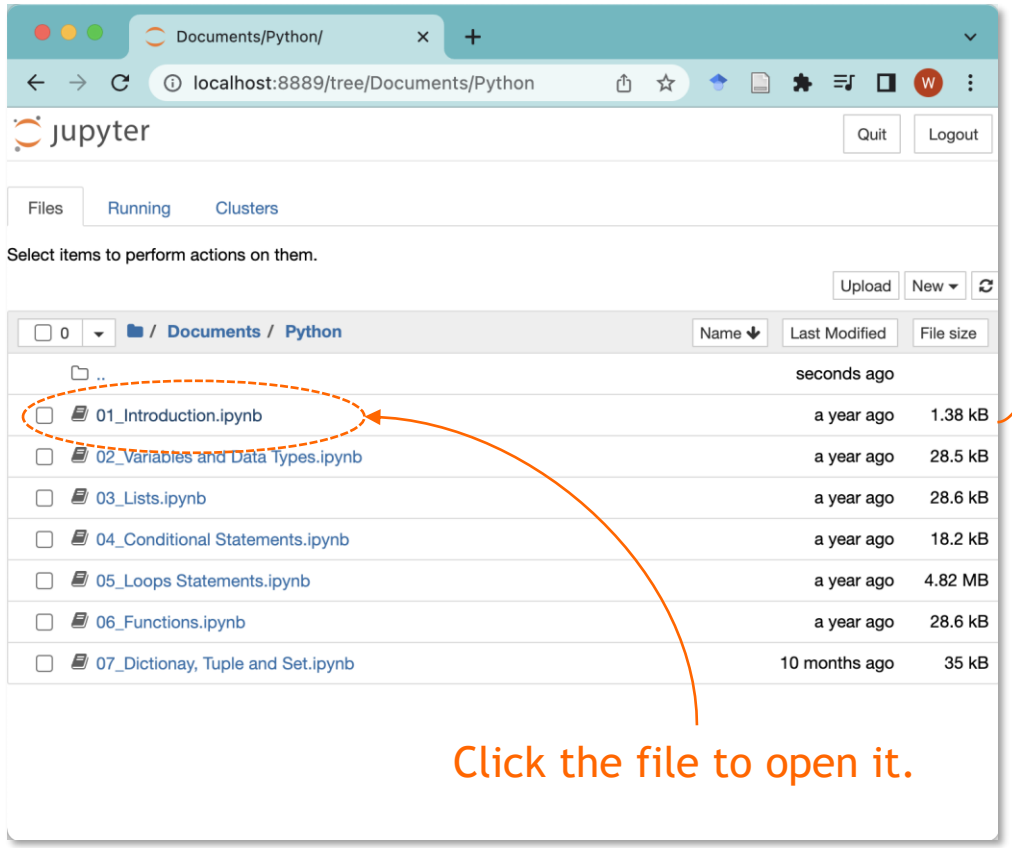
Open an existing Jupyter notebook

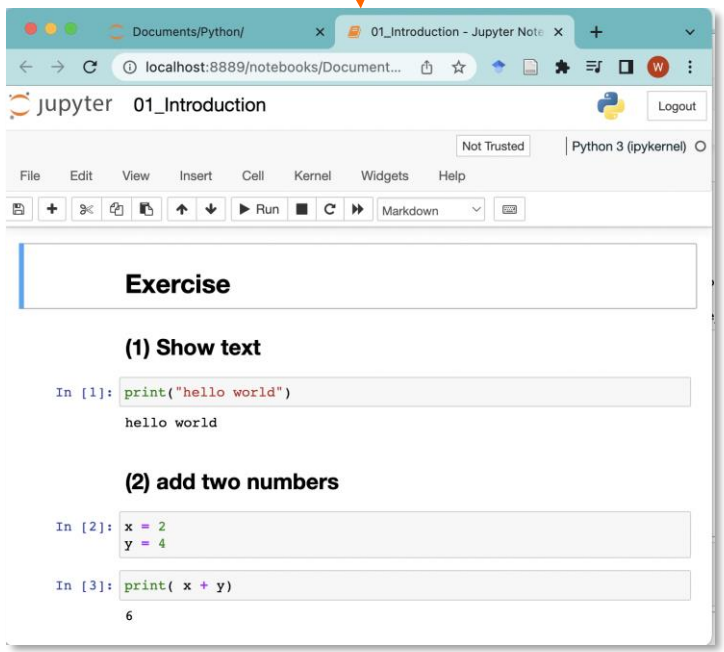
Step3
jupyter notebook
file tree



Open an existing Jupyter notebook

Step4.2
Open an existing notebook

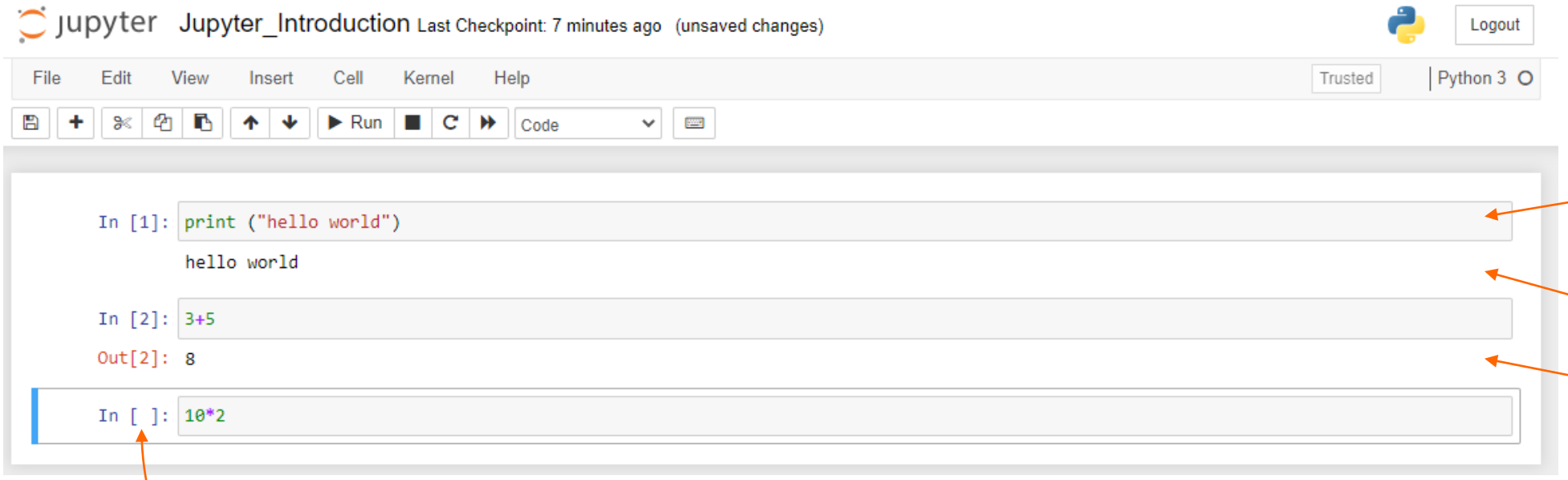




Introduction to Jupyter notebooks

Jupyter Notebook - Cells

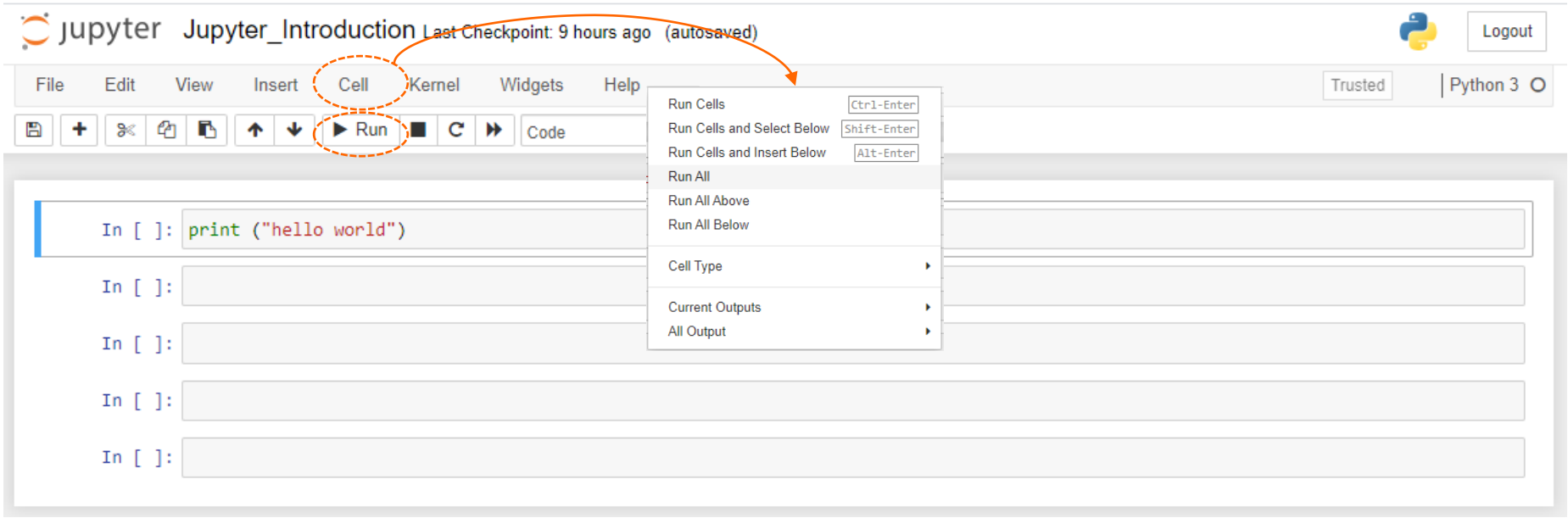
- The entire contents of a notebook is composed of only **cells**, code cells and text cells.



Number: The cell has ~~not~~ been executed and the number is the order of execution.
Asterisk(*): During the execution of a cell.
Blank: The cell has not been executed yet

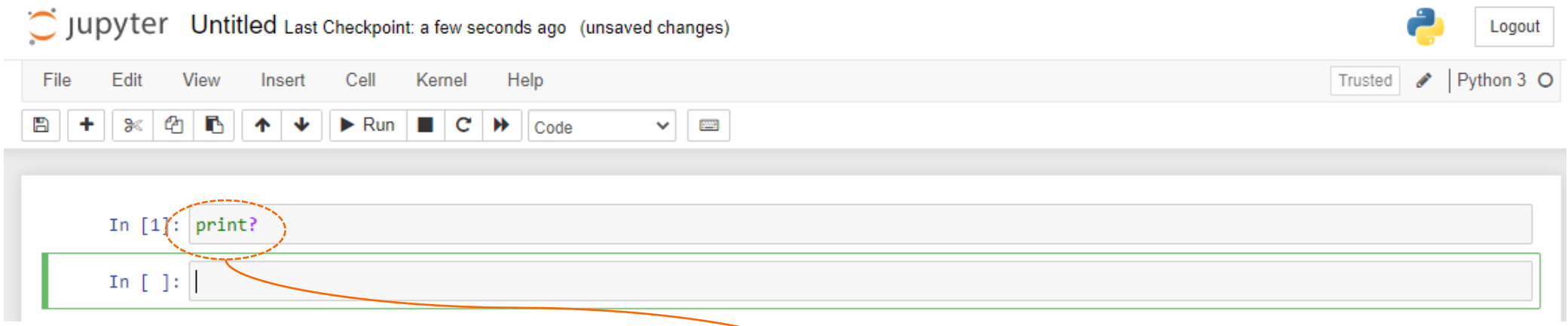
Jupyter Notebook - Execution

- To run a piece of code, click on the cell to select it, then
  - [option1] click the **play button** in the toolbar above.
  - [option2] click the **Cell dropdown menu**
  - [option3] keyboard shortcut **Ctrl+Enter**



Jupyter Notebook - Documentation

- Accessing documentation in the notebook



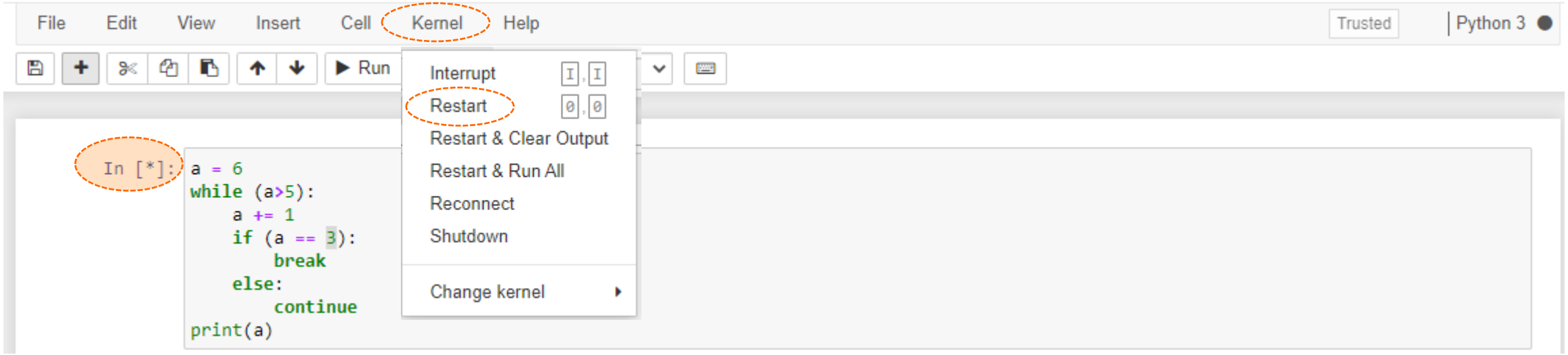

```
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
Type:      builtin_function_or_method
```

A question-mark after a method or function will bring up the documentation after the cell is run

Jupyter Notebook – Restart kernel

When to restart the kernel

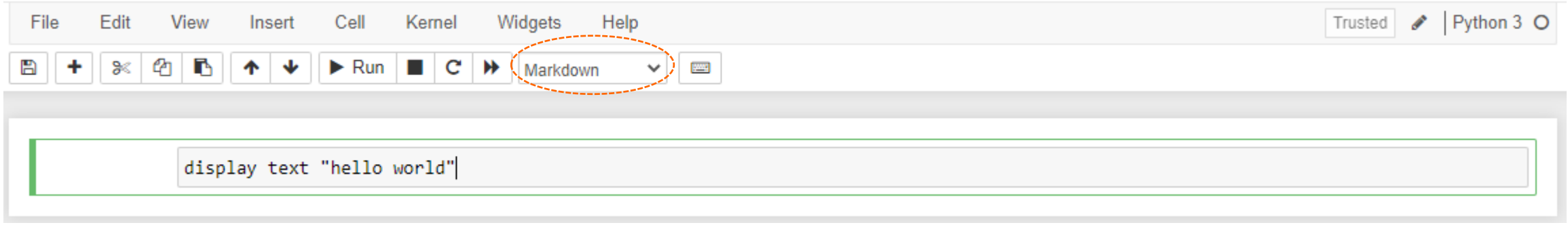
- The notebook is non-responsive.
  - You may write code that can go into an infinite loop.
- To start over a computation from scratch.

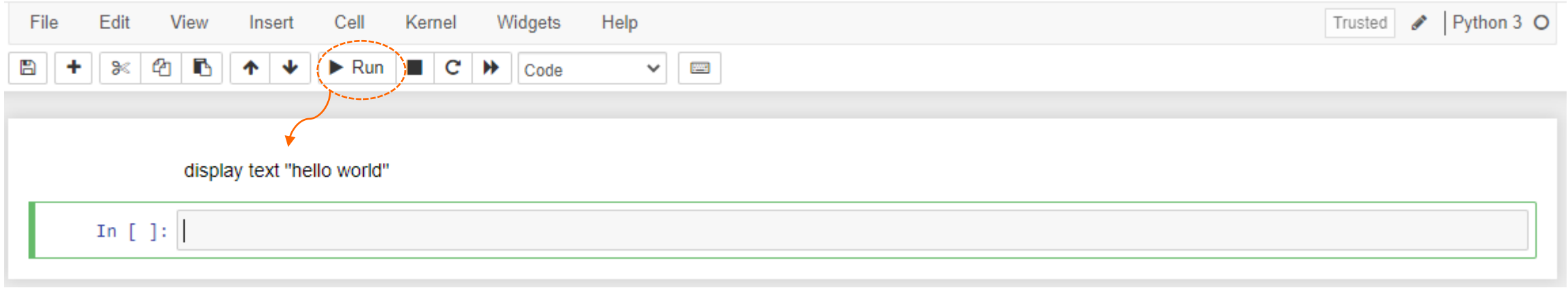


Keyboard shortcut: ESC + 0 0

Jupyter Notebook – Text cells

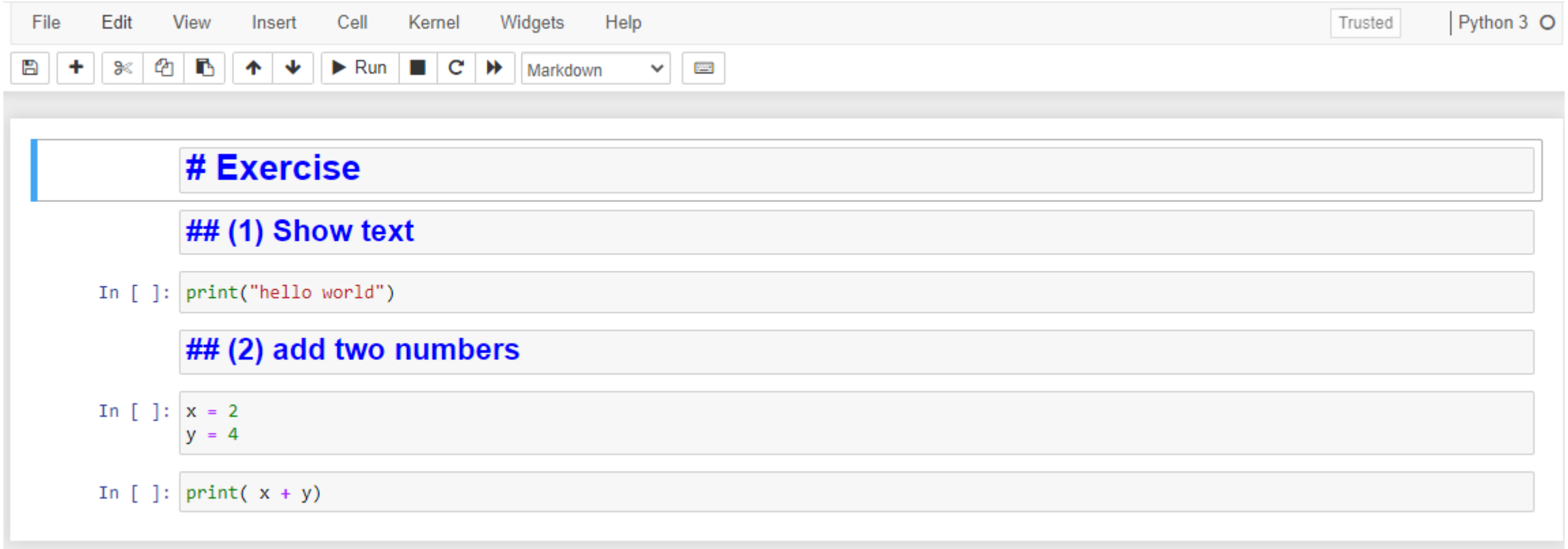
- Text can be added to Jupyter Notebooks using Markdown cells.





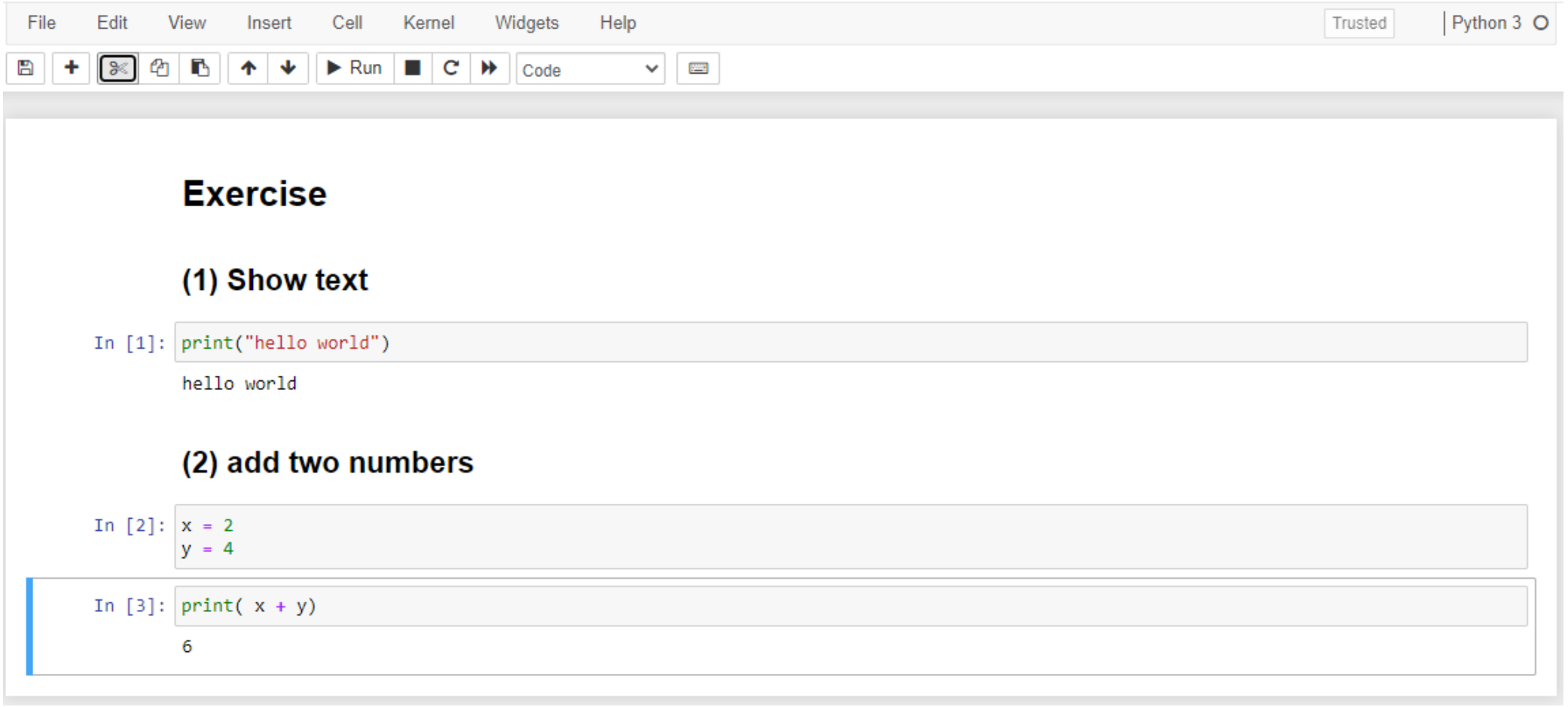
Jupyter Notebook – Text cells

- You can add headings by starting a line with one (or multiple) # followed by a space



https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/Working%20With%20Markdown%20Cells.html

Jupyter Notebook – Text cells



Comments

- As programs get bigger and more complicated, they get more difficult to read. It is a good idea to add notes to your programs to explain what the program is doing.
- Comments
  - Begin with a # character
  - Ignored by Python interpreter
  - Intended for a person reading the code

```
# display the text "hello world"
print ("hello world")
```

```
hello world
```

```
x = 2           #assign 2 to x
y = 5           #assign 5 to y
print (x+y)     #sum of x and y
```

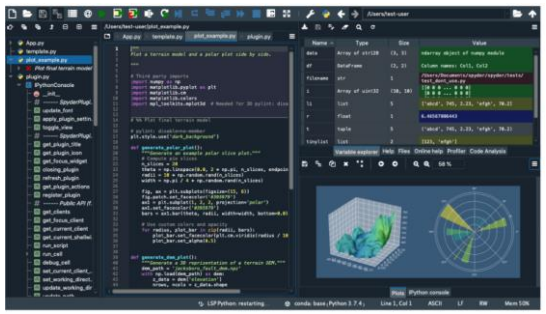
```
7
```

Programming language ≠ Jupyter

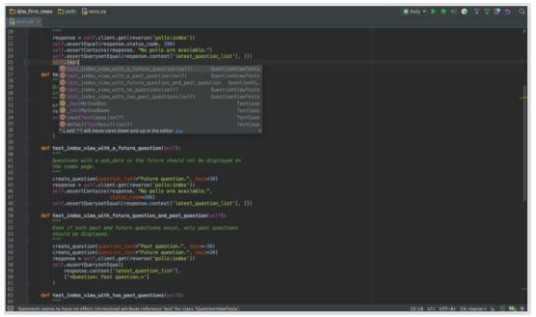
- Does programming require the notebook? No
  - Notebook is just a type of development environment.

- More python development environments
  - Spyder, PyCharm, Thonny, IDLE, VScode, Atom.

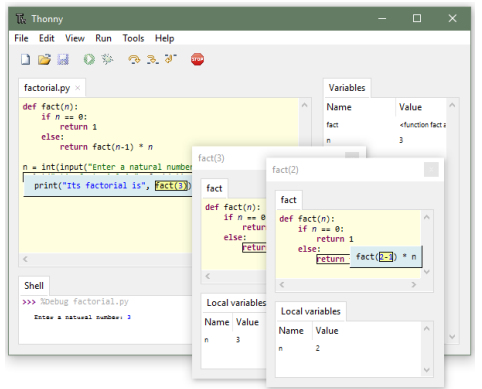
Spyder



PyCharm



Thonny

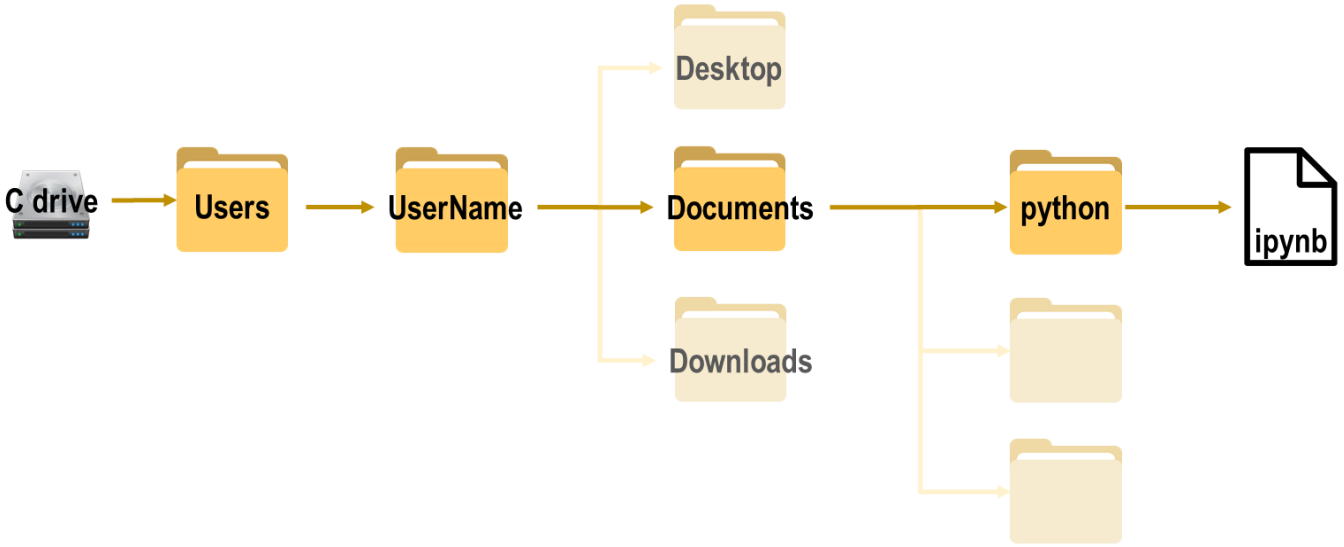


Tips for this course

- Learn by doing, practicing. You cannot learn programming only by following lectures.
- Do not just read solutions and be satisfied when you understand it. Write it yourself.
- Programming can be hard, frustrating and time consuming. Free up space in your calendar every week.
- Different people find different things easy. You are encouraged to help each other, but everyone should program on their own.

Backup

File location



- Use `getcwd()` to get the current working directory of the file.

```
import os
os.getcwd()
```

```
'C:\\Users\\UserName\\Documents\\python'
```