EBA3400
Programming, Data Extraction and Visualization

Wei-Ting Yang

Department of Data Science and Analytics

Outline

- Course information
- Introduction to programming
- Getting started with Jupyter notebooks

Weekly schedule

- Synchronous classroom lectures
  - Every Thursday 10:00-12:00

- Asynchronous learning activities
  - Exercises, quizzes, videos, and reading
  - Solution for the exercises will be uploaded to itslearning every Monday at 15:00

- TA sessions
  - Tuesday 16:00-17:00 (Zacky Dhaffa Pratama)
  - Friday 15:00-16:00 (Danielle Ajoc)

Classroom

week	Lecture	TA session-1	TA session -2
	Thursday 10:00-12:00	Tuesday 16:00-17:00	Friday 15:00-16:00
34	A1-030	D3-141	D3-037
35	B2-040	D3-141	D3-037
36	A1-030	D3-141	D3-037
37	A1-030	D3-141	D3-037
38	A1-030	D3-141	D3-037
39	A1-030	D3-141	D3-037
40	A1-030	D3-141	D3-037
41	A1-030	D3-141	D3-037
42	D3-090	D3-141	D3-037
43	A1-030	D3-141	D3-037
44	C2-010	D3-141	D3-037
45	A1-030	D3-141	D3-037
46	A1-030	D3-141	D3-037
47	A1-030	D3-141	D3-037

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

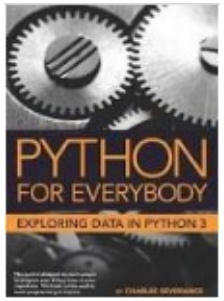
Assessment

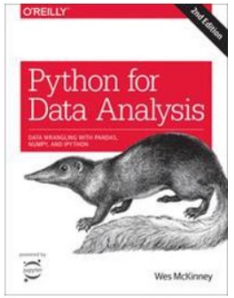
Exam	Weight	Date	Duration	Group/Individual	Exam type
Exam-1	30%	Week 42 Hand-out: 20/10/2023 12:00 Hand-in: 23/10/2023 12:00	3 days	Group(1~3)	Written submission (Jupyter notebook)
Exam-2	70%	Week 48 Hand-out: 28/11/2023 12:00 Hand-in: 28/11/2023 09:00	3 hours	Individual	Written submission (Jupyter notebook)

Other resources (optional)

- Online resources
  - https://docs.python.org/3/tutorial/index.html
  - https://www.kaggle.com/learn
  - https://www.geeksforgeeks.org/python-programming-language/?ref=shm
  - https://www.w3schools.com/python/default.asp
  - https://automatetheboringstuff.com/

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





Outline

- Course information
- Introduction to programming
- Getting started with Jupyter notebooks

What's Programming

Programming is the process of giving instructions to a computer to perform specific 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** is used to write a set of instructions that computers can understand.

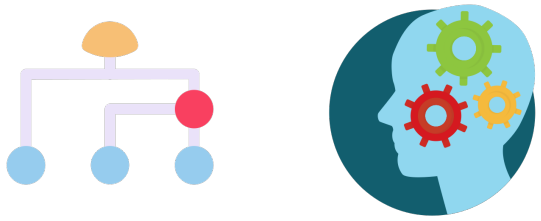
Python
JavaScript
C#
Java
Swift
Scala
C++
Go
C
Ruby
Rust
Kotlin
Julia
R
Matlab

Why we should learn programming

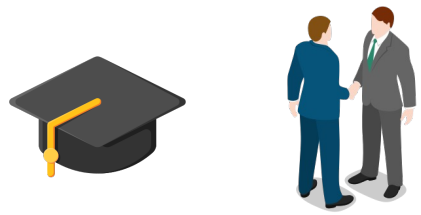
Understand the technology is our daily life

Make tasks easier for you

Develop structured thinking and logical skill

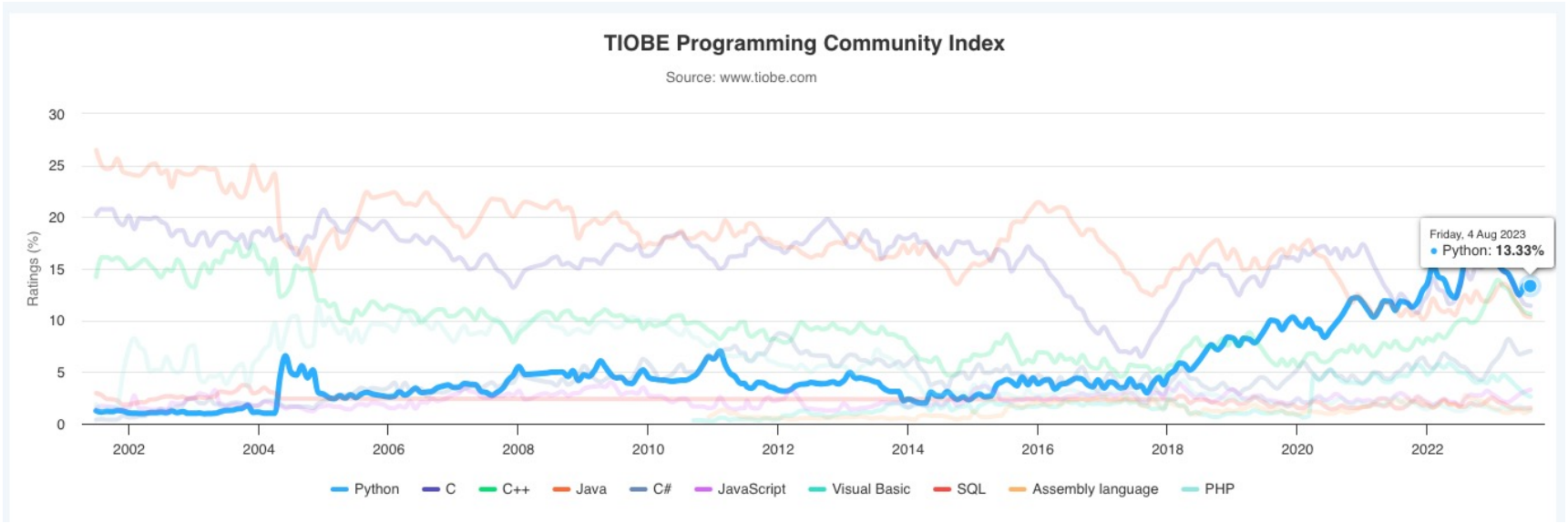


Make your job application stand out



Python

- Python is a high-level programming language known for its simple syntax.
- Created by Guido van Rossum and first released in 1991.
- Python is widely used in various fields such as data analysis, web development and scientific computing.



2023 Top3

1. Python (13.33%)
2. C (11.41%)
3. C++ (10.63%)

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

Get your programming skills ready

1. Year

Autumn
- Fundamentals of Accounting and Finance
- Organizational Behavior and Leadership
- Economics 1
- Programming, Data Extraction and Visualisation
- Mathematics for Data Science

Spring
- Mathematics for Data Science
- AI and Data Ethics
- Statistics with Programming
- Databases

2. Year

Autumn
- Economics 2
- Data Analysis with Programming
- Decision modelling using spreadsheet
- Data driven management accounting

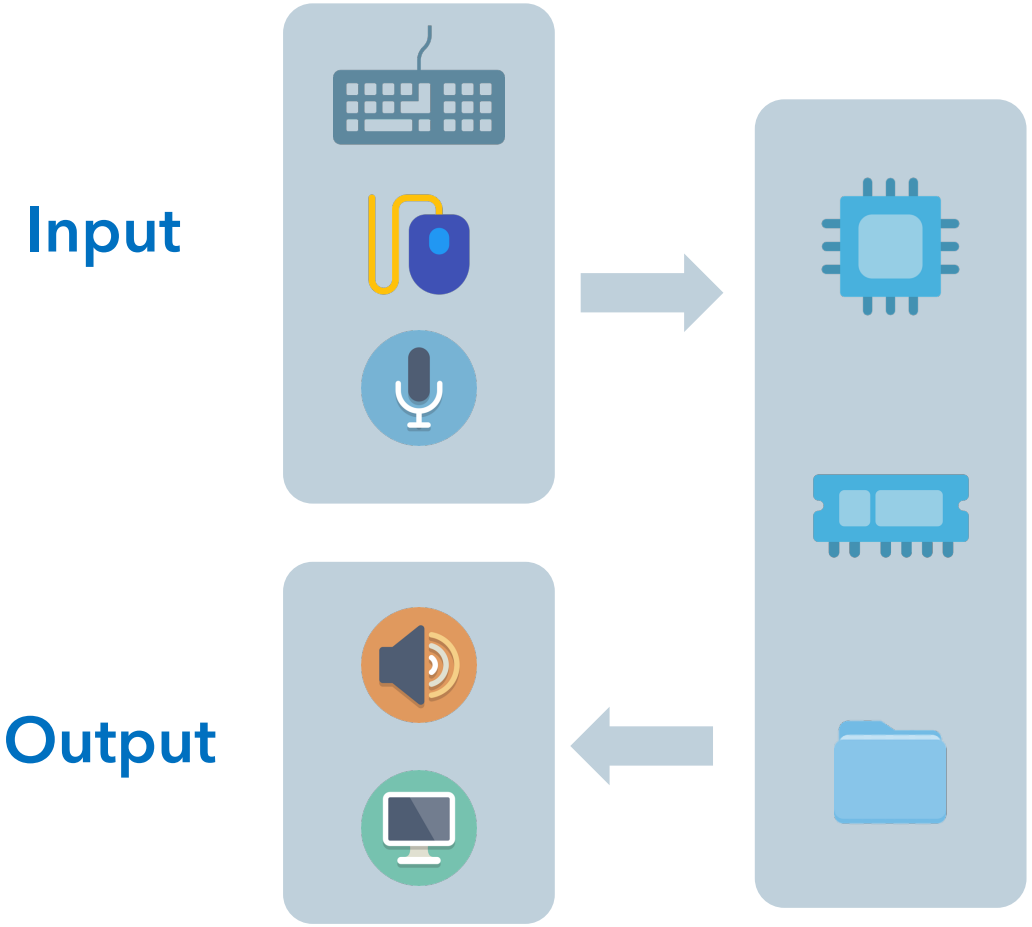
Spring
- Basic Financial Management
- Marketing and Strategy
- Electives 1 and 2 / Internship

3. Year

Autumn
- Electives 3, 4, 5 and 6 / Internship / Exchange

Spring
- Doing Sustainable Business
- Causality, Machine learning and Forecasting
- Marketing Analytics
- Quantitative Economics

How computers work



Processing & Storage

Central Processing Unit (CPU)
- Executes instructions and performs calculations.

Main memory (RAM)
- Provides fast and temporary storage for data and instructions currently in use by the CPU.
- Data will be deleted when the computer is turned off.

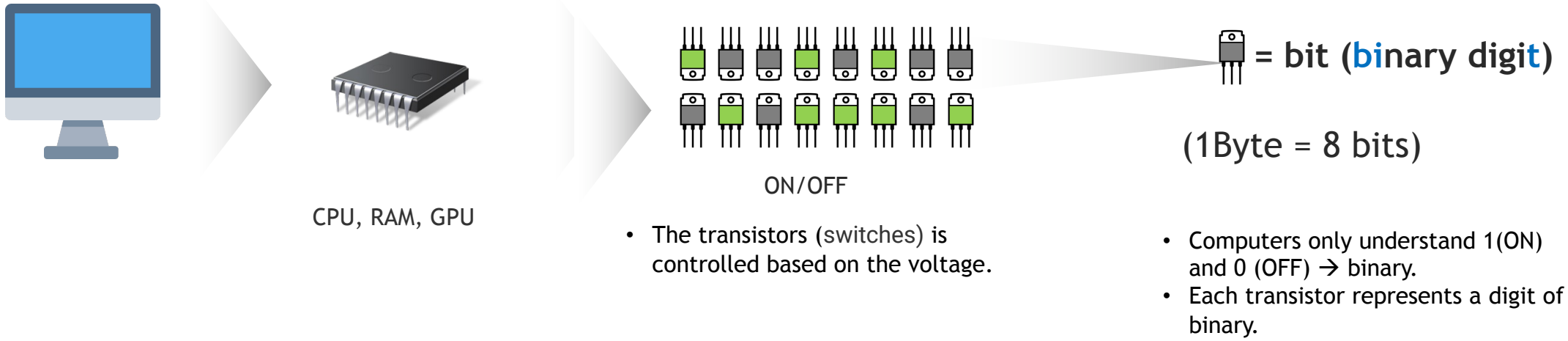
Secondary memory
- Provides long-term storge for programs and data.

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.



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**

$$(5\times1000) + (1\times100) + (3\times10) + (6\times1) = 5136$$

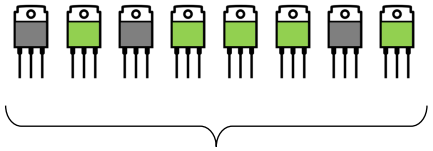
Binary system

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

Binary to Decimal

$$(1\times8) + (0\times4) + (0\times2) + (1\times1) = 9$$

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 scheme that assigns unique numerical values (in binary) to different characters.

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

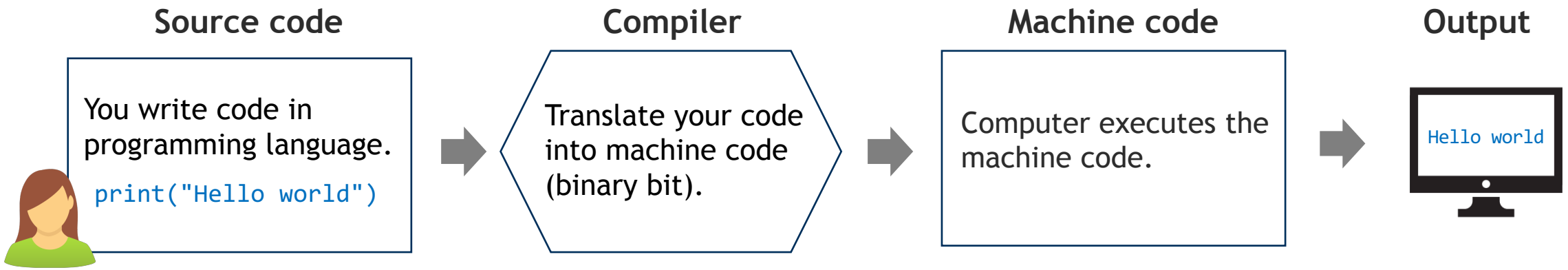
```
01101000
01100101
01101100
01101100
01101111
00100000
01110111
01101111
01110010
01101100
01100100
```

hello world

- https://www.ascii-code.com/
- Modern systems have expanded on ASCII with more complex character encodings like UTF-8 and UTF-16, which support a wider range of characters from different languages and even emojis.

Programming language

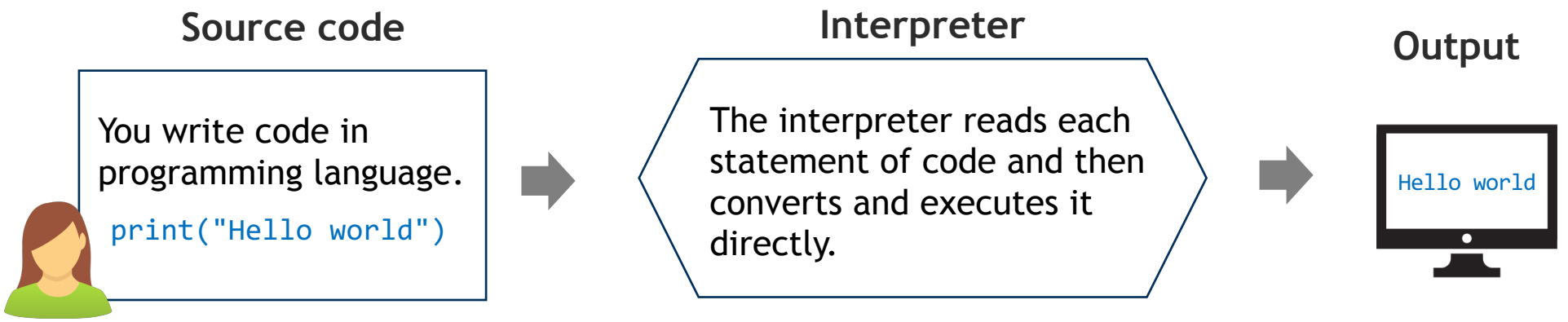
- Programming languages provide an interface between programmers and machine language (binary code).
- You can use syntax that is **English like** and **easier to understand** to express what you want the computer to do.



Programming language

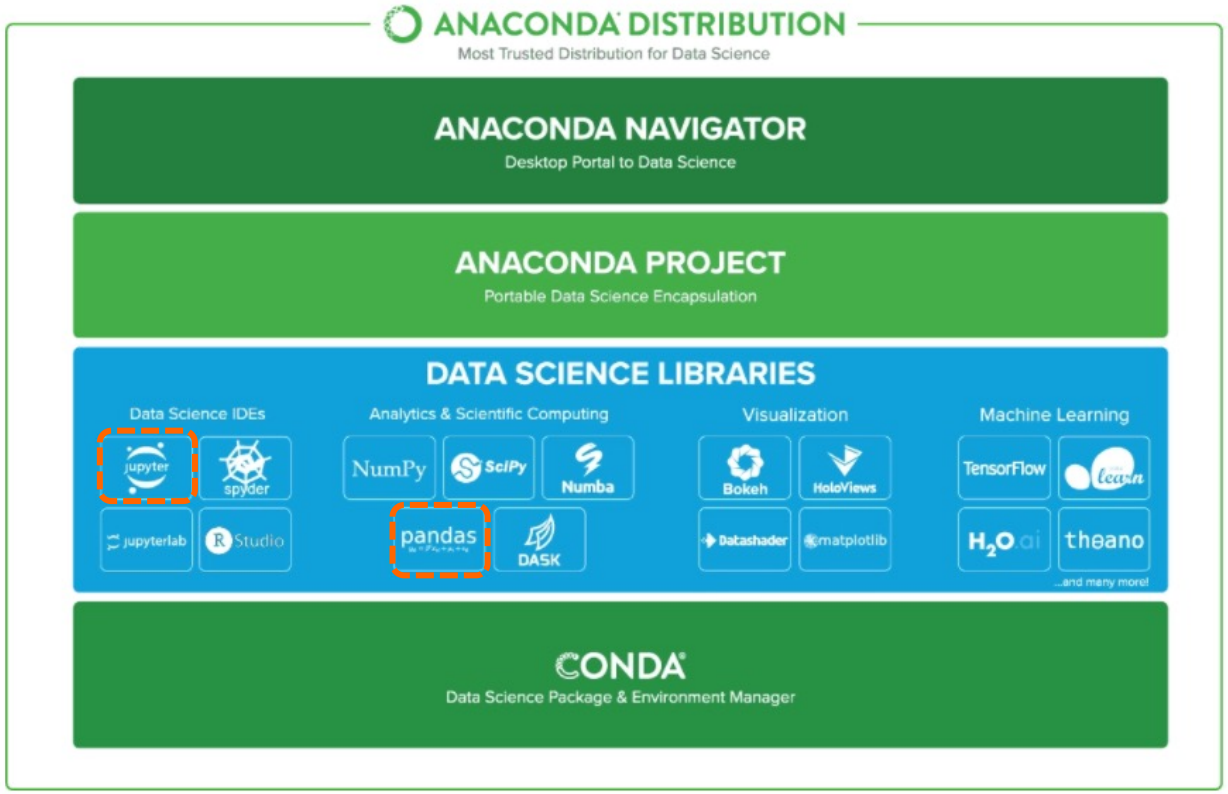
- **A Compiled language**: A programming language whose programs are typically translated into machine language by a compiler before being executed. (e.g., C, C++)

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



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.

Outline

- Course information
- Introduction to programming
- Getting started with Jupyter notebooks

Project Jupyter

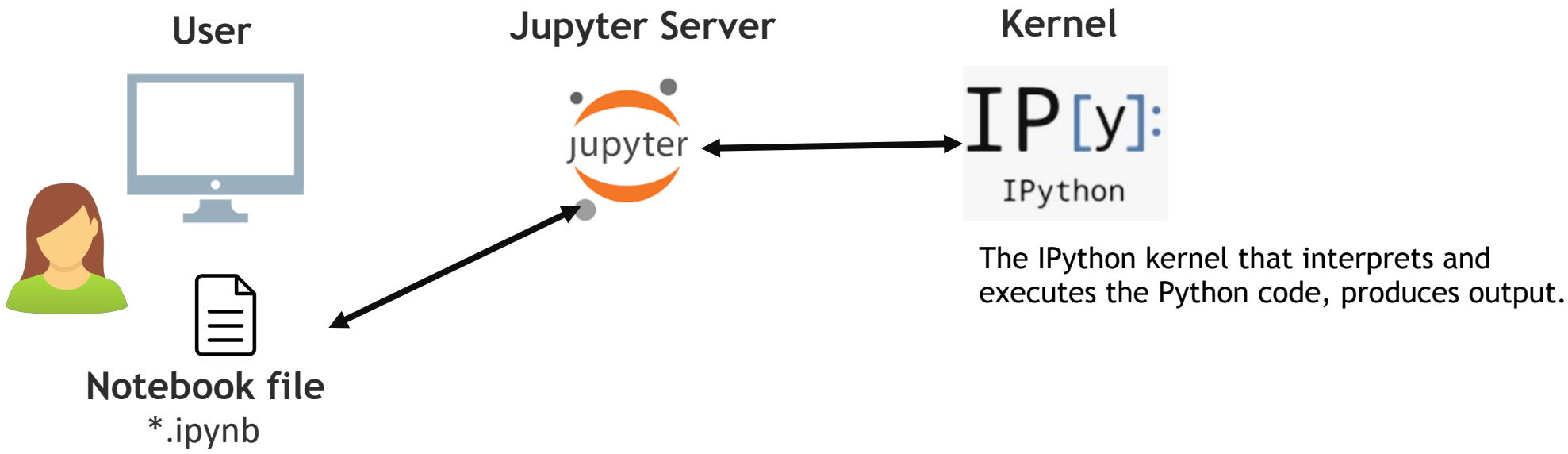
- A project and community whose goal is to "develop **open-source** software, open-standards, and services for interactive computing across dozens of programming languages“
- “Jupyter” is a reference to the three core programming languages supported by Jupyter, which are **Julia, Python, R.** Nowadays,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

- Jupyter 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

- Jupyter Notebooks are a web-based interactive coding environment that allows you to create and share documents that contain code, equations, visualizations and text.

How do Jupyter Notebooks work?



The IPython kernel that interprets and executes the Python code, produces output.

- Jupyter notebooks are automatically saved and stored on disk in the open-source JavaScript Object Notation (JSON) format and with a .ipynb extension.
- The kernel is the computational engine that executes the code and produces results. Each notebook has its own kernel, which can be a different programming language (Python, R, Julia, etc.)

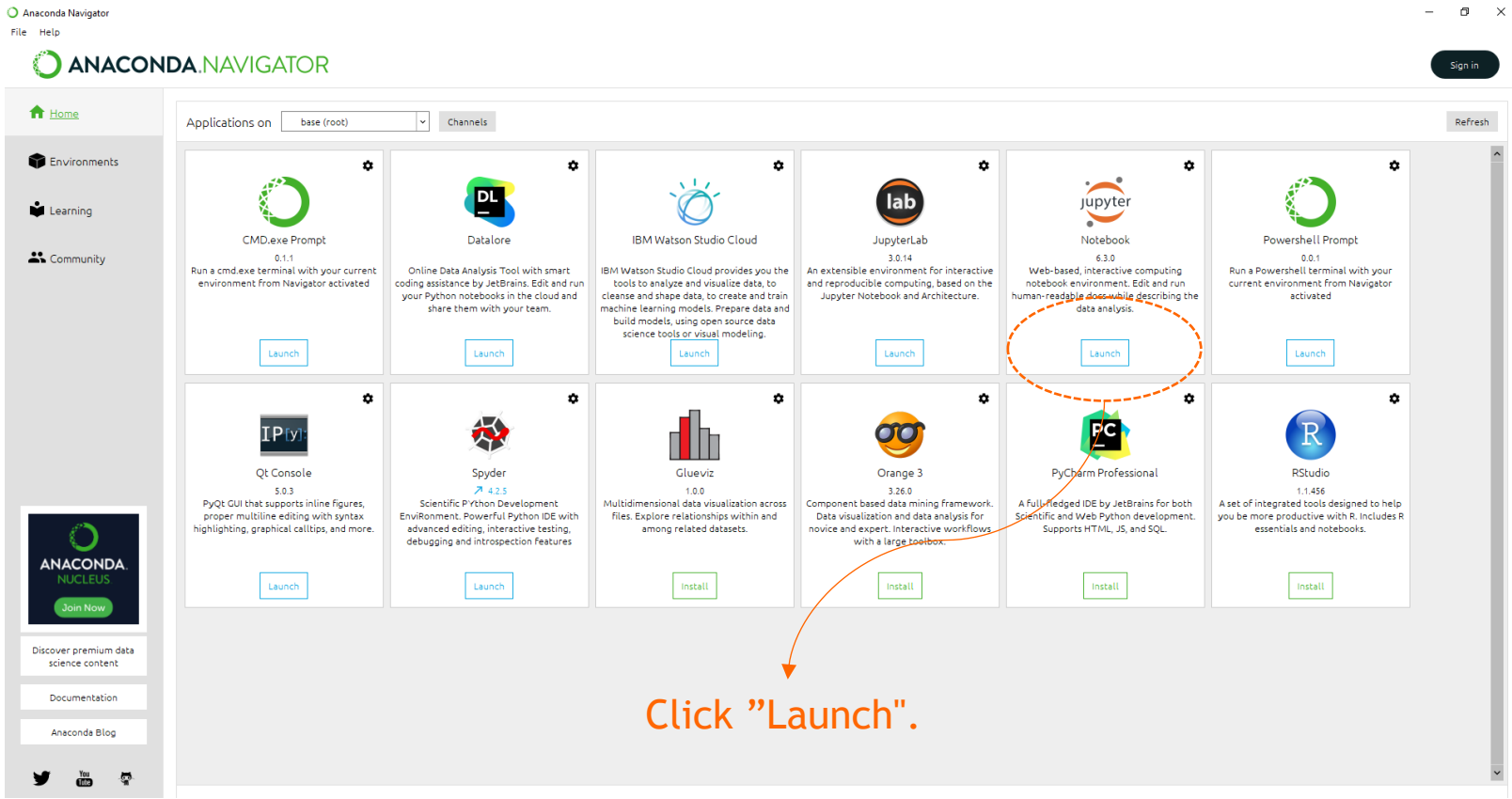
How to create or access a jupyter notebook?

Launch jupyter notebook

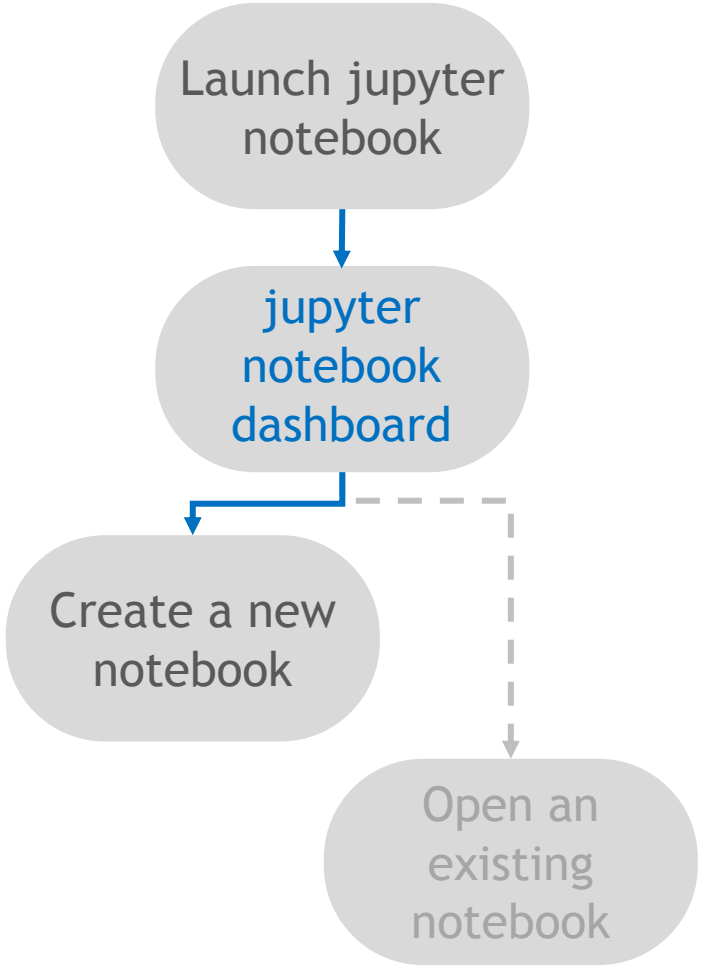
jupyter notebook dashboard

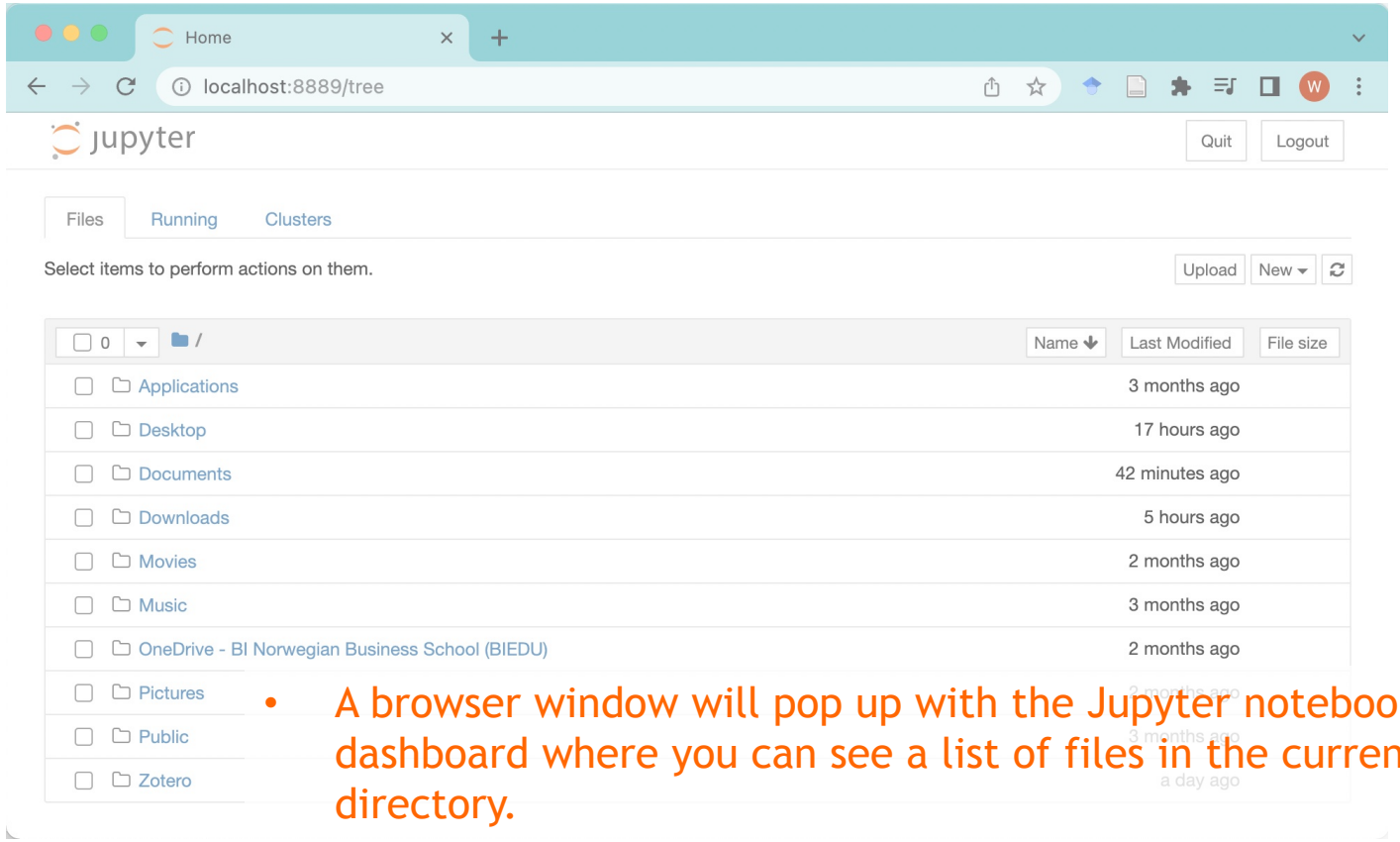
Create a new notebook

Open an existing notebook



How to create or access a jupyter notebook?





- A browser window will pop up with the Jupyter notebook dashboard where you can see a list of files in the current directory.

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

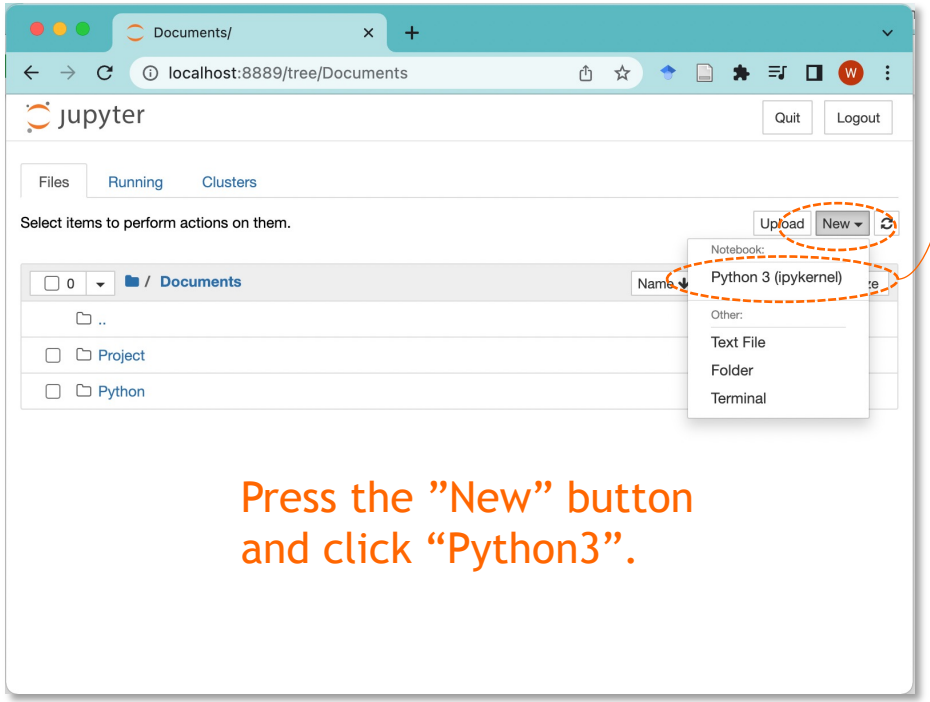
How to create or access a jupyter notebook?

Launch jupyter notebook

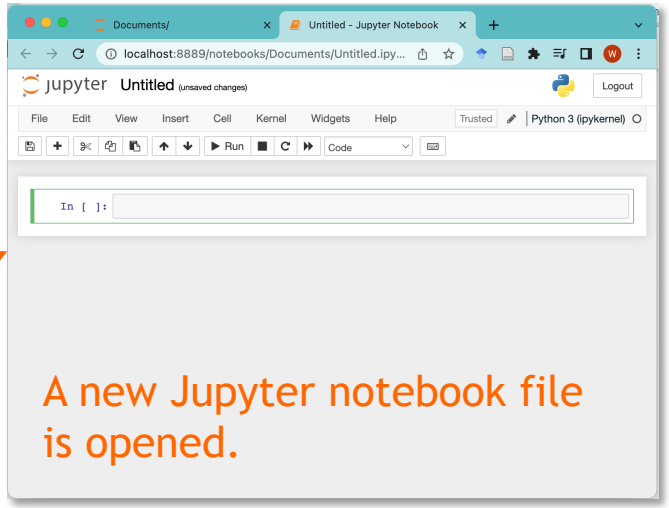
jupyter notebook dashboard

Create a new notebook

Open an existing notebook



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



A new Jupyter notebook file is opened.

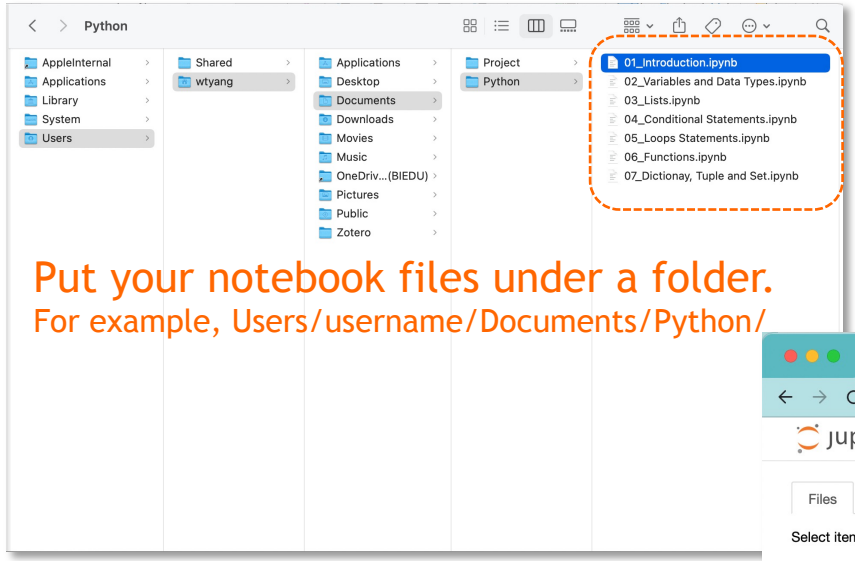
How to create or access a jupyter notebook?

Launch jupyter notebook

jupyter notebook dashboard

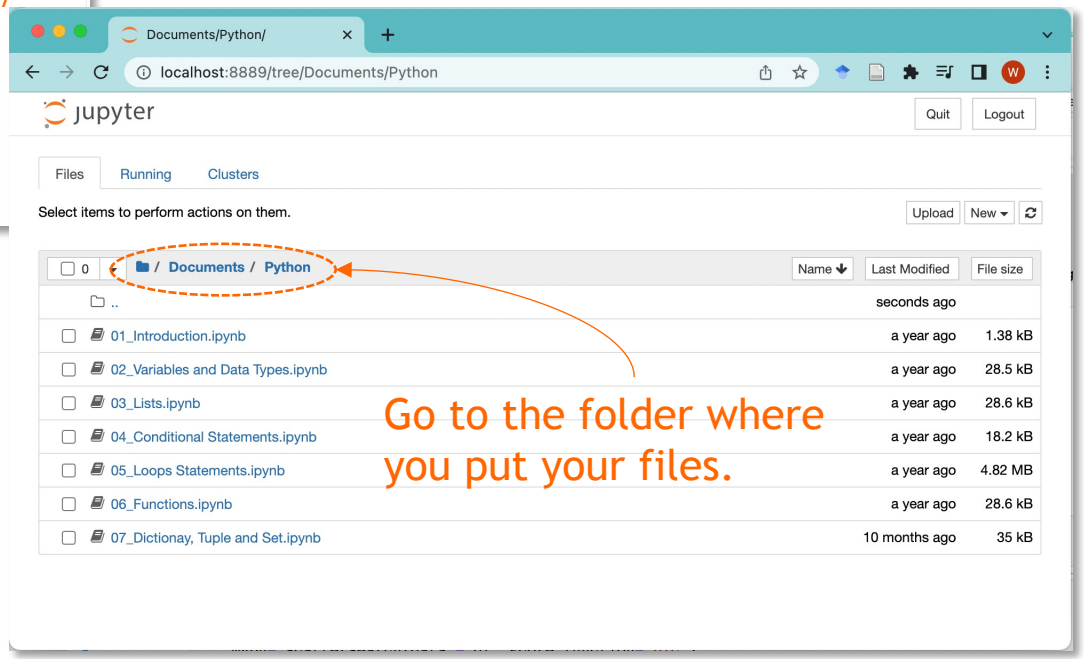
Create a new notebook

Open an existing notebook



Put your notebook files under a folder.

For example, Users/username/Documents/Python/



Go to the folder where you put your files.

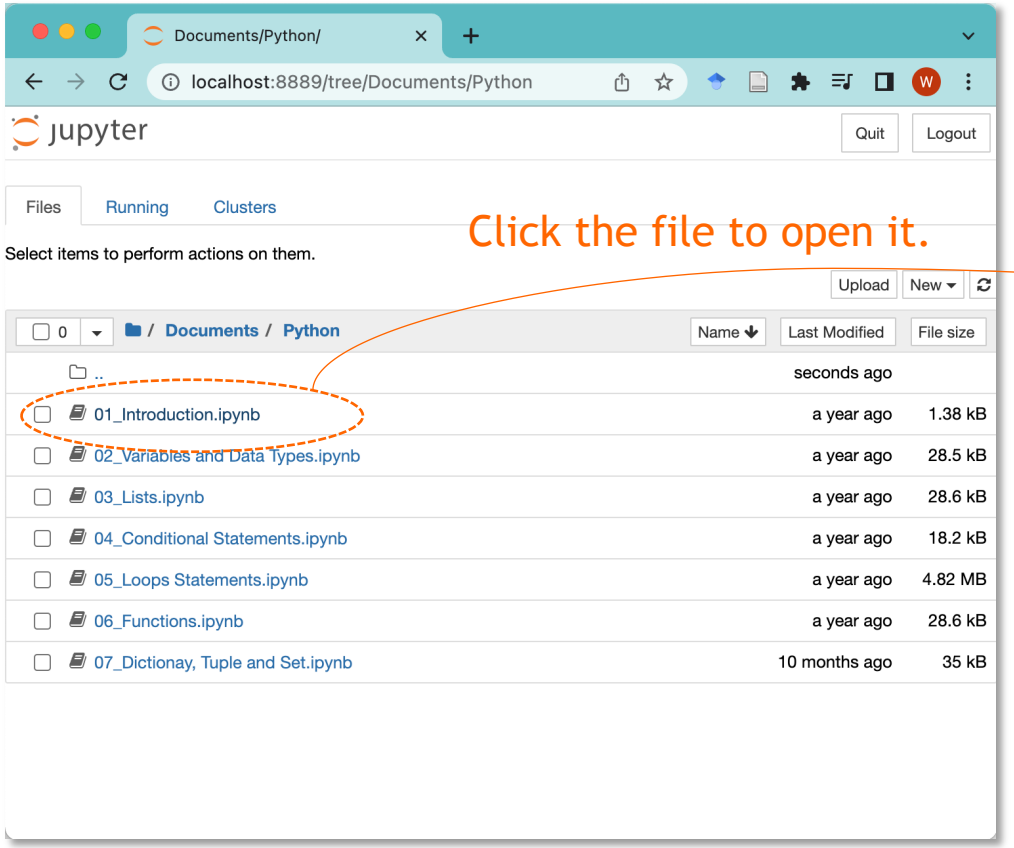
How to create or access a jupyter notebook?

Launch jupyter notebook

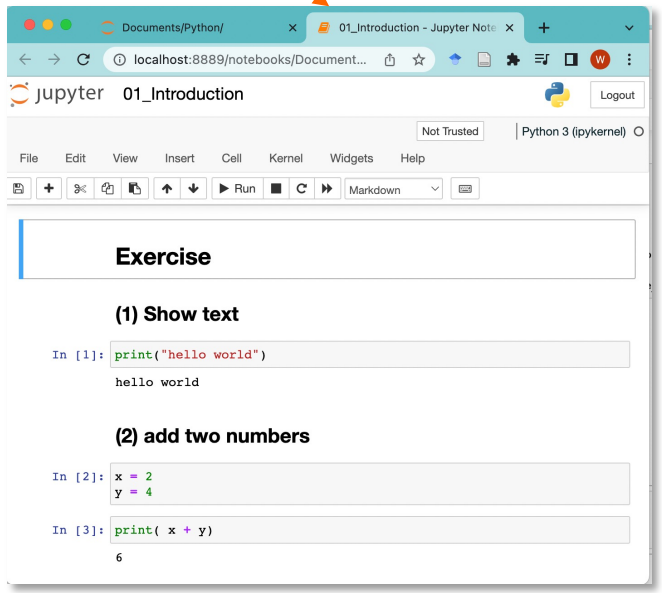
jupyter notebook dashboard

Create a new notebook

Open an existing notebook

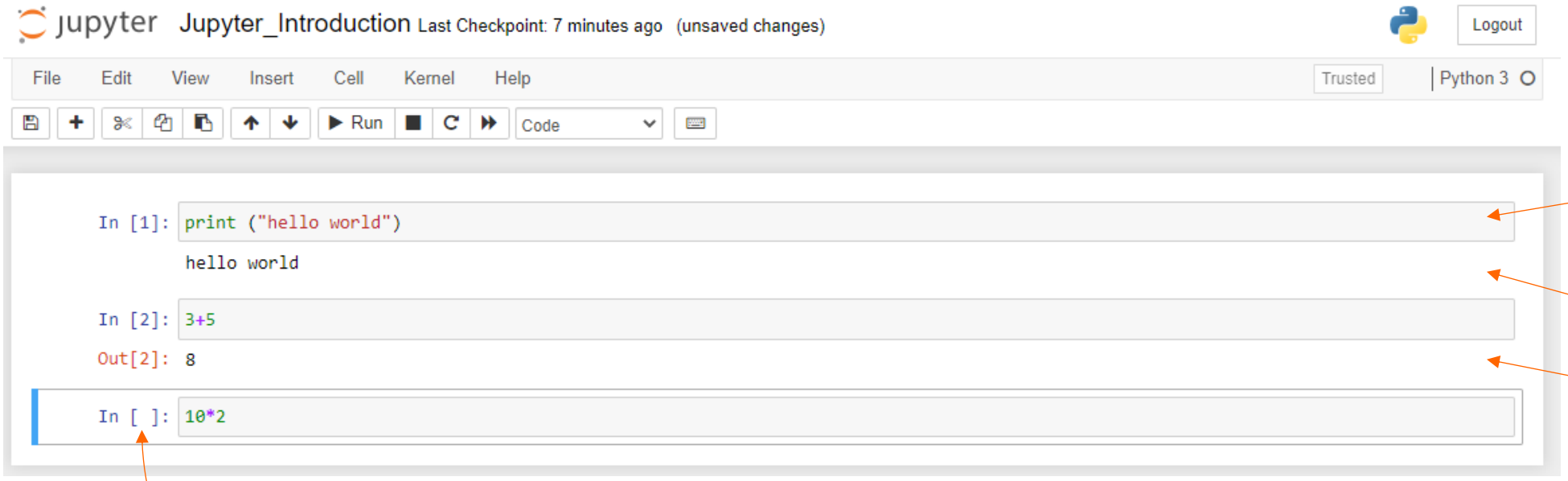


Click the file to open it.



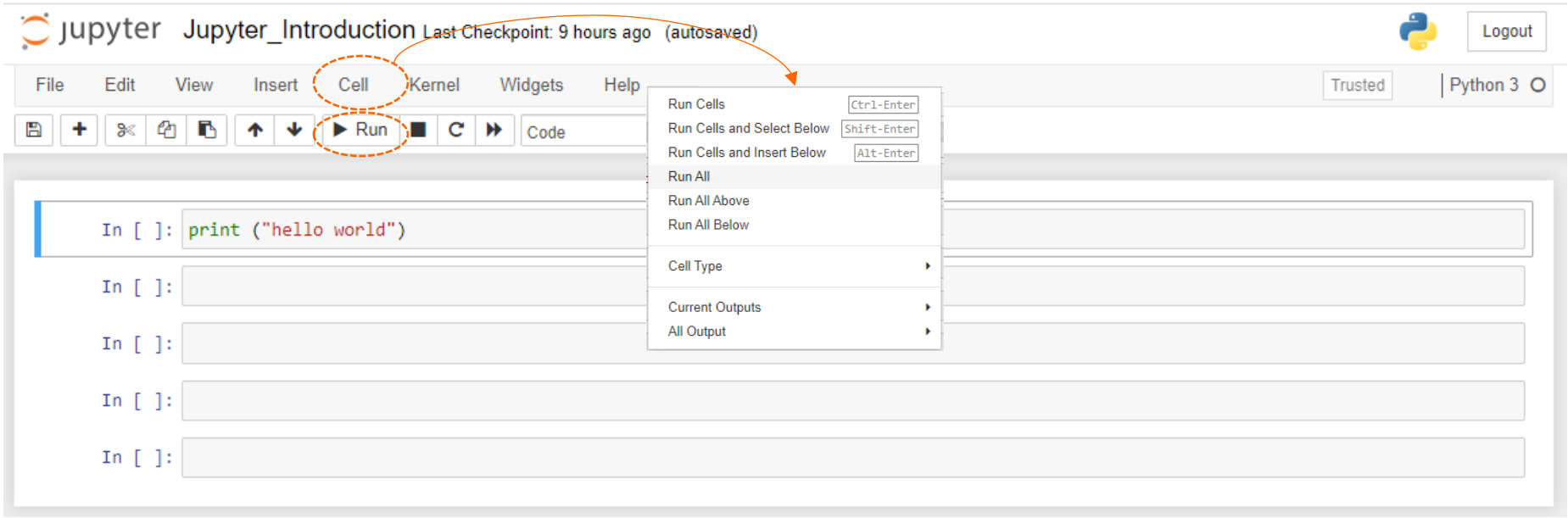
Code cells

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



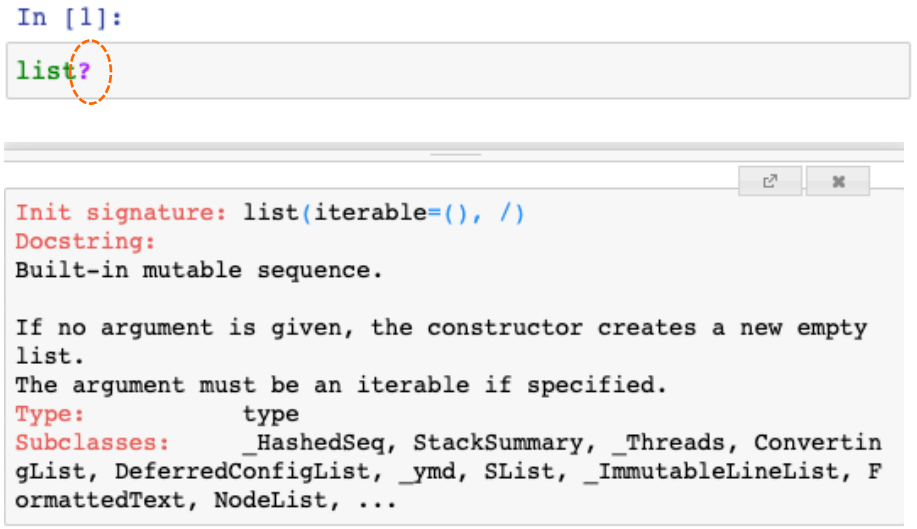
Execution

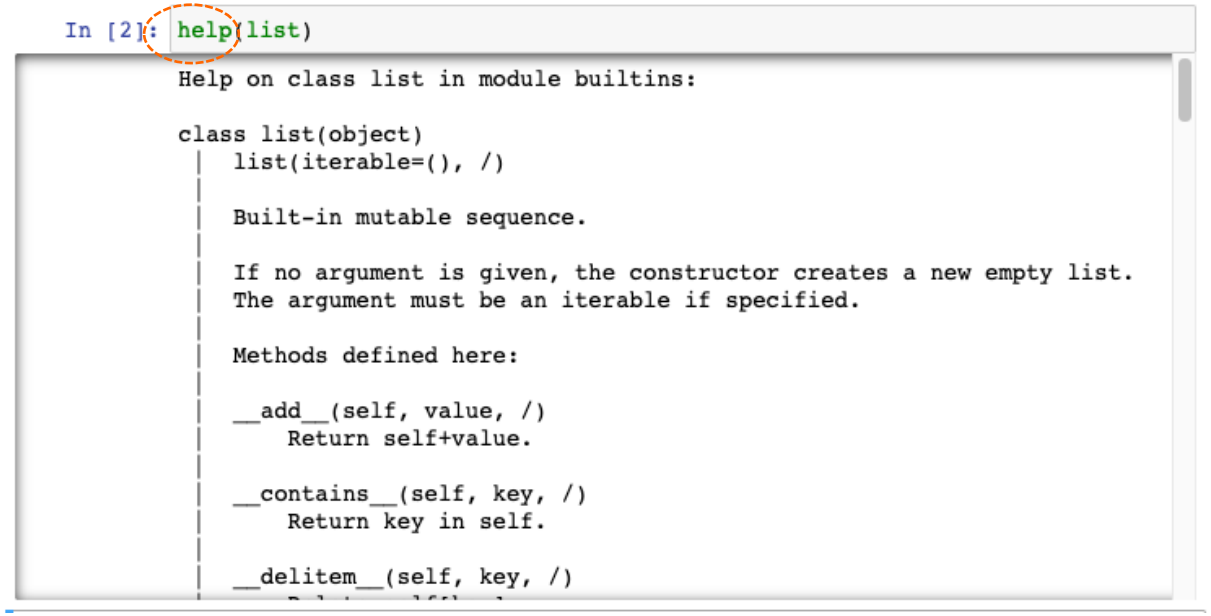
- To run a piece of code, click on the cell to select it, then
  - [option1] click the play button in the toolbar above.
  - [option2] click the Cell dropdown menu
  - [option3] keyboard shortcut Ctrl+Enter or Shift+Enter.



Documentation

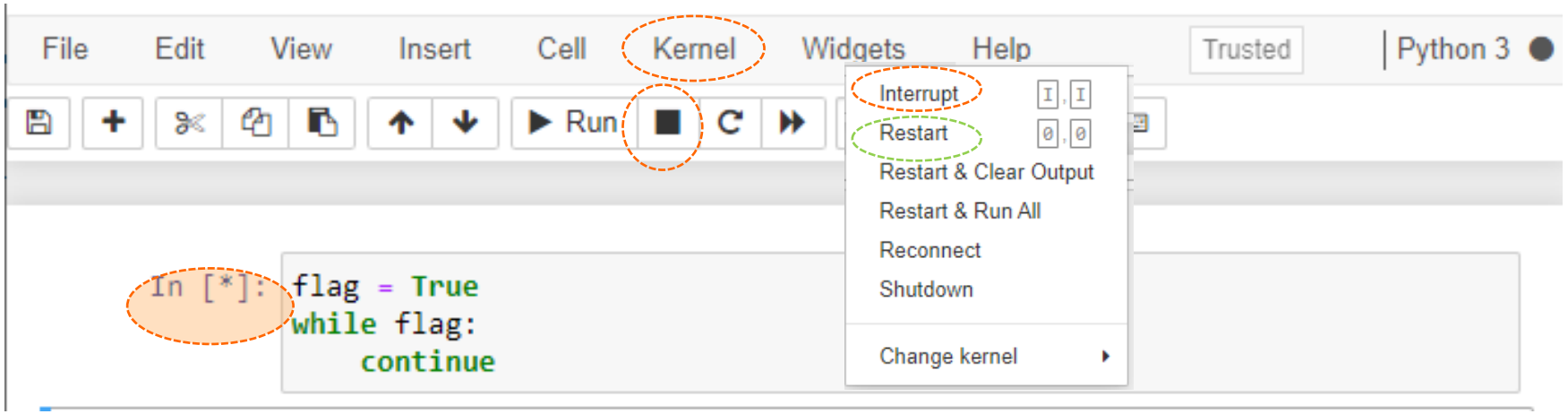
- Access documentation in Jupyter notebook.
  - `?`: Provides a quick view of basic information.
  - `help()`: Provides more comprehensive and detailed information.





Interrupt or restart kernel

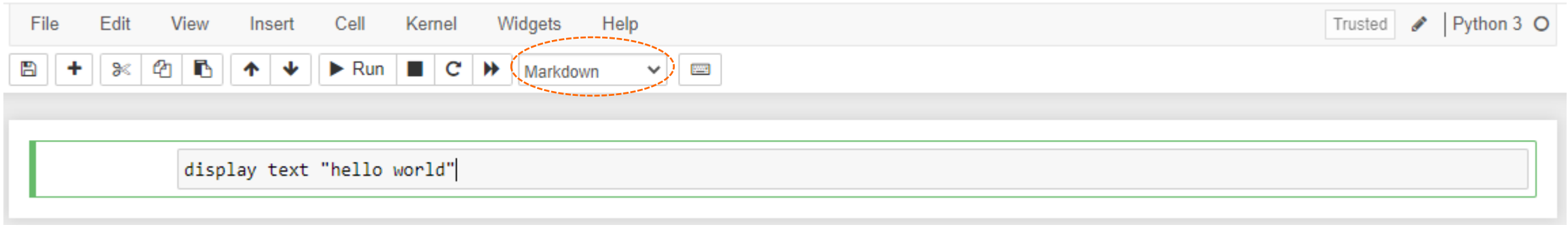
- Interrupt kernel
  - Stop the execution of code that is currently running in a cell.
  - You can interrupt the kernel when the current execution takes too long time.

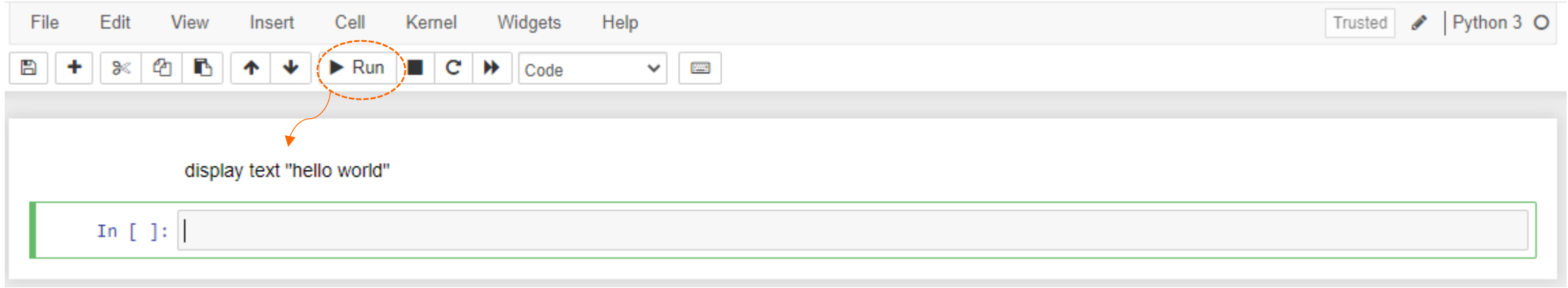


- Restart kernel
  - Reset the notebook's environment.
  - You can restart the kernel when you want to delete all the variables, functions, and data stored in the memory.

Text cells

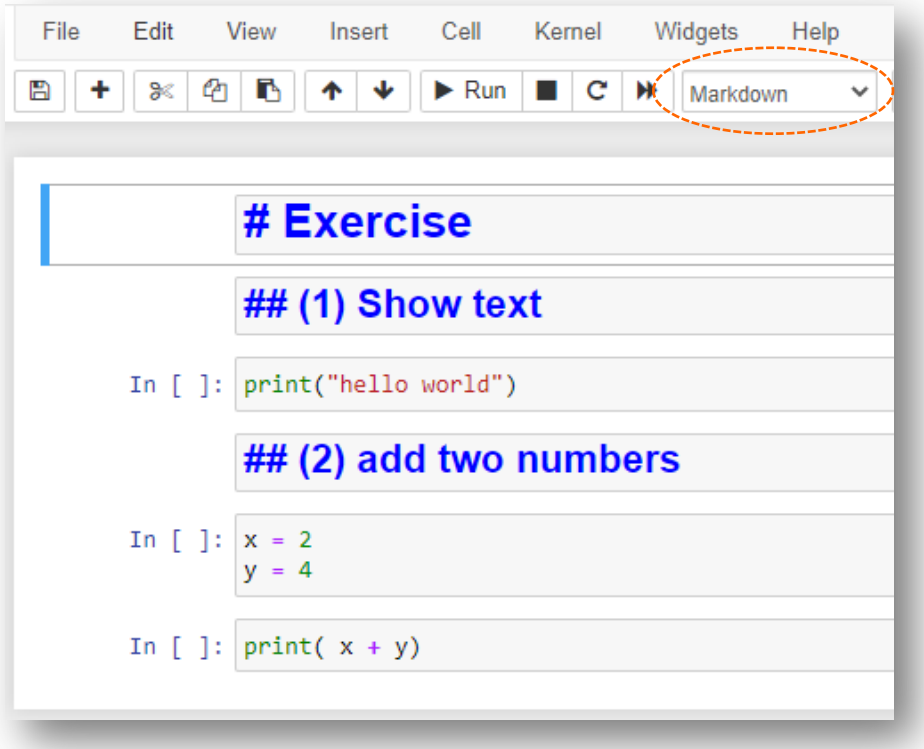
- Text can be added to Jupyter Notebooks using Markdown cells.

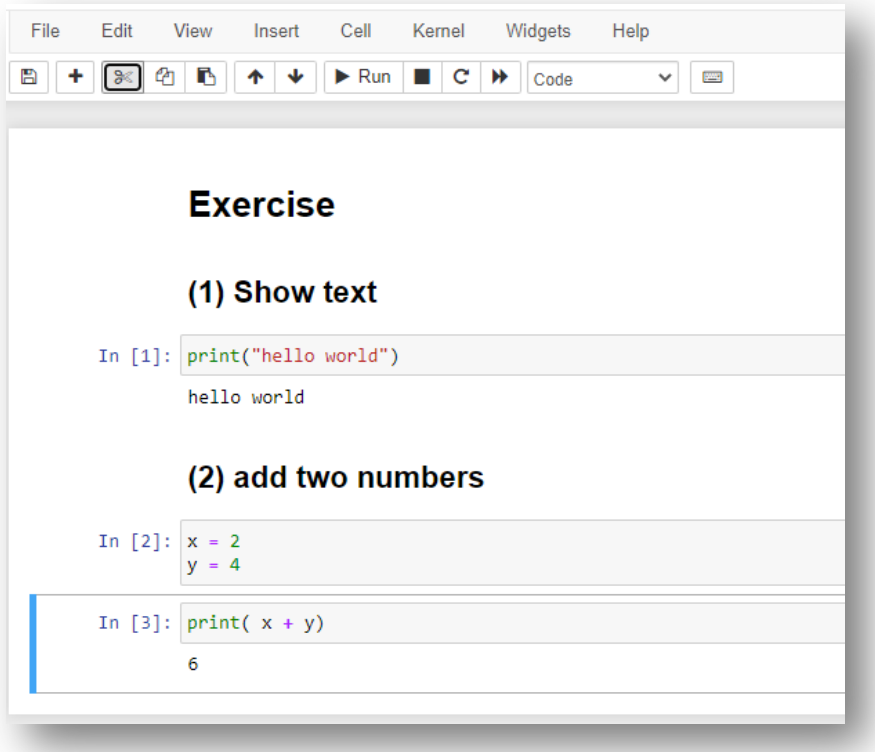




Text cells

- 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

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
```

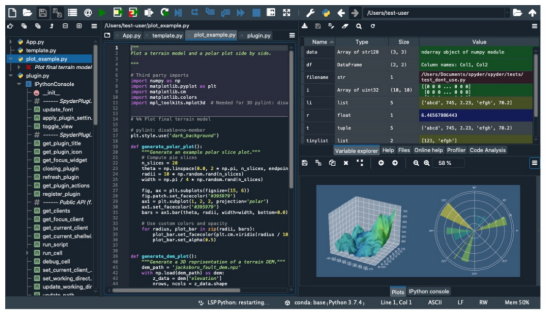
```
7
```

Programming language ≠ Jupyter notebook

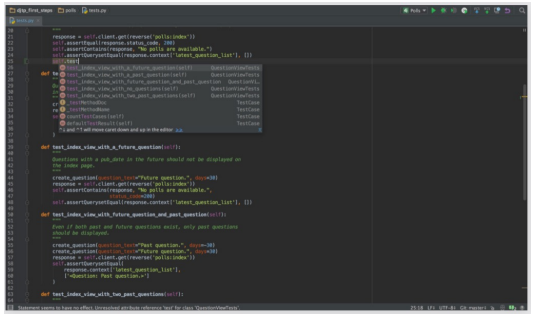
- Is a jupyter notebook necessary for programming? No
  - A notebook is simply one type of coding environment.

- More coding environments
  - Spyder, PyCharm, Thonny, IDLE, VScode, Atom.

Spyder



PyCharm



Thonny