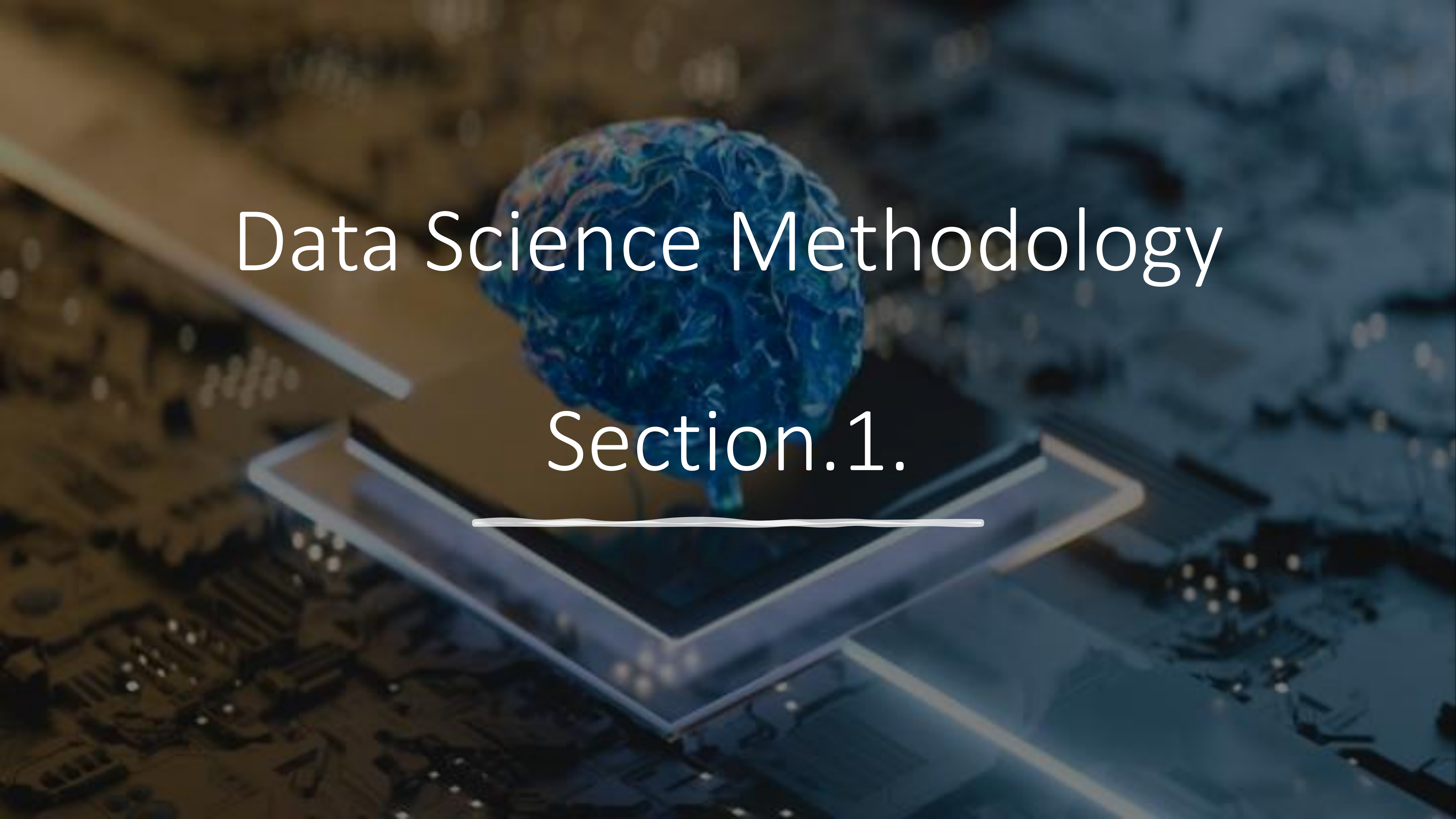


Data Science Methodology

Section.1.

Agenda

- Introduction To python.
- Download Anaconda.
  - Python IDEs.

Python programming language

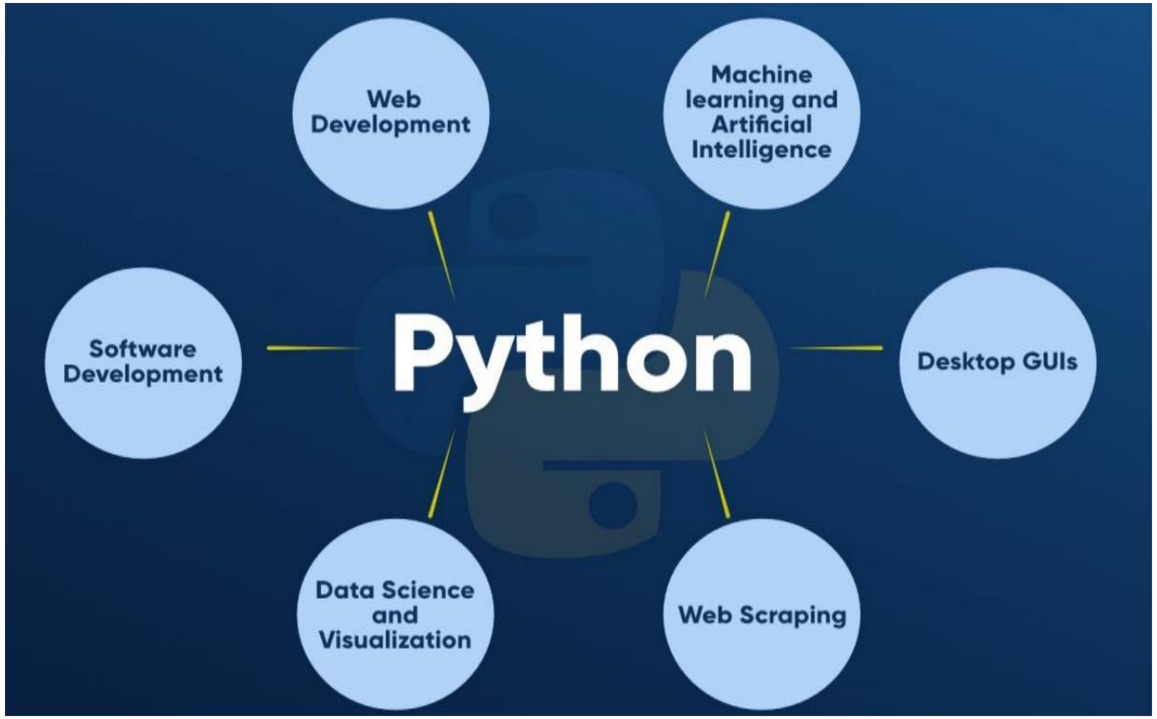
What is Python?

- Python is an open-source, object-oriented, and high-level programming language for web and app development.
- The language is very simple and easy to learn.
- The language was founded by developer Guido Van Rossum in 1991.

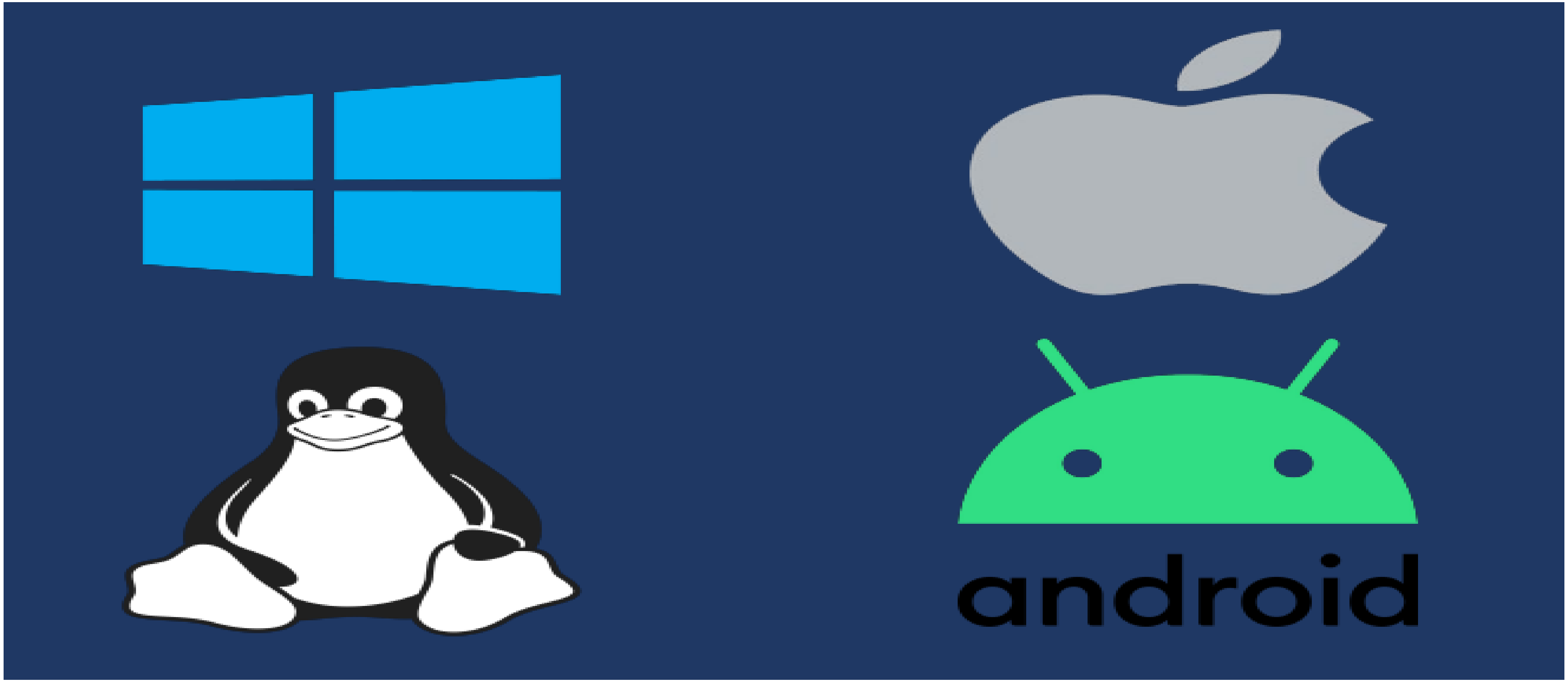
Python Features

Python Features

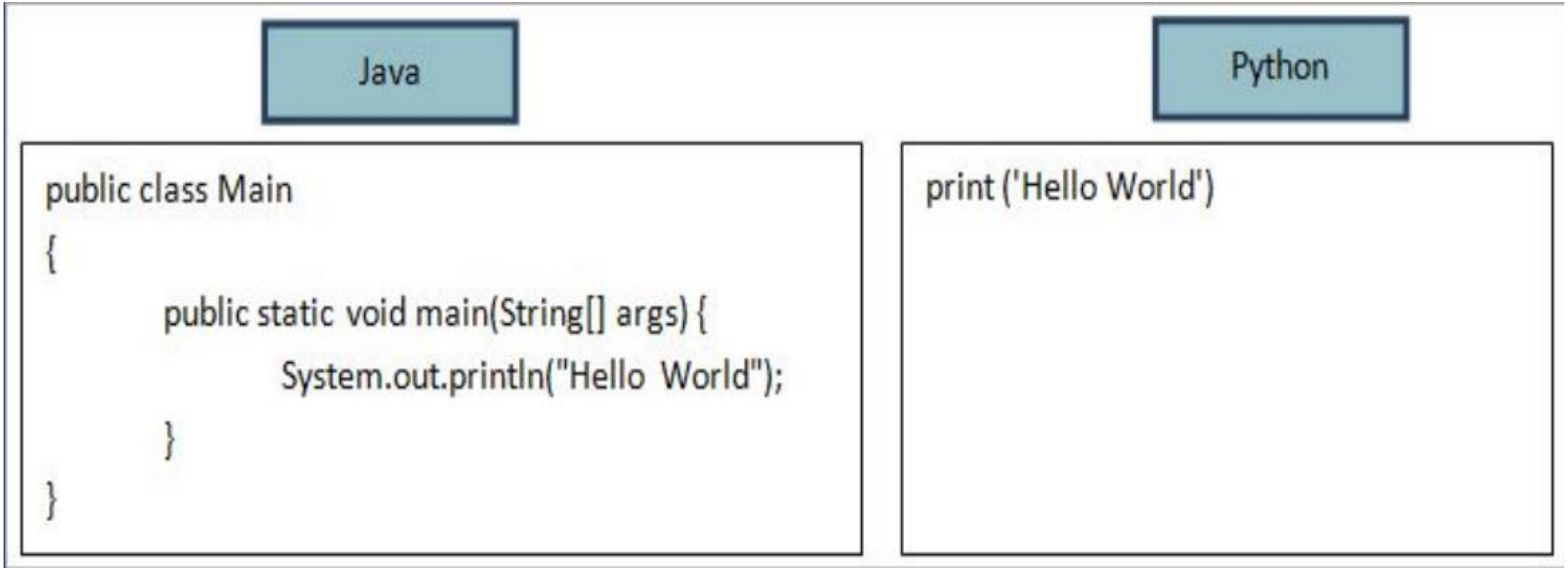
- General-purpose programming language
  - Used in various fields: Data Science, Machine Learning, Artificial intelligence, Web development, Mobile and Desktop applications.


Web Development
Machine learning and Artificial Intelligence
Software Development
Python
Desktop GUIs
Data Science and Visualization
Web Scraping

- Cross-platform
  - Python can run equally well on different platforms such as Windows, Linux, Unix, Macintosh, and so on.



- Easy to learn and use
  - High-level programming language.
  - Uses an elegant syntax, making the programs easy to read.



Java

```
public class Main
{
        public static void main(String[] args) {
                System.out.println("Hello  World");
        }
}
```

Python

```
print ('Hello World')
```

- Interpreted
  - Python is an interpreted language that executes the code line by line.
  - This makes debugging easy and suitable for beginners.

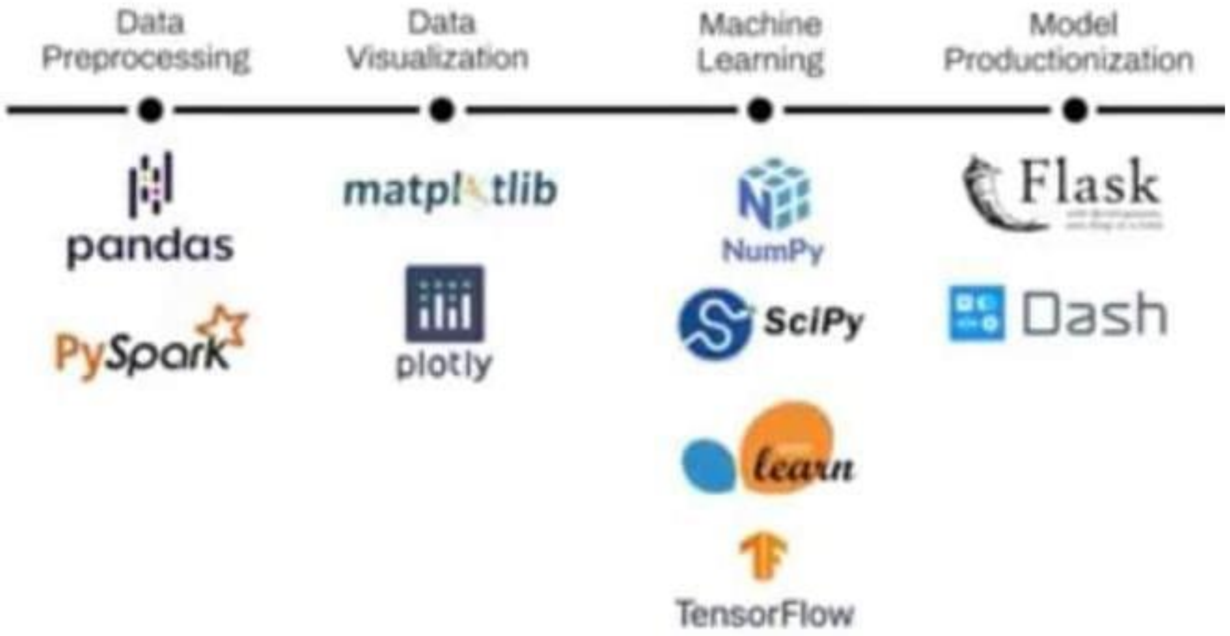
- Free and Open-source
  - Python language is freely available at https://www.python.org .
  - The Source code is also available.

- Integrated
  - It can be easily integrated with languages such as C, C++, Java, and more.

- Large standard library
  - It comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, and reading and modifying files.

Top Python libraries a Data Scientist need to know

- 01 Pandas
- 02 NumPy
- 03 SciPy
- 04 Scrapy
- 05 Matplotlib
- 06 Seaborn
- 07 Scikit-Learn
- 08 TensorFlow
- 09 Scikit-Image
- 10 Librosa



Data Preprocessing

Data Visualization

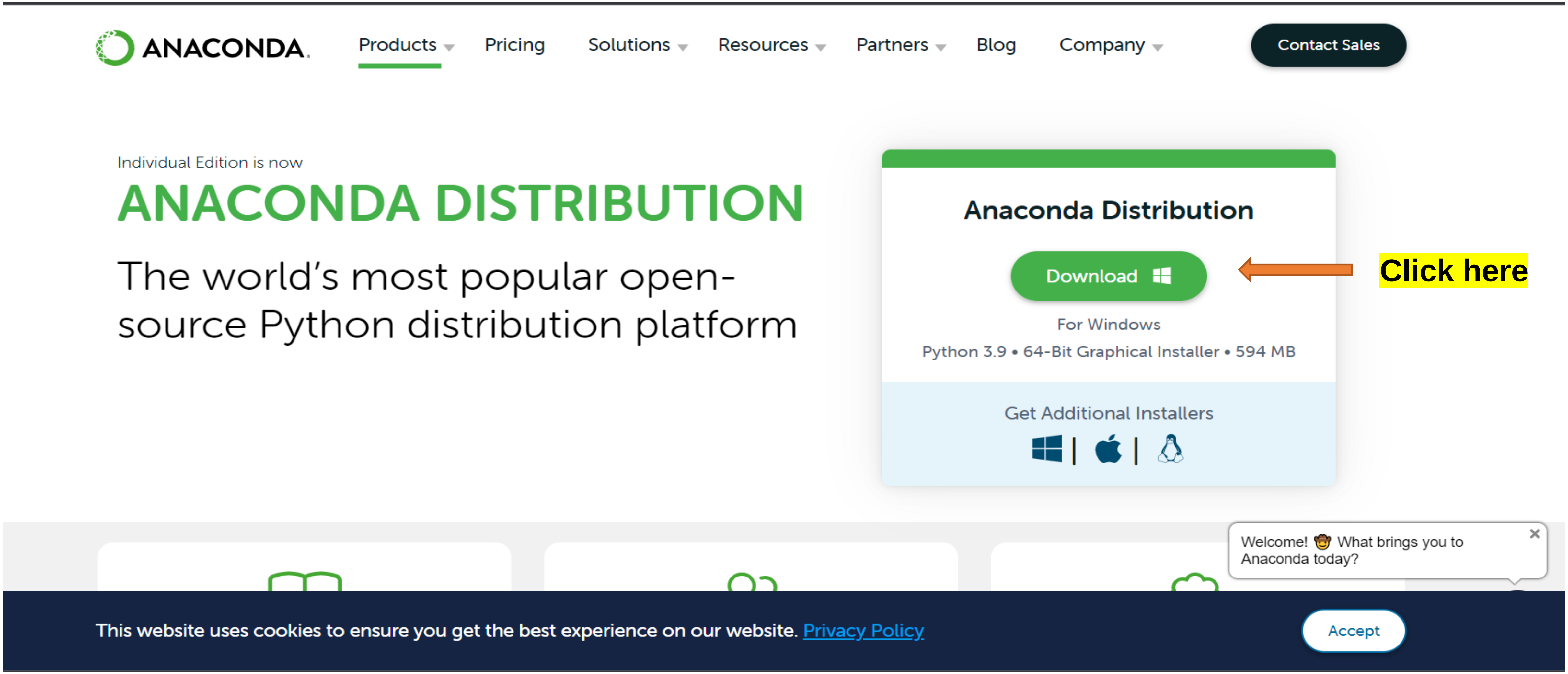
Machine Learning

Model Productionization

Anaconda

Steps for downloading Anaconda

- First visit **Anaconda Website**



Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

Get Additional Installers

Welcome! 🤠 What brings you to Anaconda today?

This website uses cookies to ensure you get the best experience on our website. Privacy Policy

Click here

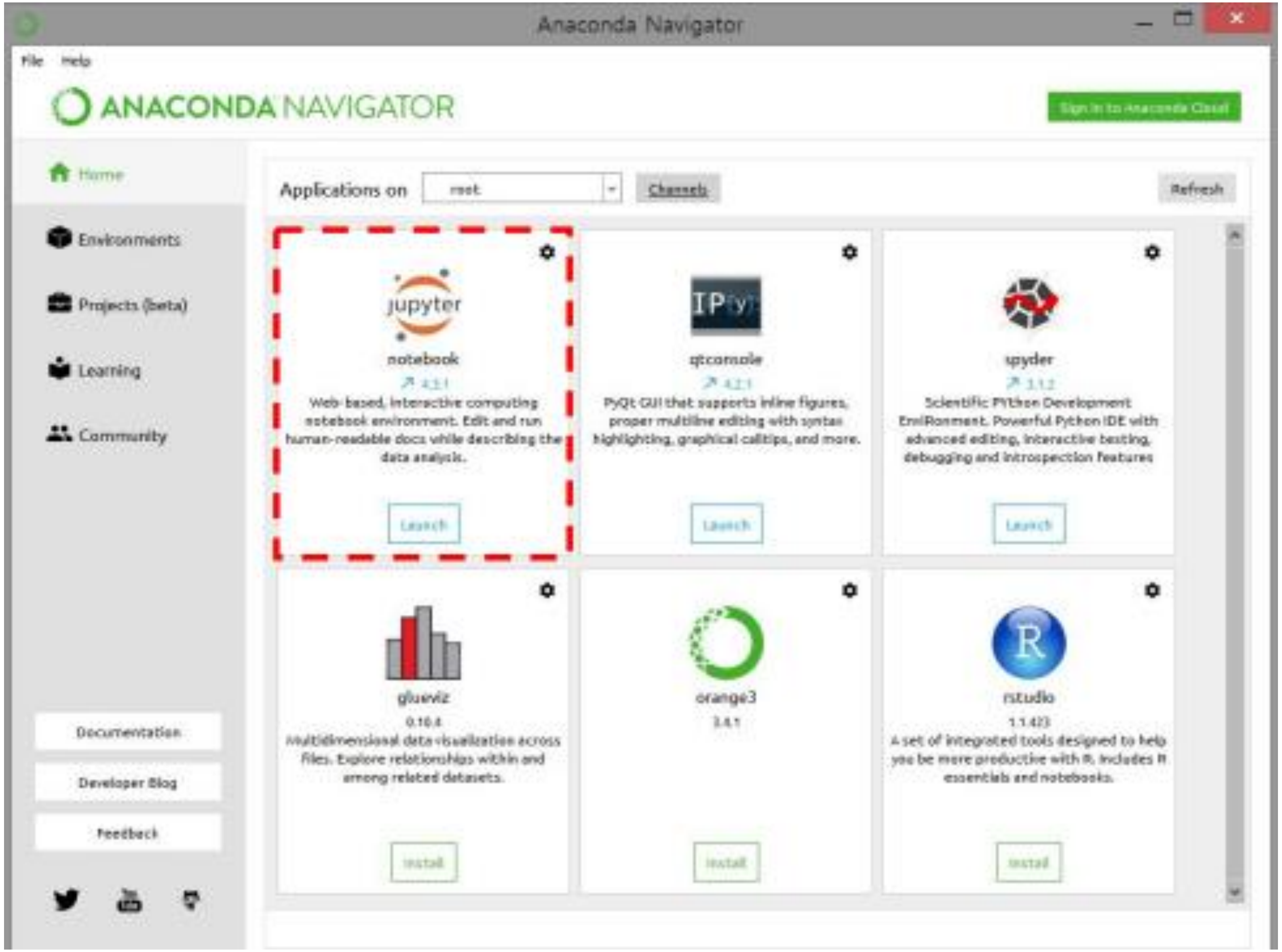
Anaconda IDEs

Python IDEs

- IDE
  - An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI).
- Examples Of Python IDEs
  - Spyder
  - Jupyter notebook

How to open a Jupyter notebook on Windows

- Step 1. Find and open the Anaconda Navigator app using the search bar,
- Step 2. Launching Jupyter Notebook from Anaconda Navigator



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

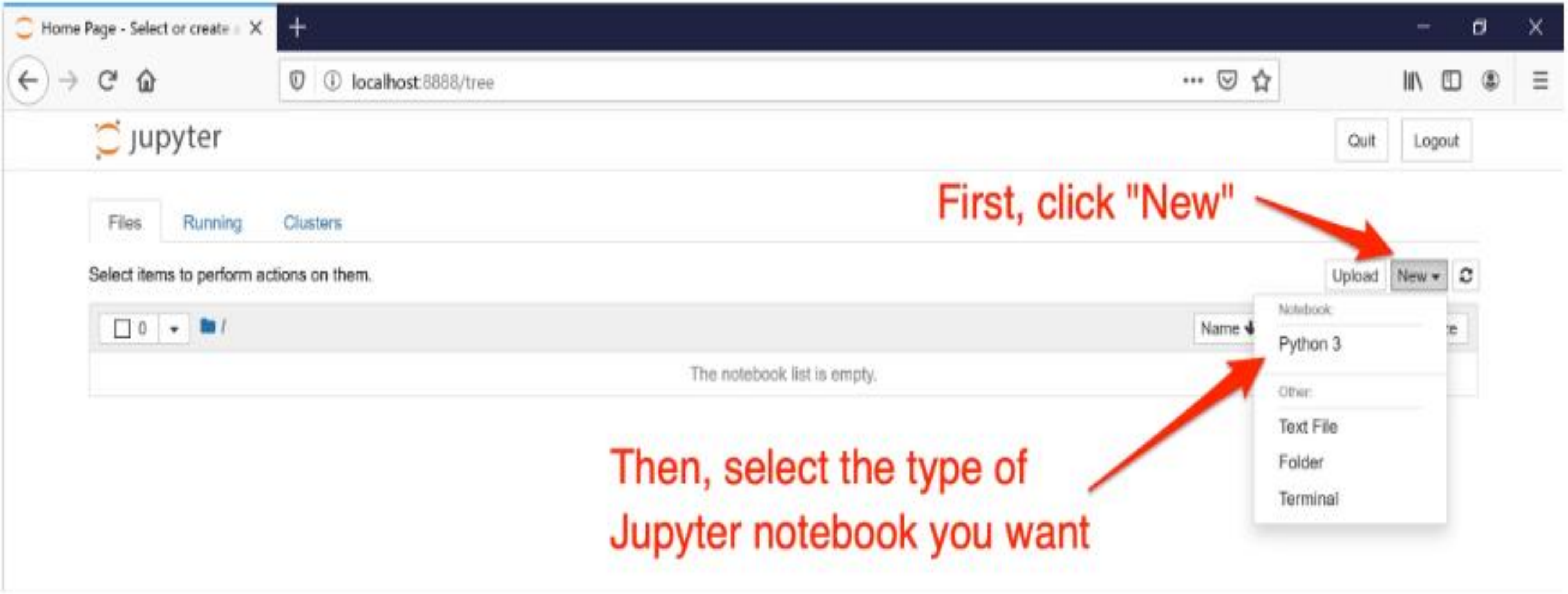
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Step 3. Create a new Jupyter notebook.



Select items to perform actions on them.

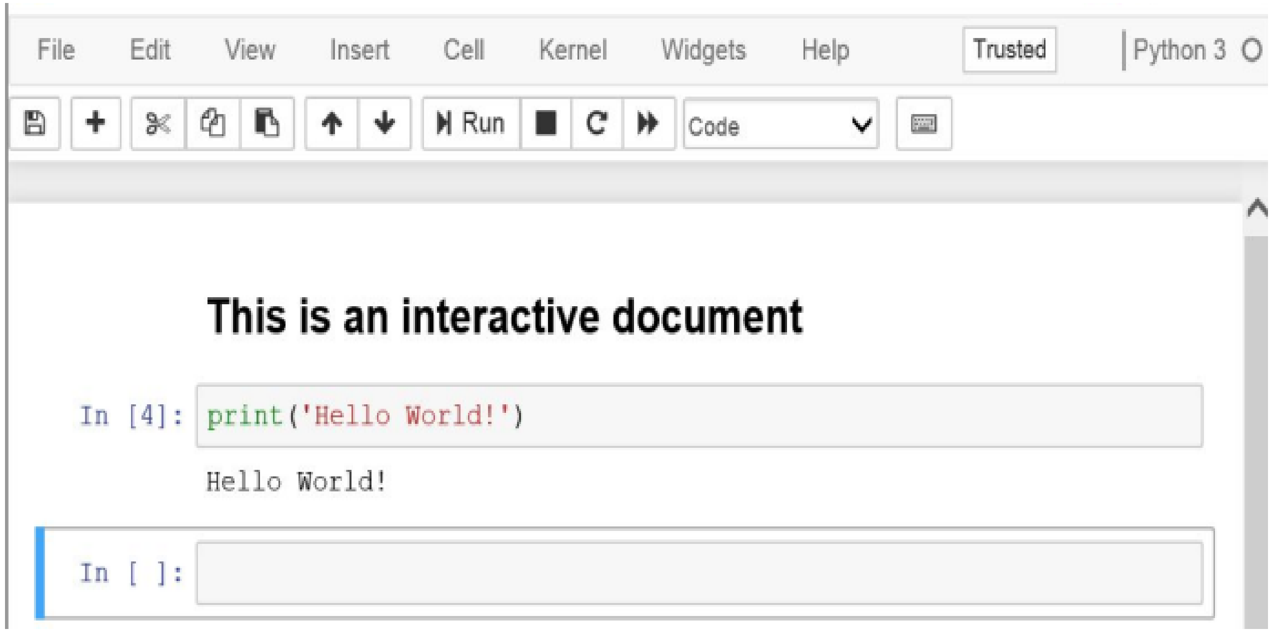
The notebook list is empty.

First, click "New"

Then, select the type of Jupyter notebook you want

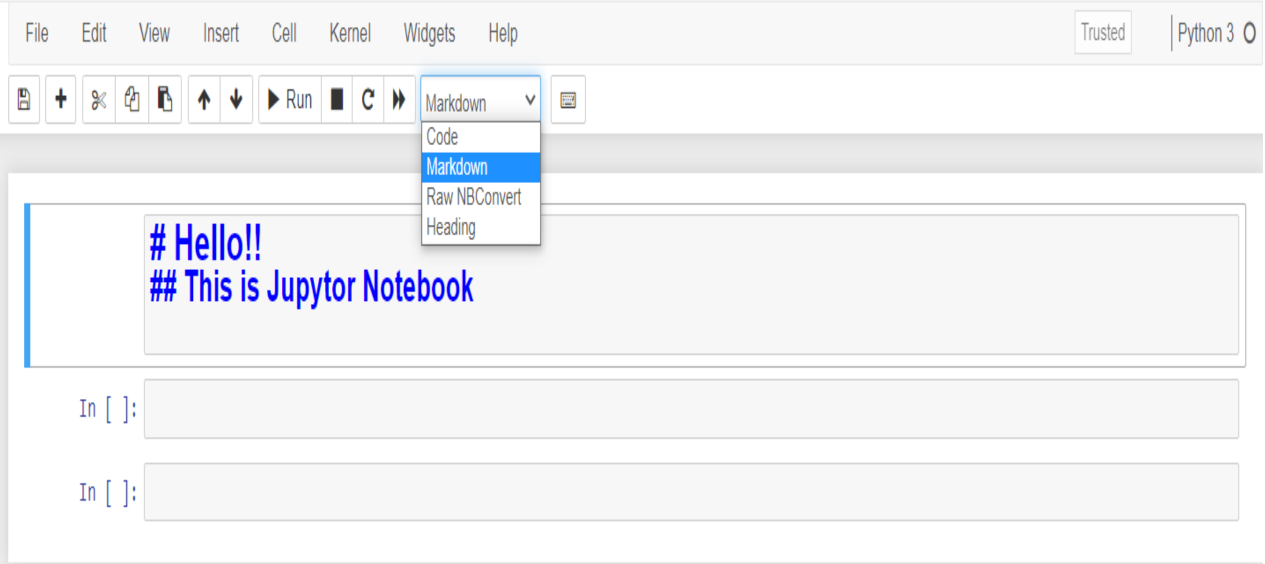
Jupyter Notebook Quick Introduction

You can switch between cell types: Code, Markdown, etc.

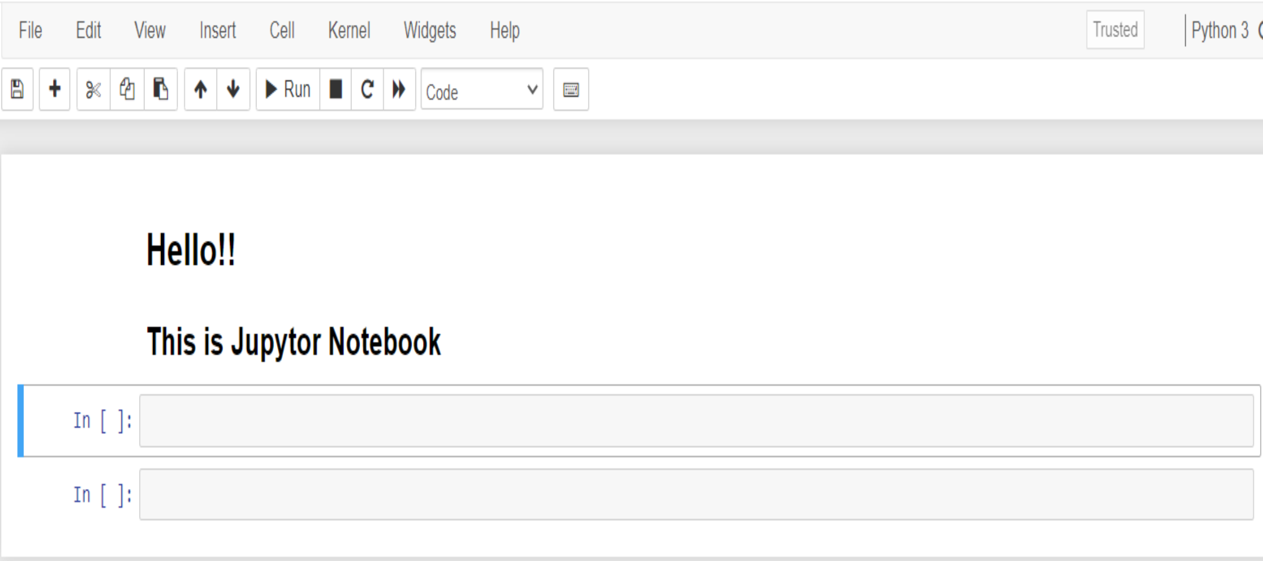


This is an interactive document

```
In [4]: print('Hello World!')

Hello World!

In [ ]:
```



```
# Hello!!
## This is Jupytor Notebook

In [ ]:

In [ ]:
```

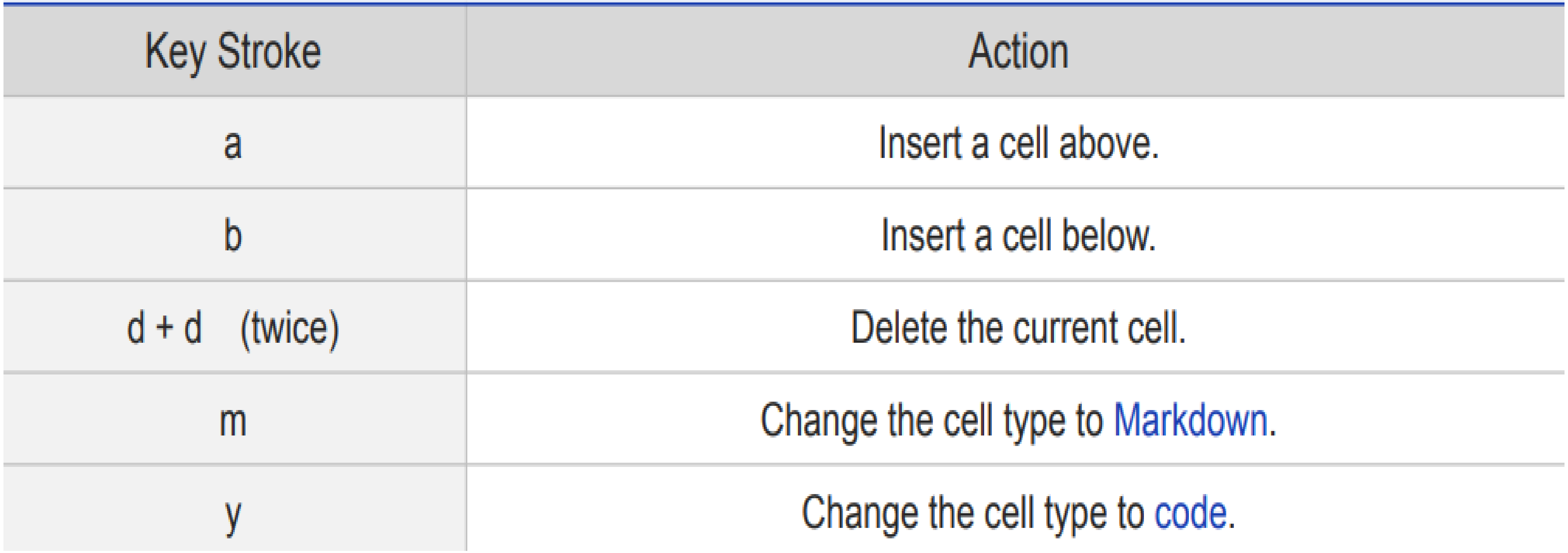


Hello!!

This is Jupytor Notebook

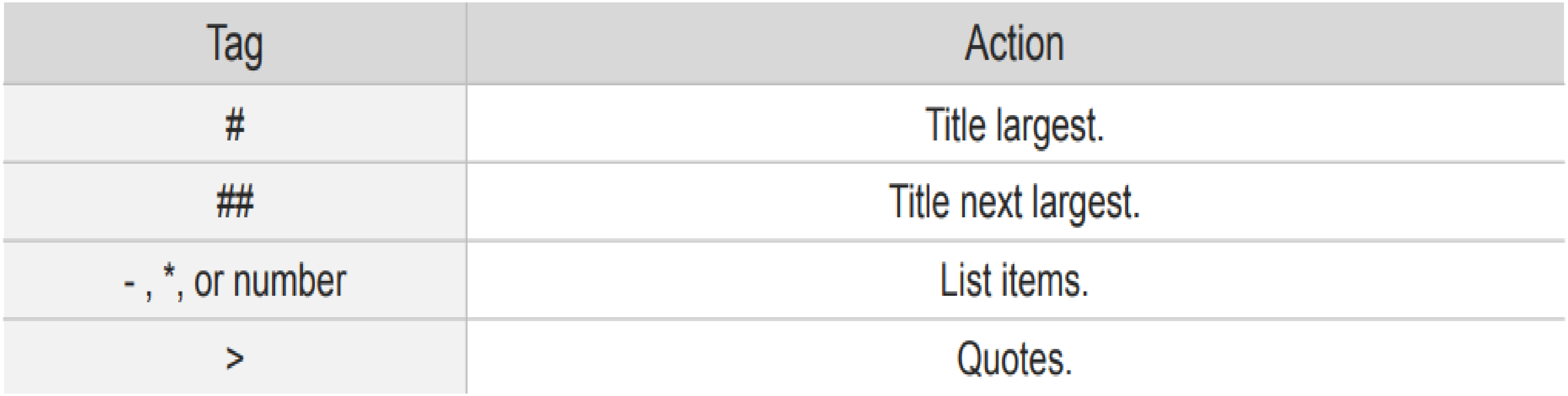
```
In [ ]:

In [ ]:
```

You can enter into the command mode by selecting a cell and then pressing the [ESC] key.



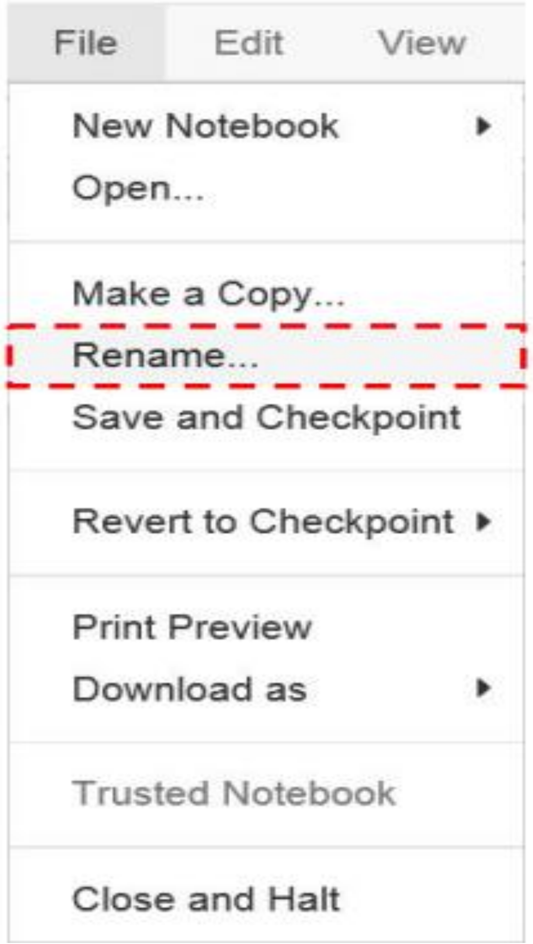
Key Stroke	Action
a	Insert a cell above.
b	Insert a cell below.
d + d (twice)	Delete the current cell.
m	Change the cell type to Markdown.
y	Change the cell type to code.

Some of the useful Markdown tags:

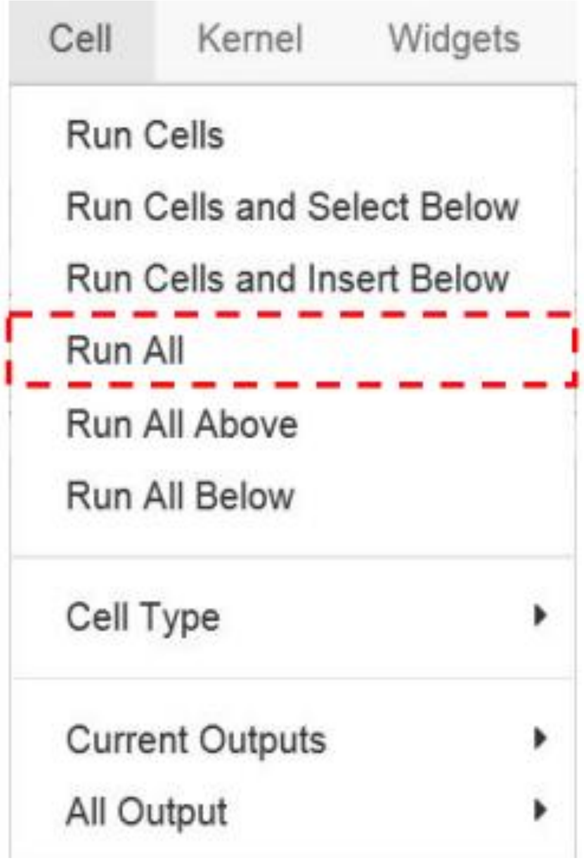


Tag	Action
#	Title largest.
##	Title next largest.
- , *, or number	List items.
>	Quotes.

Enter into a notebook, then select Files → Rename in order to change the default name.

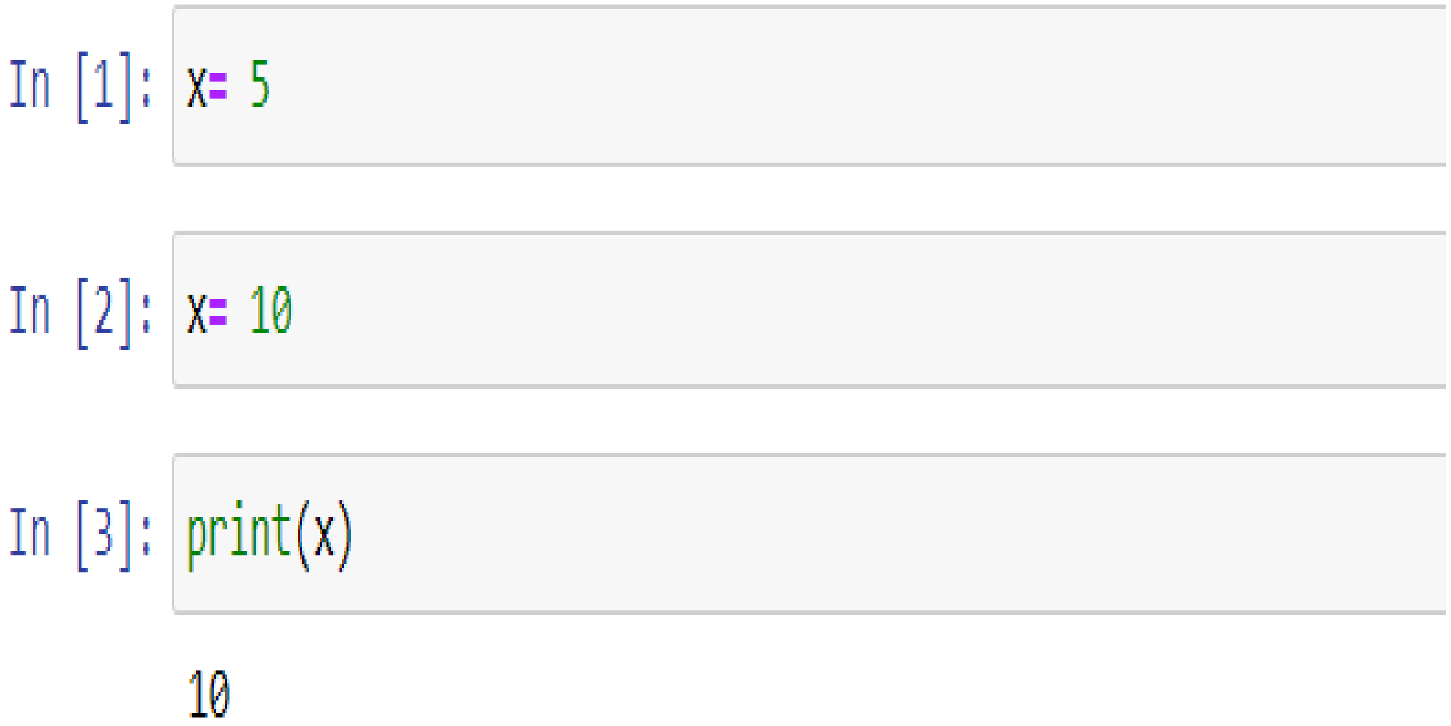


Enter into a notebook, then select Cell → Run All to execute all the cells from top to bottom in a row



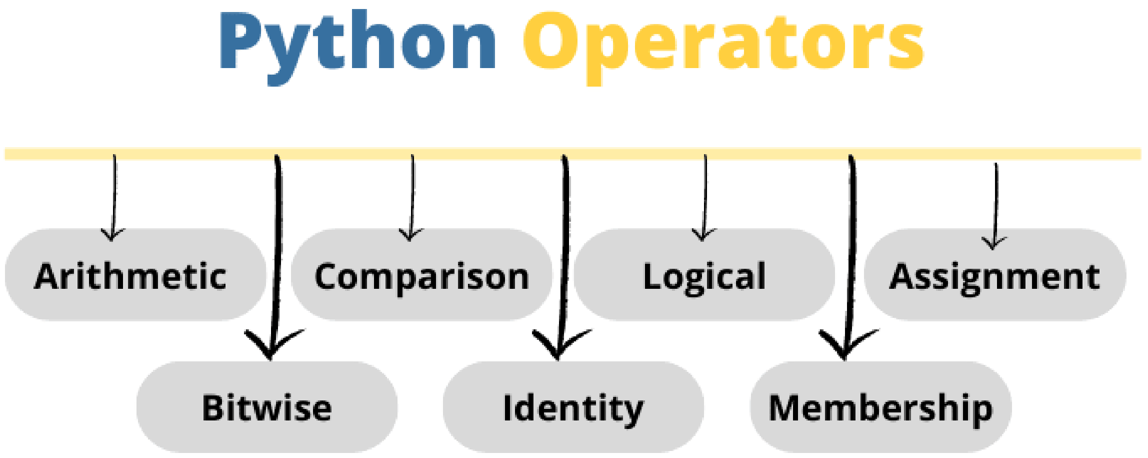
Variables, Expressions, and Statements

Variables



```
In [1]: x= 5

In [2]: x= 10

In [3]: print(x)

        10
```

Expressions and Operators



Python Operators

Arithmetic

Bitwise

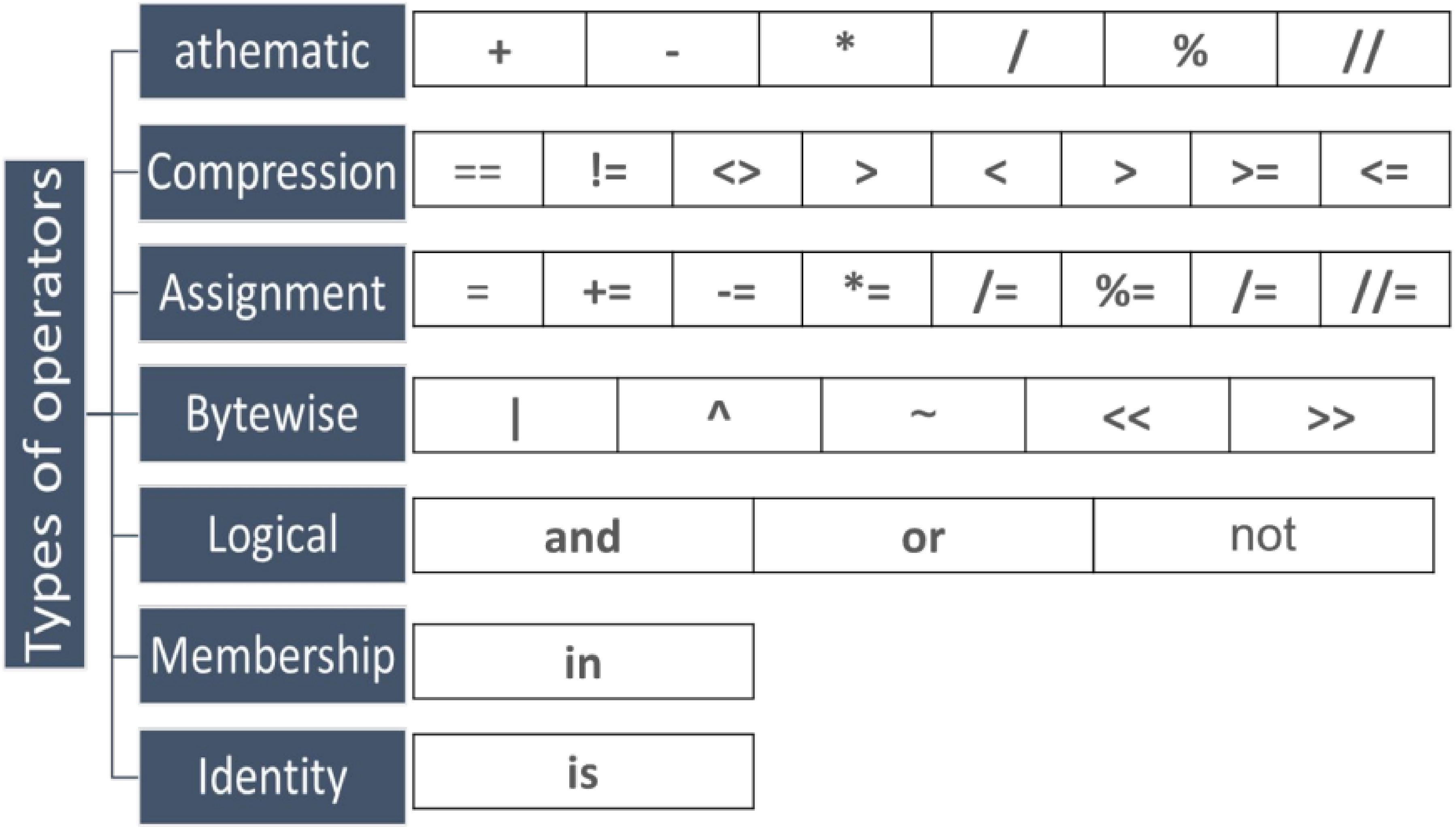
Comparison

Identity

Logical

Membership

Assignment

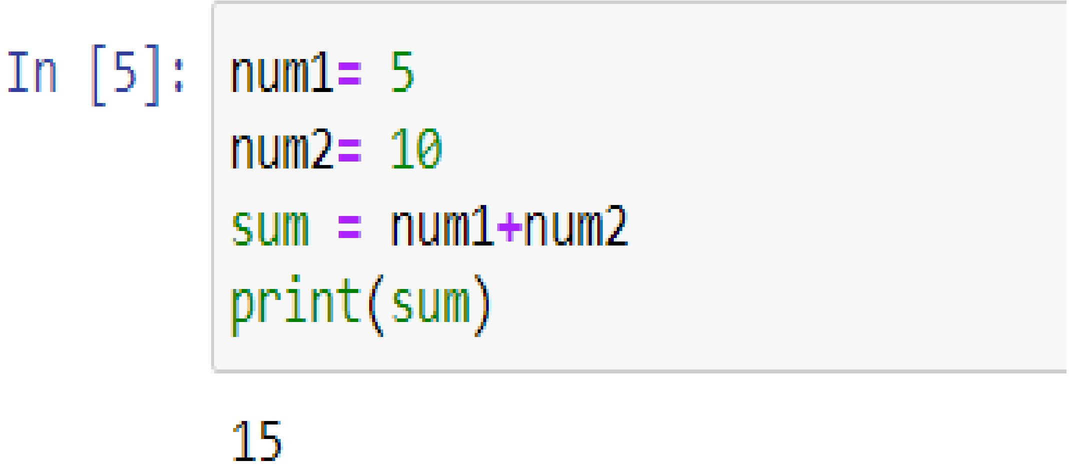


Types of operators

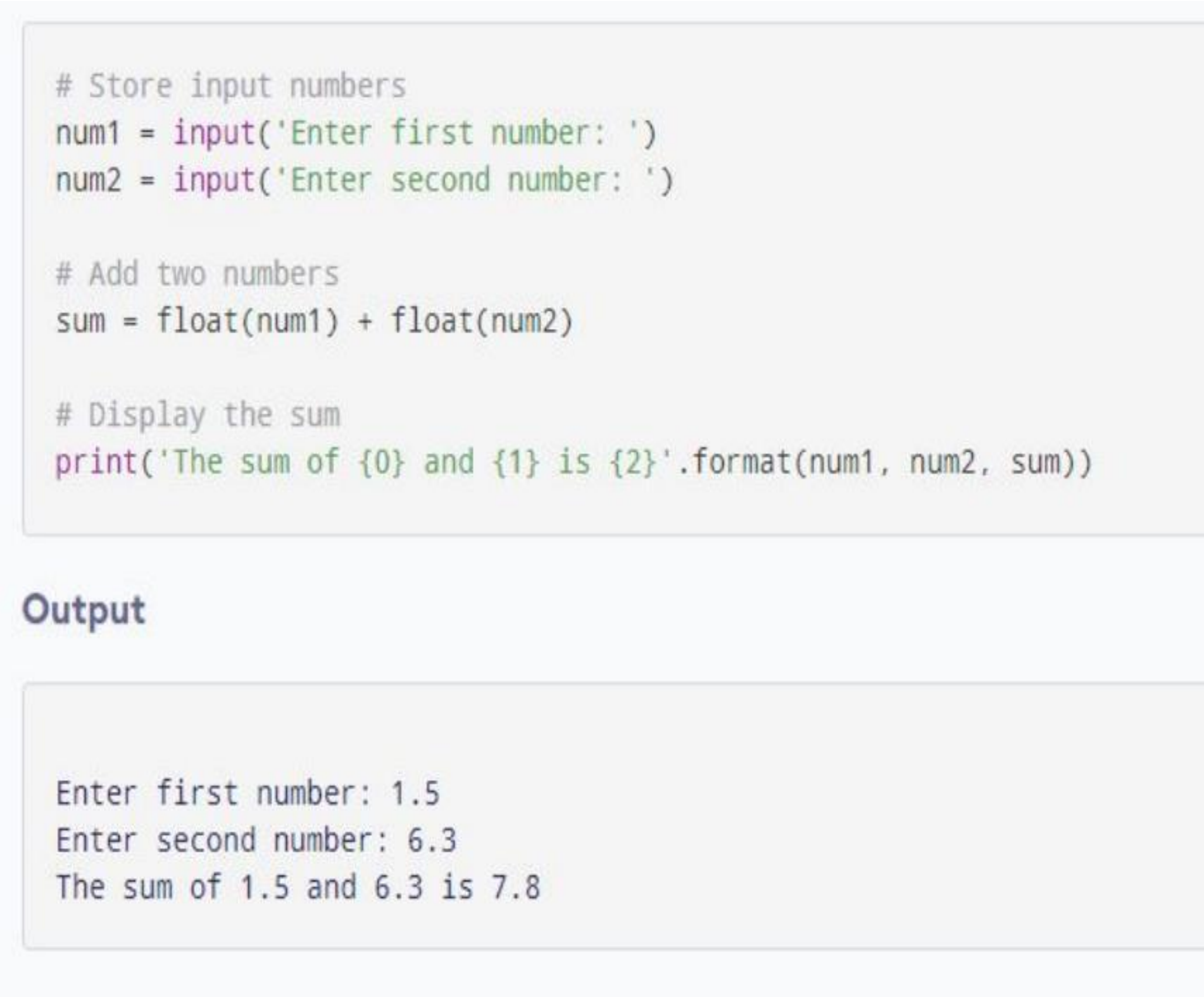
athematic	+	-	*	/	%	//		
Compression	==	!=	<>	>	<	>	>=	<=
Assignment	=	+=	-=	*=	/=	%=	/=	//=
Bytewise	|	^	~	<<	>>			
Logical	and	or	not					
Membership	in							
Identity	is							

Solve the following problems

1. Write a program that adds two number.



```
In [5]: num1= 5
        num2= 10
        sum = num1+num2
        print(sum)

        15
```

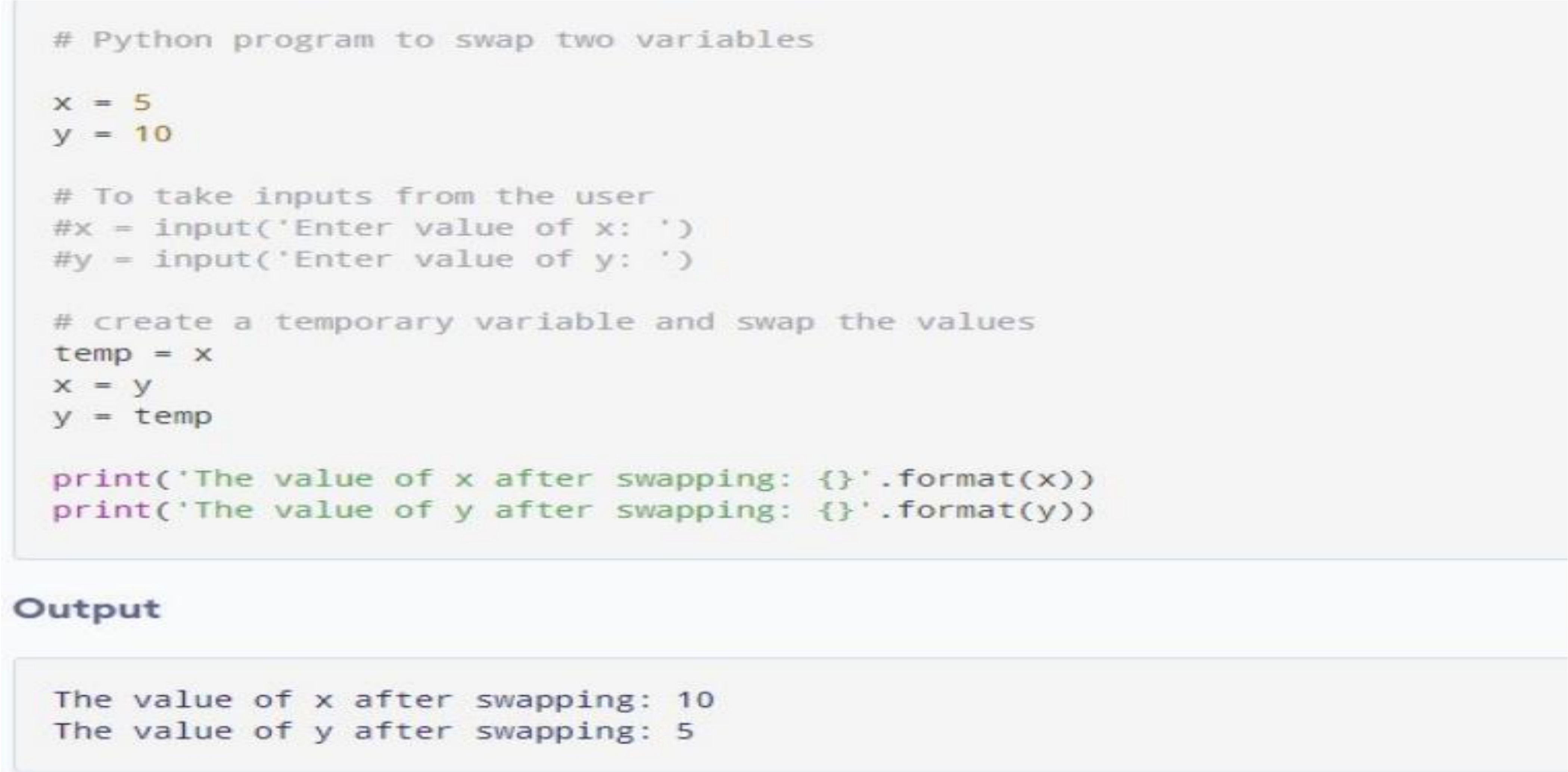


```
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)

# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

Output

```
Enter first number: 1.5
Enter second number: 6.3
The sum of 1.5 and 6.3 is 7.8
```

2. Write a program to swap the value of two variables.

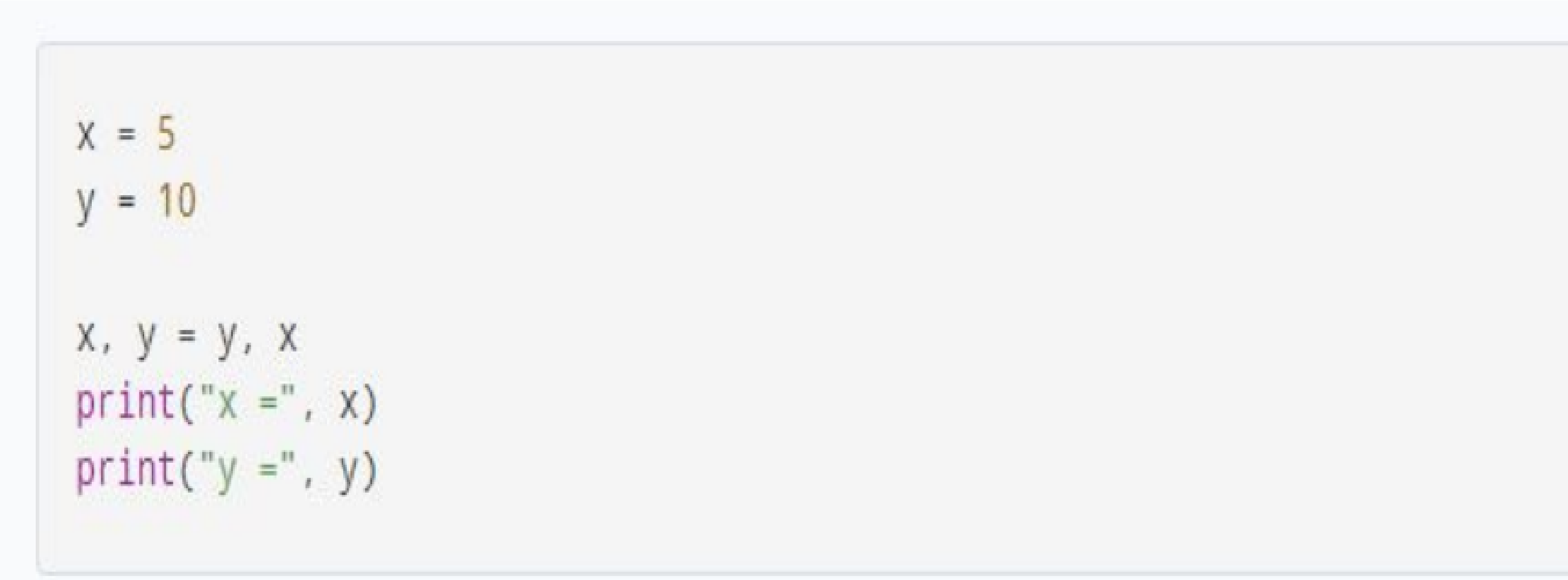


```
# Python program to swap two variables

x = 5
y = 10

# To take inputs from the user
#x = input('Enter value of x: ')
#y = input('Enter value of y: ')

# create a temporary variable and swap the values
temp = x
x = y
y = temp

print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

Output

```
The value of x after swapping: 10
The value of y after swapping: 5
```

In Python, there is a simple construct to swap variables. The following code does the same as above but without the use of any temporary variable



```
x = 5
y = 10

x, y = y, x
print("x =", x)
print("y =", y)
```

Thank you