

Introduction to Python

Arvind Rajendra Pagire

Junior research fellow

Climate monitoring and Prediction Group,

Climate Research and Services,

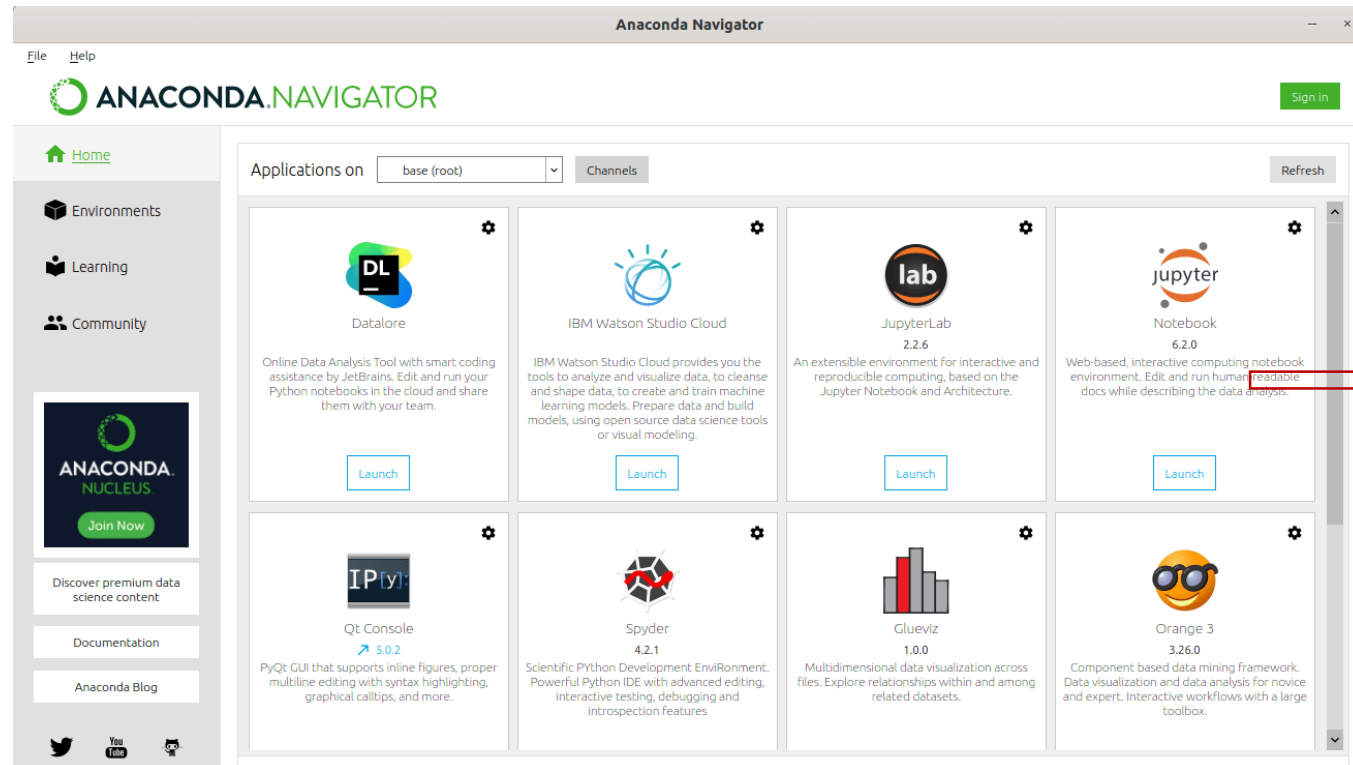
India Meteorological Department, Pune

Python training at CRS Pune (11-15 September 2023)

How to Launch Jupyter-Notebook

- (1) At first launch Anaconda-navigator from your PC
- (2) Then launch Jupyter-Notebook as shown in the below image

Anaconda-navigator: Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands.



**Jupyter-
Notebook**

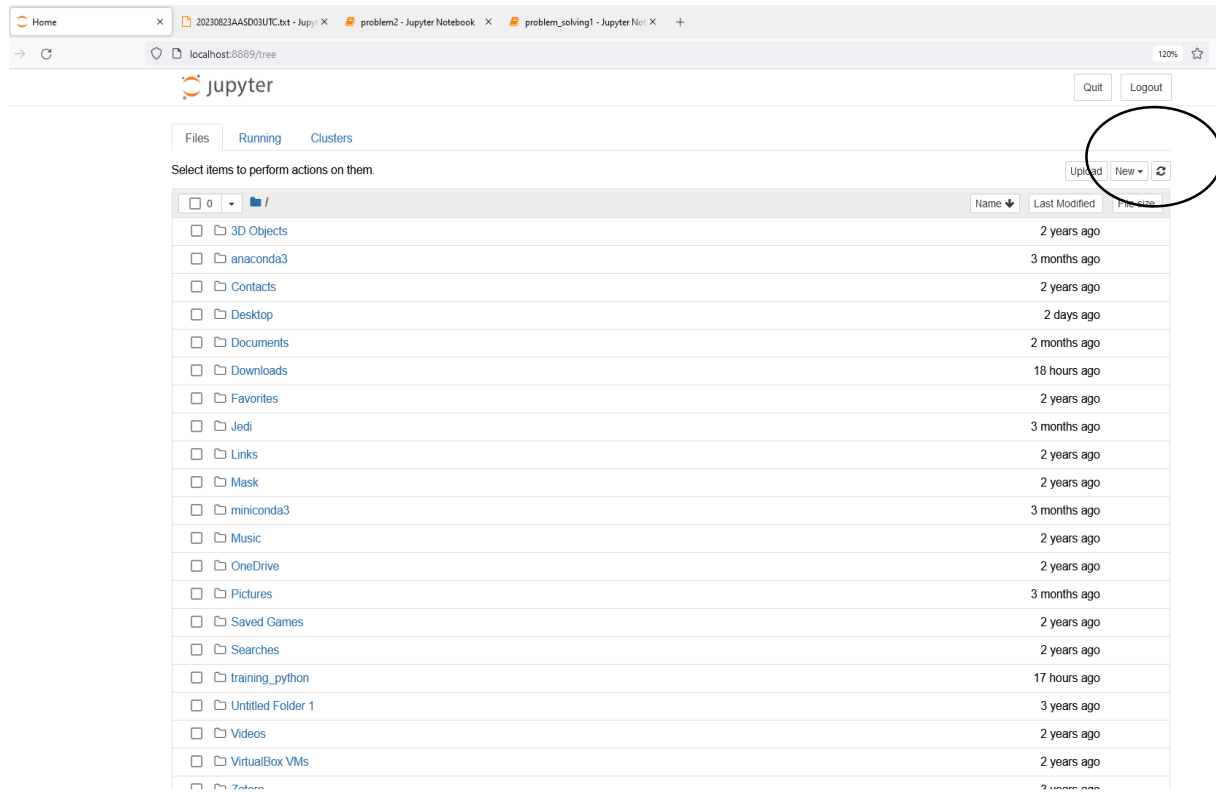
Once an anaconda-navigator is installed, you can directly open Jupyter-notebook through anaconda-prompt as well as from search tab

- ❖ Key Features: Interactive, Support for Various Programming Languages (Python, R, Julia, etc.), Rich Text Formatting, Visualization Integration.

Advantage of using Jupyter-notebook

- 1.All in one place
- 2.Easy to share
- 3.Easy to convert
- 4.Language independent
- 5.Easy to create kernel wrappers
- 6.Easy to customize
- 7.Extensions with custom magic commands
- 8.Effective teaching-cum-learning tool
- 9.Interactive code and data exploration

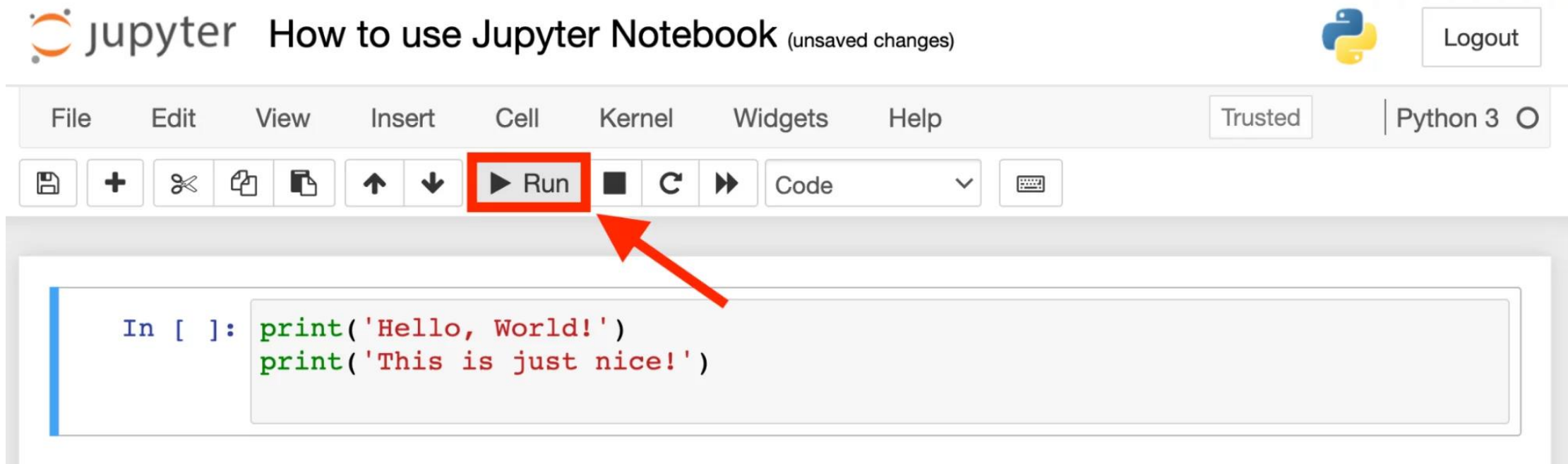
Working with Jupyter-notebook



To create a new notebook.

Jupyter-Notebook: 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.

Running a program in Jupyter-notebook

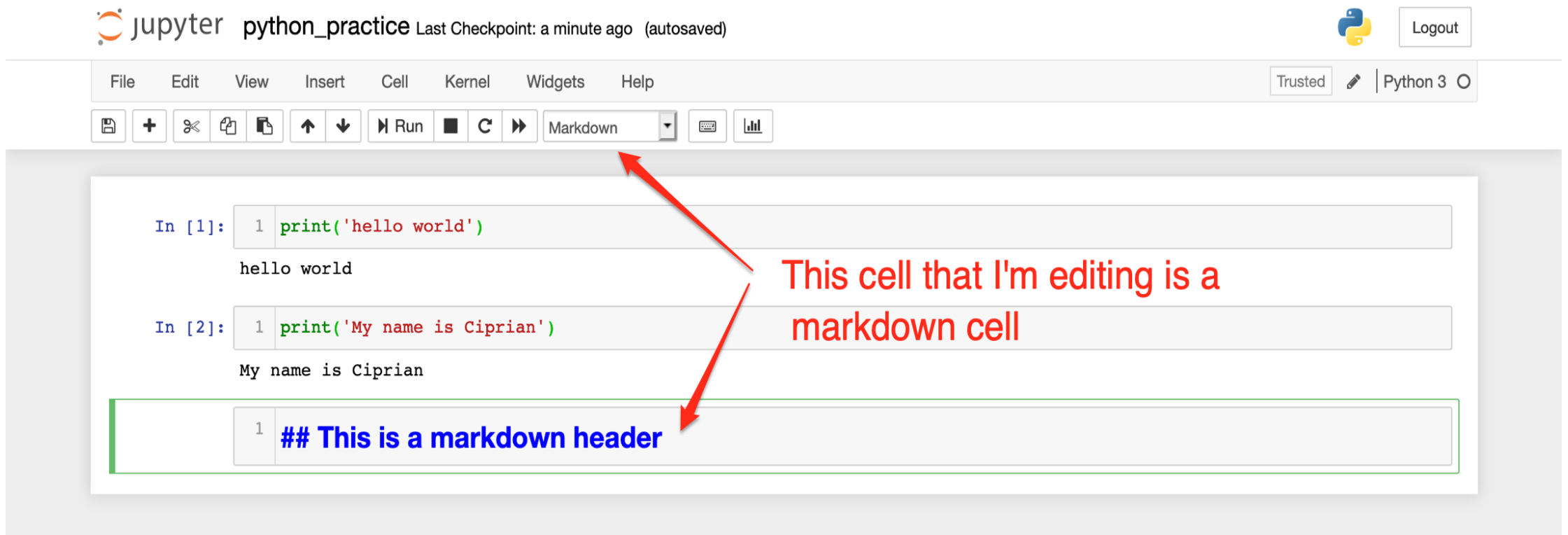


The screenshot shows the Jupyter Notebook interface. At the top, there's a header with the Jupyter logo, the text "How to use Jupyter Notebook (unsaved changes)", a Python logo, and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" buttons. Below the menu bar is a toolbar with various icons. The "Run" button, which is a play icon followed by the text "Run", is highlighted with a red box. A red arrow points from the "Run" button to a code cell below. The code cell contains the following Python code:

```
In [ ]: print('Hello, World!')
        print('This is just nice!')
```

- Write the above two lines in the cell and then run the program
- To run the Python code in a code cell push the [Run] button or type [Shift]+[Enter].
- Hitting [Enter] when the cursor is inside a code cell brings the cursor down to a new line.

Continued:



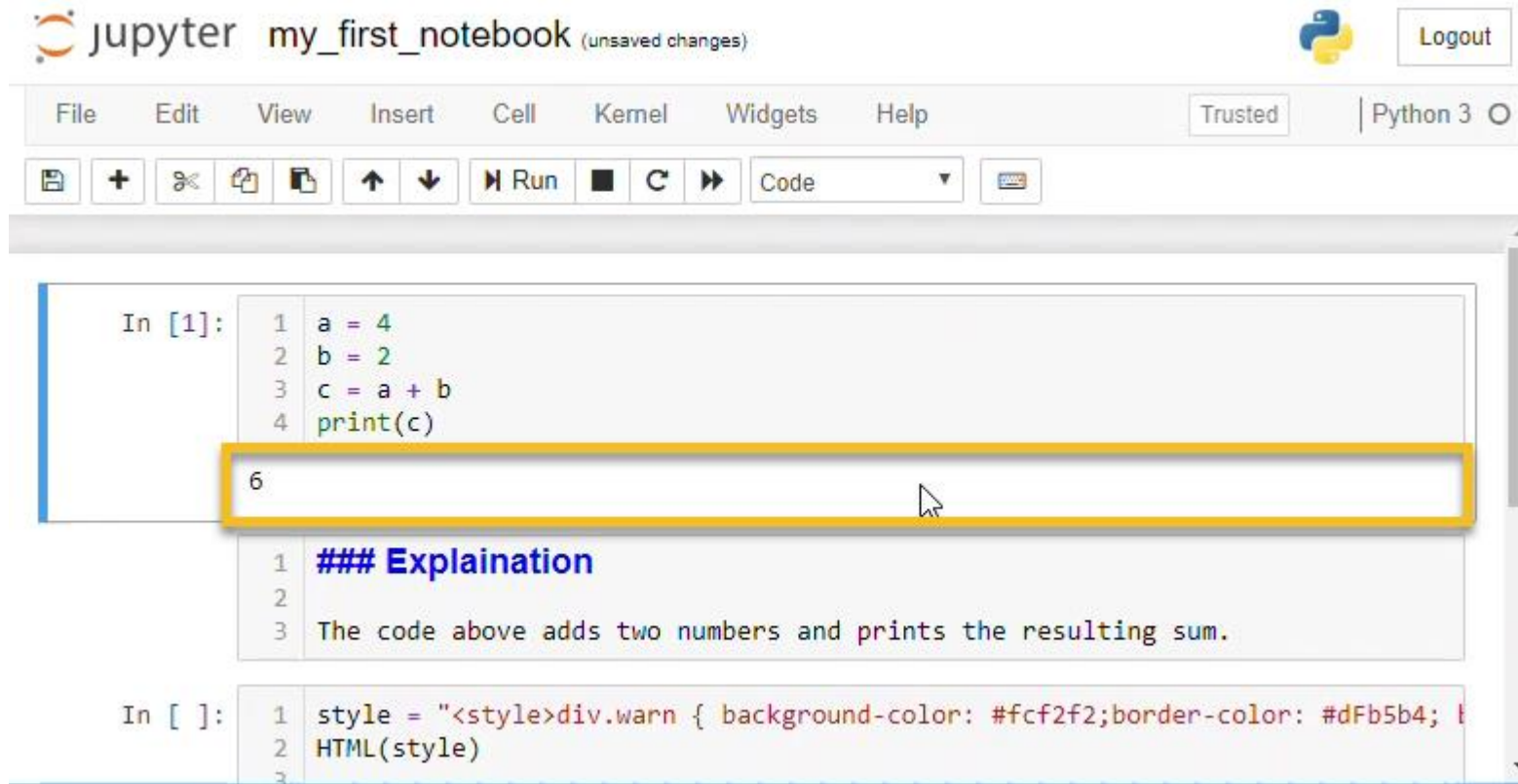
The screenshot shows a Jupyter Notebook interface with the following elements:

- Top bar: Jupyter logo, "python_practice", "Last Checkpoint: a minute ago (autosaved)", Python logo, and a "Logout" button.
- Menu bar: File, Edit, View, Insert, Cell, Kernel, Widgets, Help.
- Toolbar: Icons for saving, adding, deleting, copying, pasting, undo, redo, running, and a dropdown menu currently set to "Markdown".
- Code cells:
 - In [1]: `print('hello world')` followed by the output "hello world".
 - In [2]: `print('My name is Ciprian')` followed by the output "My name is Ciprian".
- Markdown cell (highlighted with a green border):
 - 1 **## This is a markdown header**

A red text box on the right says "This cell that I'm editing is a markdown cell". Two red arrows point from this text box: one to the "Markdown" dropdown in the toolbar and another to the markdown cell.

- Markdown cells don't contain Python code. Markdown cells contain text written in Markdown format.
- Text in markdown cells can be formatted to show **bold** or *italic* text. Tables, images, and lists can also be included in markdown cells.

Continued



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "jupyter my_first_notebook (unsaved changes)" with a Python logo and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are buttons for "Trusted" and "Python 3". Below the menu bar is a toolbar with icons for saving, adding, deleting, and running cells, as well as a dropdown menu currently set to "Code".

The main area of the notebook contains three cells:

- Code Cell (In [1]):** Contains the following Python code:

```
1 a = 4
2 b = 2
3 c = a + b
4 print(c)
```

The output of this cell is the number "6", which is displayed in a white box with a yellow border.
- Text Cell:** Contains the following text:

```
1 ### Explanation
2
3 The code above adds two numbers and prints the resulting sum.
```
- Code Cell (In []):** Contains the following HTML code:

```
1 style = "<style>div.warn { background-color: #fcf2f2;border-color: #dFb5b4;
2 HTML(style)
3
```

- After a code cell is run, an output cell can be produced below the code cell. The output cell contains the output from the code cell above it.
- Not all code produces output, so not all code cells produce output cells.
- The results in output cells can't be edited. If a code cell produces plots, charts or images, these outputs are shown in output cells

Variables and Data types

- Variables store data in memory.
- Data types include:
 - Integers (int)
 - Floating-point numbers (float)
 - Strings (str)
 - Booleans (bool)

Operator

1. Arithmetic: +, -, *, /, %, **
2. Comparison: ==, !=, <, >, <=, >=
3. Logical: and, or, not

Conditional statement

if condition:

 # Code to execute if condition is True

elif another_condition:

 # Code to execute if another_condition
is True

else:

 # Code to execute if no conditions are
True

e.g . For if statement :

Input a number

number = float(input("Enter a number: "))

Check if the number is greater than 10

if number > 10:

 print("The number is greater than 10.")

else:

 print("The number is not greater than 10.")

List and Dictionaries

- *Lists*: Ordered collections of items.

```
numbers = [1, 2, 3, 4, 5]
```

Dictionaries: Key-value pairs for efficient data storage.

```
person = {"name": "Alice", "age": 30, "city": "New York"}
```

for loop

- Loops
- For Loop:
- for item in iterable:
- # Code to repeat for each item

Functions

Functions are blocks of reusable code.

Improve code organization and readability.

Defined using the def keyword.

```
def greet(name):  
    return "Hello, " + name + "!"
```

Module and libraries

Modules: Separate Python files containing functions and variables.

Libraries: Collections of modules offering specialized functionality.

Import using import keyword.

```
# simple program for mathematical calculation
```

```
import math  
result = math.sqrt(16)
```

Important modules for handling weather and climate data

- (i) *Numpy* : array type dataset
- (ii) *Pandas* : csv, excel, text
- (iii) *Xarray* : netcdf
- (iv) *Scipy* : for scientific and technical computing.
- (v) *Matplotlib* : data visualization

```
import numpy as np
```

```
# Create a NumPy array  
arr = np.array([1, 2, 3, 4, 5])
```

```
# Perform mathematical operations  
result = arr * 2
```

```
print(result)
```

Conclusion

- Python's simplicity and versatility make it a popular programming language
- Ideal for beginners and professionals
- Continuously evolving with a vast community and extensive resources.

Thank you