

AI/ML/Python



AI Programming with Python

Build Intelligent Systems



About the Author



- Created By: Mohammad Salman
- Experience: 19 Years +
- Designation: Corporate Trainer



Icons Used



Questions



Tools



Hands-on Exercise



Coding Standards



Questions?



Reference



Try it Out



Informative
Slide



Mandatory
Slide



Welcome Break



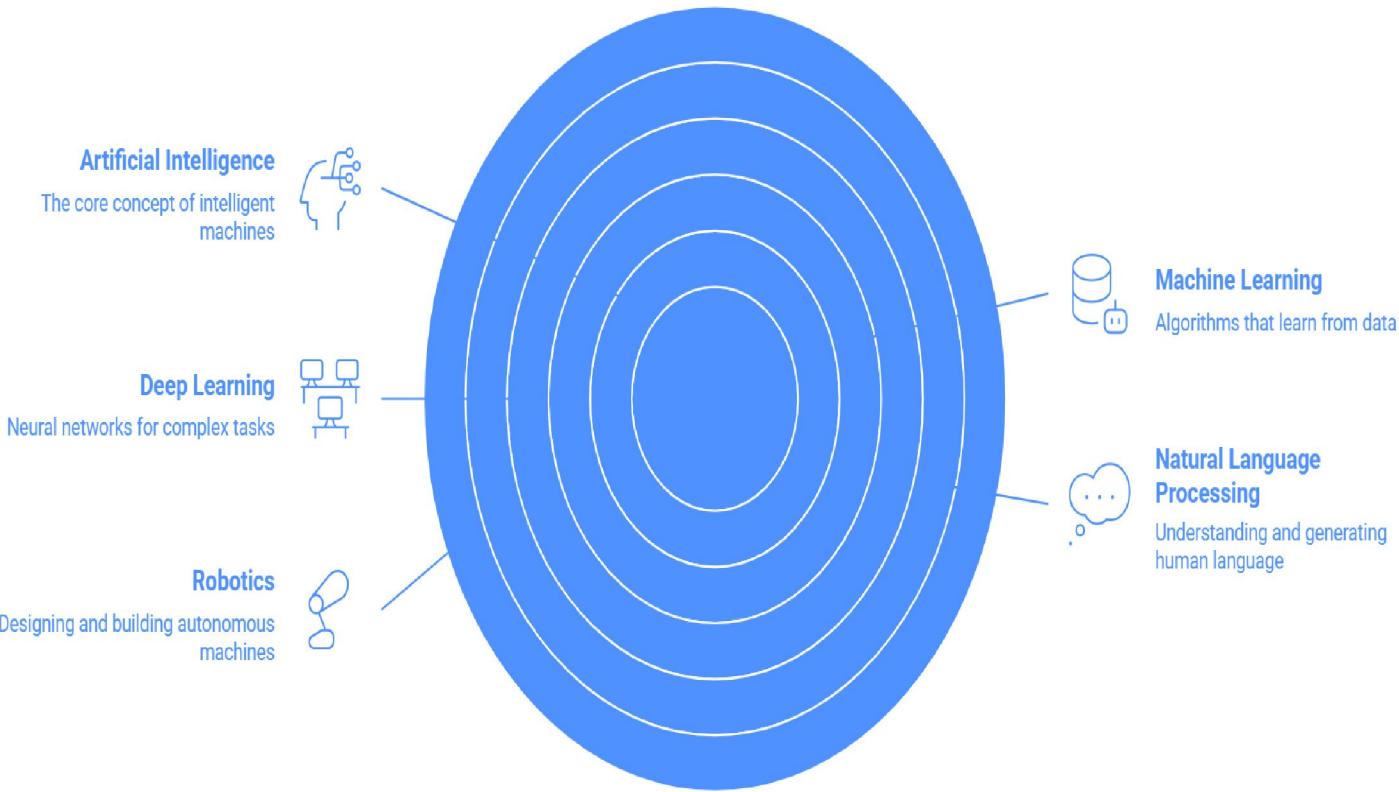
AI THE FUTURE

AI PROGRAMMING. WITH PYTHON

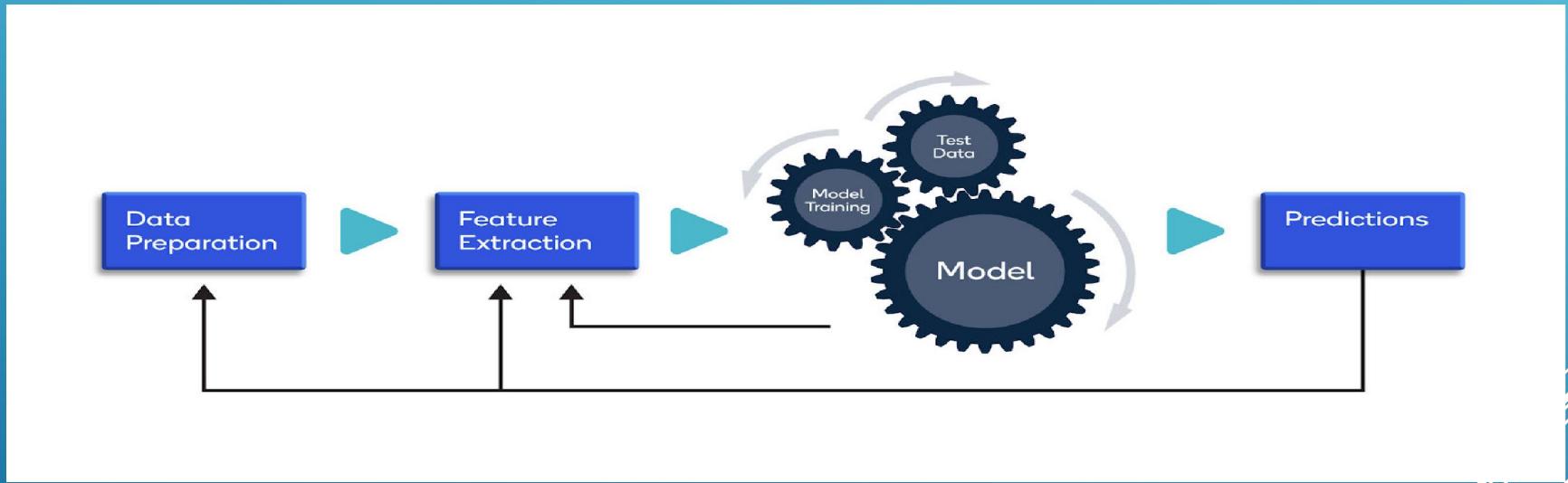
To Build Intelligent Systems



AI and Its Subfields



- ▶ Machine Learning (ML) is a subset of Artificial Intelligence that enables systems to learn from data and make predictions without explicit programming.



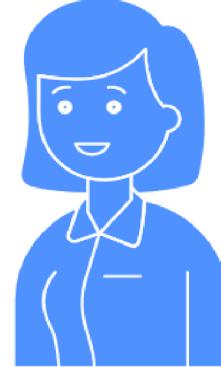
WHAT IS MACHINE LEARNING

Machine Learning Applications



Where is machine learning used today?

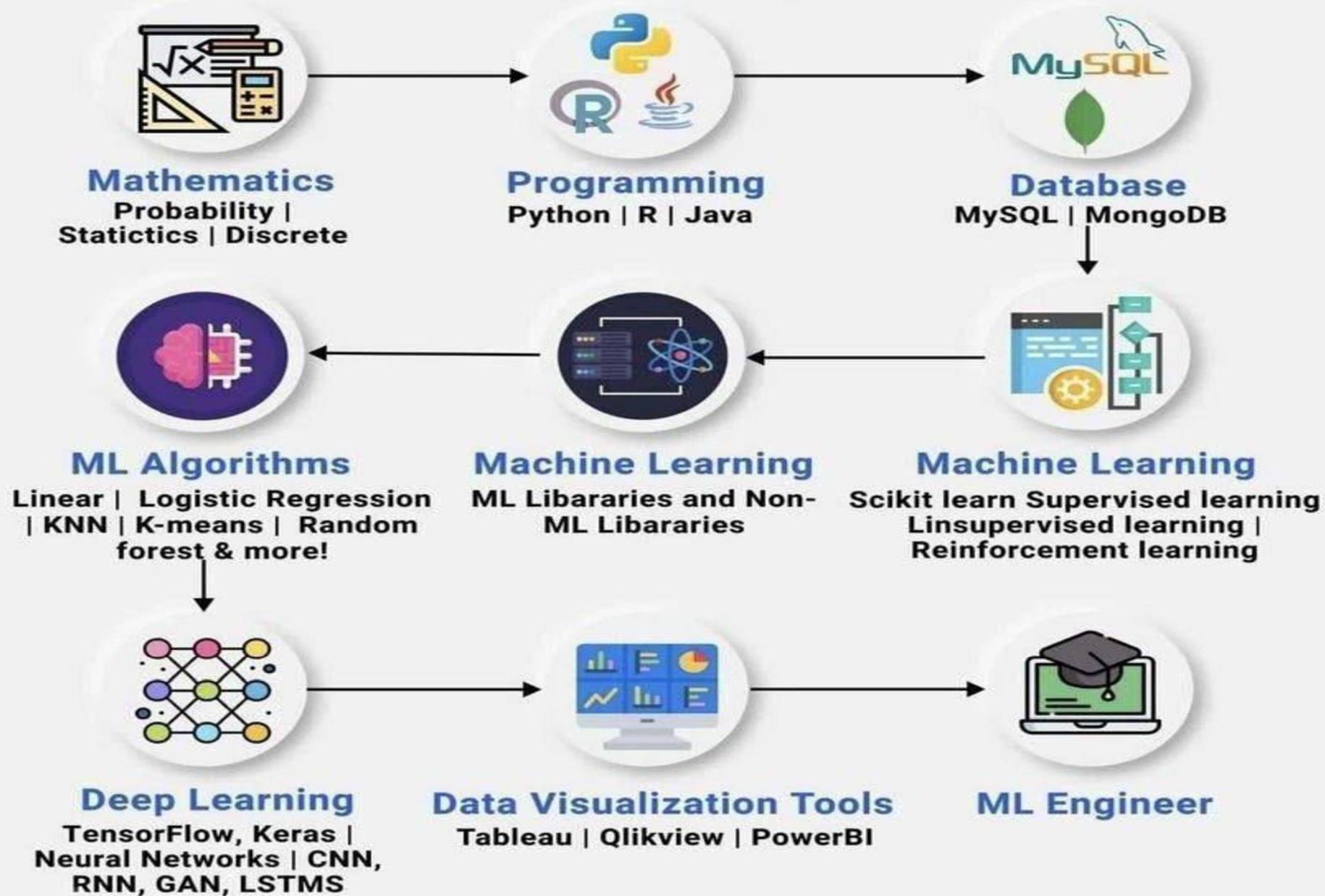
It's used in fraud detection, spam filters, product recommendations, medical diagnosis, and voice assistants.

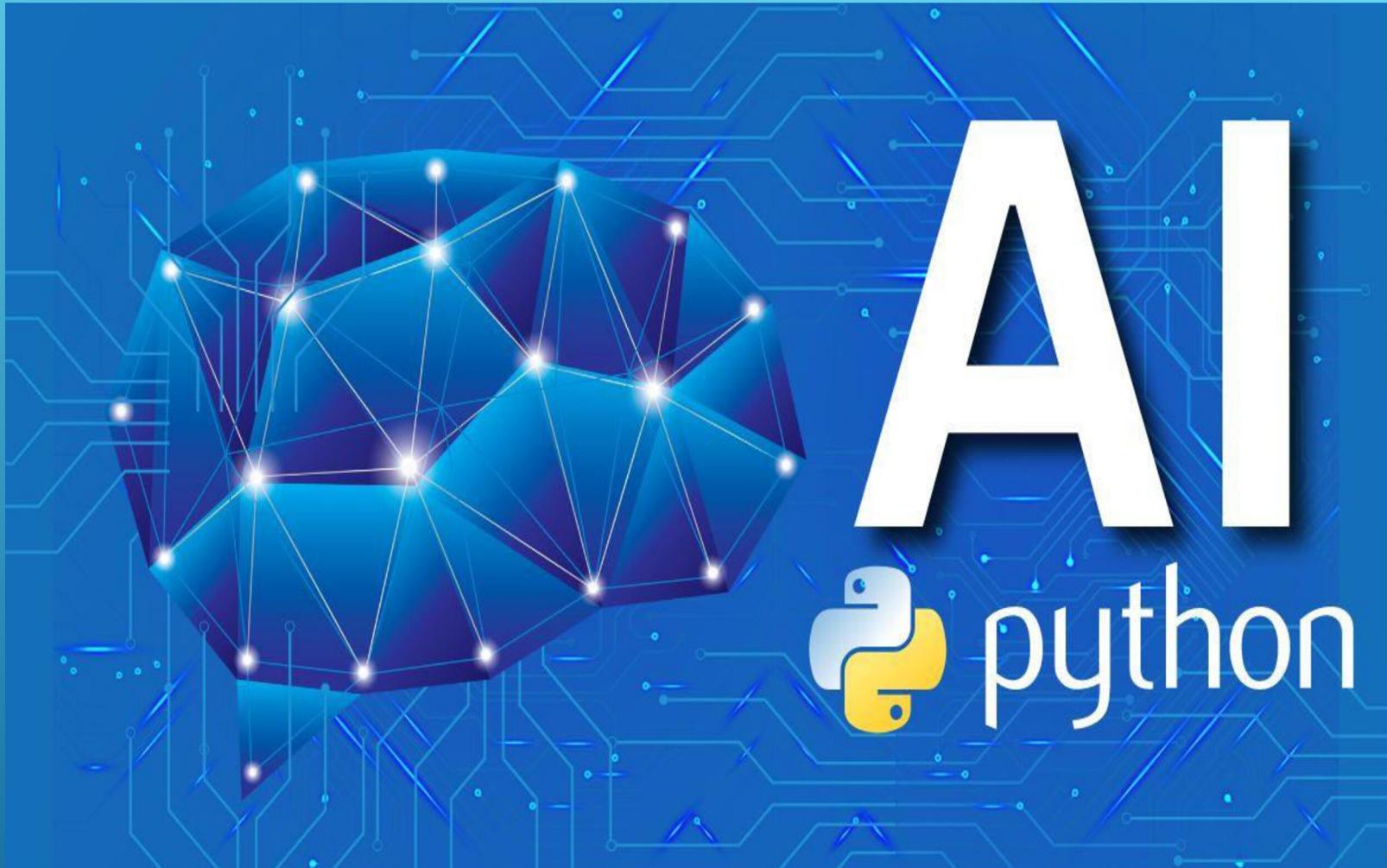


What does machine learning do?

It automates decisions and learns from large datasets.

ML ENGINEER ROADMAP





- ❑ Python is widely used in the field of artificial intelligence because of its simplicity and readability.
- ❑ Python is easy to learn.
- ❑ Python has a variety of libraries and frameworks, making it easier to develop AI applications.



□ Why Python

LIBRARIES

Numpy

Pandas

Matplotlib

Scikit-Learn

LIBRARIES

All applications on base (root) Channels

 PyCharm The only Python IDE you need – built for data and AI/ML professionals. Supercharged with an AI-enhanced IDE experience. Free forever, plus one month of Pro included. Install	 Anaconda AI Navigator Access various large language models (LLMs) curated by Anaconda, and start leveraging secure local AI today. Install	 Anaconda Toolbox 4.40.0 Anaconda Assistant JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot. Install	 Anaconda Cloud Notebooks Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage. Launch	 anaconda_prompt 1.1.0 Opens a terminal instance with conda activated (requires menuinst 2.1.1 or greater). Launch
 JupyterLab 4.3.4 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 Notebook 7.3.2 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 Qt Console 5.6.1 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Launch	 Spyder 6.0.7 Scientific PYthon Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Launch	 VS Code 1.104.1 Streamlined code editor with support for development operations like debugging, task running and version control. Launch

MACHINE LEARNING SOFTWARE



TensorFlow

Google
colab



ANACONDA®



POWER BI



COPILOT

Install Python For Windows

1. Go to the official Python website <https://www.python.org/downloads/>
2. Download the latest version You'll see a big yellow button like  [Download Python 3.x.x](#) Click it — it will automatically detect your Windows version (64-bit is most common).
3. Run the Installer ([python-3.x.x.exe](#)) When the installer opens: Check the box — “Add Python 3.x to PATH” (very important) Then click Install Now
4. Wait for installation to complete Once it says “Setup was successful”, close the window.
5. Verify installation Open Command Prompt (CMD) or PowerShell, then type:

python --version



python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Windows

Download Python install manager

Or get the standalone installer for [Python 3.14.0](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/Unix](#), [macOS](#), [Android](#), [other](#)

Want to help test development versions of Python 3.15? [Pre-releases](#), [Docker images](#)

Install For macOS Users

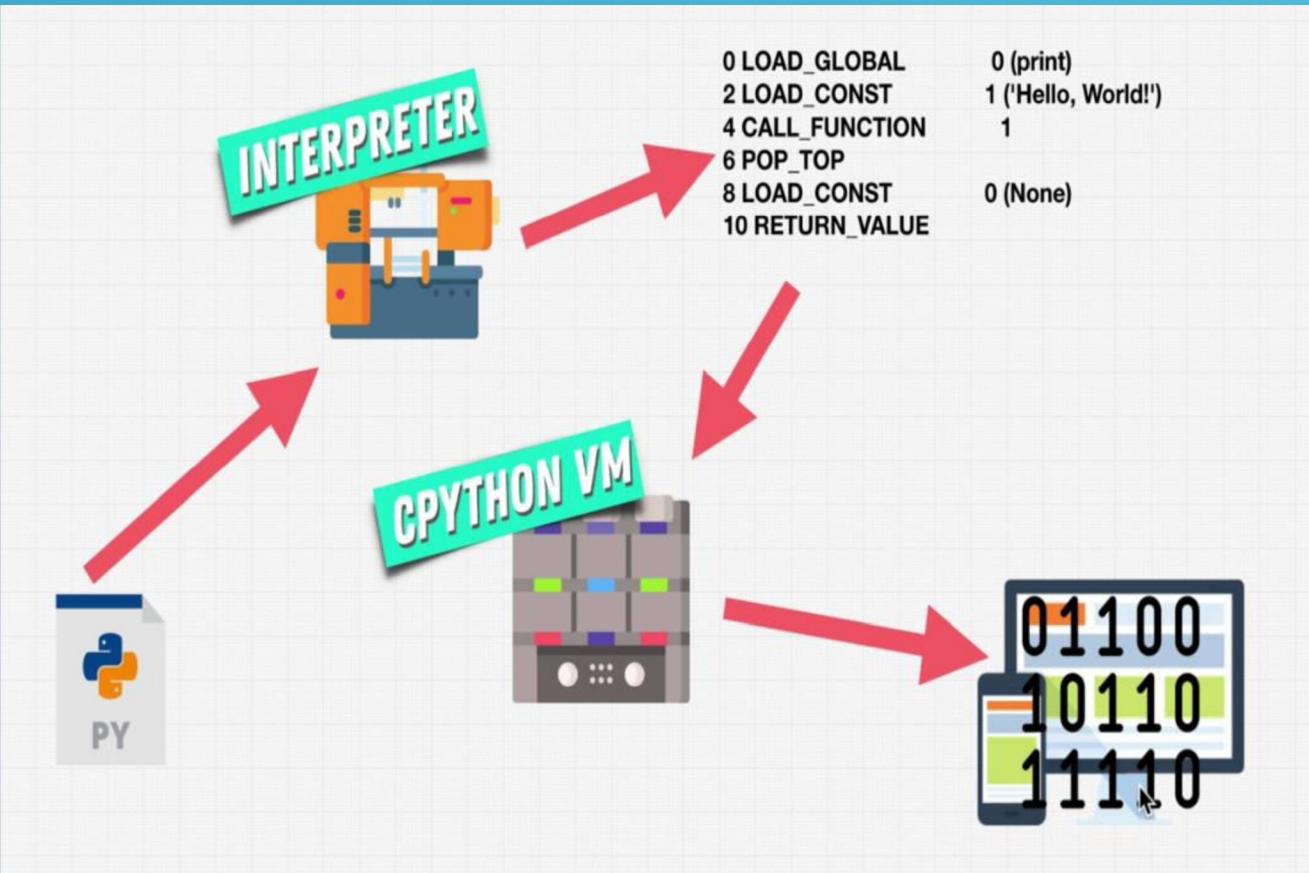
1. Open the official Python site <https://www.python.org/downloads/mac-osx/>
2. Download the latest .pkg installer For example: python-3.12.x-macos11.pkg
3. Open the downloaded file Follow the installation steps → “Continue”, “Agree”, “Install”.
4. Verify installation Open Terminal and type:

python3 --version



The screenshot shows the official Python website (<https://www.python.org>) with a dark blue header. The Python logo is on the left, followed by the word "python" in white. On the right of the header are buttons for "Donate", "Search" (with a magnifying glass icon), "GO", and "Socialize". Below the header is a navigation bar with links: "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". A large yellow button labeled "Download the latest version for Windows" is prominently displayed. Below it, text says "Or get the standalone installer for [Python 3.14.0](#)". Further down, it says "Looking for Python with a different OS? Python for [Windows](#), [Linux/Unix](#), [macOS](#), [Android](#), [other](#)". At the bottom, it says "Want to help test development versions of Python 3.15? [Pre-releases](#), [Docker images](#)". To the right of the text is a cartoon illustration of two parachutes, one yellow and one white, each carrying a brown cardboard box.

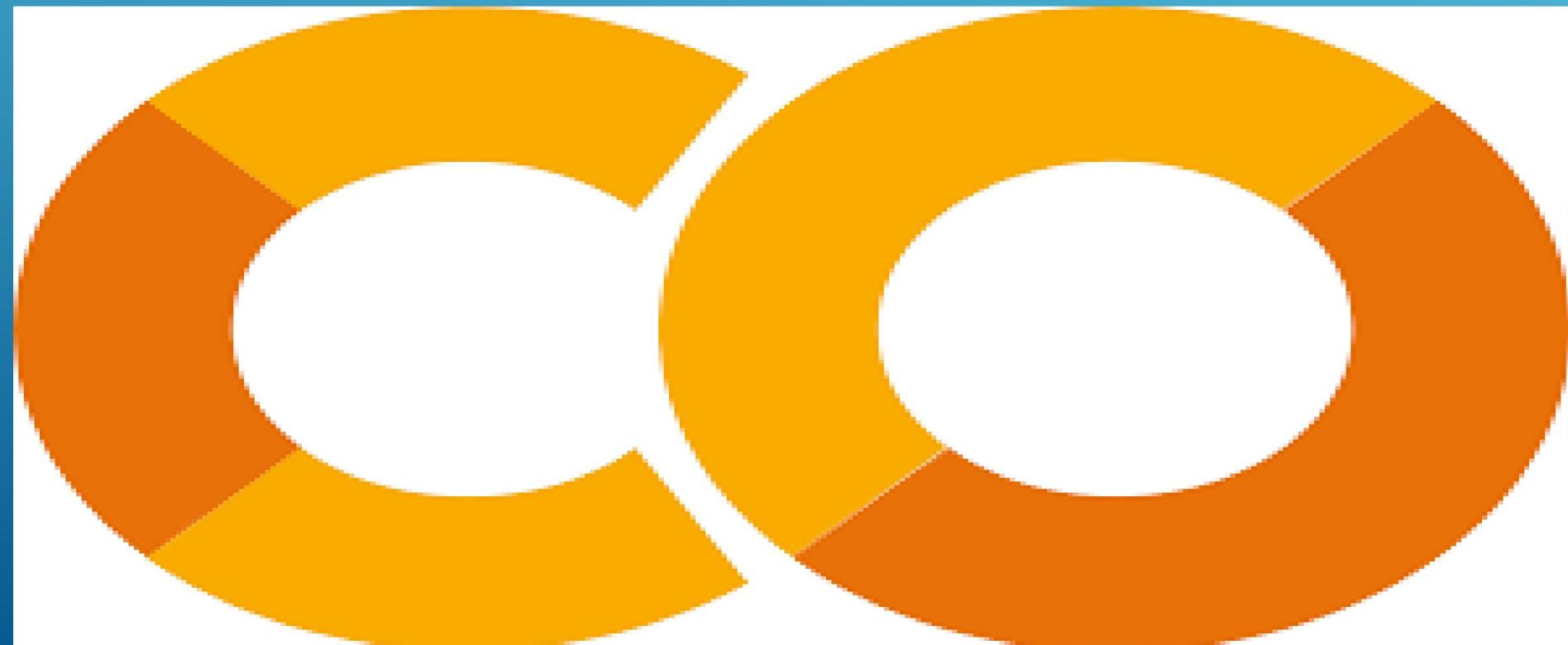
Python Interpreter



Google Collaboratory:

Or "Colab," is a free, hosted Jupiter notebook environment that requires no setup and runs entirely in the cloud.

It's particularly popular among data scientists, machine learning engineers, and students



Google Collaboratory:



Google Colab

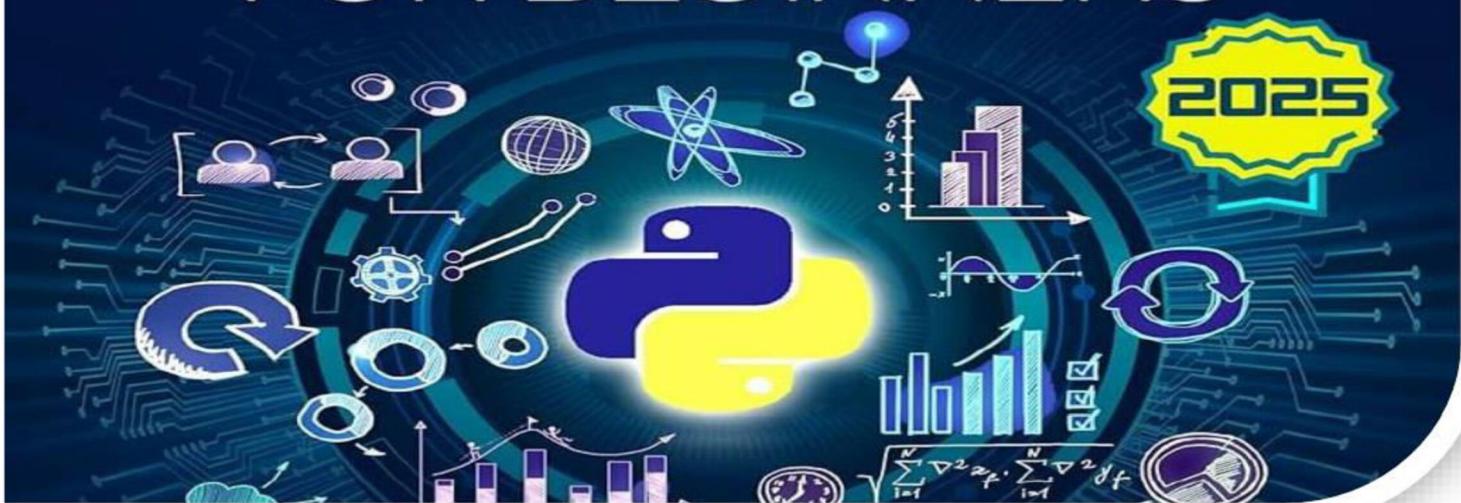
Pros

- Free access**
- No setup**
- Cloud-based**
- Collaboration**
- Pre-installed libraries**

Cons

- Internet dependency**
- Limited resources**
- Privacy concerns**

PYTHON PROGRAMMING FOR BEGINNERS



By: MOHD SALMAN

Python Syntax

Output:

```
print("Hello, World!")  
print("Welcome to Python Programming")  
print(10 + 5)
```

Python Syntax

Input:

```
name = input("Enter your name: ")  
print("Hello,", name)
```

Python DataTypes

Text Type: str

Numeric Types: int, float, complex

Sequence Types: list, tuple, range

Mapping Type: dict

Set Types: set, frozenset

Boolean Type: bool

Binary Types: bytes, bytearray, memoryview

None Type: NoneType

Python Datatypes

Numeric Types	Text Types	Collection Types																									
<p>int :</p> <p>int represents a whole number. It can be positive or negative and can be of any length.</p> <p>int variables are created when you assign an integer value to them.</p> <pre>a = 10 #Must be whole Number, No Decimals b = 123456789987654321 #Can be of any length c = -721 #Can be +ve or -ve</pre> <p>int variables can also be created using int().</p> <pre>d = int(10) #10 is assigned to d e = int(12.21) #only 12 is assigned to e, decimal value is ignored f = int("123") #123 is assigned to f g = int("abc") #Error : Invalid integer value. "abc" can't be converted to int</pre> <p>float :</p> <p>float represents decimal point numbers. They can be positive or negative and can be of any length.</p> <p>float variables are created when you assign decimal value to them.</p> <pre>a = 12.21 #Must be decimal point number b = 123456789.987654321 #can be of any length c = -45.67 #Can be +ve or -ve</pre> <p>float can also be used to represent scientific numbers with 'e' or 'E' indicating power of 10.</p> <pre>d = 854e5 #Valid e = 17E21 #Valid f = -65.23e6 #Valid</pre> <p>float variables can also be created using float() method.</p> <pre>g = float(12.21) #12.21 is assigned to g h = float("21.12") #21.12 is assigned to h i = float(10) #10.0 is assigned to i</pre> <p>complex :</p> <p>complex represents complex numbers with 'j' indicating imaginary part.</p> <pre>a = 5+7j print(a) #Output : (5+7j) print(a.real) #Output : 5.0 print(a.imag) #Output : 7.0</pre> <p>Complex numbers can also be constructed using complex() method.</p> <pre>a = complex(10) print(a) #Output : (10+0j) b = complex(12, 21) print(b) #Output : (12+21j)</pre>	<p>str :</p> <p>str represents strings in Python which are either enclosed within single quotation mark (' ') or double quotation mark (" ").</p> <p>It is the only one type which represents texts in Python.</p> <pre>a = "String" #Valid b = 'String' #Valid</pre> <p>Multiline strings are represented by three double quotes or three single quotes.</p> <pre>a = """First Line Second Line Third Line ... And So On""" #Valid b = ''''First Line Second Line Third Line And So On''' #Valid</pre> <p>Like many other programming languages, strings in python are treated as an array of characters. So, you can retrieve individual characters or iterate whole string.</p> <p>#Retrieving a character</p> <pre>a = "String" print(a[2]) #Output : r</pre> <p>#Iterating using for loop</p> <pre>for x in "String" : print(x)</pre> <p>Python doesn't have a data type which represents a single character. Characters in Python are treated as str type with length 1.</p>	<p>Collection types represents a collection of values.</p> <p>list, tuple, set, dict are important collection types in Python.</p> <p>All 4 have different usage and characteristics.</p> <table border="1"> <thead> <tr> <th></th><th>list</th><th>tuple</th><th>set</th><th>dict</th></tr> </thead> <tbody> <tr> <td>Indexed</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td></tr> <tr> <td>Ordered</td><td>Yes</td><td>Yes</td><td>No</td><td>Yes</td></tr> <tr> <td>Changeable</td><td>Yes</td><td>No</td><td>No</td><td>Yes</td></tr> <tr> <td>Duplicates</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td></tr> </tbody> </table> <p>list : Indexed, ordered, changeable and can have duplicates</p> <pre>stationeryList = ["Pen", "Pencil", "Eraser", "Pen", "Sharpner", "Pencil"]</pre> <p>tuple : Indexed, ordered, unchangeable and can have duplicates</p> <pre>nameTuple = ("John", "Arun", "Ben", "Zia", "Arun")</pre> <p>set : Unindexed, unordered, unchangeable and no duplicates</p> <pre>numberSet = {1.1, 2.2, 3.3, 4.4, 5.5}</pre> <p>dict : Unindexed, ordered, changeable and no duplicates</p> <pre>anyDictionary = { "One" : 1, 2 : "Two", "Three" : 3.0, "Four" : True, }</pre>		list	tuple	set	dict	Indexed	Yes	Yes	No	No	Ordered	Yes	Yes	No	Yes	Changeable	Yes	No	No	Yes	Duplicates	Yes	Yes	No	No
	list	tuple	set	dict																							
Indexed	Yes	Yes	No	No																							
Ordered	Yes	Yes	No	Yes																							
Changeable	Yes	No	No	Yes																							
Duplicates	Yes	Yes	No	No																							
		<p>Binary Types</p> <p>Python provides 3 data types to handle binary data. They are bytes, bytearray and memoryview.</p> <pre>a = b"any_Char" print(a) #Output : b'any_Char' b = bytearray(10) print(b) #Output : bytearray(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00') c = memoryview(bytes(10)) print(c) #Output : <memory at 0x0000000002E2B580></pre>																									
		<p>type()</p> <p>type() function is used to check the type of any variable.</p> <pre>a = 10 print(type(a)) #Output : <class 'int'> b = 12.21 print(type(b)) #Output : <class 'float'> c = complex(7, 5) print(type(c)) #Output : <class 'complex'> d = "String" print(type(d)) #Output : <class 'str'> e = 10 > 5 print(type(e)) #Output : <class 'bool'> f = None print(type(f)) #Output : <class 'NoneType'></pre>																									