



# *Intro. To Python & Development environment*

*By : Eng. Shereen Ebrahim*

# Hello



Join our  
Google classroom



Google Meet



Google Drive



VS Code

First of all, we would like to make sure that you have all the **essential apps** that we will need throughout our course and that you have joined our **Classroom**.

# INTRODUCTION

## Artificial Intelligence

- What is Artificial Intelligence?
- Brief history of AI development
- Difference between AI, Machine Learning (ML), and Deep Learning (DL)

# Importance of AI?

The technology that will change the life as we know it.

- AI in daily life (voice assistants, recommendation systems, automation)
- AI in industries (healthcare, finance, robotics, cybersecurity)
- How AI is shaping the future

# STEPS TO LEARN AI

---

Programming

01

Math &  
Statistics

02

Machine  
Learning

03

Deep  
Learning

04

Projects

05

1

# Programming

- Best programming language for AI: Python
- Why Python? (Easy syntax, rich libraries, strong community)
- Essential Python libraries for AI: NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, PyTorch

# STEPS TO LEARN AI

---

Programming

01

Math &  
Statistics

02

Machine  
Learning

03

Deep  
Learning

04

Projects

05

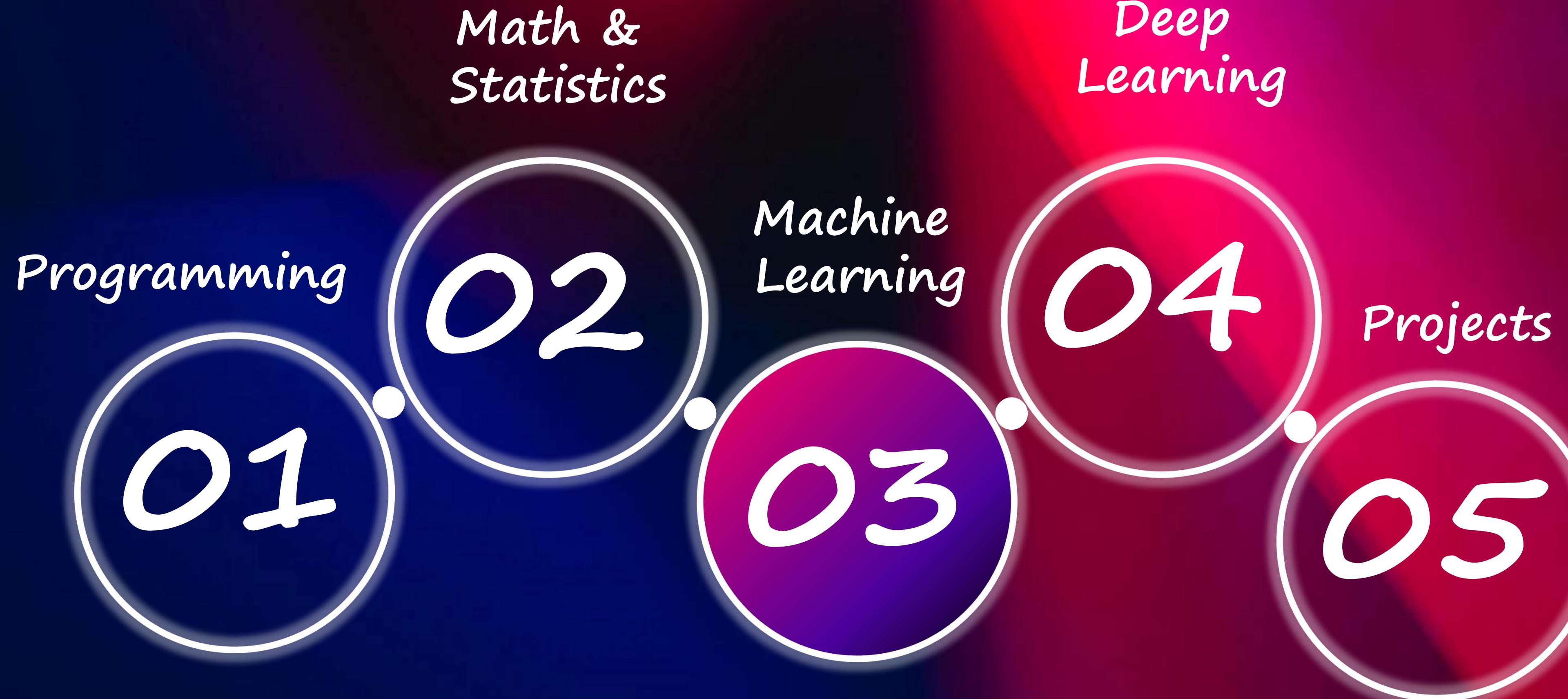
0  
2

# *Mathematics & Statistics*

- Linear algebra (vectors, matrices, operations)
- Probability and statistics (mean, variance, distributions)
- Calculus basics (derivatives, integrals)

# STEPS TO LEARN AI

---



3

# Machine Learning

- Supervised vs. unsupervised learning
- Common algorithms (Linear Regression, Decision Trees, Neural Networks)
- Building and training models using Scikit-learn

# STEPS TO LEARN AI

---

Programming

01

Math &  
Statistics

02

Machine  
Learning

03

Deep  
Learning

04

Projects

05

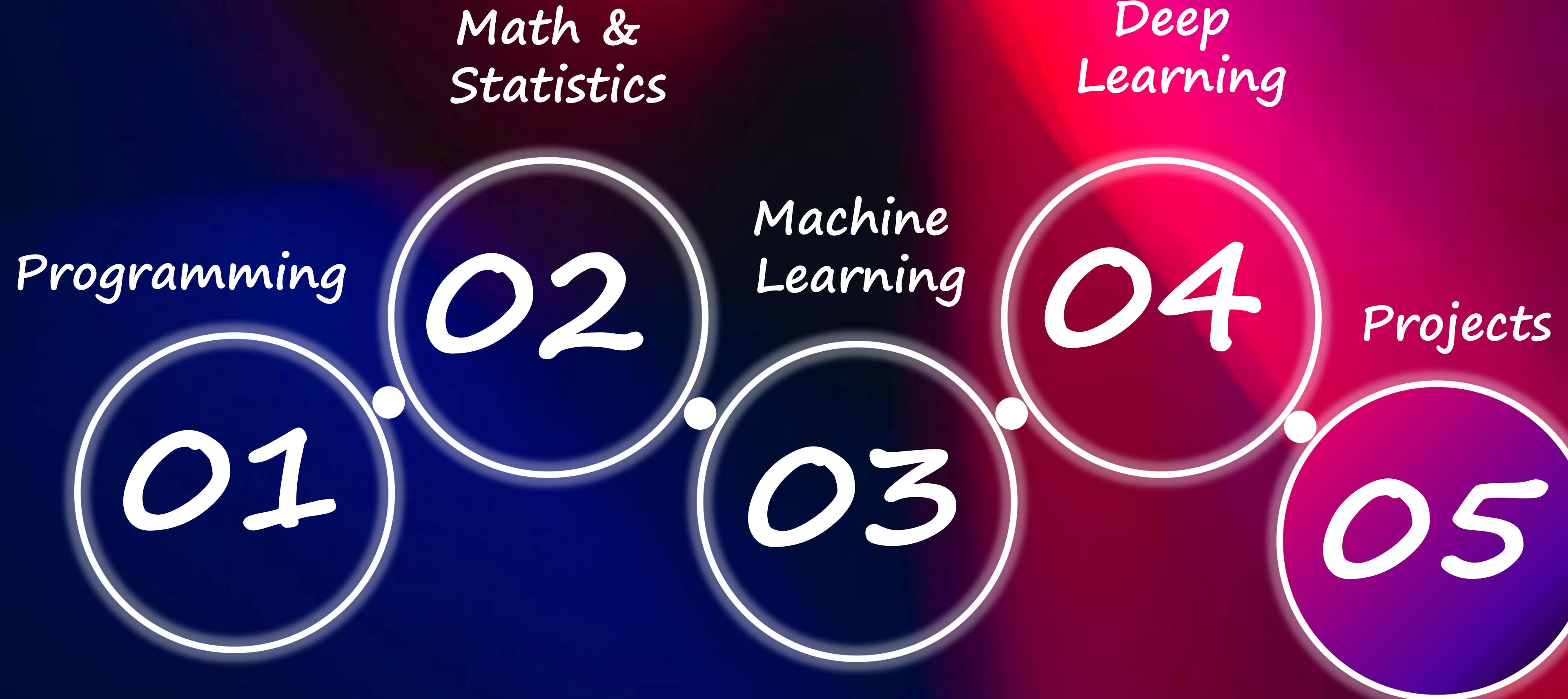
4

# Deep Learning

- Introduction to neural networks
- Deep learning frameworks: TensorFlow, Keras, PyTorch
- Hands-on projects (image classification, natural language processing)

# STEPS TO LEARN AI

---



O  
S

## AI Projects

- Start with small projects (predicting house prices, sentiment analysis)
- Explore real-world applications (chatbots, self-driving cars, AI in gaming)
- Participate in Kaggle competitions for hands-on learning



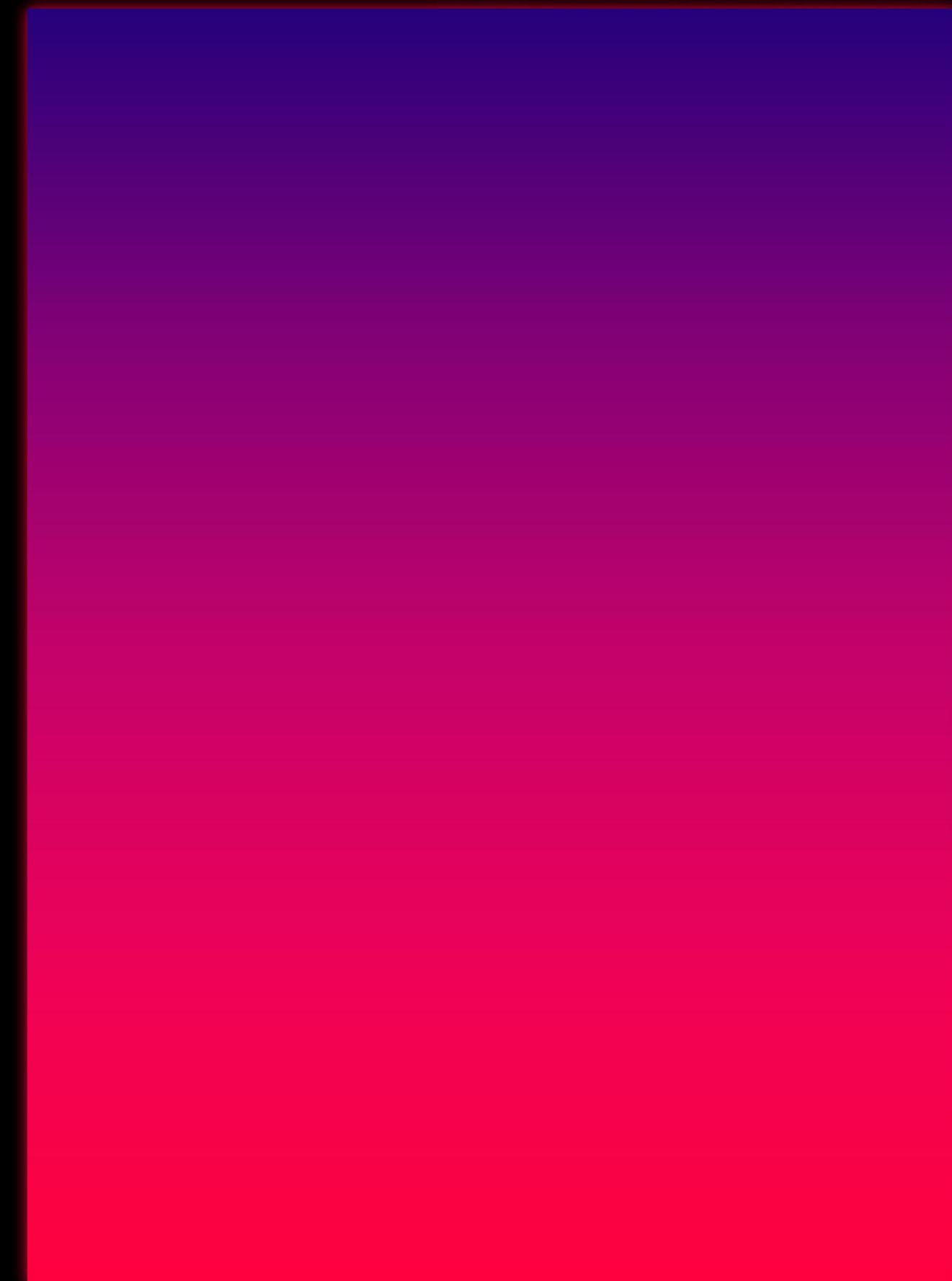
## Python in Artificial Intelligence

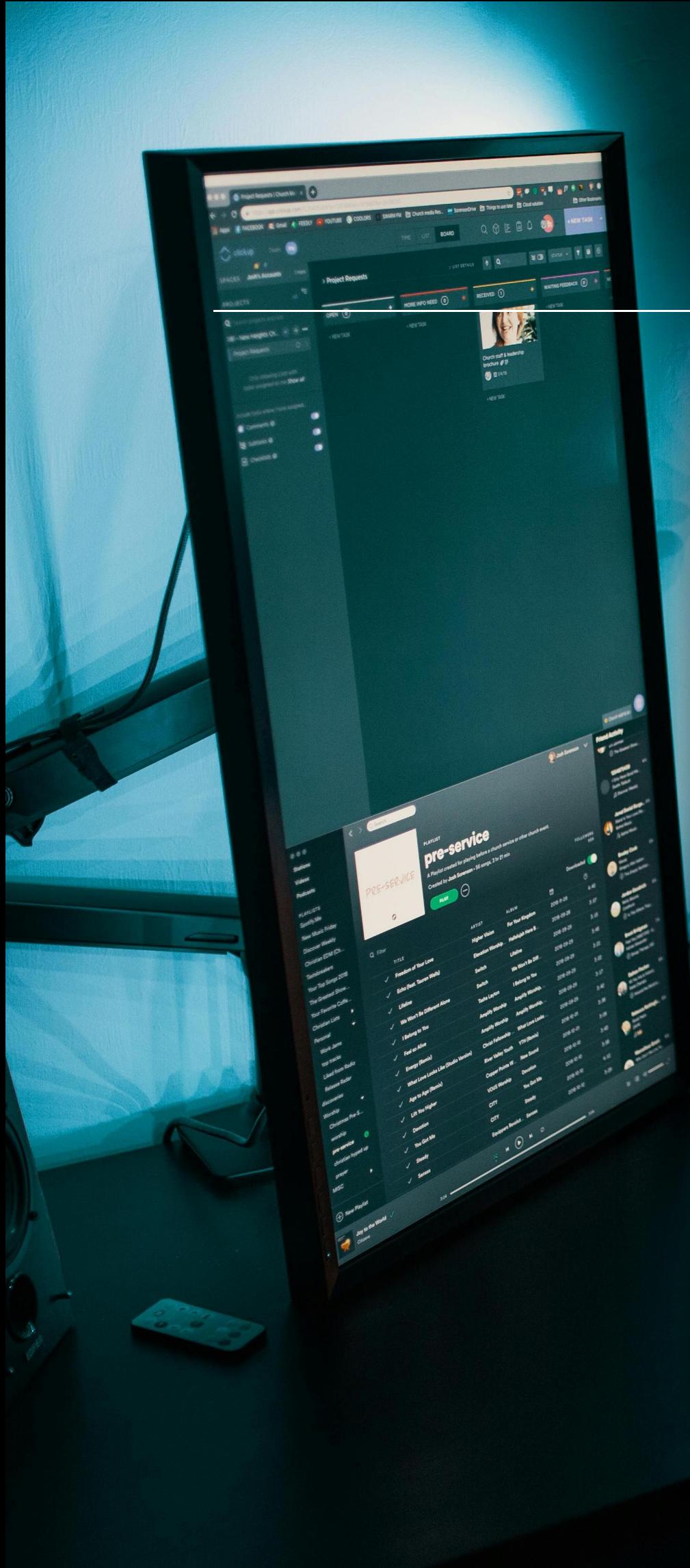
Python is the top choice for developing artificial intelligence (AI) applications due to its ease of use, powerful data-handling capabilities, and extensive libraries that simplify machine learning and deep learning model development.

Core of AI

# INTRODUCTION TO PYTHON

- 
- What is Python?
  - When was it created, and who is its founder?
  - Why is it one of the most popular programming languages today?





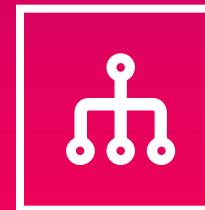
## FEATURES OF PYTHON



**Easy to learn and read**



**Powerful libraries and frameworks**



**VERSATILE AND  
MULTI-PURPOSE**



**Open-source and  
large community  
support**



# Python uses

- Web development (Django, Flask)
- Data analysis & AI (Pandas, NumPy, TensorFlow)
- Game development & graphics
- Task automation and scripting

# Python Basics (Syntax & Fundamentals)

- Printing text (`print()`)
- Variables and data types
- Arithmetic and logical operations
- Conditional statements (`if-else`)
- Loops (`for, while`)

```
    resp_iter = self.stub.GetStatuses(stub_context)
    statuses = {}
    async for data in resp_iter:
        status = Status(
            status_id=data.id, name=data.name,
            value=data.value)
        statuses[status.name] = status
    return statuses
```

# Functions and Libraries in Python

- Defining functions (def) and their importance
- Built-in libraries like math and random
- Importing external libraries



# Why is Python Used in AI?

- Easy to Learn and Read: Python has a simple and clear syntax, making it beginner-friendly.
- Strong Community Support: A large number of developers contribute solutions on platforms like Stack Overflow and GitHub.
- Powerful Libraries for Data Processing: Python offers ready-to-use libraries that help in data handling:  
**NumPy & Pandas:** For data analysis and manipulation.  
**Matplotlib & Seaborn:** For data visualization.
- Advanced Libraries for Machine Learning and Deep Learning:
  - **Scikit-learn:** For building simple machine learning models.
  - TensorFlow & PyTorch:** For creating deep neural networks.

jupyter main Last Checkpoint: Last Tuesday at 8:14 PM (unsaved changes)

In [1]: 1 print('Hello world')  
Hello world

In [2]: 1 print('This is another code cell')  
This is another code cell

In [ ]: 1



# Printing output

```
In [1]: 1 print('Hello world')
          2
          3 print("We can use double quotes")
```

```
Hello world
We can use double quotes
```

```
In [2]: 1 print("""
          2           We can also
          3           use triple quotes
          4 """)
```

```
We can also
use triple quotes
```

# Comments

In [10]:

```
1 # This is a single line comment
2 print('Hello world 1')
3
4 """
5 This is a
6 multi-line comment
7 """
8 print('Hello world 2')
```

```
Hello world 1
Hello world 2
```

# Variables, arithmetic operators

In [1]:

```
1 x = 7
2 y = 2
3 z = x + y
4 print("z equals", z)
```

z equals 9

In [9]:

```
1 print( x + y )
2 print( x - y )
3 print( x * y )
4 print( x / y )
5 print( x // y )
6 print( x % y )
7 print( x ** y )
```

9  
5  
14  
3.5  
3  
1  
49

# Spaces are important in Python

```
In [11]: 1 print('Hello world 1')
           2 print('Hello world 2')
```

```
File "<ipython-input-11-f87c9ede6d98>", line 2
  print('Hello world 2')
^
```

**IndentationError: unexpected indent**

# Error message

Read from  
bottom to top

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

File "<ipython-input-11-f87c9ede6d98>", line 2

print('Hello world 2')

^

IndentationError: unexpected indent

# Error message

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

```
File "<ipython-input-11-f87c9ede6d98>", line 2
    print('Hello world 2')
    ^

```

IndentationError: unexpected indent

Error reason  
(this means our spaces are wrong)

# Error message

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

Line

```
File "<ipython-input-11-f87c9ede6d98>", line 2
    print('Hello world 2')
    ^
```

IndentationError: unexpected indent

# Error message

In [11]:  
Here it is

```
1 print('Hello world 1')
2 print('Hello world 2')
```

```
File "<ipython-input-11-f87c9ede6d98>", line 2
    print('Hello world 2')
 ^
IndentationError: unexpected indent
```

# Error message

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

Filename  
(this means  
our cell)

```
File "<ipython-input-11-f87c9ede6d98>", line 2
    print('Hello world 2')
    ^

```

IndentationError: unexpected indent

# Error message

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

File "<ipython-input-11-f87c9ede6d98>", line 2

```
    print('Hello world 2')
    ^
```

Line number

IndentationError: unexpected indent

# Error message

In [11]:

```
1 print('Hello world 1')
2 print('Hello world 2')
```

File "<ipython-input-11-f87c9ede6d98>", line 2  
print('Hello world 2')  
^

IndentationError: unexpected indent

If this is not your code  
(like when you import  
a library), continue  
reading from bottom  
to top

# Order of running cells is important

```
In [ ]: 1 myvariable = 5
```

```
In [13]: 1 print(myvariable)
```

```
-----  
NameError
```

```
Traceback (most recent call last)
```

```
<ipython-input-13-9eb3f011738c> in <module>
```

```
----> 1 print(myvariable)
```

```
NameError: name 'myvariable' is not defined
```

# Order of running cells is important

```
In [ ]: 1 myvariable = 5
```

```
In [13]: 1 print(myvariable)
```

```
-----  
NameError
```

```
Traceback (most recent call last)
```

```
<ipython-input-13-9eb3f011738c> in <module>
```

```
----> 1 print(myvariable)
```

```
NameError: name 'myvariable' is not defined
```

This means either we forgot to run the previous cell or the variable name is wrong  
(or the variable is not defined at all)

# Order of running cells is important

We must run this first

```
In [ ]: 1 myvariable = 5
```

```
In [13]: 1 print(myvariable)
```

---

```
NameError Traceback (most recent call last)
<ipython-input-13-9eb3f011738c> in <module>
----> 1 print(myvariable)

NameError: name 'myvariable' is not defined
```

# Variables are case-sensitive

```
In [14]: 1 MyVariable = 5
```

```
In [15]: 1 print(myvariable)
```

```
NameError
```

```
Traceback (most recent call last)
```

```
<ipython-input-15-9eb3f011738c> in <module>
```

```
----> 1 print(myvariable)
```

```
NameError: name 'myvariable' is not defined
```

# Variables are case-sensitive

```
In [14]: 1 MyVariable = 5
```

```
In [15]: 1 print(myvariable)
```

```
NameError
```

```
Traceback (most recent call last)
```

```
<ipython-input-15-9eb3f011738c> in <module>
----> 1 print(myvariable)
```

```
NameError: name 'myvariable' is not defined
```

Again, this means either we forgot to run the previous cell or the variable name is wrong (or the variable is not defined at all)

# Variables are case-sensitive

```
In [14]: 1 MyVariable = 5
In [15]: 1 print(myvariable)
```

Names are different

---

```
NameError: name 'myvariable' is not defined
```

Traceback (most recent call last)  
<ipython-input-15-9eb3f011738c> in <module>  
----> 1 print(myvariable)

# If statement

In [24]:

```
1 x = 10
2 if x > 1 and x < 20:
3     print('Yes')
4     print('Condition matched')
5 else:
6     print('No')
7     print('Condition not matched')
8 print('Outside if statement')
```

Yes  
Condition matched  
Outside if statement

# Data structures

In [28]:

```
1 myTuple = (10, 20, 30)
2
3 myList = [10, 20, 30]
```

In [29]:

```
1 myDictionary = {
2     'name': 'Ahmed',
3     'age': 30,
4     'marks': [10, 20, 30],
5 }
```

# Length of a list (or a tuple)

In [32]:

```
1 myList = [10, 20, 30]
2 n = len(myList)
3 print(n)
```

3

# For loop

In [34]:

```
1 for i in range(5):  
2     print(i)
```

```
0  
1  
2  
3  
4
```

# For loop

In [30]:

```
1 myList = [10, 20, 30]
2
3 for i in range(len(myList)):
4     print(myList[i])
```

```
10
20
30
```

In [31]:

```
1 for x in myList:
2     print(x)
```

```
10
20
30
```

# While loop

In [35]:

```
1 x = 1
2 while x <= 5:
3     print(x)
4     x += 1
```

```
1
2
3
4
5
```

# Functions

In [36]:

```
1 def add(x, y):  
2     z = x + y  
3     return z
```

In [38]:

```
1 s = add(2, 3)  
2 print(s)
```

# GIT

## What is Git?



- Git is a version control system that helps track changes in code.
- It allows collaboration, branching, and rollback to previous versions.

# GIT

Setting Up a Git  
Repository in VS  
Code



This creates a .git folder  
to track changes.

Write these commands in terminal

`git init`

# GIT

Configure Git  
(First-time setup  
only)

---



Write these commands in terminal

```
git config --global user.name "Your Name"  
git config --global user.email "youremail@example.com"
```

# GIT

Connect to a Remote Repository (Github)



- Create a repository on GitHub.
- Copy the repository URL.

Write these commands in terminal

```
git remote add origin <repo_URL>
```

# GIT

## Essential Git Commands (Push & Pull Workflow)



`git status`

Shows untracked & modified files.

`git commit -m "Your  
commit message"`

Saves changes locally.

`git pull origin  
main`

Updates local code with  
the latest version.

`git add .`

Adds all modified files for commit.

`git push origin main`

Uploads changes to GitHub

# Further reading

- W3Schools:

- <https://www.w3schools.com/python/default.asp>

- Youtube:

- <https://youtube.com/playlist?list=PLIjLGHnioqmbI6eSb4tU1VdgOEq8nZaY2>

- Feel free to search for more online sources

# Python Libraries For Artificial Intelligence



Data Processing and Statistical  
Analysis Libraries



Machine Learning and  
Deep Learning Libraries



Natural Language  
Processing (NLP) Libraries



Computer Vision Libraries

# Data Processing and Statistical Analysis Libraries

Pandas → Used for data manipulation and organization in table formats (CSV, Excel, etc.).

NumPy → Used for numerical operations and handling arrays, forming the backbone of many AI libraries.

Matplotlib → Used for data visualization and creating charts.

Seaborn → Built on Matplotlib, it provides advanced statistical visualizations.

# Python Libraries For Artificial Intelligence



Data Processing and Statistical  
Analysis Libraries



Machine Learning and  
Deep Learning Libraries



Natural Language  
Processing (NLP) Libraries



Computer Vision Libraries

# Machine Learning and Deep Learning Libraries

**Scikit-learn** → Provides machine learning algorithms like classification, clustering, and regression.

**TensorFlow** → A library from Google used for building and training deep learning models.

**Keras** → Built on TensorFlow, it simplifies the process of creating deep learning models.

**PyTorch** → An open-source library from Facebook used for deep learning with dynamic computation graphs.

# Python Libraries For Artificial Intelligence



Data Processing and Statistical  
Analysis Libraries



Machine Learning and  
Deep Learning Libraries



Natural Language  
Processing (NLP) Libraries



Computer Vision Libraries

# Natural Language Processing (NLP) Libraries

**NLTK (Natural Language Toolkit)** → Used for text processing, such as sentence analysis and keyword extraction.

**spaCy** → A fast and advanced library for text analysis and Named Entity Recognition (NER).

**TextBlob** → A simple library for sentiment analysis and text correction.

**transformers (Hugging Face)** → Used for running advanced NLP models like BERT and GPT.

# Python Libraries For Artificial Intelligence

01

Data Processing and Statistical  
Analysis Libraries

02

Machine Learning and  
Deep Learning Libraries

03

Natural Language  
Processing (NLP) Libraries

04

Computer Vision Libraries

# Computer Vision Libraries

**OpenCV** → Used for image and video processing, object detection, and face recognition.

**Pillow (PIL)** → Used for image manipulation and editing.

**Torchvision** → Works with PyTorch for deep learning-based image processing.

# Assignment

- Use Python to create a simple application from you choice (Example: Simple calculator)
- Build a Python script that processes and analyzes a dataset (Example a script that calculates total store sales from excel sheet)



*See you in the next session*

