

Module-1: Introduction to Python and NumPy

◆ Introduction

Python is a powerful and easy-to-learn programming language that is widely used in data science, machine learning, and scientific computing. One of its key strengths is its rich ecosystem of libraries, such as **NumPy**, which is essential for numerical operations and array-based computing.

In this module, we explore the following:

- Installing Python and required packages
 - Setting up and managing virtual environments using the `venv` module
 - Introduction to **NumPy** arrays: creation, indexing, data types
 - Advanced array operations like broadcasting, views, and universal functions
 - Input/output operations using NumPy for reading/writing data
-



Python Programmes

◆ 1. Print a Message and Python Version

```
import sys
print("Hello, Python!")
print("Python version:", sys.version)
```

◆ 2. Check if NumPy is Installed

```
try:
    import numpy
    print("NumPy is installed.")
except ImportError:
    print("NumPy is not installed.")
```

◆ 3. Create a Virtual Environment (Instructional - not executable as script)

```
# These are terminal commands, not Python code
python -m venv myenv
# Activate the environment (on Windows)
myenv\Scripts\activate
# On Linux/Mac
source myenv/bin/activate
```

◆ 4. Create a 1D and 2D NumPy Array

```
import numpy as np
```

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([[1, 2], [3, 4]])
print("1D Array:", arr1)
print("2D Array:\n", arr2)
```

◆ 5. Indexing and Slicing a NumPy Array

```
import numpy as np
arr = np.array([[10, 20, 30], [40, 50, 60]])
print("Element at (0,1):", arr[0, 1])
print("First row:", arr[0])
print("First column:", arr[:, 0])
```

◆ 6. Check and Change Data Types

```
import numpy as np
arr = np.array([1, 2, 3])
print("Original dtype:", arr.dtype)
arr_float = arr.astype(float)
print("Changed dtype:", arr_float.dtype)
```

◆ 7. Broadcasting Example

```
import numpy as np
arr = np.array([1, 2, 3])
print("Original Array:", arr)
print("After adding 5:", arr + 5)
```

◆ 8. Copies vs Views

```
import numpy as np
original = np.array([1, 2, 3])
copy = original.copy()
view = original.view()
original[0] = 99
print("Original:", original)
print("Copy (not affected):", copy)
print("View (affected):", view)
```

◆ 9. Array Arithmetic Operations

```
import numpy as np
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
print("Addition:", a + b)
print("Multiplication:", a * b)
```

◆ 10. Universal Functions (ufuncs)

```
import numpy as np
arr = np.array([1, 4, 9])
print("Square Root:", np.sqrt(arr))
print("Exponential:", np.exp(arr))
```

◆ 11. Statistical Functions

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print("Mean:", np.mean(arr))
print("Standard Deviation:", np.std(arr))
print("Max:", np.max(arr))
```

◆ 12. Save a NumPy Array to a File

```
import numpy as np
arr = np.array([10, 20, 30])
np.save('my_array.npy', arr)
loaded = np.load('my_array.npy')
print("Loaded array:", loaded)
```

◆ 13. Load from a Text File

```
# Save this manually as 'data.txt' with numbers
# Example content:
# 1 2 3
# 4 5 6
```

```
import numpy as np
arr = np.loadtxt('data.txt')
print("Loaded array:\n", arr)
```

◆ 14. Temperature Converter (C to F)

```
import numpy as np
celsius = np.array([0, 20, 37, 100])
fahrenheit = celsius * 9/5 + 32
print("Fahrenheit:", fahrenheit)
```

◆ 15. Matrix Multiplication

```
import numpy as np
a = np.array([[1, 2], [3, 4]])
b = np.array([[2, 0], [1, 2]])
result = np.dot(a, b)
print("Matrix Multiplication Result:\n", result)
```