# PYTHON FOR DATA SCIENCE

## ASSIGNMENT – 01

Question- 01:

Which of the following variable names are INVALID in Python?

a) 1_variable
b) variable_1
c) variable1
d) variable#

Ans: (a) 1_variable

Question- 02:

Which of the following object does not support indexing?

a) Tuple
b) List
c) Dictionary
d) Set

Ans: (d) Set

Question- 03:

Explain with an example how **NumPy arrays differ from Python lists** in terms of functionality and performance. Write a short Python code snippet to demonstrate element-wise addition using both a list and a NumPy array, and explain the output.

Ans:

Python lists are general-purpose containers that can hold heterogeneous data types and are flexible but slower for numerical computations. NumPy arrays, on the other hand, are **homogeneous, multi-dimensional arrays** optimized for fast mathematical operations using vectorization and low-level C implementations.

**Key Differences:**

1. **Performance:** NumPy arrays are much faster due to vectorization.
2. **Memory Efficiency:** Arrays use less memory compared to lists for large data.
3. **Operations:** Mathematical operations are applied element-wise in NumPy arrays, while lists require explicit loops.

Ex:

```
import numpy as np

# Using Python list
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]
list_sum = [list1[i] + list2[i] for i in range(len(list1))]
print("List Addition:", list_sum)

# Using NumPy array
arr1 = np.array([1, 2, 3, 4])
arr2 = np.array([5, 6, 7, 8])
arr_sum = arr1 + arr2
print("Array Addition:", arr_sum)
```

**Output:**

```
List Addition: [6, 8, 10, 12]
Array Addition: [ 6  8 10 12]
```

Question- 04:

Explain the concept of **slicing in Python sequences** with suitable examples. Show how slicing works on both a Python string and a NumPy array, and highlight the differences in their usage.

Ans:

**Slicing** is a technique in Python used to extract a portion of a sequence (like a string, list, or array) using the syntax:

sequence[start:stop:step]

**Example with a String:**

```
text = "DataScience"
print(text[0:4])     # Output: Data
print(text[::2])     # Output: DtSine
print(text[::-1])    # Output: ecneicSataD (reversed)
```

**Example with a NumPy Array:**

```
import numpy as np

arr = np.array([10, 20, 30, 40, 50, 60])

print(arr[1:4])      # Output: [20 30 40]
print(arr[::2])      # Output: [10 30 50]
```

```
print(arr[::-1])     # Output: [60 50 40 30 20 10]
```

**Key Difference:**

- In **strings**, slicing returns a new string (immutable).
- In **NumPy arrays**, slicing returns a **view** of the original array (mutable). Changing the slice may affect the original array.