

PYTHON FOR DATA SCIENCE

ASSIGNMENT 1

Name: Budumuru Nishitha

SRN: PES2UG22EC036

MCQ-based Questions:

Q1: In Spyder, which of the following Python commands correctly removes the variables a, b, and c from the environment at once?

- a) `rm (a, b, c)`
- b) `del (a, b, c)`
- c) `rm a, b, c`
- d) `del a, b, c`

A1: d) `del a, b, c`

'rm' is a Linux/Shell command used to remove files from the operating system, not variables inside Python.

In Spyder (and Python in general), we don't use operating system commands to remove variables; we use the Python keyword 'del'.

If we use parentheses, Python first tries to evaluate it as a tuple object and then deletes the tuple itself. Since there is no single variable that stores (a, b, c) as a tuple, it raises an error instead of deleting the names individually.

Q2: What will the output of the following code be?

```
import numpy as np
arr = np.array([[1,2,3],[4,5,6]])
print(arr.ndim)
```

- a) 1
- b) 2
- c) 3
- d) Error

A2: b) 2

The array has 2 dimensions (rows and columns)

Descriptive Questions:

Q1: What are the basic libraries in Python? Mention their primary functionality. Write code to display all the modules within any one of these basic libraries.

A1: The basic libraries in Python are:

- a) numpy - numerical python
- b) pandas - dataframe python (paneled dataframe)
- c) matplotlib - visualization
- d) Sklearn - machine learning

The code to display all the modules within numpy:

```
import numpy
content = dir(numpy)
print(content)
```

Q2: Explain the differences between the shape, size, and ndim attributes of a NumPy array. Provide an example for each and discuss how these attributes help in analysing and manipulating arrays.

A2: The differences between the shape, size and ndim attributes of a NumPy array is as follows, along with their examples:

- shape: Returns a tuple showing the number of elements in each dimension.
 - Example- Helps understand the structure of an array:
`arr = np.array([[1,2,3],[4,5,6]])`
`print(arr.shape)` # (2,3) → 2 rows, 3 columns
- size: Returns the total number of elements in the array.
 - Example- Useful in memory estimation and iteration:
`print(arr.size)` # 6 (since $2 \times 3 = 6$ elements)
- ndim: Returns the number of dimensions (axes) of the array.
 - Example- Helps distinguish between 1D, 2D, or higher-dimensional arrays:
`print(arr.ndim)` # 2 → 2D array

These three attributes allow us to inspect the structure (with the help of 'shape'), check the total number of elements (with the help of 'size'), and understand the dimensionality (with the help of 'ndim'). This is crucial when reshaping, performing matrix operations, or debugging array mismatches.