# Python assignment 1

**1. What is namespace? Explain with example.**

Ans: A **namespace** in programming is a container that holds a set of identifiers (such as variable names, function names, etc.) and ensures that all identifiers within it are unique. It helps to avoid naming conflicts by differentiating between identifiers that may have the same name but are used in different contexts or scopes.

#### ****Types of Namespaces in Python****:

* **Built-in Namespace**: Contains functions and objects provided by Python (e.g., print(), len()).
* **Global Namespace**: Contains names defined at the top-level of a script or module.
* **Local Namespace**: Contains names defined inside a function or method.

**Eg:**

# Global namespace

x = "global"

def my\_function():

    # Local namespace

    x = "local"

    print("Inside function:", y)

my\_function()

print("Outside function:", x)

* x in the global namespace has the value "global".
* y in the local namespace of my\_function has the value "local"

**2. Implement a function that accepts a string and returns it reversed.**

**Ans:** def reverse\_string(s):

    return s[::-1]

input\_string = "hello"

reversed\_string = reverse\_string(input\_string)

print(reversed\_string)

# Python assignment 2

**1. Create a function that takes three numbers and returns their average. Use default parameters to make each number 0 .**

Ans: def average\_of\_three(a=0, b=0, c=0):

    return (a + b + c) / 3

print(average\_of\_three(10, 20, 30))

print(average\_of\_three(10, 20))

print(average\_of\_three())

**2. Explain how to import a specific function from a Python module. Provide an Example.**

Ans: In Python, you can import a specific function from a module using **from module\_name import** **function\_name** syntax. This allows you to use the function directly without prefixing it with the module name.

Eg.

from math import sqrt

number = 16

result = sqrt(number)

print("Square root of", number, "is", result)

# .