**ASSIGNMENT ON (LIST, TURPLE, SETS, DICTIONARIES, STATEMENT, LOOP)**

1. Print the second item in the fruits list.

fruits = ["apple", "banana", "cherry"]

print(fruits[1])

1. Use the correct syntax to print the number of items in the fruits tuple.

fruits = ("apple", "banana", "cherry")

print(len(fruits))

1. Check if "apple" is present in the fruits set.

fruits = {"apple", "banana", "cherry"}

if "apple" in fruits:

print("Yes, apple is a fruit!")

4. Use the get method to print the value of the "model" key of the car dictionary.

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(car.get('model'))

5. Change the "year" value from 1964 to 2020.

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car['year'] = 2020

6. Print "Hello World" if a is greater than b.

a = 50

b = 10

if a > b:

print("Hello World")

7. Print "Yes" if a is equal to b; otherwise, print "No."

a = 50

b = 10

if a == b:

print("Yes")

else:

print("No")

8. Print i as long as i is less than 6.

i = 1

while i < 6:

print(i)

i += 1

9.Define a

“Function  in Python”,

Function Parameters and Arguments,

Function Return Values,

Function Calling and Scope

**Function:**

Function in python is used to increase the reusability of the code. Encapsulate/ Inserting a task/code in a block. So, we can use it by calling its name whenever we require, instead of writing that code again.

**Example:**

def add (a, b):

return a + b

print (add (3, 4))

print (add (4, 8))

**Function Return Values:**

Returns value in Python is used to display the value of what the function concludes in the end by using the result using the 'return' statement.

**Example:**

As above.

**Parameters:**

Parameters are the variable or placeholder value passed to replicate or to show how to insert value in function

**Example:**

def add(a, b):

return a + b

where a and b are the parameters.

**Arguments**:

Arguments are the real values passed inside the function

**Example:**

print(add(10, 20))

where 10 and 20 are the arguments.

**Function Calling:**

Executing a function to run the code.

**Example:**

def add(a, b):

    return a + b

result = add(10, 20)  #This is function calling

print(result)

**Scope:**

Determines the accessibility of variables within a program.

There are 2 types of scopes:

**1. Local:**

Created inside a function and only accessed in the same function.

**Example:**

def my\_function(local\_var):

    local\_var = local\_var

    print("Inside the function: local\_var =", local\_var)   # Accessing local variable

my\_function(20)

# print(local\_var)

**2. Global:**

Created outside a function and can be accessed in the inside of a function or outside the function. Two methods of creating a global variable.

**Method # 01:**

**Example:**

global\_var = 10  # Global variable

def my\_function(value):

    print("Inside the function: global\_var =", global\_var)  # Accessing global variable

my\_function(20)

print(global\_var)

**Method # 02:**

**Example:**

# Using global function inside the function.

def my\_function(value):

    global global\_var

    global\_var = 10  # Global variable

    print("Inside the function: global\_var =", global\_var)  # Accessing global variable

my\_function(20)

print(global\_var)