1.

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result. Functions **reduce the need for duplicate code**. This makes programs shorter, easier to read, and easier to update.

2.

**When a function is "called" the program "**leaves" the current section of code and begins to execute the first line inside the function.

3.

A function is created with the **def keyword**. The statements in the block of the function must be indented. The def keyword is followed by the function name with round brackets and a colon.

4.

Defining a function is a lot like building a house - you give it a name, you state how to use it (the house has doors etc) and you say clearly in computer code what the function does (similar to defining the layout of the house - which rooms exist, what connects to what).

Calling a function is a lot like living in a house - you are actually making use of the function, passing the data it needs, executing the code inside the function and using the values it generates - this is similar to making use of a house that is already built.

For example :

function multiple(x,y){

return x\*y;

}

Calling a function would look like

print multiply(2,3);

5.

There are two types of variables: global variables and local variables. **The scope of global variables is the entire program whereas the scope of local variable is limited to the function where it is defined**.

def func():

x = "Python"

print(x)

print(s)

s = "Tutorialspoint"

print(s)

func()

print(x).

6.

A local variable **retains its value** until the next time the function is called

A local variable becomes **undefined** after the function call completes.

The local variable can be used outside the function any time after the function call completes.

**The local variable can be used outside the function any time after the function call completes.**

**7.**

A return is **a value that a function returns to the calling script or function when it completes its task**. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.

8.

If you want to refer to a global variable in a function, you can **use the global keyword to declare which variables are global**.

9.

# Python program showing no need to

# use global keyword for accessing

# a global value

# global variable

a = 15

b = 10

# function to perform addition

def add():

    c = a + b

    print(c)

# calling a function

add()

x = 15

def change():

    # using a global keyword

    global x

    # increment value of a by 5

    x = x + 5

    print("Value of x inside a function :", x)

change()

print("Value of x outside a function :", x)