Functions in Python

Functions in Python

- Python Functions is a block of related statements designed to perform a computational task.
- Function helps to avoid of writing the same code again and again for different inputs.
- The function calls help to reuse code repeatedly.
- Functions can be both built-in or user-defined.
- It helps the program to be concise, non-repetitive, and organized.

Built-in Dictionary Methods

```
def fact(n):
    result=1
    while n>1:
        result=result*n
        n=n-1
    return result
```

Functions in Python

Creating a Function

We can create a Python function using the **def** keyword.

```
def func():
    print("Welcome to RCCIIT")
```

```
def fact(n):
        result=1
        while n>1:
                result=result*n
                n=n-1
        return result
```

Functions in Python

Calling a Function

After creating a function we can call it by using the name of the function followed by parenthesis containing parameters of that particular function.

```
def func():
    print("Welcome to RCCIIT")

# Call a function
func()
```

Functions in Python

Arguments of a Function

Arguments are the values passed inside the parenthesis of the function. A function can have any number of arguments separated by a comma.

```
def Test(x):
       if (x % 2 == 0):
            print("even")
      else:
            print("odd")


# Call the function
Test(25)
```

Functions in Python

The return statement

The function return statement is used to exit from a function and go back to the function caller and return the specified value or data item to the caller.

Syntax: return [expression_list]

The return statement can consist of a variable, an expression, or a constant which is returned to the end of the function execution.

```
def cube(y):
   return y*y*y
```

If none of the above is present with the return statement a None object is returned.

Functions in Python

Is Python Function Pass by Reference or pass by value?

- In Python every variable name is a reference.

- When we pass a variable to a function, a new reference to the object is created.

Functions in Python

Anonymous functions:

An anonymous function means that a function is without a name. The **def** keyword is used to define the normal functions and the lambda keyword is used to create anonymous functions.

```
# normal function
def cube(y):
      return y*y*y

# Lambda function
lambda_cube = lambda y: y*y*y

# Both functions generate same output.
print(cube(5))
print(lambda_cube(5))
```

Functions in Python

Global and Local Variables in Python

- Global variables are the one that is defined and declared outside a function.
- Local variables can only be reached within their scope.

```
# This function uses global variable s
def func():
     print(s)

# Global scope
s = "RCCIIT"
func()
```

Output:

```
RCCIIT
```

As there are no locals, the value from the global will be used.

Functions in Python

Global and Local Variables in Python

```
# This function has a variable with name same as s.
def func():
    s = "KOLKATA"
    print(s)

# Global scope
s = "RCCIIT"
func()
print(s)
```

Output:

```
KOLKATA
RCCIIT
```

If a variable with the same name is defined inside the scope of function as well then it will print the value given inside the function only and not the global value.

THANK YOU