Python Functions

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

Functions

- A function is a **group of related statements** that performs a **specific task.**
- That gets executed only on a function **call**.

Why Functions??

- Functions help break our program into smaller and modular chunks
- Makes the code:
  - Ordered
  - Readable
  - Reusable

Syntax of Function

```
def function_name(parameters):
    statement(s)
```

- Function definition that consists of the following components.
  - Keyword **def** that marks the start of the function header.
  - **Function name** to uniquely identify the function.
  - **Parameters** within parenthesis: through which we **pass values to a function**.
    - They are optional.
  - A colon **(:)** to mark the **end of the function** header.
  - Function **statements** with proper indentation.
  - An optional **return** statement to **return a value** from the function.

Example of a Function

```
def sayHello():
    print("Hello, Good morning")



def greet(name):
    print("Hello, " + name + ". Good morning")
```

How to call a Python Function ?

- Calling does the **execution** of a defined function.
- Can call a Python function in two ways,
  - directly from the prompt.

```
> greet('Anoop')
Hello, Anoop. Good morning
> sayHello()
Hello, Good morning
>
```

  - from another Python function.

```
def sayHello():
    print("Hello, Good morning")

def greet(name):
    print("Hello, " + name + ". Good morning")

sayHello()
greet('Anoop')
```

The return Statement

- The return statement is used to **exit a function** and go back to the place from where it was called.

Syntax

```
return (expression)
```

- This statement can contain an **expression that gets evaluated** and the **value is returned**.

```
def addingTwoNumbers(numberOne, numberTwo):
    result = numberOne+numberTwo
    return result

n1 = 12.5
n2 = 65.3
result = addingTwoNumbers(n1, n2)
print('The sum is', result)
```

Output

The sum is 77.8

Returning Multiple Values

- Using **tuple** we can return multiple values.

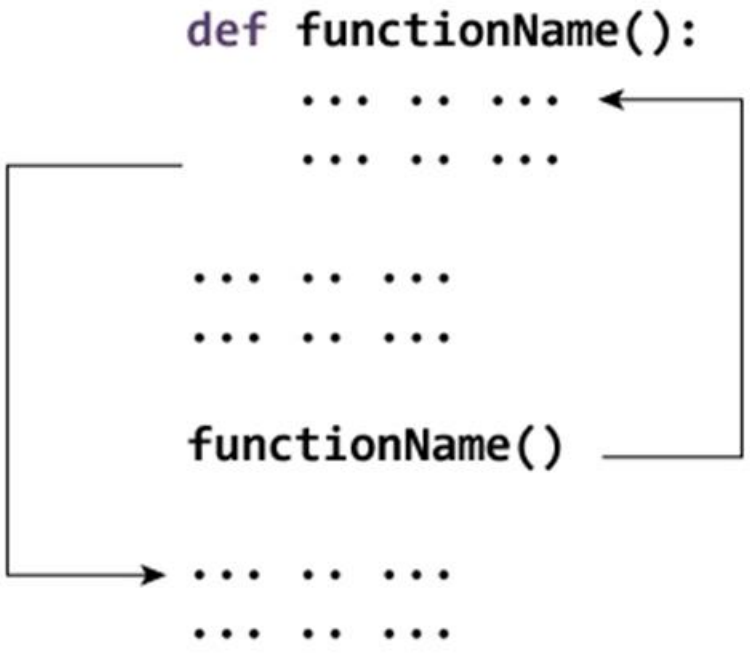
```
def calculator(x,y):
    sum = x + y
    product = x * y
    difference = x - y
    # returning multiple items as a tuple
    return(sum,product,difference)

s, p, d = calculator(7, 3) # unpacking tuple
print("Sum = {}, Product = {}, Difference = {}".format(s,p,d))
```

Output

```
Sum = 10, Product = 21, Difference = 4
```

Function: Execution Flow



Scope and Lifetime of Variables

- Parameters and variables **defined inside a function** are **not visible from outside the function**.

```
def sampleFunction():
    x = 10
    print("Value inside function:",x)

x = 20
sampleFunction()
print("Value outside function:",x)
```

Output

```
Value inside function: 10
Value outside function: 20
```

Types of Functions

- There are mainly 3 types
  - **Built-in functions.**
    - Python has several functions that are **readily available** for use.
    - These functions are called built-in functions.
    - Eg:- sum(), len(), print()
  - **User-defined functions.**
    - Functions that define by user to do certain specific task
  - **Anonymous functions**
    - À function that is defined **without a name**.