Functions

Python Functions

Python Functions is a block of statements that return the specific task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again.

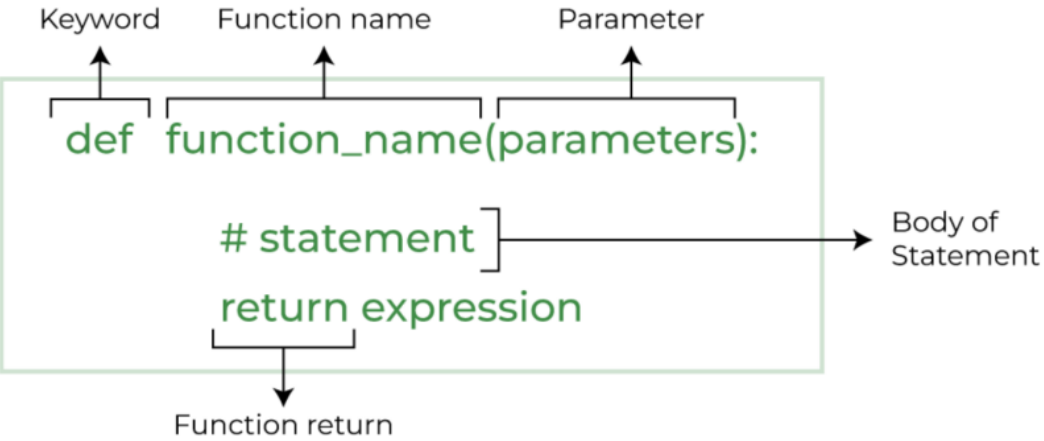
Some Benefits of Using Functions

Increase Code Readability

Increase Code Reusability

Python Function Declaration

The syntax to declare a function is:



Types of Functions in Python

- Below are the different types of functions in Python:

Built-in library function: These are Standard functions in Python that are available to use.

User-defined function: We can create our own functions based on our requirements.

Creating a Function in Python

We can define a function in Python, using the def keyword. We can add any type of functionalities and properties to it as we require. By the following example, we can understand how to write a function in Python. In this way we can create Python function definition by using def keyword.

Syntax:

```
# A simple Python function
def fun():
   print("Welcome to GFG")
```

Calling a Function in Python

After creating a function in Python we can call it by using the name of the functions Python followed by parenthesis containing parameters of that particular function. Below is the example for calling def function Python.

Example:

```
# A simple Python function
def fun():
   print("Welcome to GFG")

# Driver code to call a function
fun()
```

Python Function Arguments

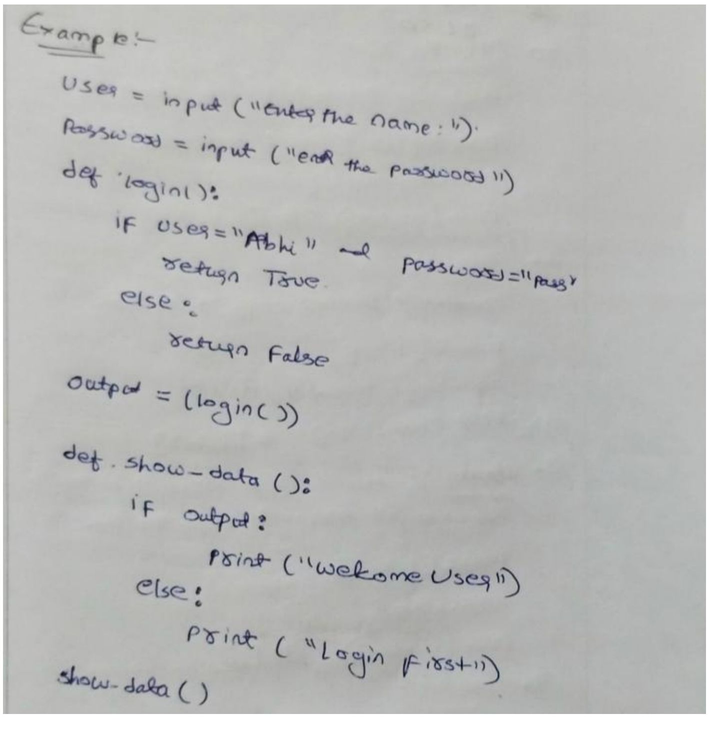
Arguments are the values passed inside the parenthesis of the function. A function can have any number of arguments separated by a comma.

In this example, we will create a simple function in Python to check whether the number passed as an argument to the function is even or odd.

Example:

```
# A simple Python function to check
# whether x is even or odd
def evenOdd(x):
    if (x % 2 == 0):
        print("even")
    else:
        print("odd")

# Driver code to call the function
evenOdd(2)
evenOdd(3)
```



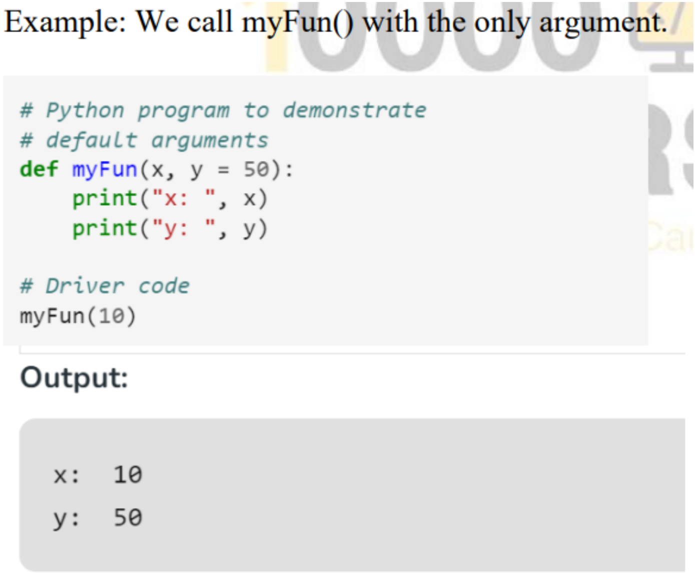
Example:-

```
User = input ("Enter the name: ").
Password = input ("enter the password ")
def login():
    if user="Abhi" and password="pass"
        return True.
    else:
        return False
Output = (login())

def. show-data ():
    if output:
        print ("welcome User")
    else:
        print ( "Login First")
show-data ()
```

Python Default arguments

A default argument is a parameter that assumes a default value if a value is not provided in the function call for that argument. The following example illustrates Default arguments.



Example: We call myFun() with the only argument.

```python
# Python program to demonstrate
# default arguments
def myFun(x, y = 50):
    print("x: ", x)
    print("y: ", y)

# Driver code
myFun(10)
```

Output:

```
x:  10
y:  50
```

Taking input in Python

input () function first takes the input from the user and converts it into a string. The type of the returned object always will be <class ‘str’>. It does not evaluate the expression it just returns the complete statement as String.

Example-1:

```
# Python program showing
# a use of input()
val = input("Enter your value: ")
print(val)
```

Example-2:

```
username = input("Enter username:")
print("Username is: " + username)
```