PYTHON FUNCTION:

- function is a group of related statements that perform a specific task.
- Functions help break our program into smaller and modular chunks.
- If the program grows larger and larger, functions make it more organized and manageable.

Syntax:

```
def function_name(parameter  list):
Statements i.e the function body

Or
def function_name(parameters):
        """docstring"""
        statement(s)
```

1. Keyword def marks the start of function header.
2. A function name to uniquely identify it. Function naming follows the same rules of writing identifiers in Python.
3. Parameters (arguments) through which we pass values to a function. They are optional.
4. A colon (:) to mark the end of function header.
5. Optional documentation string (docstring) to describe what the function does.
6. One or more valid python statements that make up the function body. Statements must have same indentation level (usually 4 spaces).
7. An optional return statement to return a value from the function.

CREATING A FUNCTION:

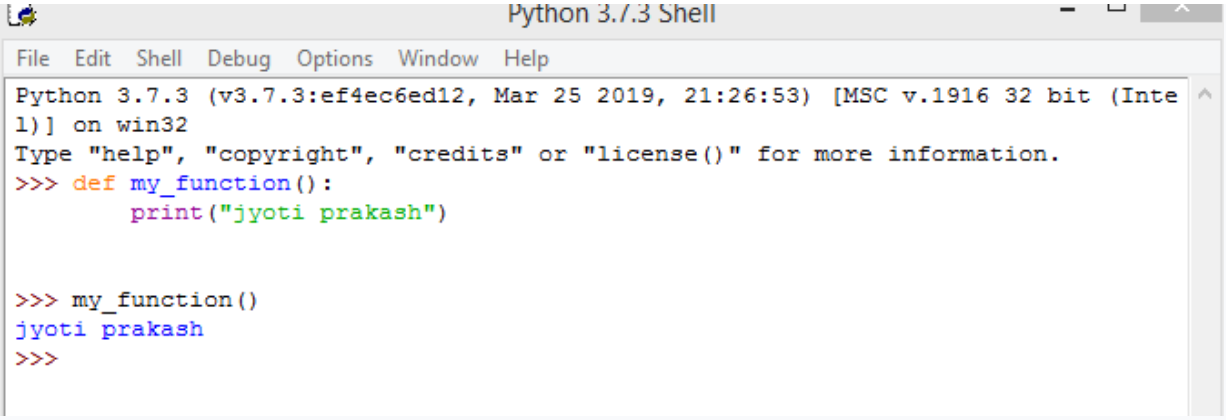
```
def  my_function( ):
        print(" hello jyoti prakash")
```

CALLING A FUNCTION

To call a function ,use the function name followed by parenthesis.

Ex:

```
def  my_function():
        Print("jyoti prakash")
my_function( )
```



Functions can be categorized into the following three types.

1. **Built-in**
2. **Module**
3. **User defined**

BUILT-IN FUNCTIONS:

Built in function are the predefined functions that are already available in python.Function provide efficiency and structure to a programming language.

TYPE CONVERSION FUNCTION

Python provides built-in functions that convert values from one type to another.which are termed as type conversion functions.

1. **int()**

   the int function takes any value and converts it into an integer.int() can convert floating point values to integers.but it doesnot round off

   int('123') o/p-123

   ` int(334.56) o/p-334

2. float()

   float converts integers and strings into floating point numbers

   float(26) o/p-26.0

   float('3.14159') o/p-3.14159

3. input()

   it enable us to accept an input string from the user without evaluating its value.it provides the most common way to gather input from the key board

   name=input("enter a name")

4. eval function

5. min and max function
6. abs function
7. type function
8. len function
9. round function

Python 3.7.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> int('123')
123
>>> int(334.56)
334
>>> float('3.14159')
3.14159
>>> name=input("enter a name")
enter a name jyoti
>>> x=eval('45+10')
>>> print(x)
55
>>> max(9,12,6,15)
15
>>> min(23,-109,5,2)
-109
>>> abs(-50)
50
>>> type(10)
<class 'int'>
>>> str='jyoti prakash')
SyntaxError: invalid syntax
>>> str="jyoti prakash"
>>> len(str)
13
>>> round(12.452)
12
>>> range(5)
range(0, 5)
>>> list(range(5))
[0, 1, 2, 3, 4]
>>>
```

RETURN STATEMENT:

- A return statement is used to end the execution of the function call and "returns" the result (value of the expression following the return keyword) to the caller.
- The statements after the return statements are not executed. If the return statement is without any expression, then the special value none is returned.

Note: Return statement can not be used outside the function.

Syntax:

```
def  fun():
      statements
      …..
      Return [expression]
```

Example1: python programme to demonstrate return function.

```
def  add(a, b):

# return sum of a nd b

      return a+b
```

Example2: area of rectangle

```
def  areaRectangle(length, breadth):

     area=length*breadth

     return area
```

Example3:WAP to add or subtract two values and to return them

```
def  add_diff( x,y):

   add=x+y

   deff=x-y

   return add,diff

   a,b=add_diff(200,180)

   print("the sum of two number is:",a)

   print("the difference of two number is ",b)
```

addsub.py - C:/Users/jps/Desktop/addsub.py (3.7.3)

```
def add_diff(x,y):
    add=x+y
    diff=x-y
    return add,diff
a,b=add_diff(200,180)
print("the sum of two numbers is:",a)
print("the diffference of two numbers is ",b)
```

Python 3.7.3 Shell

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Users/jps/Desktop/addsub.py ==================
the sum of two numbers is: 380
the diffference of two numbers is  20
>>>
```

Example4:**WAP to findout sum of two variables**

```
def sum (a,b):
    return a+b;

#taking values from the user
a = int(input("Enter a: "))
b = int(input("Enter b: "))

#printing the sum of a and b
print("Sum = ",sum(a,b))
```

addsub1.py - C:/Users/jps/Desktop/addsub1.py (3.7.3)

```
def add(a,b):
    s=a+b
    return s;
a=int(input("enter a"))
b=int(input("enter b"))
result =add(a,b)
print("sum=",result)
```

Python 3.7.3 Shell

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Users/jps/Desktop/addsub1.py ==================
enter a 45
enter b67
sum= 112
>>>
```

addsub2.py - C:/Users/jps/Desktop/addsub2.py (3.7.3)

File Edit Format Run Options Window Help

```
def sum(a,b):
    return a+b;
a=int(input("enter a"))
b=int(input("enter b"))
print("sum=",sum(a,b))
```

Python 3.7.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Users/jps/Desktop/addsub2.py ==================
enter a34
enter b89
sum= 123
>>>
```