Functions

- A function is a set of statements that take inputs, do some specific computation and produces output.
- Functions are reusable.
- Python provides built-in functions like int(), float(), input(), print() etc. but we can also create your own functions. These functions are called user-defined functions.

Parameters

- A parameter is a variable used to define a particular value during a function definition.

Arguments

- An argument is a value passed to a function at the time function calling.

Defining a Function

- In python a function is defined using the 'def' keyword.

```
Syntax: def functionName(define parameters):
            statement1
            statement2
            statement3
            .
            .
            statementn

        functionName(pass arguments)
```

In [13]:

```
def hello(y): # y=5
    print("Hello")
    print(y+2)

hello(5)
```

```
Hello
7
```

In [17]:

```
def evenOrOdd(n): #n=12
    if(n%2 == 0):
        print(n,"is even")
    else:
        print(n,"is not even")

a = int(input("Enter a number: ")) #a=12
evenOrOdd(a) # evenOrOdd(12)
```

```
Enter a number: 15
15 is not even
```

In [18]:

```
n = int(input())
if(n%2 == 0):
    print(n,"is even")
else:
    print(n,"is not even")
```

```
12
12 is even
```

In [1]:

```
s = "SRIT"
print(type(s))
```

```
<class 'str'>
```

In [2]:

```
s = input("Enter your name: ")
print(s)
print(type(s))
```

```
Enter your name: srit
srit
<class 'str'>
```

In [3]:

```
s = int(input())
print(s)
print(type(s))
```

```
12345
12345
<class 'int'>
```

In [5]:

```
f = float(input())
print(f)
print(type(f))
```

```
12.5
12.5
<class 'float'>
```

In [11]:

```
a,b = 12, 14
if(a>b):
    print(a,"is less than",b)
else:
    print(a,"is not greater than",b)
```

```
12 is not greater than 14
```

Types of functions in python

1. Without arguments & without return values
2. Without arguments & with return value
3. With arguments & without return value
4. with arguments & with return value

In [19]:

```
# 1.Without arguments & without return values

def add():
    a,b = 5,3
    print(a+b)

add()
```

```
8
```

In [22]:

```
# 2.Without arguments & with return value

def mul():
    a,b = 3,6
    mul = a*b
    return mul

print(mul())
```

```
18
```

In [30]:

```
def hi():
    print(5)
    #return 13
def hello():
    return 7

x = hi()
y = hello()
print(x)
print(y)
```

```
5
None
7
```

In [31]:

```
# 3.With arguments & without return value

def mod(a,b): #a=n1, b=n2 (parameters)
    mod = a%b
    print(mod)

n1 = int(input())
n2 = int(input())
mod(n1,n2) # arguments
```

```
3
7
3
```

In [32]:

```
def mod(a,b): #a=n2, b=n1
    mod = a%b
    print(mod)

n1 = int(input())
n2 = int(input())
mod(n2,n1)
```

```
3
7
1
```

In [34]:

```
# 4.with argument & with return value

def sum(a,b,c): #a=b, b=a, c=n3
    s = a+b+c
    return s

a = int(input())
b = int(input())
n3 = int(input())
print(sum(b,a,n3))
```

```
3
5
8
16
```

In [ ]:

```

```