

FUNCTIONS

set of statements which are used to perform specific operations on variables is called as function.

Functions are two types

1. predefine or inbuilt function {max(), min(), input(), print(), len(), ..ect}
2. User define function

SYNTAX

```
def funname():  
    print("my name is umesh")  
funname()
```

FUNCTION DECLARATION

```
def funname():  
    print("my name is umesh")
```

FUNCTION CALLING

funname()

```
In [27]: def show():  
        print("Hello")  
        print("This is Umesh")  
        print("Welcome to my world")  
  
        # function declaration done but we didn't call function so it will not show  
        # the output.
```

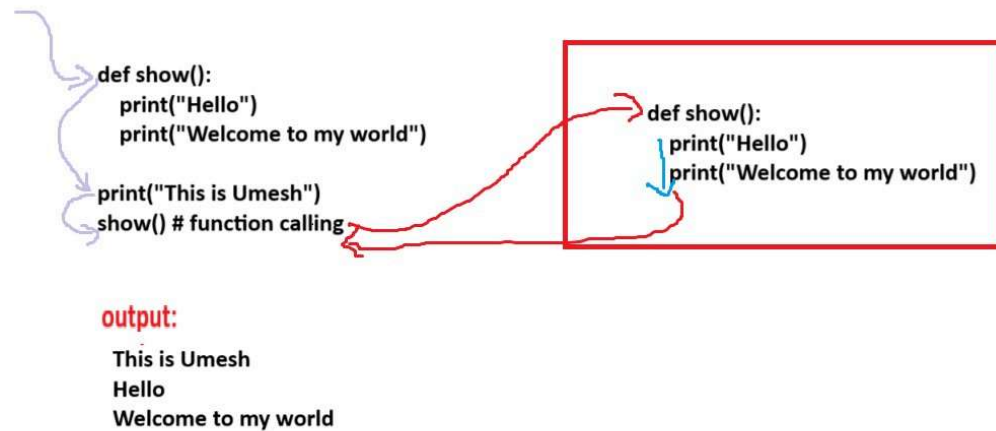
```
In [28]: def show():  
        print("Hello")  
        print("This is Umesh")  
        print("Welcome to my world")  
        show() # function calling
```

```
Hello  
This is Umesh  
Welcome to my world
```

```
In [29]: def show():  
        print("Hello")  
        print("Welcome to my world")  
  
        print("This is Umesh")  
        show() # function calling  
  
        # In python it will run in line by line
```

This is Umesh
Hello
Welcome to my world

HERE WE GO HOW THE ACTUAL FUNCTION CODE WILL RUN :)



Different types of Arguments/ Parameters:-

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable length Arguments
5. Keyword-Length Arguments

1. Positional Arguments:

THE ORDER HOW YOU PASS VALUES ACCORDING TO THAT IT WILL ALLOCATED TO PARAMETERS.

```
In [30]: def myself(name,clname,per):
          print("My name is ",name)
          print("I completed my graduation in ",clname)
          print("With percentage of",per)
          myself("UMESH",'SCSVMV UNIVERSITY',93)
          print()
          myself("UMESH",93,'SCSVMV UNIVERSITY')
          # IF WE CHANGE ARGUMENTS POSITIONS IT WILL TAKE LINE WISE.
```

My name is UMESH
I completed my graduation in SCSVMV UNIVERSITY
With percentage of 93

My name is UMESH
I completed my graduation in 93
With percentage of SCSVMV UNIVERSITY

2. Keyword Arguments:

AT THE TIME OF CALLING FUNCTION WE NEED TO ADD PARAMETERS NAME.

```
In [31]: def myself(name,clname,per):
          print("My name is ",name)
          print("I completed my graduation in ",clname)
          print("With percentage of",per)
          myself(name="UMESH",clname='SCSVMV UNIVERSITY',per=93)
          print()
          myself(per=93,name="UMESH",clname='SCSVMV UNIVERSITY')
```

My name is UMESH
I completed my graduation in SCSVMV UNIVERSITY
With percentage of 93

My name is UMESH
I completed my graduation in SCSVMV UNIVERSITY
With percentage of 93

3. Default Arguments:

IF YOU ASSIGN ANY VALUES TO PARAMETERS WHILE DECLARATION IT IS CALLED AS DEFAULT ARGUMENTS.

```
In [32]: def myself(name="UMESH",clname='SCSVMV UNIVERSITY',per=93):
          print("My name is ",name)
          print("I completed my graduation in ",clname)
          print("With percentage of",per)
          myself()
          print()
          myself(per=100,name='KRISH')
          # IF YOU PASS ANY VALUE THEN IT ILL TAKE THOSE VALUES OTHER WISE IT WILL
          # PERFORM OPERATIONS WITH DEFAULT VALUES.
```

My name is UMESH
I completed my graduation in SCSVMV UNIVERSITY
With percentage of 93

My name is KRISH
I completed my graduation in SCSVMV UNIVERSITY
With percentage of 100

4. Variable length Arguments:

IF YOU WANT TO TAKE MULTIPLE VALUES FROM USER FOR PARAMETERS USE VARIABLE LENGTH ARGUMENTS.

```
In [33]: def num(*p):  
          print(p)  
          num(100,40,3)
```

(100, 40, 3)

5. Keyword-Length Arguments:

IF YOU WANT PASS VALUE ALONG WITH KEY THEN USE KEYWORD-LENGTH ARGUMENTS.

```
In [34]: def call(**u):  
          print(u)  
          call(name="UMESH",clgname='SCSVMV UNIVERSITY',per=93)
```

{'name': 'UMESH', 'clgname': 'SCSVMV UNIVERSITY', 'per': 93}