In [1]:

```
#functions
#block of code  => reuse
```

In [ ]:

```
"""
def func():
    statement
"""
```

In [3]:

```
def msg():
    print("hello")
```

In [5]:

```
msg()
```

```
hello
```

In [7]:

```
msg()
msg()
msg()
```

```
hello
hello
hello
```

In [8]:

```
# error will occur in this because we did define the value in fun(x)
msg("tushar")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 msg("tushar")

TypeError: msg() takes 0 positional arguments but 1 was given
```

In [16]:

```
# x is parameter
def msg(x):
    print("hell0",x)
```

In [19]:

```
# "tushar"  / aman  => argument
msg("tushar")
msg(1000)
# python is a dynamic programming
```

```
hell0 tushar
hell0 1000
```

In [29]:

```
def ya(x,y):
    print("x and y",x)

ya(20,20)
```

```
x and y 20
```

In [23]:

```
def msg(x,y):
    print("x and y",x,y)


msg(10,20)
msg(20,50)
```

```
x and y 10 20
x and y 20 50
```

In [31]:

```
def msg(x,y):
    print("x and y",x,y)


print(msg(10,20))
```

```
x and y 10 20
None
```

In [33]:

```
def msg(x,y):
    print("x and y",x,y)

out=msg(10,20)
print(out)
```

```
x and y 10 20
None
```

In [35]:

```
def msg(x,y):
    print("x and y",x,y)
    return x+y
out=msg(10,20)
print(out)
```

```
x and y 10 20
30
```

In [36]:

```
def msg(x,y):
    print("x and y",x,y)
    return x+y

out=msg(10,20)
print("out value is ",out)
```

```
x and y 10 20
out value is  30
```

In [38]:

```
def msg(x,y):
    print("x and y",x,y)
    return x+y
    print("###########")

out=msg(10,20)
print("out value is ",out)
```

```
x and y 10 20
out value is  30
```

In [40]:

```
# return ke niche kch bhe likh do vo nhi run hoga
def msg(x,y):
    print("x and y",x,y)
    return x+y
    return 0

out=msg(10,20)
print("out value is ",out)
```

```
x and y 10 20
out value is  30
```

In [47]:

```
def fun(x):
    x+5


x=10
print("befor func : ",x)
fun(x)
print("after func : ",x)
```

```
befor func :  10
after func :  10
```

In [51]:

```
def func(x):
    print(x,id(x))

mylist=[10,20]
print("Mylist",mylist,id(mylist))
func(mylist)
```

```
Mylist [10, 20] 2207195330688
[10, 20] 2207195330688
```

In [52]:

```
# taking value from list in function
def func(x):
    print(x,id(x) , x[0])

mylist=[10,20]
print("Mylist",mylist,id(mylist))
func(mylist)
```

```
Mylist [10, 20] 2207194845440
[10, 20] 2207194845440 10
```

In [56]:

```
# updating the value of list in fun and printing the address
def func(x):
     x[0] = "aman"

mylist=[10,20]
print("Mylist",mylist,id(mylist))
func(mylist)
print(mylist,id(mylist))
```

```
Mylist [10, 20] 2207195484928
['aman', 20] 2207195484928
```

In [57]:

```
# updating the value of list in fun
def func(x):
     x[0] = "aman"

mylist=[10,20]
func(mylist)
print(mylist)
```

```
['aman', 20]
```

In [64]:

```
def put(x,y,z):
    if(x > y and x > z):
        print("excellent")
    elif(y > z and y > x):
        print("good")
    elif(z > x and z > y):
        print("he")
    else:
        print("lol")

x=int(input("enter the 1st value:"))
y=int(input("enter the 2nd value:"))
z=int(input("enter the 3rd value:"))
put(x,y,z)
```

```
enter the 1st value:74
enter the 2nd value:54
enter the 3rd value:61
excellent
```

In [63]:

```
#or
def put(x,y,z):
    if(x > y and x > z):
        print("excellent")
    elif(y > z and y > x):
        print("good")
    else:
        print("lol")

x=int(input("enter the 1st value:"))
y=int(input("enter the 2nd value:"))
z=int(input("enter the 3rd value:"))
put(x,y,z)
```

```
enter the 1st value:41
enter the 2nd value:21
enter the 3rd value:10
excellent
```

In [73]:

```
# lcm
def lcm(x,y):
    if(x>y):
        greatest=x
    else:
        greatest=y

    while(True):
        if(greatest%x==0 and greatest%y==0):
            print("lcm is :",greatest)
            return
        else:
            greatest+=1

lcm(10,9)
```

```
lcm is : 90
```

In [74]:

```
#types of arguments
```

In [83]:

```
#required argument
# all value should be define in argument
# employee() missing 3
def
employee(eid,ename,email):
    print(f" Eid is :{eid} \n Ename is : {ename} \n Email is: {email}")

employee()
```

```
  Cell In[83], line 4
    def
       ^
SyntaxError: invalid syntax
```

In [84]:

```python
def employee(eid,ename,email):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee(1,"yash","yashsankhla@gmail.com")
```

```
Eid is :1
Ename is : yash
Email is: yashsankhla@gmail.com
```

In [85]:

```python
# positional argument

def employee(eid,ename,email):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee("yash","yashsankhla@gmail.com",90)
```

```
Eid is :yash
Ename is : yashsankhla@gmail.com
Email is: 90
```

In [86]:

```python
# keyword argument

def employee(eid,ename,email):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee(ename="yash",email="yashsankhla@gmail.com",eid=90)
```

```
Eid is :90
Ename is : yash
Email is: yashsankhla@gmail.com
```

In [87]:

```python
# keyword argument
# id is not is func() so it will give error

def employee(eid,ename,email):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee(ename="yash",email="yashsankhla@gmail.com",id=90)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[87], line 6
      3 def employee(eid,ename,email):
      4     print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")
----> 6 employee(ename="yash",email="yashsankhla@gmail.com",id=90)

TypeError: employee() got an unexpected keyword argument 'id'
```

In [88]:

```python
# default parameter
# required value

def employee(eid,ename,email):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee(90,"naina")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[88], line 6
      3 def employee(eid,ename,email):
      4     print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")
----> 6 employee(90,"naina")

TypeError: employee() missing 1 required positional argument: 'email'
```

In [89]:

```python
# default parameter

def employee(eid,ename,email="regex@gmail.com"):
    print(f"Eid is :{eid} \nEname is : {ename} \nEmail is: {email}")

employee(90,"naina")
employee(180,"ashwani","ash@gmail.com")
```

```
Eid is :90
Ename is : naina
Email is: regex@gmail.com
Eid is :180
Ename is : ashwani
Email is: ash@gmail.com
```

In [96]:

```python
# variable length argument
#index value define krna hard he thoda

def facebook( *data ):
    print(data, type(data))

facebook("tushargoyal02","regex",100)
facebook("tushargoyal02","XYZ")
facebook("Amazon","Nidhi")
```

```
('tushargoyal02', 'regex', 100) <class 'tuple'>
('tushargoyal02', 'XYZ') <class 'tuple'>
('Amazon', 'Nidhi') <class 'tuple'>
```

In [99]:

```
# keyword variable length argument
# args , kwargs
def facebook( **data ):
    print(data, type(data))

facebook(username="tushargoyal02",salary=100)
facebook(salary=150,username="nidhi2")
```

```
{'username': 'tushargoyal02', 'salary': 100} <class 'dict'>
{'salary': 150, 'username': 'nidhi2'} <class 'dict'>
```

In [100]:

```
# for confi..
# keyword variable length argument
# args , kwargs
def facebook( **kwargs ):
    print(kwargs)

facebook(username="tushargoyal02",salary=100)
facebook(salary=150,username="nidhi2")
```

```
{'username': 'tushargoyal02', 'salary': 100}
{'salary': 150, 'username': 'nidhi2'}
```

In [101]:

```
def func():
    print("hello")

x=func
print(func)
print(x)
x()
```

```
<function func at 0x00000201E8177A60>
<function func at 0x00000201E8177A60>
hello
```

In [102]:

```
#high order function
# which take another function as a argument
```

In [106]:

```
def square(x):
    print(x*x)

print(square(10))
```

```
100
None
```

In [108]:

```
def square(x):
    print(x*x)

def addno(x,func):
    print(x,func)

addno(10,20)
```

```
10 20
```

In [109]:

```
def square(x):
    print(x*x)

def addno(x,func):
    print(x,func)

addno(10,square(5))
```

```
25
10 None
```

In [111]:

```
def square(x):
    return x*x

def addno(x,func):
    print(x,func)

addno(10,square(5))
```

```
10 25
```

In [112]:

```
def square(x):
    return x*x

def addno(x,func):
    print(x,func)

addno(10, square )
```

```
10 <function square at 0x00000201E7E7C540>
```

In [115]:

```
# error
def square(x):
    return x*x

def addno(x,func):
    print(x,func())

addno(10, square )
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[115], line 8
      5 def addno(x,func):
      6     print(x,func())
----> 8 addno(10, square )

Cell In[115], line 6, in addno(x, func)
      5 def addno(x,func):
----> 6     print(x,func())

TypeError: square() missing 1 required positional argument: 'x'
```

In [114]:

```
# high order func()
def square(x):
    return x*x

def addno(x,func):
    print(x,func(3))

addno(10, square )
```

```
10 9
```

In [ ]:

In [ ]: