

Data Science & AI & NIC - Param

Python-For Data Science
Functions

Lecture No.- 01

By- Pankaj Sharma Sir



Recap of Previous Lecture



Topic

Dictionary



Topics to be Covered



Topic

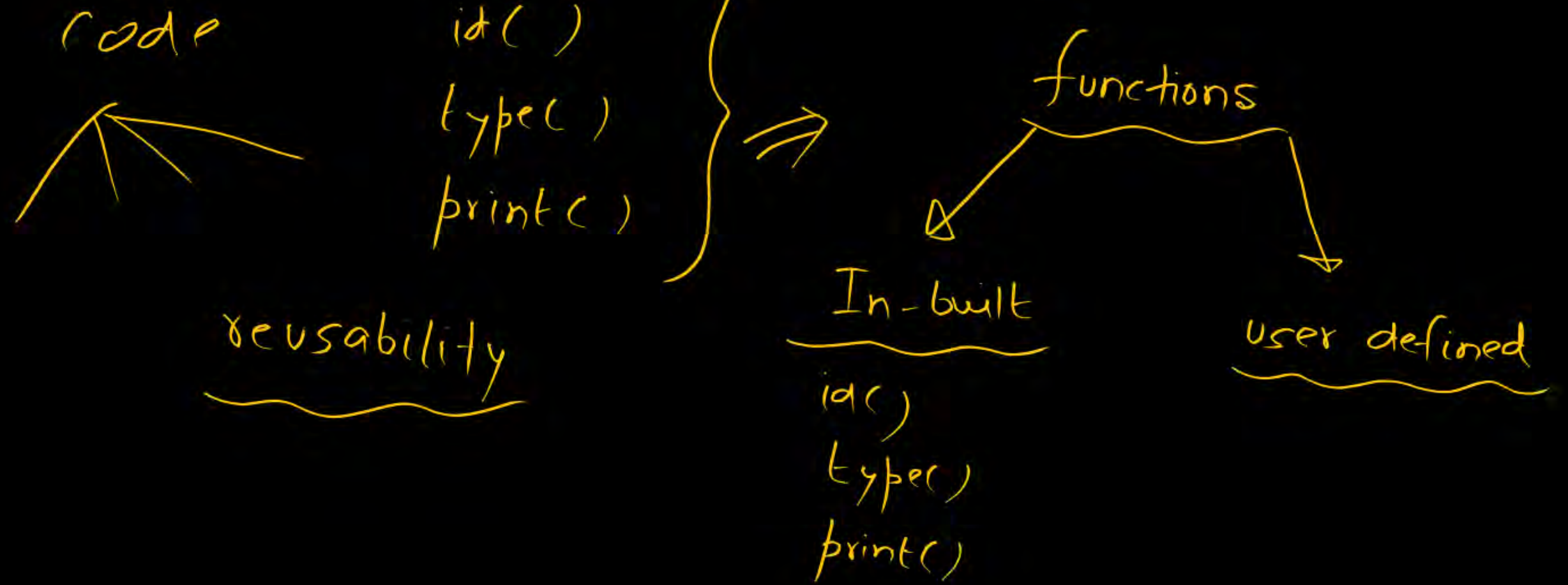
Functions Part-01





Topic : Functions





$$S = \frac{1!}{2!} + \frac{2!}{3!} + \frac{3!}{4!} + \frac{4!}{5!} + \frac{5!}{6!}$$

1, 2, 3, 4, 5

Sum = 0.0

for i in range(1, 6):

Prod = 1

for j in range(1, i+1):

Prod = Prod * j

Prod1 = 1

for k in range(1, i+2):

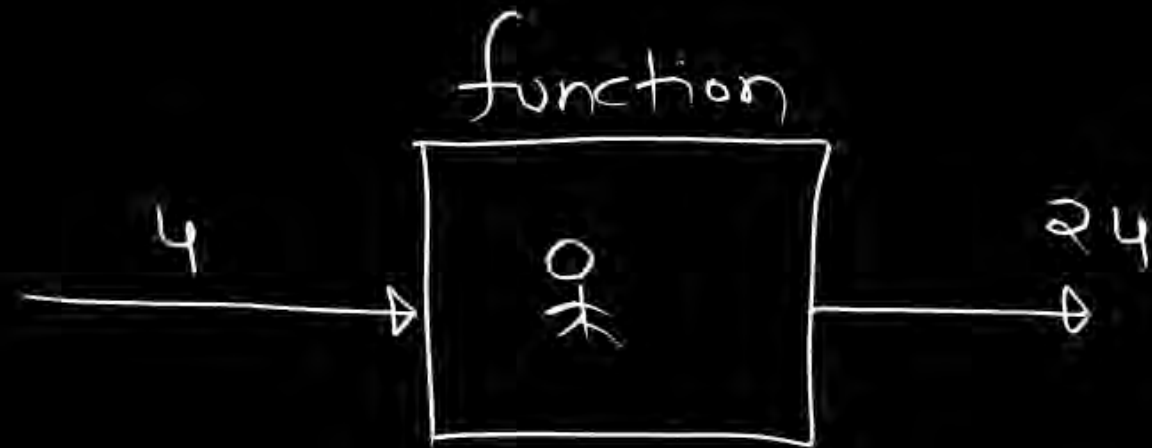
Prod1 = Prod1 * k

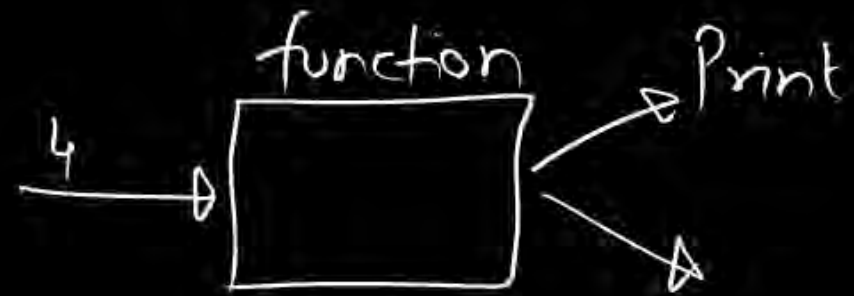
Sum = Sum + Prod/Prod1

i = 1, 2, 3, 4, 5

defined \Rightarrow def

def factorial





```
n = int(input())  
prod = 1  
for i in range(1, n+1):  
    prod = prod * i  
print(prod)
```


function
definition

```
def factorial(n):
```

```
    Prod = 1
```

```
    for i in range(1, n+1):
```

```
        Prod = Prod * i
```

```
    return Prod
```

```
x = factorial(4)
```

```
print(x)
```

```
def display(string):  
    print(string)
```

Not returning anything

```
display("Pankaj")
```

```
display(1)
```

```
x = display(12.4)
```

```
def function_name(parameters):  
    """  
        doc string  
        """
```

```
    _____  
    |  
    |  
    |  
    |  
    |
```

```
    return _____
```

Optional

if no return \Rightarrow None

$x = \text{factorial}()$

def factorial(n):

~~(1, 9)~~



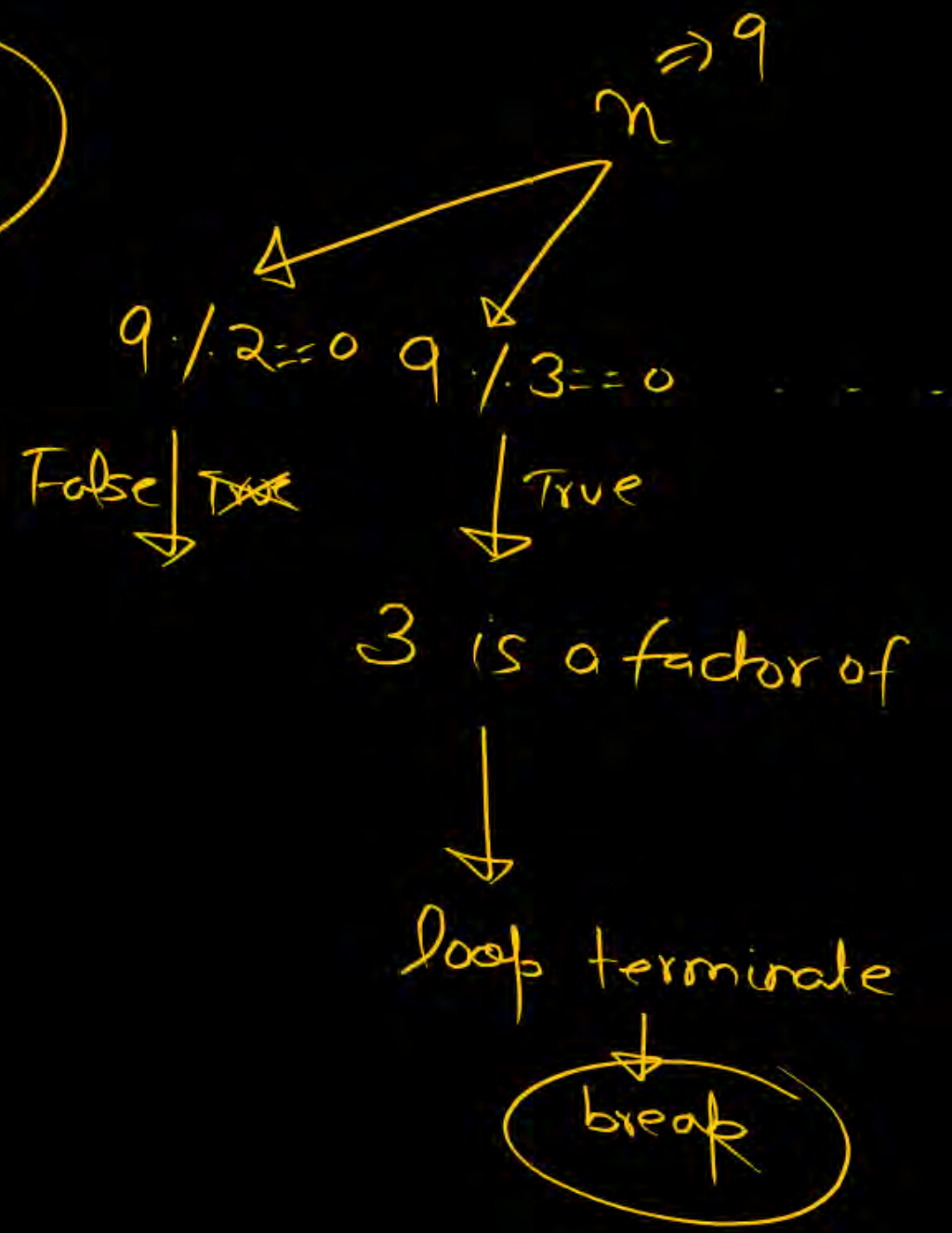
Jupyter Notebook

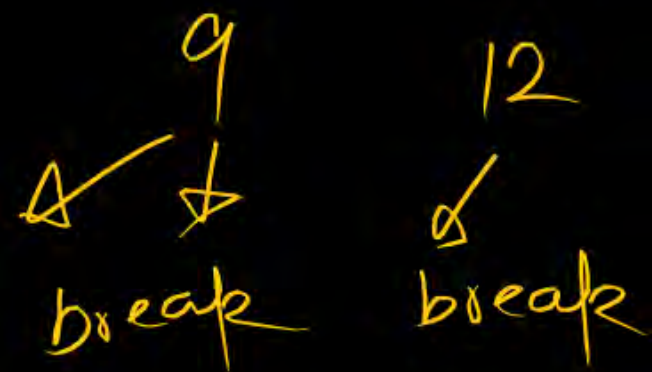
↓
print x

x = 10

x ×
out[] 10

factor other
than 1 and n





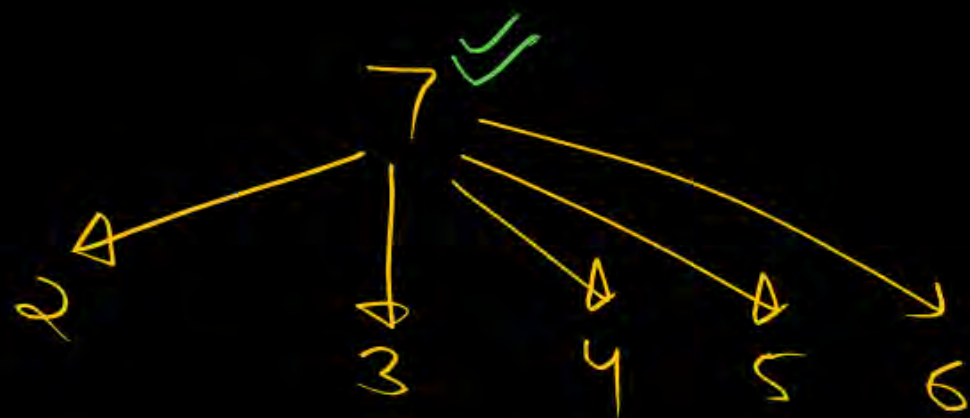
def IsPrime(n) :

n == 1

for i in range(2, n) :

if n % i == 0 :

break




else :

return True

return False

$n=10$ ^{False}
`IsPrime(10)`
`if (False)`

`for i in range(10, 101):`
`if (IsPrime(i)):`


`def IsPrime(n):`

$n==1$

`if n == 1:`
`return False`

`for i in range(2, n):`

`if n % i == 0:`

`break`

`else:`

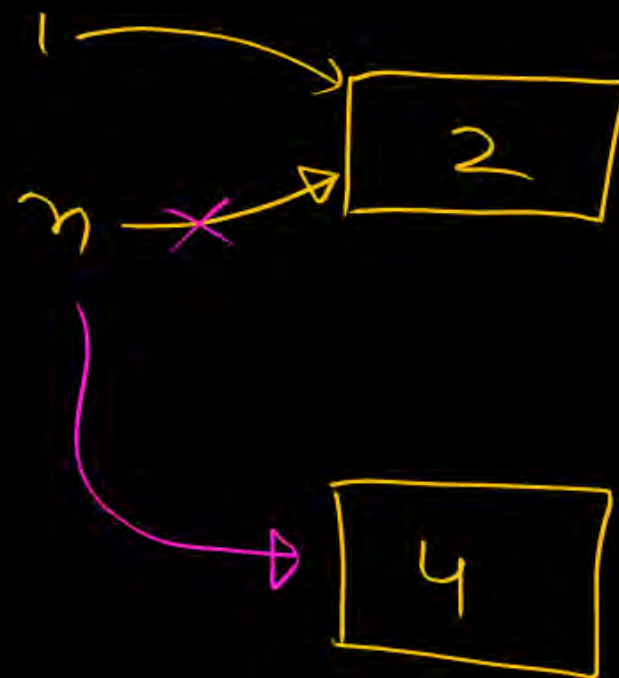
`return True`

`return False`

```
def Twice(n):
```

$n = 2 * n$

```
    i = 2  
    Twice(i)  
    Print(i)
```




```
def Twice(l):
```

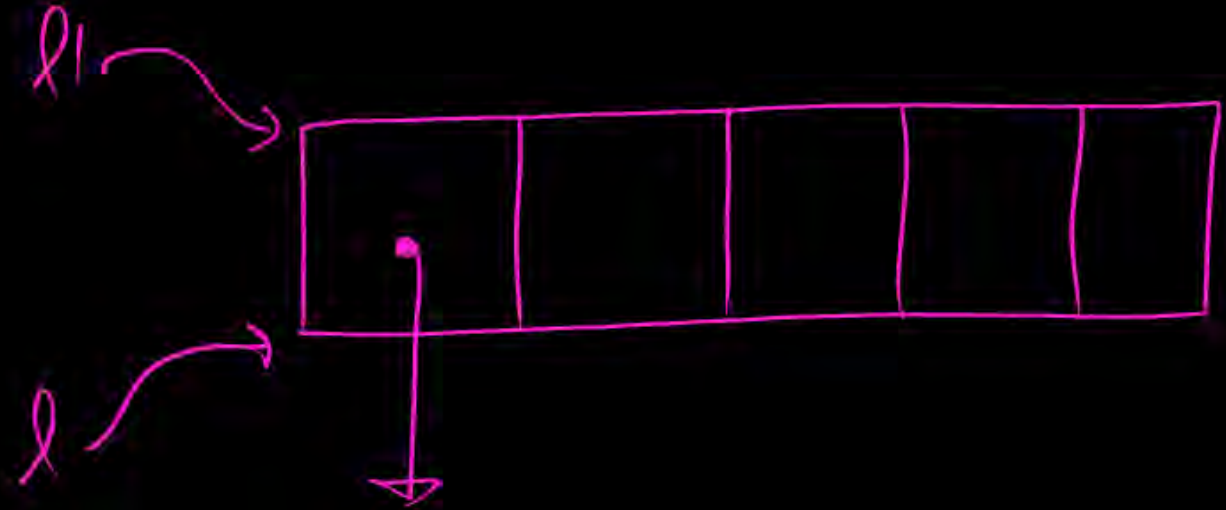
```
    for i in range(len(l)):
```

```
        l[i] = l[i] * 2
```

```
l = [1, 2, 3, 4, 5]
```

```
Twice(l)
```

```
print(l)
```



local/global

outside

functions

```
def f()
```

```
    a = 10
```

```
    a = a + 2
```

```
    print(a)
```

```
f()
```

```
print(a)
```

→ error

```
a = 10 # global
```

```
def f():
```

```
    ==  
    ==  
    ==
```

```
f()
```

```
✓ print(a)
```

local/global

outside

functions

```
def f()  
    a = 10  
    a = a + 2  
    print(a)  
  
f()  
print(a)  → error
```

```
def f():  
    _____  
    _____  
    _____  
    a = 10 # global ✓  
    f() ✓  
    print(a)
```


local/global

outside

functions

```
def f()  
    a = 10  
    a = a + 2  
    print(a)  
  
f()  
print(a)
```

→ error

```
def f():  
    print(a)
```

≡≡≡

#global

✓ f()
print(a)

a = 10

1580 → ①

$a = 10$

def f():

$a = 90$

→ 90

print(a)



③ n : 3

*
* *
* * *

n : 7

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *

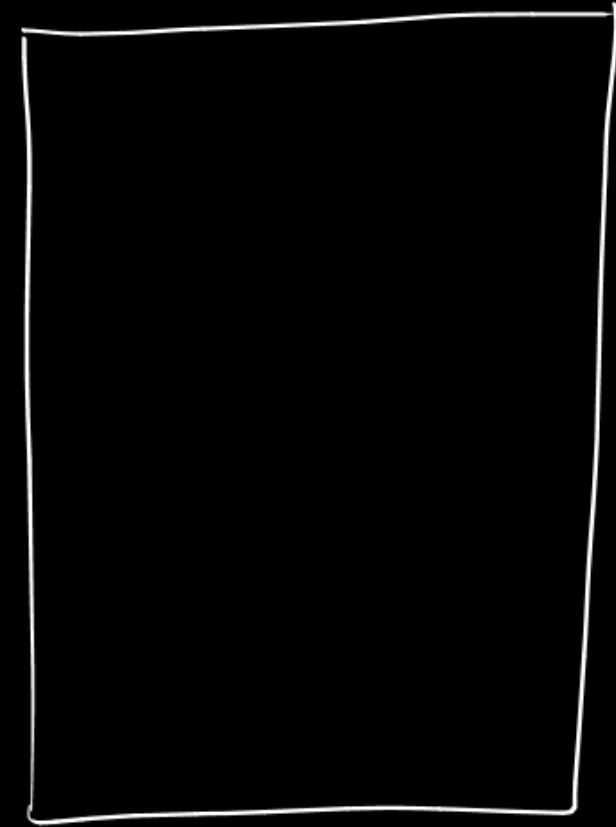
① function to calculate

$$\frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} \dots n \text{ terms}$$

I/P : n

O/P : value

② $\sin x$
 $\cos x$



✓

Review }
complaint }

Day 16 functions in python

```
In [2]: def factorial(n):  
        prod=1  
        for i in range(1,n+1):  
            prod=prod*i  
        return prod
```

```
In [3]: x=factorial(4)  
print(x)
```

24

```
In [4]: print(factorial(6))
```

720

```
In [5]: for n in range(1,11):  
        print(factorial(n))
```

1

2

6

24

120

720

5040

40320

362880

3628800

```
In [6]: def display(s):  
        print(s)  
        display("pankaj")  
        display(1)
```

pankaj

1

```
In [7]: display(12.34)
```

12.34

```
In [8]: x=display(10)
```

10

```
In [9]: print(x)
```

None

```
In [10]: x=factorial()
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[10], line 1  
----> 1 x=factorial()  
  
TypeError: factorial() missing 1 required positional argument: 'n'
```

```
In [11]: factorial(4) #we are calling a function/using a function
```

```
Out[11]: 24
```

```
In [12]: factorial(10)#calling
```

```
Out[12]: 3628800
```

```
In [13]: #while calling a function ,if the function contains parameters and we are not providing  
#those parameters ==>ERROR
```

```
In [14]: def mul(a,b):  
         x=a*b  
         return x
```

```
In [15]: mul(10,20)
```

```
Out[15]: 200
```

```
In [16]: mul(10)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[16], line 1  
----> 1 mul(10)  
  
TypeError: mul() missing 1 required positional argument: 'b'
```

```
In [17]: mul()
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[17], line 1  
----> 1 mul()  
  
TypeError: mul() missing 2 required positional arguments: 'a' and 'b'
```

```
In [18]: def Isprime(n):  
         for i in range(2,n):  
             if n%i==0:  
                 break  
         else :  
             return True  
         return False
```

```
In [19]: Isprime(10)
```

```
Out[19]: False
```

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
```

```
In [20]: Isprime(9)
```

```
Out[20]: False
```

```
In [21]: Isprime(11)
```

```
Out[21]: True
```

```
In [22]: Isprime(2)
```

```
Out[22]: True
```

```
In [24]: for i in range(10,101):  
         if Isprime(i):  
             print(i,"is Prime")  
         #printing all prime numbers from 10 to 100
```

```
11 is Prime  
13 is Prime  
17 is Prime  
19 is Prime  
23 is Prime  
29 is Prime  
31 is Prime  
37 is Prime  
41 is Prime  
43 is Prime  
47 is Prime  
53 is Prime  
59 is Prime  
61 is Prime  
67 is Prime  
71 is Prime  
73 is Prime  
79 is Prime  
83 is Prime  
89 is Prime  
97 is Prime
```

```
In [25]: def twice(n):  
         n=2*n  
         i=2  
         twice(i)  
         print(i)
```

```
2
```

```
In [27]: def twice(l):  
         for i in range(len(l)):  
             l[i]=2*l[i]  
         l1=[1,2,3,4,5]  
         twice(l1)  
         print(l1)
```

```
[2, 4, 6, 8, 10]
```

```
In [28]: def f():
```

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
```

```
    a=a+2
```

```
print(a)
f()
print(a)
```

14

```
-----
NameError                                Traceback (most recent call last)
Cell In[28], line 6
      4 print(a)
      5 f()
----> 6 print(a)

NameError: name 'a' is not defined
```

```
In [30]: a=11111 #global
def f():
    a1=10
    print(a1)
f()
print(a) #global wala
```

10
11111

```
In [33]: x=10 #global
def f():
    y=200
    print(y)
    print(x)

f()
x=234
print(x)
```

200
10
234

```
In [34]: def f():
    x=10
    print(x)
    print(y)
y=200 #global
print(y)
f()
```

200
10
200

```
In [35]: y=200 #global
def f():
    x=10
    print(x)
    print(y)

print(y)
f() #call ==> before call anywhere outside all the function y=200
```


200
10
200

```
In [37]: def f():
          x=10
          print(x)
          print(z)

          print(z)
          f() #call ==>before call anywhere outside all the function y=200
          z=200 #global
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[37], line 6
      3     print(x)
      4     print(z)
----> 6 print(z)
      7 f() #call ==>before call anywhere outside all the function y=200
      8 z=200

NameError: name 'z' is not defined
```

In [40]:

In [43]: f()

90
140713888617544

Out[43]:

In [44]: id(a)

Out[44]: 140713888617544

```
In [45]: a=10
          def f():
              a=90
              print(id(a))
```

In [46]: f()

140713888620104

In [47]: id(a)

Out[47]: 140713888617544

In []:

THANK - YOU