In [2]:
```
print("this is my prog")
```

```
this is my prog
```

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

In [4]:
```
len(l)
```

Out[4]:
```
5
```

In [5]:
```
def test():
    pass
```

In [6]:
```
def test1():
    print("This is my first function program ")
```

In [7]:
```
test1()
```

```
This is my first function program
```

In [8]:
```
test1()+"sneha"
```

```
This is my first function program
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1936\942559475.py in <module>
----> 1 test1()+"sneha"

TypeError: unsupported operand type(s) for +: 'NoneType' and 'str'
```

In [9]:
```
def test2():
    return "This is my first function program"
```

In [11]:
```
test2()+" Sneha"
```

Out[11]:
```
'This is my first function program Sneha'
```

In [12]:
```
def test3():
    return 1,4.5,"sneha"
```

In [14]:
```
test3()
```

Out[14]:
```
(1, 4.5, 'sneha')
```

In [15]:
```
a = 1,2,3,4,5,6
```

In [16]:
```
a
```

Out[16]:
```
(1, 2, 3, 4, 5, 6)
```

In [45]:
```
a,b,c,d = 1,3.4,"sneha",True
```

In [46]:
```
a
```

Out[46]:
```
1
```

In [47]:
```
b
```

Out[47]:
```
3.4
```

In [48]:
```
c
```

Out[48]:
```
'sneha'
```

In [49]:
```
d
```

Out[49]:
```
True
```

In [50]:
```
test3()[0]
```

Out[50]:
```
1
```

In [51]:
```
test3()[1]
```

Out[51]:
```
4.5
```

In [52]:
```
test3()[2]
```

Out[52]:
```
'sneha'
```

In [55]:
```
test3()[3]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1936\2551648335.py in <module>
----> 1 test3()[3]

IndexError: tuple index out of range
```

In [56]:
```
a,b,c,d = test3()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1936\631285073.py in <module>
----> 1 a,b,c,d = test3()

ValueError: not enough values to unpack (expected 4, got 3)
```

In [57]:
```
def test4():
    a = 3*4+5
    return a
```

In [58]:
```
type(test4())
```

Out[58]:
```
int
```

In [59]:
```
def test5(a,b):
    c = a+b
    return c
```

In [61]:
```
test5(1,3)
```

Out[61]:
```
4
```

In [64]:
```
test5("Sneha ","Singh")
```

Out[64]:
```
'Sneha Singh'
```

In [65]:
```
test5([1,2,3,4,5],[9,8,7,6,54])
```

Out[65]:
```
[1, 2, 3, 4, 5, 9, 8, 7, 6, 54]
```

In [68]:
```
test5([1,2,3,4,5],[9,8,7,6])
```

Out[68]:
```
[1, 2, 3, 4, 5, 9, 8, 7, 6]
```

In [70]:
```
test5(b = "Singh",a = "Akash ")
```

Out[70]:
```
'Akash Singh'
```

In [71]:
```
l = [1,2,3,4,5,"sneha" , "singh" , [9,8,7,6]]
```

# create a function which will take list as a input and give me a final list with all the numeric value

In [72]:
```
def test6(a):
    n = []
    for i in a :
        if type(i) == int or type(i) == float :
            n.append(i)
    return n
```

In [73]:
```
test6(l)
```

Out[73]:
```
[1, 2, 3, 4, 5]
```

In [74]:
```
l
```

Out[74]:
```
[1, 2, 3, 4, 5, 'sneha', 'singh', [9, 8, 7, 6]]
```

In [75]:
```
def test7(a) :
    n = []
    for i in a :
        if type(i) == list :
            for j in i :
                if type(j) == int or type(j) == float:
                    n.append(j)
        else:
            if type(i) == int or type(i) == float :
                n.append(i)
        return n
```

In [78]:
```
test7(l)
```

Out[78]:
```
[1]
```

In [79]:
```
l
```

Out[79]:
```
[1, 2, 3, 4, 5, 'sneha', 'singh', [9, 8, 7, 6]]
```

In [80]:
```
def test8(a,b,c,d,e):
    pass
```

In [81]:
```
test8(1,2,3,4,5,6,7,8,9,12,15)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1936\3841440927.py in <module>
----> 1 test8(1,2,3,4,5,6,7,8,9,12,15)

TypeError: test8() takes 5 positional arguments but 11 were given
```

In [82]:
```
def test9(*args):
    return args
```

In [83]:
```
test9(1,2,3,4,5,6,7,8,9)
```

Out[83]:
```
(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

In [84]:
```
def test10(*number):
    return number
```

In [86]:
```
test10(1,2,3,4,5,6,7,89,1,1,2,2,3,45,5,6,)
```

Out[86]:
```
(1, 2, 3, 4, 5, 6, 7, 89, 1, 1, 2, 2, 3, 45, 5, 6)
```

In [88]:
```
def test11 (*args, a):
    return args , a
```

In [90]:
```
test11(2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1936\3641381080.py in <module>
----> 1 test11(2)

TypeError: test11() missing 1 required keyword-only argument: 'a'
```

In [91]:
```
test11(1,2,3,4,5 ,a = "sneha")
```

Out[91]:
```
((1, 2, 3, 4, 5), 'sneha')
```

In [92]:
```
def test12(**kwargs):
    return kwargs
```

In [93]:
```
test12()
```

Out[93]:
```
{}
```

In [94]:
```
type(test12())
```

Out[94]:
```
dict
```

In [95]:
```
test12( a = 12 , b = 13 , c = [18,19,20], d = ("sneha" , "singh") )
```

Out[95]:
```
{'a': 12, 'b': 13, 'c': [18, 19, 20], 'd': ('sneha', 'singh')}
```

In [96]:
```
def test13(**kwargs):
    for i  in kwargs.keys():
        if type(kwargs[i]) == list:
            return i , kwargs[i]
```

In [97]:
```
test13(a = 1, b = 2 ,c =[9,7,8,7] , d = ("sneha","singh"))
```

Out[97]:
```
('c', [9, 7, 8, 7])
```

In [98]:
```
def test14(*args, **kwargs):
    return args , kwargs
```

In [99]:
```
test14(2,3,4,a= 34 ,b = 24)
```

Out[99]:
```
((2, 3, 4), {'a': 34, 'b': 24})
```

# Generator Function :-

In [100…
```
range(1,10)
```

Out[100]:
```
range(1, 10)
```

In [101…
```
for i in range(1,10):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
```

In [102…
```
l = [1,2,3,4,5,6,7,8,9,"sneha" , "singh"]
```

In [103…
```
def test15(a) :
    n = []
    for i in a:
        if type(i) == int :
            n.append(i)
    return n
```

In [104…
```
test15(l)
```

Out[104]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

# Fibonacci :

0,1,1,2,3,4,5,8,13,21,34

In [106…
```
def test_fib(n) :
    a,b = 0,1
    for i in range(n):
        yield a
        a,b = b , a+b
```

In [107…
```
test_fib(10)
```

Out[107]:
```
<generator object test_fib at 0x00000190E76A4270>
```

In [108…
```
for i in test_fib(10) :
    print(i)
```

```
0
1
1
2
3
5
8
13
21
34
```

In [110…
```python
def test_fib2(n):
    a,b = 0,1
    for i in range(n):
        yield a
        a,b = b , a+1
```

In [111…
```python
test_fib2(20)
```

Out[111]:
```
<generator object test_fib2 at 0x00000190E76FA9E0>
```

In [113…
```python
for i in test_fib2(20):
    print (i)
```

```
0
1
1
2
2
3
3
4
4
5
5
6
6
7
7
8
8
9
9
10
```

In [ ]:
```python

```