

Day_7_Function(02)_(DATA_MINDS)

September 9, 2023

```
[7]: def test6(a,b):  
      return a+b
```

```
[9]: test6(2,4)
```

```
[9]: 6
```

```
[12]: test6("Virat"," yash")
```

```
[12]: 'Virat yash'
```

```
[13]: test6([1,2,3],[ 4,5,6])
```

```
[13]: [1, 2, 3, 4, 5, 6]
```

```
[14]: l=[1,2,3,4,5,"Virat Tiwari",[7,8,9]]
```

```
[15]: l
```

```
[15]: [1, 2, 3, 4, 5, 'Virat Tiwari', [7, 8, 9]]
```

```
[16]: type(l)
```

```
[16]: list
```

```
[17]: len(l)
```

```
[17]: 7
```

```
[20]: l1=[]  
      for i in l:  
          if type(i)==int or type(i)==float:  
              l1.append(i)
```

```
[21]: l1
```

```
[21]: [1, 2, 3, 4, 5]
```

```
[25]: def test7(l):  
      l1=[]  
      for i in l:  
          if type(i)==int or type(i)==float:  
              l1.append(i)  
      return l1
```

```
[27]: test7(l)
```

```
[27]: [1, 2, 3, 4, 5]
```

```
[28]: 1
```

```
[28]: [1, 2, 3, 4, 5, 'Virat Tiwari', [7, 8, 9]]
```

```
[37]: def test8(a):  
      l=[]  
      for i in a:  
          if type(i)==list:  
              for j in i:  
                  l.append(j)  
          else:  
              if type(i)==int or type(i)==float:  
                  l.append  
      return l
```

```
[41]: test8(l)
```

```
[41]: [7, 8, 9]
```

```
[46]: def test9(a):  
  
      """DOCSTRING - This is how we extract the num the list"""  
  
      l=[]  
      for i in a:  
          if type(i)==list:  
              for j in i:  
                  l.append(j)  
          else:  
              if type(i)==int or type(i)==float:  
                  l.append  
      return l
```

```
[ ]: test9()
```

WE WRITE DOCSTRING FOR THE FUNCTION FOR UNDERSTAD THE CODE OR FUNCTION IN FUTURE BY OTHER DEVELOPERS THOSE WHO WORK ON OUR CODE OR PROGRAMS IN FUTURE

```
[47]: def test10(a,b):  
      return a+b
```

```
[48]: test10(145,36478)
```

```
[48]: 36623
```

NOTE - *ARGS MEANS WE PASS N NUMBER OF ARGUMENTS INSIDE A FUNCTION

Args is just a convention

NOTE - IT REPRESENT DATA INTO THE FORM OF TUPLES

```
[49]: def test11(*args):  
      return args
```

```
[50]: test11()
```

```
[50]: ()
```

```
[52]: type(test11())
```

```
[52]: tuple
```

```
[53]: test11(2,3,4)
```

```
[53]: (2, 3, 4)
```

```
[54]: test11(125,54,86,36,79,2)
```

```
[54]: (125, 54, 86, 36, 79, 2)
```

```
[56]: test11(1,5,"Virat tiwari",254.6,[145,678])
```

```
[56]: (1, 5, 'Virat tiwari', 254.6, [145, 678])
```

```
[58]: def test12(*virat):  
      return virat
```

```
[59]: test12("Virat tiwari",[12,45,78],"Data science",25.01)
```

```
[59]: ('Virat tiwari', [12, 45, 78], 'Data science', 25.01)
```

```
[60]: def test14(c,d,a=10,b=25):  
      return a,b,c,d
```

```
[61]: test14(4,6)
```

```
[61]: (10, 25, 4, 6)
```

NOTE - **Kwargs ARE USED FOR PASSING N NUMBER OF ARGUMENTS IN A FUNCTION

ITS GIVES THE OUTPUT IN THE FORM OF LIST

```
[ ]: def test15(**kwargs):  
      return kwargs
```

```
[63]: test15()
```

```
[63]: {}
```

```
[64]: type(test15())
```

```
[64]: dict
```

```
[65]: test15(a=[1,2,3],b="Virat tiwari",c=52.4)
```

```
[65]: {'a': [1, 2, 3], 'b': 'Virat tiwari', 'c': 52.4}
```