

print

```
In [2]: print (10)
```

10

```
In [3]: a=34  
print(a)
```

34

```
In [4]: print(346.64778)
```

346.64778

```
In [5]: print(4554,66474)
```

4554 66474

```
In [7]: print(77,76451,"Rana",4546.776,"Saeed")
```

77 76451 Rana 4546.776 Saeed

```
In [10]: student=77,77.77,"Rana"  
print(student)
```

(77, 77.77, 'Rana')

```
In [11]: # print m hum her type ki value type ya print karwa sakti hn
```

type

```
In [12]: # python m hum variable declare nhi krty ,is m variable type python khud chek  
krta hy
```

```
In [13]: a=29  
type(a)
```

Out[13]: int

```
In [14]: a=6577.78  
type(a)
```

Out[14]: float

```
In [15]: type(78)
```

```
Out[15]: int
```

```
In [16]: type("Rana Saeed")
```

```
Out[16]: str
```

Operatoers +,-,*,/

```
In [17]: print(12+18)
```

```
30
```

```
In [18]: a=29  
b=31  
print(a+b)
```

```
60
```

```
In [19]: a=70  
b=32  
print(a-b)
```

```
38
```

```
In [20]: a=200      # result in python in floating /point ,  
b=100  
print(a/b)
```

```
2.0
```

```
In [21]: print(a//b) # agr hum dubl slash lgayn gy to result integer m ayga
```

```
2
```

```
In [22]: a=20  
b=15.5  
print(a/b)
```

```
1.2903225806451613
```

```
In [25]: print(a//b)
```

```
1.0
```

```
In [26]: # if we add string in pythn
```

```
In [28]: print("Rana"+"Saeed")
```

RanaSaeed

```
In [29]: a=100  
b=200  
print(a/b)
```

0.5

```
In [30]: print(a//b)      #duble sirf integer value print krwata hy
```

0

```
In [31]: # if we add number in str so the result will be
```

```
In [32]: a="57"  
b="85"    # in string compiler print the number as written  
print(a+b)
```

5785

If Else Statement

```
In [33]: a=100  
b=200    # NOTE; if condition true block will be execute while (:) colon cal  
         # led indent in program and print when goes begin so then  
         # then block ended and (super rana saeed) is out of block.  
if a<b:  
    print("Rana")  
    print("Saeed")  
    print("roll no",77)  
print("Super Rana Saeed")
```

Rana
Saeed
roll no 77
Super Rana Saeed

```
In [34]: if a>b:  
    print(77)    # if codition is false ,block will not be executed,only ou  
    t of block print will be executed  
    print("nice")  
    print("yes")  
print("Ohoo")
```

Ohoo

Comments

```
In [35]: # starting with hash ,all sentences called comments
```

```
In [37]: # Rana Saeed please doing hard work
```

Input

```
In [40]: # in pythn programming languages,
# we take input it is also string input stored in data type.
# marks=int(input("Enter the value"))
```

```
In [41]: # value=str(input("Enter value as string"))
```

Lists

```
In [42]: names=["Rana","saeed","Ramzan","Ijaz","Irfan","Ali","Abbas","Imran","Amjad"]
```

```
In [43]: names
```

```
Out[43]: ['Rana', 'saeed', 'Ramzan', 'Ijaz', 'Irfan', 'Ali', 'Abbas', 'Imran', 'Amjad']
```

Access the Index in list

```
In [48]: names.index("Ali") # index("") value k zraye hm index maloom kr sakti hn
```

```
Out[48]: 5
```

```
In [51]: names.index("Saeed") # pythn is a case sensitive languages, it does not track the value ,
# Saeed ka S capital hy but list m Saeed ka s small hy
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-51-60c8c8e64027> in <module>
----> 1 names.index("Saeed")
```

```
ValueError: 'Saeed' is not in list
```

```
In [52]: names.index("saeed")
```

```
Out[52]: 1
```

```
In [55]: names.index("Amjad")
```

```
Out[55]: 8
```

Access the value in Lists

```
In [57]: names[1]    # value find krny k liy list name bracket m index numbr write kry gy
```

```
Out[57]: 'saeed'
```

```
In [58]: names[7]
```

```
Out[58]: 'Imran'
```

Length

```
In [59]: len(names)
```

```
Out[59]: 9
```

```
In [60]: # list ki lenth 9 hy
```

```
In [61]: # len() it is called function, istrah hum kisi bhi list ki length check kr saktay hn
```

```
In [62]: # len(list name)
```

Add a value in list

```
In [63]: # list m value add krny k liey ye "list name.append('jo value add krni hy wo w rite kren')"
```

```
In [64]: names.append("Shoban") # jo value add krengy wo list k end m add ho gy
```

```
In [65]: names
```

```
Out[65]: ['Rana',  
          'saeed',  
          'Ramzan',  
          'Ijaz',  
          'Irfan',  
          'Ali',  
          'Abbas',  
          'Imran',  
          'Amjad',  
          'Shoban']
```

```
In [66]: names.append("kamran")
```

```
In [67]: names
```

```
Out[67]: ['Rana',  
          'saeed',  
          'Ramzan',  
          'Ijaz',  
          'Irfan',  
          'Ali',  
          'Abbas',  
          'Imran',  
          'Amjad',  
          'Shoban',  
          'kamran']
```

Insert a value in Lists

```
In [69]: # insert function k zreye hm lists m kisi bhi index pr koi value store krwa s  
          akty hn
```

```
In [70]: names.insert(3,"Shakeel")
```

```
In [71]: names
```

```
Out[71]: ['Rana',  
          'saeed',  
          'Ramzan',  
          'Shakeel',  
          'Ijaz',  
          'Irfan',  
          'Ali',  
          'Abbas',  
          'Imran',  
          'Amjad',  
          'Shoban',  
          'kamran']
```

Extend Function

```
In [72]: # append or insert function k zraye hm single value lists m store krwa saky h  
          n  
          # Extend Function k zraye hum multiple value list m store krwa saky hn
```

```
In [73]: names.extend(["Saddique", "Asghar", "Shahzad"])
```

```
In [74]: names
```

```
Out[74]: ['Rana',  
          'saeed',  
          'Ramzan',  
          'Shakeel',  
          'Ijaz',  
          'Irfan',  
          'Ali',  
          'Abbas',  
          'Imran',  
          'Amjad',  
          'Shoban',  
          'kamran',  
          'Saddique',  
          'Asghar',  
          'Shahzad']
```

Count

```
In [75]: # count function k zraye hm lists k value count kr sakty hn keh konsi value list m kitny mrtba ay hy
```

```
In [76]: names.count("saeed")
```

```
Out[76]: 1
```

```
In [77]: names.count("Ali")
```

```
Out[77]: 1
```

Clear

```
In [78]: # list m clear function k zraye hm value parmanant delete kr sakty hn
```

```
In [79]: number=[3,5,7,9,9]
```

```
In [81]: number.clear()
```

```
In [82]: number
```

```
Out[82]: []
```

```
In [83]: # clear function k zraye hm lists k all value parmanant delete kr skty hn
```

```
In [85]: number.clear()
```

```
In [86]: number
```

```
Out[86]: []
```

Copy

```
In [4]: city=["gogran","lodhran","bwp","multan"]
```

```
In [5]: city
```

```
Out[5]: ['gogran', 'lodhran', 'bwp', 'multan']
```

```
In [6]: city1=city.copy()      # copy by value in the lists
```

```
In [7]: city1
```

```
Out[7]: ['gogran', 'lodhran', 'bwp', 'multan']
```

```
In [9]: city3=city      # copy by refference
```

```
In [10]: city3
```

```
Out[10]: ['gogran', 'lodhran', 'bwp', 'multan']
```

```
In [11]: city.append("lahore")
```

```
In [12]: city3
```

```
Out[12]: ['gogran', 'lodhran', 'bwp', 'multan', 'lahore']
```

```
In [16]: city1
```

```
Out[16]: ['gogran', 'lodhran', 'bwp', 'multan']
```

```
In [17]: city
```

```
Out[17]: ['gogran', 'lodhran', 'bwp', 'multan', 'lahore']
```

Removing

```
In [15]: # removing an item from lists
```

```
In [19]: city.remove("bwp")
```



```
In [20]: city3
Out[20]: ['gogran', 'lodhran', 'multan', 'lahore']

In [21]: city
Out[21]: ['gogran', 'lodhran', 'multan', 'lahore']

In [22]: city1
Out[22]: ['gogran', 'lodhran', 'bwp', 'multan']
```

del

```
In [23]: # it is not a function, but we can delete value

In [27]: del city[3]

In [28]: city
Out[28]: ['gogran', 'lodhran', 'multan']

In [30]: del city[2]

In [31]: city
Out[31]: ['gogran', 'lodhran']

In [32]: city.extend(["multan", "bwp", "faisalbad"])

In [33]: city
Out[33]: ['gogran', 'lodhran', 'multan', 'bwp', 'faisalbad']
```

POP

```
In [35]: # pop , its mean that remove or delete value but its advantages is that we can return value

In [36]: pop1=city.pop(3)

In [37]: pop1
Out[37]: 'bwp'
```

```
In [38]: city
Out[38]: ['gogran', 'lodhran', 'multan', 'faisalbad']

In [41]: city.append(pop1)

In [42]: city
Out[42]: ['gogran', 'lodhran', 'multan', 'faisalbad', 'pop1', 'bwp']

In [43]: city.append(pop1)

In [44]: city
Out[44]: ['gogran', 'lodhran', 'multan', 'faisalbad', 'pop1', 'bwp', 'bwp']
```

Sort

```
In [45]: city
Out[45]: ['gogran', 'lodhran', 'multan', 'faisalbad', 'pop1', 'bwp', 'bwp']

In [46]: city.sort()

In [47]: city
Out[47]: ['bwp', 'bwp', 'faisalbad', 'gogran', 'lodhran', 'multan', 'pop1']

In [48]: city.reverse()      #Reverse function used the value to be reversed

In [49]: city
Out[49]: ['pop1', 'multan', 'lodhran', 'gogran', 'faisalbad', 'bwp', 'bwp']

In [52]: city.sort(Revese=True)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-52-b49ab04807e4> in <module>
----> 1 city.sort(Revese=True)

TypeError: 'Revese' is an invalid keyword argument for sort()
```

```
In [ ]:

In [53]: city
Out[53]: ['pop1', 'multan', 'lodhran', 'gogran', 'faisalbad', 'bwp', 'bwp']
```

SLICING

In [54]: `# in slicing ,we can get more value at a time`

In [55]: `# -14,-13,-12,-11,-10,-9,-8,-7,-6,-5,-4, -3, -2, -1, -0 # Negative Index`
`number=[2 , 4, 3, 5, 6, 8, 7, 9, 30, 45, 27, 90, 96, 20, 10]`
`# 0 , 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,12, 13, 14 # index positive`

In [56]: `number`

Out[56]: `[2, 4, 3, 5, 6, 8, 7, 9, 30, 45, 27, 90, 96, 20, 10]`

In [58]: `print(number[1:4]) # if we get values 4,3,5 # we give one extra value`
`[4, 3, 5]`

In [59]: `print(number[-13:-10])`
`[3, 5, 6]`

In [60]: `print(number[-1:])`
`[10]`

In [61]: `print(number[:])`
`[2, 4, 3, 5, 6, 8, 7, 9, 30, 45, 27, 90, 96, 20, 10]`

In [62]: `print(number[2::2])`
`[3, 6, 7, 30, 27, 96, 10]`

In [63]: `print(number[::3])`
`[2, 5, 7, 45, 96]`

indexing

`print(number[14])`

In []: `print`