

## Display the “data type” of a variable

*Data types in python are → Int, float, str, list, tuple, dict*

```
>>> x=23
>>> type(x)
<class 'int'>
```

```
>>> y=2.5
>>> type(y)
<class 'float'>
```

```
>>> name='Kuma'
>>> type(name)
<class 'str'>
```

```
>>> num=[34,67,2,9]
>>> num
[34, 67, 2, 9]
>>> names=['Kuma','thilini']
>>> names
['Kuma', 'thilini']
```

```
>>> type(num)
<class 'list'>
>>> type(names)
<class 'list'>
```

```
>>> height=[5.6,6.3,6,6.1]
>>> type(height)
<class 'list'>
```

```
>>> height=(5.6,6.3,6,6.1)
>>> type(height)
<class 'tuple'>
```

```
>>> d={"name":"ravi","age":33}
>>> d
{'age': 33, 'name': 'ravi'}
>>> type(d)
<class 'dict'>
```

## For Statement

Eg 1  
for x in range (1,6):  
 print(x)

```
Output Eg 1
1
2
3
4
5
```

Eg 2  
for x in range (2,11,2):  
 print(x)

```
Output Eg 2
2
4
6
8
10
```

Eg 3  
for x in range (10,0,-1):  
 print(x,end=' ,')

```
Output - Eg 3
10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
```

Eg 4  
sum=0  
print('Odd number from 1 - 10: ')  
for x in range (1,11,2):  
 print(x)  
 sum=sum+x;  
print('Total: ',sum)

```
Output of Eg 4
Odd number from 1 - 10:
1
3
5
7
9
Total: 25
```

Eg 5  
for x in range(1.5, 3.5):  
 print ("Number : ",x)

```
Output – Eg - 5
Traceback (most recent call last):
  File "E:/nayana python/for3.py", line 1, in <module>
    for x in range(1.5, 3.5):
TypeError: 'float' object cannot be interpreted as an integer
```

Eg 6  
for x in range(1, 13):  
 print ("2 x %d=%d" % (x,2\*x))

Eg 7

```
#Nested for statement eg 1
for x in range(1,4):
    for y in range(1,x+1):
        print(x,end=' ')
    print('\n')
```

```
Output – Eg 7
1
2 2
3 3 3
```

Eg 8

```
#Nested for statement eg 2
for x in range(1,4):
    for y in range(1,x+1):
        print(y,end=' ')
    print('\n')
```

```
Output – Eg 8
1
1 2
1 2 3
```

Eg 9

```
#Nested for statement eg 3
for x in range(1,4):
    for y in range(1,x+1):
        print('*',end=' ')
    print('\n')
```

```
Output – Eg 9
*
* *
* * *
```

## While Statement (While Loop)

Eg (a)

```
c=1
while (c<=5):
    print(c)
    c=c+1
```

```
Output Eg (a)
1
2
3
4
5
```

Eg (b)

```
t=1.0
while t<=5:
    print (t)
    t+=.5
```

```
Output Eg (b)
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
```

## Lists

List is like an array in Pascal language. More data can be store under the same name

### Creating a simple list

```
>>>num = [5,76,32,9]
```

5	76	32	9
num[0]	num[1]	num[2]	num[3]

### Print the List

```
>>>num
```

```
[5, 76, 32, 9]
```

### Print the value stored in 1<sup>st</sup> location in the List

```
>>>num[0]
```

```
>>>5
```

```
>>>num[-1] //reverse display
```

```
>>>9
```

```
>>>num [:3]
```

```
>>>[5, 76, 32,]
```

```
>>> num [3:]
```

```
>>>9
```

## Readymade Functions used in python

### Length of the list

```
>>> len (num)
```

```
>>>4
```

### Delete a list

```
>>>del (num)
```

Delete some items in the list

```
>>>del (num[2])
```

### **Maximum value in the list**

```
>>>max(num)
```

```
>>>76
```

### **Minimum value in the list**

```
>>>min(num)
```

```
>>>5
```

## **Methods used in python**

### **To add new item to the end of the list**

```
>>>num.append(555) or >>>num.extend([555])
```

### **To remove an item from the list**

```
>>> num.remove(67)
```

### **To Sort a list in ascending order**

```
>>>num.sort()
```

```
>>>num
```

```
>>>[5,9,32,76]
```

### **To Sort a list in descending order**

```
>>>num.reverse()
```

```
>>>num
```

```
>>>[76,32,9,5]
```

### **Count an item in the list**

```
>>>num.count(32)
```

```
>>>1
```

### **List with different data types**

```
>>> mix=[3,4.5,'Kuma']
```

```
>>> mix
```

```
[3, 4.5, 'Kuma']
```

## Tuples

- Same as list but tuples are immutable. So values can't assign or delete in tuples.
- They can't sort.
- Can store values with different data types.

### Creating a simple Tuple

```
>>>n = (5,76,32,9)
```

5	76	32	9
n[0]	n[1]	n[2]	n[3]

### Print the Tuple

```
>>>n
```

```
(5, 76, 32, 9)
```

### Print the value stored in 1<sup>st</sup> location in the Tuple

```
>>>n[0]
```

```
>>>5
```

### store values with different data types

```
>>> mix2=(3,4.5,'Kuma')
```

```
>>> mix2
```

```
(3, 4.5, 'Kuma')
```

## Dictionaries

```
>>> dic1={"Name":"Ravi","Age":40,"Dept":"IT"} #create a dictionary
```

```
>>> print (dic1) #print a dictionary
```

```
{'Age': 40, 'Name': 'Ravi', 'Dept': 'IT'}
```

```
>>> print (dic1.keys()) #print keys of a dictionary
```

```
dict_keys(['Age', 'Name', 'Dept'])
```

```
>>> print (dic1.items()) #print Items of a dictionary
```

```
dict_items([('Age', 40), ('Name', 'Ravi'), ('Dept', 'IT')])
```

### Update values in dictionary

```
>>>dic1['Age']=35
```

### Add new values to a dictionary

```
>>>dic1['Job']='Programmer'
```

### Delete dictionary elements

```
del dic1['Name'] # remove entry with key 'Name'  
dic1.clear()    # remove all entries in dict  
del dic1        # delete entire dictionary
```

### Find the length of a dictionary

```
>>>len(dic1)
```

```
>>>3
```

## List with For Statement (For Loop)

Eg 1

```
for x in [1,2,3,4,5]:  
    print (x)
```

Eg 2

```
marks=[78,98,65,69]  
for x in marks:  
    print (x)
```

Eg 3

```
for x in ['Ravi','Karu','Susi']:  
    print (x)
```

