

Lists

Lists are just like dynamically sized arrays

Lists need **not** be homogeneous always which makes it the most powerful tool **in** Python.

A single list may contain DataTypes like Integers, Strings, **as** well **as** Objects.

Lists are mutable, **and** hence, they can be altered even after their creation.

```
# Lists
# We can store multiple datatyped elements in a single list
# We use [] to denote a list
```

```
# writing an empty list
x = []
print(x)
print(type(x))
```

```
# list maintains insertion order
x = [11,23.45,'hello','a',True]
print(x)
```

```
# list support loops
x = [11,23.45,'hello','a',True]
for i in x:
    print(i)
```

```
# list elements have indexes
x = [11,23.45,'hello','a',True]
print(x)
print(x[0])
print(x[1])
print(x[4])
```

```
# processing list elements using loop
x = [11,23.45,'hello','a',True]
for i in range(0,5):
    print(x[i])
```

```
# list supports reverse indexing
x = [11,23.45,'hello','a',True]
print(x)
print(x[0])
print(x[1])
```

```
print("*****")
#reverse indexing
print(x[-1])
print(x[-2])
```

```
# list can have same datatyped values
x = ['a','b','c','d','e']
print(x)
```

```
# replacing a value in a list
x = [11,23.45,'hello','a',True]
print(x)
```

```
x[4] = False
print(x)
```

```
# reverse of a list
l = [11,22,33,44,55]
print(l)
l.reverse()
print(l)
```

.....

```
# List Functions
# NOTE: changes made to a list will be affected to the original list
# NOTE: By copying you cannot take backup of a list
```

```
# adding a new value to the existing list using fun append()
# by using append() the new will be appended as the list index
# append fun returns 'None'
x = [11,23.45,'hello','a',True]
print(x)
x.append(100)
print(x)
```

```
# adding a new element at a desired position using insert() function
# insert(position, new element)
x = [11,23.45,'hello','a',True]
print(x)
x.insert(2,1000)
print(x)
```

```
# Add multiple values to a list
list = [1, 2, 3]
list.extend([4, 5, 6])
print(list)
```

```
# deleting an element using fun pop()
# pop() deletes the last element of the list by default
# pop() also returns the deleted element
x = [11,23.45,'hello','a',True]
print(x)
print(x.pop())
print(x)
```

```
# deleting desired element using pop(index)
x = [11,23.45,'hello','a',True]
print(x)
y = x.pop(1)
print(x)
print("Deleted element: ", y)
```

```
# deleting an element using the value
# fun used is remove(element)
x = [11,23.45,'hello','a',True]
```

```
print(x)
x.remove('a')
print(x)
```

```
# getting index for any element of a list using fun index(element)
x = [11,23.45,'hello','a',True]
print(x)
print(x.index('hello'))
```

```
# list support duplicate values
x = [11,23.45,'hello','a',True,11,2,'a']
print(x)
```

```
# you will get IndexError
x = [11,23.45,'hello','a',True]
print(x)
print(x[50])
```

```
# sorting elements of a list using fun sort()
# by default sort() fun arranges the elements in ascending order
x = [11,1,22,3,44,5,66]
print("Before sorting: ", x)
x.sort() # ascending order
print("After sorting: ", x)

x.sort(reverse=True)
print("Descending Order: ", x)
```

```
# for string values sort() fun arranges in dictionary order
x = ['a','c','e','f','g','b']
print(x)
x.sort()
print(x)
```

```
x = ['apple','astronut','mango','banana','apricot']
print(x)
x.sort()
print(x)
```

```
# any operation on a list or its copy will change both the lists
x = [1,22,3,44,5,66,7]
y = x # getting copy of list x in y
print(x)
print(y)

x.sort()
print(x)
print(y) # y also will be sorted
```

```
# sorting elements of a list using fun 'sorted()'
# sorted() fun will return sorted list, without disturbing the
original list
x = [1,22,3,44,5,66,7]
y = sorted(x)
print(x)
print("sorted list: ", y)
```

```
# To find the max and min value of a list
prices = [589.36, 237.81, 230.87, 463.98, 453.42]
print(max(prices))
print(min(prices))
```

```
# to find the no of elements of a list
list_1 = [50.29]
list_2 = [76.14, 89.64, 167.28]
print('list_1 length is ', len(list_1))
print('list_2 length is ', len(list_2))
```

```
# To wipe out all the elements of a list
months = ['January', 'February', 'March', 'April', 'May']
print(months)
months.clear()
print(months)
```

```
# To find no of occurrences of an element
fruits = ['cherry', 'apple', 'cherry', 'banana', 'cherry']
x = fruits.count("cherry")
```

```
print(x)
```

```
# To remove a value from the list
```

```
fruits = ['apple', 'banana', 'cherry', 'orange', 'pineapple']  
print(fruits)  
fruits.remove("banana")  
print(fruits)
```

```
# To copy the elements of a list
```

```
fruits = ['apple', 'banana', 'cherry', 'orange']  
x = fruits.copy()  
print(x)  
x.append('kiwi')  
print(x)  
print(fruits)
```

.....

```
# list in list(nested lists)
```

```
l = [11,2,33,4,55,['one',2,'3',45.67],77,8,9]  
print(l)  
print(l[4])  
print(l[5]) # gets inner list  
print("First element of inner list: ", l[5][0])  
print("First element of inner list: ", l[5][0][0])  
print("First element of inner list: ", l[5][0][1])  
print("First element of inner list: ", l[5][0][2])
```

```
# multiple values stored directly in a variable will get 'tuple'  
datatype
```

```
# to convert a tuple to list using fun list()  
l = 11,2,33,4,55,6,77  
print(l)  
print(type(l)) # tuple
```

```
l = list(l)  
print(l)
```

```
# appending loop generated values to an empty list
l = []
print(l)
for i in range(1,11):
    l.append(i)
    pass

print(l)
```

```
# list as a fun parameter
l = [11,22,3,44,5,66]

def test(l):
    for i in l:
        print(i)
        pass
    pass

test(l)
```

```
# list as return value
l = []
def test():
    for i in range(1,11):
        l.append(i)
        pass
    return l

result = test()
print(result)
```