

Create a List

```
In [6]: li = [1, 2.0, 3, "arya"]
        tup = 1, 2, 3.0, 4, "arya"

        print(type(li))
        print(type(tup))

<class 'list'>
<class 'tuple'>
```

Access and Change elements in a List

```
In [10]: # indexing starts from '0'
         li[1]
```

Out[10]: 2.0

```
In [13]: li[1]=5.0
         print(li)
```

[1, 5.0, 3, 'arya']

Slicing of a List

```
In [16]: li=[1, 2, 3, 4, 5, 6, 7]
         li1 = li[1 : 5 : 2] # start : end : steps

         print(li)
         print(li1)
```

[1, 2, 3, 4, 5, 6, 7]  
[2, 4]

```
In [19]: li2 = li[0 : 100 : 2]
         print(li2)
```

[1, 3, 5, 7]

appending in a list

```
In [25]: li=[1, 2, 3, 4, 5, 6, 7]
         li.append(9)
         li.append('arya')
         print(li)
```

[1, 2, 3, 4, 5, 6, 7, 9, 'arya']

```
In [29]: # appending occurs at last whereas inserting occurs anywhere
         li=[1, 2, 3, 4, 5, 6, 7]
         li.insert(1, 'tito')
         print(li)
```

[1, 'tito', 2, 3, 4, 5, 6, 7]

```
In [31]: li=[1, 2, 3, 4, 5, 6, 7]
         li.insert(100, 'tito') # automatically adds at last
         print(li)
```

[1, 2, 3, 4, 5, 6, 7, 'tito']

```
In [33]: li=[1, 2, 3, 4, 5, 6, 7]
         li.append([8, 9, 10])
         print(li)
```

[1, 2, 3, 4, 5, 6, 7, [8, 9, 10]]

```
In [35]: li=[1, 2, 3, 4, 5, 6, 7]
         li.extend([8, 9, 10]) # extend function allows us to retrieve data from [] and add it to original list
         print(li)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Removing elements from List

```
In [41]: # you cannot remove elements not present in the list

         li=[1, 2, 3, 2, 4, 5, 6, 7]
         li.remove(2) # removes the first occurrence of 2
         print(li)
```

[1, 3, 2, 4, 5, 6, 7]

```
In [44]: li=[1, 2, 3, 2, 4, 5, 6, 7]
         li.pop() # removes from the last
         print(li)
```

[1, 2, 3, 2, 4, 5, 6]

```
In [46]: li=[1, 2, 3, 2, 4, 5, 6, 7]
         li.pop(2)
```

Out[46]: 3

```
In [48]: li=[1, 2, 3, 2, 4, 5, 6, 7]
         len(li)
```

Out[48]: 8

Looping through a List

```
In [55]: # using index

         li=[5, 4, 2, 1]
         for i in range(len(li)) :
             print(li[i])
```

5  
4  
2  
1

```
In [54]: # using iterator

         li=[5, 4, 2, 1]
         for itr in li :
             print(itr)
```

5  
4  
2  
1

```
In [59]: # using slicing
         li=[5, 4, 2, 1, 3, 7, 9, 11, 34, 23, 45]
         for itr in li[2 : 10 : 2] :
             print(itr)
```

2  
3  
9  
34

Negative indexing of List

```
In [62]: li=[5, 4, 2, 1, 3, 7, 9, 11, 34, 23, 'arya']
         print(li[-2]) # prints -xth element in the list
```

23

Sequencing

```
In [66]: # list_name [start : end : steps]
         # we've discussed abt it earlier

         li=[5, 4, 2, 1, 3, 7, 9, 11, 34, 23, 'arya']
         li1=li[-9 : -1 : 2]
         print(li1)
```

[2, 3, 9, 34]

```
In [68]: li[-1:]
```

Out[68]: ['arya']

Taking input in a List

Line Seperated Input

```
In [70]: # input() by default converts everything to a list

         n=input()
         type(n)
```

Out[70]: 6  
str

```
In [72]: # although, we can specify the type of the input

         n=int(input()) # actually, this is nothing but typecasting
         type(n)
```

Out[72]: 3  
int

```
In [74]: # now we take input in list accordingly

         li = [] # empty list
         n=int(input())
         for i in range(n) :
             ele=int(input())
             li.append(ele)
         print(li)
```

3  
12  
11  
45  
[12, 11, 45]

Space separated input

```
In [77]: # using split function

         str=input()
         str1=str.split(' ') # OR, str.split()
         str1
```

Out[77]: 1 2 23 12 1  
['1', '2', '23', '12', '1']

```
In [81]: # OR

         li=[int(itr) for itr in input().split(' ')] # shortcut

         print(li)
         print(type(li[0]))
```

1 2 23 12  
[1, 2, 23, 12]  
<class 'int'>