

Python List

Python List

- A list in Python is used to store the sequence of various types of data. Python lists are mutable type its mean we can modify its element after it created. However, Python consists of six data-types that are capable to store the sequences, but the most common and reliable type is the list.
- A list can be defined as a collection of values or items of different types. The items in the list are separated with the comma (,) and enclosed with the square brackets [].

```
L1 = ["John", 102, "USA"]
```

```
L2 = [1, 2, 3, 4, 5, 6]
```

Characteristics of Lists

The lists are ordered.

The element of the list can access by index.

The lists are the mutable type.

```
a = [1,2,"Peter",4.50,"Ricky",5,6]
```

```
b = [1,2,5,"Peter",4.50,"Ricky",6]
```

```
a == b
```

Output:

False

- Both lists have consisted of the same elements, but the second list changed the index position of the 5th element that violates the order of lists. When compare both lists it returns the false.

```
a = [1, 2,"Peter", 4.50,"Ricky",5, 6]
```

```
b = [1, 2,"Peter", 4.50,"Ricky",5, 6]
```

```
a == b
```

Output:

True

```
emp = ["John", 102, "USA"]
Dep1 = ["CS",10]
Dep2 = ["IT",11]
HOD_CS = [10,"Mr. Holding"]
HOD_IT = [11, "Mr. Bewon"]

print("printing employee data...")

print("Name : %s, ID: %d, Country: %s"%(emp[0],emp[1],emp[2]))

print("printing departments...")

print("Department 1:\nName: %s, ID: %d\nDepartment 2:\nName: %s, ID: %s"%(Dep1[0],Dep2[1],Dep2[0],Dep2[1]))

print("HOD Details ....")

print("CS HOD Name: %s, Id: %d"%(HOD_CS[1],HOD_CS[0]))

print("IT HOD Name: %s, Id: %d"%(HOD_IT[1],HOD_IT[0]))

print(type(emp),type(Dep1),type(Dep2),type(HOD_CS),type(HOD_IT))
```

```
printing employee data...
Name : John, ID: 102, Country: USA
printing departments...
Department 1:
Name: CS, ID: 11
Department 2:
Name: IT, ID: 11
HOD Details ....
CS HOD Name: Mr. Holding, Id: 10
IT HOD Name: Mr. Bewon, Id: 11
<class 'list'> <class 'list'> <class 'list'> <class 'list'> <class 'list'>
```

List indexing and splitting

- The indexing is processed in the same way as it happens with the strings. The elements of the list can be accessed by using the slice operator [].
- The index starts from 0 and goes to length - 1. The first element of the list is stored at the 0th index, the second element of the list is stored at the 1st index, and so on.

List = [0, 1, 2, 3, 4, 5]

0	1	2	3	4	5
---	---	---	---	---	---

List[0] = 0

List[0:] = [0,1,2,3,4,5]

List[1] = 1

List[:] = [0,1,2,3,4,5]

List[2] = 2

List[2:4] = [2, 3]

List[3] = 3

List[1:3] = [1, 2]

List[4] = 4

List[:4] = [0, 1, 2, 3]

List[5] = 5

sub-list of the list using the following syntax.

list_variable(start:stop:step)

The **start** denotes the starting index position of the list.

The **stop** denotes the last index position of the list.

The **step** is used to skip the nth element within a **start:stop**

```
list = [1,2,3,4,5,6,7]
```

```
print(list[0])
```

```
print(list[1])
```

```
print(list[2])
```

```
print(list[3])
```

```
# Slicing the elements
```

```
print(list[0:6])
```

```
# By default the index value is 0 so its starts from the 0th element and goes for index -1.
```

```
print(list[:])
```

```
print(list[2:5])
```

```
print(list[1:6:2])
```

Output

Output:

1

2

3

4

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

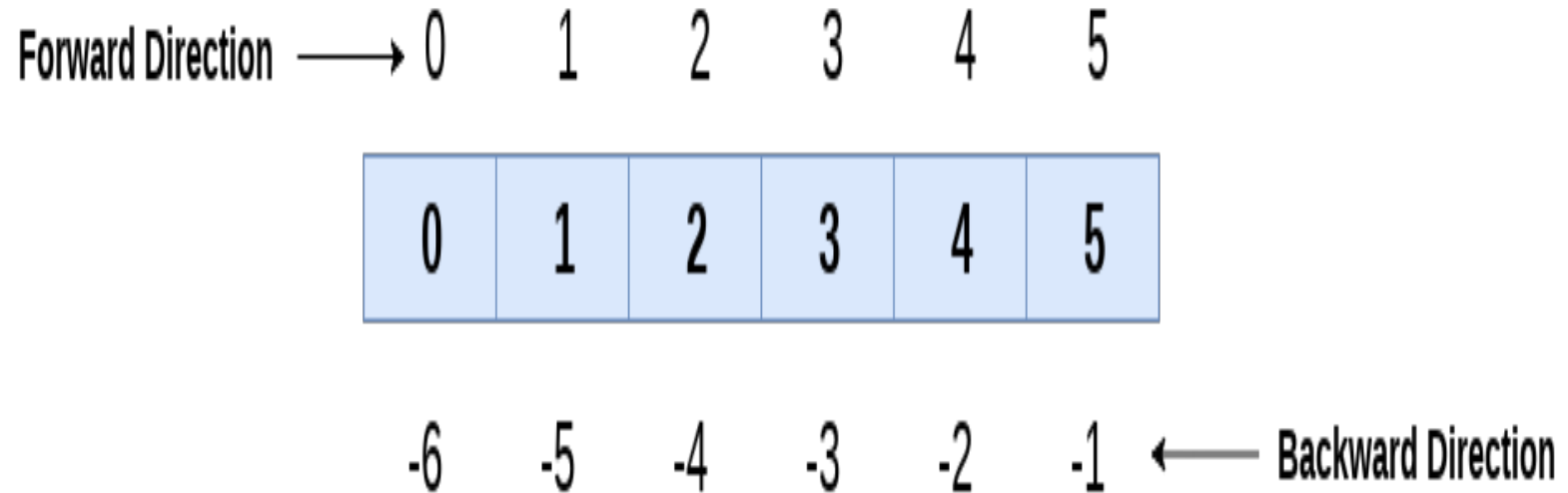
[1, 2, 3, 4, 5, 6, 7]

[3, 4, 5]

[2, 4, 6]

- Unlike other languages, Python provides the flexibility to use the negative indexing also.
- The negative indices are counted from the right. The last element (rightmost) of the list has the index -1; its adjacent left element is present at the index -2 and so on until the left-most elements are encountered.

List = [0, 1, 2, 3, 4, 5]




```
list = [1,2,3,4,5]
```

```
print(list[-1])
```

```
print(list[-3:])
```

```
print(list[:-1])
```

```
print(list[-3:-1])
```

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

Updating List values: Lists are the most versatile data structures in Python since they are mutable, and their values can be updated by using the slice and assignment operator. Python also provides `append()` and `insert()` methods, which can be used to add values to the list.

```
list = [1, 2, 3, 4, 5, 6]
```

```
print(list)
```

```
# It will assign value to the value to the second index
```

```
list[2] = 10
```

```
print(list)
```

```
# Adding multiple-element
```

```
list[1:3] = [89, 78]
```

```
print(list)
```

```
# It will add value at the end of the list
```

```
list[-1] = 25
```

```
print(list)
```

Output:

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

```
[1, 2, 10, 4, 5, 6]
```

```
[1, 89, 78, 4, 5, 6]
```

```
[1, 89, 78, 4, 5, 25]
```

Python List Operations

The concatenation (+) and repetition (*) operators work in the same way as they were working with the strings. Consider a Lists l1 = [1, 2, 3, 4], and l2 = [5, 6, 7, 8] to perform operation

Operator	Description	Example
Repetition	The repetition operator enables the list elements to be repeated multiple times.	<code>l1*2 = [1, 2, 3, 4, 1, 2, 3, 4]</code>
Concatenation	It concatenates the list mentioned on either side of the operator.	<code>l1+l2 = [1, 2, 3, 4, 5, 6, 7, 8]</code>
Membership	It returns true if a particular item exists in a particular list otherwise false.	<code>print(2 in l1)</code> prints True.
Iteration	The for loop is used to iterate over the list elements.	<code>for i in l1: print(i)</code> Output 1 2 3 4
Length	It is used to get the length of the list	<code>len(l1) = 4</code>

Iterating a List

A list can be iterated by using a for - in loop. A simple list containing four strings, which can be iterated as follows.

```
list = ["John", "David", "James", "Jonathan"]
```

```
for i in list:
```

```
    # The i variable will iterate over the elements of the List and contains each element in each iteration.
```

```
    print(i)
```

Output:

John

David

James

Jonathan

Adding elements to the list

Python provides `append()` function which is used to add an element to the list. However, the `append()` function can only add value to the end of the list.

```
#Declaring the empty list
l = []

#Number of elements will be entered by the user
n = int(input("Enter the number of elements in the list:"))

# for loop to take the input
for i in range(0,n):
    # The input is taken from the user and added to the list as the item
    l.append(input("Enter the item:"))

print("printing the list items..")

# traversal loop to print the list items
for i in l:
    print(i, end = " ")
```

Output:

```
Enter the number of elements in the list:5
Enter the item:25
Enter the item:46
Enter the item:12
Enter the item:75
Enter the item:42
printing the list items
25 46 12 75 42
```

Removing elements from the list

Python provides the **remove()** function which is used to remove the element from the list.

```
list = [0,1,2,3,4]
print("printing original list: ");
for i in list:
    print(i,end=" ")
list.remove(2)
print("\nprinting the list after the removal of first element...")
for i in list:
    print(i,end=" ")
```

Output:

printing original list:

0 1 2 3 4

printing the list after the removal of first element...

0 1 3 4

Python List Built-in functions

SN	Function	Description	Example
1	<code>cmp(list1, list2)</code>	It compares the elements of both the lists.	This method is not used in the Python 3 and the above versions.
2	<code>len(list)</code>	It is used to calculate the length of the list.	<code>L1 = [1,2,3,4,5,6,7,8] print(len(L1)) 8</code>
3	<code>max(list)</code>	It returns the maximum element of the list.	<code>L1 = [12,34,26,48,72] print(max(L1)) 72</code>
4	<code>min(list)</code>	It returns the minimum element of the list.	<code>L1 = [12,34,26,48,72] print(min(L1)) 12</code>
5	<code>list(seq)</code>	It converts any sequence to the list.	<code>str = "Johnson" s = list(str) print(type(s)) <class list></code>

Write the program to remove the duplicate element of the list.

```
list1 = [1,2,2,3,55,98,65,65,13,29]
# Declare an empty list that will store unique values
list2 = []
for i in list1:
    if i not in list2:
        list2.append(i)
print(list2)
```

Output:

```
[1, 2, 3, 55, 98, 65, 13, 29]
```

Write a program to find the sum of the element in the list.

```
list1 = [3,4,5,9,10,12,24]
sum = 0
for i in list1:
    sum = sum+i
print("The sum is:",sum)
```

Output:

The sum is: 67

Write the program to find the lists consist of at least one common element.

```
list1 = [1,2,3,4,5,6]
```

```
list2 = [7,8,9,2,10]
```

```
for x in list1:
```

```
    for y in list2:
```

```
        if x == y:
```

```
            print("The common element is:",x)
```

Output:

The common element is: 2