



List

Agenda

1

List

2

Accessing and updating list items

3

Slicing

4

Add, Delete, Remove, Clear operations

5

for loop with list

6

in keyword with list

List



List

- List is almost similar to array in C/C++/Java.
- It maintains the insertion order and supports index based access.
- It's **mutable**, elements can be added or removed from the original list.
- It can contain elements of different types.
- Elements are enclosed in square brackets [].

Examples:

```
li_1 = [1, 2]
```

```
li_2 = ['abc', 'xyz']
```

```
li_3 = [100, '@', 200, '$']
```

Accessing and updating list items



Accessing and updating list items

- List items are accessed and updated using index numbers.
- Index starts from 0.

Program:

```
          #      0          1          2
mylist = ["India", "Canada", "France"]
print(mylist[1]) #printing second item
mylist[2] = 'Mexico' #updating 2 index
print(mylist)
```

Output:

```
Canada
['India', 'Canada',
'Mexico']
```

Accessing and updating list items

- List items can be accessed and updated with negative indexing also.
- Index starts from -1 from the last element.

Program:

```
          #   -3           -2           -1
mylist = ["India", "Canada", "France"]
print(mylist[-1]) #printing last item
mylist[-3] = 'Chicago' #updating -3 index
print(mylist)
```

Output:

```
France
['Chicago', 'Canada',
'France']
```

Slicing




Slicing

- We can extract a set of items from a list by mentioning the start and end indexes.
- While specifying this start and end range, the returned value will be a new list with the specified items.

Program:

```
          #  0           1           2           3           4           5
mylist = "India", "Canada", "France", "Mexico", "London", "Ireland"]
print(mylist[2:5])
```



Output:

```
['France', 'Mexico', 'London']
```

Ending index is excluded.

Slicing

Negative slicing:

Program:

```
          # -6      -5      -4      -3      -2      -1
mylist = "India", "Canada", "France", "Mexico", "London", "Ireland"]
print(mylist[-6:-3])
```

Output:

```
['India', 'Canada', 'France']
```

Add, Delete, Remove, Clear operations



Add, Delete, Remove, Clear operations

Program:

```
mylist = ['A', 'B', 'C']
```

```
mylist.append('D') #Appending 'D' at the end
```

```
del mylist[-1] #delete 'D' by mentioning the index
```

```
mylist.remove('A') #delete 'A' by mentioning the item
```

```
mylist.clear() #empties the list
```

```
del mylist #deleting the list itself
```

Other important functions

Function syntax	Description
<code>list_name.copy()</code>	Returns a copy of the list.
<code>list_name.count(item)</code>	Return the number of times the specified item appears in the list.
<code>list_name.insert(index, item)</code>	Adds the item at the specified index.
<code>list_name.pop()</code>	Removes the last item from the list.
<code>list_name.reverse()</code>	Reverses the order of items in the list.
<code>list_name.sort()</code>	Sorts the list items in ascending order by default.
<code>sum(list_name)</code>	Returns the sum of items.
<code>max(list_name)</code>	Returns the maximum number from the list.
<code>min(list_name)</code>	Returns the minimum number from the list.

for loop with list



for loop with list

You can loop through the list items using a for loop:

Program:

```
mylist = ["India", "Canada", "France"]  
for item in mylist:  
    print(item)
```

Output:

```
India  
Canada  
France
```

in keyword with list



in keyword with list

To determine if a specified item is present in the list use the in keyword:

Program:

```
mylist = ["India", "Canada", "France"]  
if "Canada" in mylist:  
    print("Present")  
else:  
    print("Not Present")
```

Output:

Present



Thank you