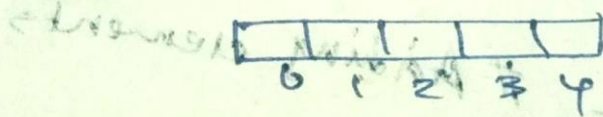


List

- ⇒ List is used to store sequence of data - (It is a collection of elements)
- ⇒ List is heterogeneous & it can contain different datatypes like int, string etc.
- ⇒ The elements in a list are indexed.
- ⇒ First index is 0.



- ⇒ List allows duplicating of elements.

because each element has a definite place in the list.

- ⇒ List = [1, 2, 3]

- ⇒ List doesn't need a built-in function to create a list. (unlike sets)

append() method

- ⇒ It is used to add elements to the list.
- ⇒ It adds elements at the end of the list.
- ⇒ Tuples can be added list using append.
- ⇒ List can be added to another list using append.

eg: - # Adding elements

```
list1 = [1, 2]
```

```
Print("Initial list:", list1)
```

```
list1.append(3)
```

```
list1.append(4)
```

```
list1.append(5)
```

```
Print("Final list:", list1)
```


Adding elements using iterator

for i in range(1,4):

list.append(i)

print("list after adding elements:", list)

Adding tuple to list

list = [1, 2, 3]

list.append((5, 6))

print("list after adding tuple:", list)

Adding list to list

list1 = [1, 2, 3]

list2 = [4, 5, 6]

list1.append(list2)

print("list1 after adding list2:", list1)

insert method

→ It adds elements at the desired

position

⇒ It requires two arguments

(position, value)

eg:-

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

```
print("Initial list:", list)
```

```
list.insert(1, 6)
```

```
print("Final list:", list)
```

output

Final list = [1, 6, 2, 3, 4]

⇒ list = [1, 2, 3, 4]

list.insert(0, "internet")

print(list)

output

[internet, 1, 2, 3, 4]

extend() method

⇒ It is used to add multiple elements at the same time at the end of the list.

eg:-

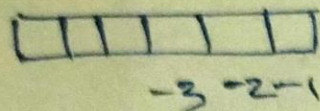
list = [1, 2, 3, 4]

print("initial list:", list)

list.extend(0, 'IoT', 'Bigdata')

print(list)

Negative indexing



list = [1, 2, 3]

print(list[-1])

output: 3

remove() method

=> It is used to remove element from the list.

=> It removes first occurrence of the searched element.

eg:- list = [1, 2, 3, 4, 5, 6, 7]

print("Initial list", list)

list.remove(3)

print(list)

output: Initial list - [1, 2, 3, 4, 5, 6, 7]
[1, 2, 4, 5, 6, 7]

pop() method

⇒ It is used to remove & return an element.

⇒ By default it removes last element.

⇒ can remove element ^{from} a specific position in the list, by passing ~~arguments~~ the index of element as argument in pop().

eg: - List = [1, 2, 3, 4], 1 = tail

List.pop()

print("List:", List)

(tail).print

output

List: [1, 2, 3]

=> List = [1, 2, 3, 4, 5, 6]

List.pop(2)

print("List:", List)

output

List: [1, 2, 4, 5, 6]

Slicing

=> slice operation is used to print a specific range of element from list.

=> `len()` is used for slicing.

- `[: index]` - from beginning to a range
- `[: - index]` - from ~~end~~ ^{begin} to ~~range~~ ^{indexed} position
- `[index :]` - specific index to end
- `[start index : end index]`
- `[:]` - to print whole list
- `[:: -1]` - to reverse a list.

eg:- List = ['a', 'e', 'e', 'k', 's', 'f', 'o', 'r']

Print("Initial list:", List)

sliced_list = List[3:8]

print("list after slicing:", sliced_list)

Initial list: ['a', 'e', 'e', 'k', 's', 'f', 'o', 'r',
'a', 'e', 'e', 'k', 's']

list after slicing: ['k', 's', 'f', 'o', 'r']

sort list

=> uses `sort()` method

eg:- `list1 = [6, 11, 1, 5]`

`list1.sort()`

`print(list1)`

output

`[1, 5, 6, 11]`

reverse list

=> use `reverse()`

- `list.reverse()`

copy list

=> `thislist = ["apple", "banana"]`

`mylist = thislist.copy()`

`print(mylist)`

clear()

Removes all elements in the list

count()

[2, 1, 1, 1, 1] = list

number of elements - (2) two 1s

(1) one 2

[1, 1, 1, 1, 1]

list