

Python Lists

Ordering, Indexing and Finding the Number of Objects

Key Point: A Python list maintains the order of its objects and allows individual objects to be accessed using indexes.

1. Lists Are Ordered

A list maintains the order of its objects. Python does not automatically arrange the objects in numerical or alphabetical order.

Consider:

```
numbers = [30, 10, 50, 20, 40]
```

```
print(numbers)
```

```
[30, 10, 50, 20, 40]
```

The order in which we place the objects is retained.

Key Point: [30, 10, 50, 20, 40] is different from [10, 20, 30, 40, 50].

2. Accessing Objects in a List

A very important property of a Python list is that we can access individual objects using their index.

Consider:

```
numbers = [10, 20, 30, 40, 50]
```

The positions are:

Object:	10	20	30	40	50
Positive:	0	1	2	3	4
Negative:	-5	-4	-3	-2	-1

Notice that Python starts indexing from 0, not 1.

Therefore, numbers[0] means the first object, numbers[1] means the second object, and numbers[2] means the third object.

3. Example: Accessing Individual Objects

```
numbers = [10, 20, 30, 40, 50]
```

```
print(numbers[0])  
print(numbers[1])  
print(numbers[2])
```

Output:

```
10  
20  
30
```

We can also access the last two objects:

```
print(numbers[3])  
print(numbers[4])
```

Output:

```
40  
50
```

Key Point: The index determines which object is accessed. Index 0 refers to 10, while index 4 refers to 50.

4. Negative Indexing

Python also allows us to access objects from the end of a list using negative indexes.

For example:

```
numbers = [10, 20, 30, 40, 50]
```

The negative index -1 refers to the last object, -2 refers to the second-last object, and so on.

Therefore:

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

Output:

```
50
```

And:

```
print(numbers[-2])
```

Output:

```
40
```

Key Point: Negative indexing is useful when we want to access objects from the end of a list.

5. Finding the Number of Objects in a List

Python provides the len() function to find the number of objects in a list.

Example:

```
numbers = [10, 20, 30, 40, 50]  
  
print(len(numbers))
```

Output:

```
5
```

Therefore, the list contains five objects.

Another example:

```
names = ["Ram", "John", "Ali"]  
  
print(len(names))
```

Output:

```
3
```

Key Point: The len() function returns the number of objects present in a list.

6. Quick Revision

- **Lists are ordered:** A list retains the order in which its objects are placed.
- **Positive indexing:** Indexing starts from 0. The first object has index 0.
- **Negative indexing:** Index -1 refers to the last object, -2 to the second-last object, and so on.
- **len() function:** len(list) returns the number of objects in the list.

Key Point: Remember: Python lists preserve order, use zero-based positive indexing, support negative indexing, and can be counted using len().

