

Python List - Complete Explanation

A list in Python is an ordered, mutable collection of items. It allows duplicate elements and can store multiple data types.

Basic Syntax:

```
my_list = [1, 2, 3, 'Python', 4.5]
```

Key Features:

- Ordered and indexed
- Mutable (can be changed)
- Allows duplicates
- Can hold different data types

Creation Methods:

1. Using `[]`: `my_list = [1, 2, 3]`
2. Using `list()`: `my_list = list((1, 2, 3))`
3. Empty list: `my_list = []`

Accessing Elements:

```
print(my_list[0])  
print(my_list[-1])
```

Slicing:

```
print(my_list[1:4])  
print(my_list[:3])  
print(my_list[::-1]) # Reverse
```

Adding Elements:

```
append(), insert(), extend()
```

Example:

```
my_list.append(6)  
my_list.insert(2, 'Hello')  
my_list.extend([7, 8])
```

Removing Elements:

```
remove(), pop(), del, clear()
```

Example:

```
my_list.remove('Hello')  
my_list.pop(1)  
del my_list[0]  
my_list.clear()
```

Updating Elements:

```
my_list[1] = 'Updated'
```

Looping Through List:

```
for item in my_list:  
    print(item)
```

Common Functions:

```
len(), min(), max(), sum(), sorted(), reversed()
```

Common Methods:

```
append(), extend(), insert(), remove(), pop(), index(), count(), sort(), reverse(), copy(), clear()
```

List Comprehension:

```
squares = [x**2 for x in range(5)]
```

Nested Lists:

```
matrix = [[1, 2], [3, 4]]  
print(matrix[0][1]) # Output: 2
```

Concatenation & Repetition:

```
a = [1, 2] + [3, 4]  
b = [0] * 4
```

Membership:

```
print(3 in [1, 2, 3]) # True
```

Example - Filtering Even Numbers:

```
nums = [1, 2, 3, 4, 5, 6]  
evens = [x for x in nums if x % 2 == 0]  
print(evens)
```

Summary:

Create: `l = [1, 2, 3]`

Access: `l[0]`

Add: `l.append(4)`

Remove: `l.remove(2)`

Update: `l[1] = 10`

Loop: `for x in l:`

Sort: `l.sort()`

Copy: `l.copy()`

Clear: `l.clear()`