

◆ Python List – Introduction

Python में **list** एक **ordered, mutable, heterogeneous** data structure होता है।

List को square brackets (`[]`) में लिखा जाता है:

python

CopyEdit

```
my_list = [1, 2, 3, "hello", 4.5]
```

यह एक **dynamic array** की तरह काम करता है (C/C++ की तरह fixed size array नहीं होता)।

◆ Key Features of List

Feature	Description
Ordered	Elements insertion order maintained
Mutable	Elements को change कर सकते हैं
Heterogeneous	अलग-अलग type के elements रख सकते हैं
Dynamic	Size को runtime पर change कर सकते हैं

◆ Internal Working of List in Python

✓ 1. List is an Object (PyObject)

हर list Python में एक object होता है जिसे internally `PyListObject` से represent किया जाता है (CPython में):

c

CopyEdit

```
struct PyListObject {  
    PyObject_VAR_HEAD  
    PyObject **ob_item; // pointer to array of pointers  
    Py_ssize_t allocated; // allocated memory slots  
};
```

मतलब:

- `ob_item` एक pointer होता है actual list elements के लिए
- Python internally list को एक **array of pointers** की तरह रखता है

✓ 2. Dynamic Resizing

Python list में जब आप `append()` करते हैं, तो यह automatic memory allocate करता है:

python

CopyEdit

```
lst = []  
lst.append(10)  
lst.append(20)
```

🔗 **CPython implementation** extra space allocate करता है future appends के लिए।

For example:

Items Inserted	Allocated Memory
0	0
1	4
5	9
9	16
...	Exponentially increases

✦ इसलिए `append()` बार-बार करने पर भी list efficient रहती है।

✓ 3. List is Mutable

List के elements को हम change कर सकते हैं:

python

CopyEdit

```
lst = [1, 2, 3]
```

```
lst[0] = 100
```

```
print(lst) # [100, 2, 3]
```

📝 लेकिन ध्यान रहे, ये सिर्फ references change करता है — original data नहीं copy करता।

✓ 4. List Stores References (Not actual objects)

python

CopyEdit

```
a = [1, 2]
```

```
b = a
b[0] = 100
print(a) # [100, 2]  ✓ Same object
```

✓ इसका मतलब list shallow copy होती है, actual data नहीं copy होता।

✓ 5. Garbage Collection & Reference Count

Python में जब किसी list का reference count 0 हो जाता है, तो Garbage Collector उसे memory से हटा देता है:

python

CopyEdit

```
a = [1, 2, 3]
a = None # अब पुरानी list unreachable है, GC उसे हटा देगा
```

✓ 6. Common List Methods

Method	Description
<code>append(x)</code>	Add item at end
<code>insert(i, x)</code>	Insert at index
<code>extend(iterable)</code>	Add multiple values
<code>remove(x)</code>	Remove first occurrence

<code>pop([i])</code>	Remove and return element at index
<code>sort()</code>	Sort list (in-place)
<code>reverse()</code>	Reverse list
<code>copy()</code>	Shallow copy
<code>clear()</code>	Remove all elements

◆ Diagram: Internal Memory Representation

python

CopyEdit

```
a = [10, 20, 30]
```

Internally:

```
a --> [Pointer to obj1] --> 10
      [Pointer to obj2] --> 20
      [Pointer to obj3] --> 30
```

✓ हर list सिर्फ object का **reference** store करती है, actual object नहीं।

◆ Performance Tips

Operation	Time Complexity
Index access	$O(1)$

Append Amortized $O(1)$

Insert/Delete at middle $O(n)$

Iteration $O(n)$

✦ अगर आपको fast insertion/deletion चाहिए तो `deque` (from `collections`) का use करें।

◆ Summary Table

Feature	Value
Type	list
Mutable	Yes
Ordered	Yes
Stores	References to objects
Memory	Dynamic array (resizable)
Implementation	PyListObject in CPython

✅ Bonus Tools to Inspect

python

CopyEdit

```
import sys
```

```
lst = [1, 2, 3]
```

```
print(sys.getsizeof(lst))                      # Memory size in bytes
```

```
print(lst.__sizeof__())                        # Without garbage overhead
```

◆ Conclusion:

Python list एक flexible और powerful data structure है जो internally dynamic array की तरह काम करता है और references को manage करता है।