Everything about Python List

Creating

```
# Empty list
my_list = []

# 1D
numbers = [1, 2, 3, 4, 5]

# 2D
num = [1, 2, 3, [4, 5]]

# Mixed data types
mixed = [1, "hello", 3.14]

# Using list() function
new_list = list((1, 2, 3))
```

Accessing Elements

```
# Indexing
print(numbers[0])

# Negative Indexing
print(numbers[-1])

# Slicing
print(numbers[1:4])
```

Editing a List

```
# Changing element
numbers[2] = 10

# Adding elements
numbers.append(6)       # Adds at end
numbers.insert(2, 99)   # Inserts at index 2

# Extending list
numbers.extend([7, 8, 9])
```

Deleting a List

```
# Removing by value
numbers.remove(99)

# Removing by index
del numbers[2]

# Using pop (returns removed element)
last_item = numbers.pop()

# Clearing all elements
numbers.clear()
```

List Operations

```
# Concatenation
new_list = [1, 2] + [3, 4]

# Repetition
repeat_list = [Hi] * 5

# Membership check
print(3 in new_list)  # True

# Iterating
for item in numbers:
    print(item)
```

Two Ways to Traverse a List

a) Item-wise

```
fruits = ["apple", "banana", "cherry"]

for fruit in fruits:            apple
    print(fruit)                banana
                                cherry
```

b) Index-wise

```
fruits = ["apple", "banana", "cherry"]

for i in range(len(fruits)):
    print(f"Index {i}: {fruits[i]}")
           Index 0: apple
           Index 1: banana
           Index 2: cherry
```

List Comprehension

new_list = [Expression for item in iterable if condition == True]

List comprehension is a concise way to create lists using a single line of code.

```
# Square of numbers from 1 to 5
squares = [x**2 for x in range(1, 6)]
print(squares)                          # [1, 4, 9, 16, 25]
```

zip()

Pairs elements from multiple lists.

```
names = [‘Rudra’, ‘Python’]
ages = [18, 34]

for names, ages in zip(names, ages):
    print(f’{names} is {ages} yr old.’)

# Rudra is 18 yr old
```

map()

Applies a function to all elements in an iterable.

```
# Convert list of strings to uppercase
w = ["hello", "world"]
uup_w= list(map(str.upper, words))
print(up_w)

# ['HELLO', 'WORLD']
```

List Functions

When/why	Function	Input	Output
Add elements	append(x)	lst.append(5)	[1, 2, 3, 4, 5]
Insert at index	insert(i, x)	lst.insert(2, 9)	[1, 2, 9, 3, 4]
Merge lists	extend(lst2)	lst.extend([5,6])	[1,2,3,4,5,6]
Remove by value	remove(x)	lst.remove(3)	[1, 2, 4, 5]
Remove by index	pop(i)	lst.pop(2)	Returns removed item
Find index	index()	lst.index(2)	1 (index of 2)
Count occurrences	count()	lst.count()	1 (occurrences)
Sort list	sort()	lst.sort()	[1, 2, 3, 4]
Reverse list	reverse()	lst.reverse()	[4, 3, 2, 1]
copy list	copy()	new_lst = lst.copy()	Copy of list
Clear all	clear()	lst.clear()	[] (empty list)
Length	len()	len([1, 2, 3])	3
Minimum/ Maximum	min(), max()	min([1, 2, 3) , max([1, 2, 3)	1, 3

Looking for more valuable content ? Follow me

Rudra Prasad Bhuyan

Ai | Business | Creative | Data