

🔗 Lists in Python

🔗 1. What is a List?

A list is a collection of items that are **ordered**, **mutable** (changeable), and **allow duplicate elements**. Lists can hold items of different data types, such as integers, strings, or even other lists.

🔗 Syntax:

```
my_list = [element1, element2, ...]
```

🔗 Example:

```
fruits = ["apple", "banana", "orange"]  
numbers = [1, 2, 3, 4, 5]  
mixed = ["apple", 3, True]
```

🔗 2. Accessing List Elements

You can access individual elements in a list using **indexing**. Remember that Python uses **zero-based indexing**, so the first item is at index 0.

🔗 Syntax:

```
list_name[index]
```



🔗 Example:

```
fruits = ["apple", "banana", "orange"]  
print(fruits[0]) # Output: apple  
print(fruits[2]) # Output: orange
```



You can also use **negative indexing** to access elements from the end of the list:

```
print(fruits[-1]) # Output: orange  
print(fruits[-2]) # Output: banana
```



🔗 3. Modifying Lists

Lists are mutable, which means you can change the value of items in a list.

🔗 Changing a specific element:

```
fruits[1] = "orange"
print(fruits) # Output:
```



🔗 Adding elements:

- `append()` : Adds an element to the **end** of the list.

```
fruits.append("grape")
print(fruits) # Outp
```



- `insert()` : Inserts an element at a **specific index**.

```
fruits.insert(1, "kiw")
print(fruits) # Outp
```



Removing elements:

- `remove()` : Removes the first occurrence of an element.

```
fruits.remove("orange")  
print(fruits) # Outp
```

- `pop()` : Removes the element at a specific index (or the last item if no index is provided).

```
fruits.pop() # Remove last element  
print(fruits) # Outp
```

```
fruits.pop(0) # Remove first element  
print(fruits) # Outp
```


- `clear()` : Removes all elements from the list.

```
fruits.clear()  
print(fruits) # Outp
```

🔗 4. Slicing Lists


You can extract a portion of a list using **slicing**.

🔗 Syntax:

```
list_name[start:stop:step] 
```

- **start** : The index to start the slice (inclusive).
- **stop** : The index to stop the slice (exclusive).
- **step** : The number of steps to skip elements (default is 1).

🔗 Examples:


```
numbers = [0, 1, 2, 3, 4,   
print(numbers[1:4]) # Out  
print(numbers[:4]) # Out  
print(numbers[2:]) # Out  
print(numbers[::2]) # Ou
```

5. List Functions and Methods


Python provides several built-in functions and methods for working with lists.

5.1 Common Functions:


- `len(list)` : Returns the number of elements in the list.

```
print(len(fruits)) # 
```

- `sorted(list)` : Returns a new sorted list without changing the original list.

```
numbers = [5, 2, 9, 1]   
print(sorted(numbers))  
print(numbers) # Ori
```

- `sum(list)` : Returns the sum of elements in a list (for numerical lists).

```
numbers = [1, 2, 3, 4]   
print(sum(numbers))
```

🔗 5.2 Common Methods:

- `index(element)` : Returns the index of the first occurrence of the specified element.

```
print(fruits.index("a")
```

- `count(element)` : Returns the number of occurrences of an element in the list.

```
numbers = [1, 2, 3, 1]  
print(numbers.count(1
```

- `reverse()` : Reverses the elements of the list in place.

```
fruits.reverse()  
print(fruits) # Outp
```

- `sort()` : Sorts the list in place (ascending by default).

```
numbers = [5, 2, 9, 1]  
numbers.sort()  
print(numbers) # Out
```

🔗 6. Nested Lists

Lists can contain other lists, allowing you to create **nested lists**. This can be useful for storing matrix-like data structures.

🔗 Example:

```
matrix = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]
```



```
# Accessing elements in a  
print(matrix[0]) # Output: [1, 2, 3]  
print(matrix[1][1]) # Output: 5
```


Homework

1. List Manipulation Exercise:

- Create a list of 5 items (strings or numbers).
- Add a new item to the end of the list and another at the second position.
- Remove the third item from the list.
- Print the list after each operation.

2. Reverse and Sort a List: Create a list of numbers and:

- Sort it in descending order.
 - Reverse the sorted list and print it.
-