1. List in Python

A list in Python is a versatile and mutable collection data type that can hold a mixture of different data types, including integers, floats, strings, and even other lists. Unlike arrays in other languages, Python lists are dynamic and can resize themselves, making them more akin to arrays in Java or vectors in C++.

Operations of list

1. Creating a List

In [1]:
```
sample_list = [1, 2, 3, 4, 5]
print("Original List:", sample_list)
```

```
Original List: [1, 2, 3, 4, 5]
```

2. Adding an Element to the End of the List (Append)

In [3]:
```
sample_list.append(6)
print("\nList After Append:", sample_list)
```

```
List After Append: [1, 2, 3, 4, 5, 6, 6]
```

3. Adding an Element at a Specific Position (Insert)

In [4]:
```
sample_list.insert(2, 2.5)  # This inserts 2.5 at the 3rd position
print("\nList After Insert:", sample_list)
```

```
List After Insert: [1, 2, 2.5, 3, 4, 5, 6, 6]
```

4. Removing an Element from the List (Remove)

In [5]:
```
sample_list.remove(2.5)  # Removes the first occurrence of 2.5
print("\nList After Remove:", sample_list)
```

```
List After Remove: [1, 2, 3, 4, 5, 6, 6]
```

5. Accessing Elements from the List

In [6]:
```
print("\nThird Element in List:", sample_list[2])
```

```
Third Element in List: 3
```

6. Slicing a List

In [7]:
```
print("\nFirst Three Elements:", sample_list[:3])
```

```
First Three Elements: [1, 2, 3]
```

7. List Comprehension (Generating a new list by performing operations)

In [9]:
```
squared_list = [x**2 for x in sample_list]
print("\nSquared List:", squared_list)
```

```
Squared List: [1, 4, 9, 16, 25, 36, 36]
```

2. Tuple in Python

Tuples in Python are immutable collections used to group together multiple items. Defined using parentheses (), they can hold different data types. Unlike lists, once created, their content cannot be changed. Tuples are typically used for data that shouldn't be altered.

Tuple Operations in Python

1. Creating a Tuple

In [10]:
```
sample_tuple = (1, 2, 3, 4, 5)
print("Original Tuple:", sample_tuple)
```

```
Original Tuple: (1, 2, 3, 4, 5)
```

2. Accessing Elements from the Tuple

In [11]:
```
print("\nThird Element in Tuple:", sample_tuple[2])
```

```
Third Element in Tuple: 3
```

3. Slicing a Tuple

In [12]:
```
print("\nFirst Three Elements:", sample_tuple[:3])
```

```
First Three Elements: (1, 2, 3)
```

4. Counting Occurrences of a Value

In [12]:
```
count_twos = sample_tuple.count(2)
print("\nNumber of Occurrences of 2:", count_twos)
```

```
Number of Occurrences of 2: 1
```

5. Finding the Index of a Value

In [13]:
```
index_of_three = sample_tuple.index(3)
print("\nIndex of Value 3:", index_of_three)
```

```
Index of Value 3: 2
```

In [14]:
```
# Note: Tuples are immutable, so operations like append, insert, or re
```

In [ ]:
```

```