List in Python

In [4]:

```
my_list = [1, 2, 3, 4, 5]
print(my_list)
print(type(my_list))
```

```
[1, 2, 3, 4, 5]
<class 'list'>
```

In [6]:

```
a = [1, "hello", 3.5, True]
print(a)
print(type(a))
```

```
[1, 'hello', 3.5, True]
<class 'list'>
```

Empty List

In [9]:

```
b = []
print(b)
print(type(b))
```

```
[]
<class 'list'>
```

List Constructor

In [17]:

```
c = list((1, 2, 3))
print(c)
print(type(c))
```

```
[1, 2, 3]
<class 'list'>
```

Properties of list

1. Ordered

In [22]:

```
fruits = ["mango", "orange", "apple"]

print(fruits[0])
print(fruits[1])
print(fruits[2])
```

```
mango
orange
apple
```

2. Heterogeneous

Lists can contain elements of different data types, such as integers, strings, and even other lists.

In [25]:

```
mixed_list = [1, "hello", 3.14, [1, 2, 3]]

print(mixed_list[0])
print(mixed_list[1])
print(mixed_list[2])
print(mixed_list[3])
```

```
1
hello
3.14
[1, 2, 3]
```

3. Indexed

Elements in a list can be accessed by their index, which starts at 0. You can also use negative indices to access elements from the end of the list.

In [29]:

```
# Example list
animals = ["cat", "dog", "bird", "fish"]

# Accessing elements by index
print(animals[0])  # Output: cat
print(animals[2])  # Output: bird

# Accessing elements with negative index
print(animals[-1])  # Output: fish
print(animals[-3])  # Output: dog
```

```
cat
bird
fish
dog
```

Slicing of List

In [32]:

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
subset = numbers[1:5]
print(subset)  # Output: [1, 2, 3, 4]
```

```
[1, 2, 3, 4]
```

In [42]:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
subset = numbers[0:8:2]
print(subset)
```

```
[1, 3, 5, 7]
```

Using Negative Indices

In [45]:

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

subset = numbers[-5:]
print(subset)  # Output: [5, 6, 7, 8, 9]

subset = numbers[:-3]
print(subset)  # Output: [0, 1, 2, 3, 4, 5, 6]
```

```
[5, 6, 7, 8, 9]
[0, 1, 2, 3, 4, 5, 6]
```

List Methods

List Length

In [49]:

```
a = [12,26,2.36,"books", "fruits", 2.36,78,-96,-2, True, False]
print(len(a))
```

```
11
```

mutability property of lists

Changing Elements:

You can change the value of elements within a list using indexing.

In [53]:

```
my_list = [1, 2, 3, 4, 5]
my_list[2] = 10
print(my_list)  # Output: [1, 2, 10, 4, 5]
```

```
[1, 2, 10, 4, 5]
```

Change a Range of Item Values

In [56]:

```
a = [12,14,15,16,8,9,-5,-3,41,-9]
a[2:4] = ["books","pen","copy"]
print(a)
```

```
[12, 14, 'books', 'pen', 'copy', 8, 9, -5, -3, 41, -9]
```

Add List Items

In [59]:

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

```
['apple', 'banana', 'cherry', 'orange']
```

In [61]:

```python
m = [12,23,45,69,23,47,8,5]
m.append(-20)
print(m)
```

```
[12, 23, 45, 69, 23, 47, 8, 5, -20]
```

Remove List Items

In [64]:

```python
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

```
['apple', 'cherry']
```

Join List

In [67]:

```python
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
print(list3)

list4 = list2 + list1
print(list4)
```

```
['a', 'b', 'c', 1, 2, 3]
[1, 2, 3, 'a', 'b', 'c']
```

Check if a list contains an element

In [70]:

```python
li = [1,2,3,'a','b','c']
'a' in li
```

Out[70]:

```
True
```

In [72]:

```python
li = [1,2,3,'a','b','c']
5 in li
```

Out[72]:

```
False
```

Reversing a List

In [75]:

```python
mylist = [1, 2, 3, 4, 5, 'IHH', 'Python']
mylist.reverse()
print(mylist)
```

```
['Python', 'IHH', 5, 4, 3, 2, 1]
```

Practice Question

In [ ]:

```
Data Analysis Scenario:
You have two lists containing exam scores
of students from two different classes.
The lists are class1_scores and class2_scores.
You need to analyze the performance
of the classes by calculating the average score for each class.

Write Python code to calculate the average score for each class.
```

In [78]:

```
# Given lists of exam scores for two classes
class1_scores = [85, 90, 88, 92, 78]
class2_scores = [75, 82, 80, 85, 79]

# Calculate the average score for class 1
class1_average = sum(class1_scores) / len(class1_scores)

# Calculate the average score for class 2
class2_average = sum(class2_scores) / len(class2_scores)

# Print the average scores for each class
print("Average score for Class 1:", class1_average)
print("Average score for Class 2:", class2_average)
```

```
Average score for Class 1: 86.6
Average score for Class 2: 80.2
```

In [ ]:

```
Question: Given the list numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
write Python expressions to:
Create a new list with the first half and the second half swapped.
Print the sum of the last three elements.
```

In [80]:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


half_length = len(numbers) // 2
swapped_list = numbers[half_length:] + numbers[:half_length]
print(swapped_list)  # Output: [6, 7, 8, 9, 10, 1, 2, 3, 4, 5]

# Print the sum of the last three elements
sum_last_three = sum(numbers[-3:])
print(sum_last_three)  # Output: 27
```

```
[6, 7, 8, 9, 10, 1, 2, 3, 4, 5]
27
```

In [ ]:

```
Find the Second Largest Number in a List
```

In [82]:

```
def second_largest(lst):
    unique_lst = list(set(lst))  # Remove duplicates
    unique_lst.sort()
    return unique_lst[-2]  # Second last element

numbers = [10, 20, 4, 45, 99, 45]
print(second_largest(numbers))  # Output: 45
```

```
45
```

In [ ]:

```
Task: Data Cleaning
You have a list responses containing responses
from a survey. However, some responses are recorded
```

```
as empty strings due to data entry errors.
Your task is to remove these empty responses from the list.

Write Python code to remove all
empty responses from the responses list
```

In [84]:

```
# Given list of responses
responses = ["Yes", "", "No", "", "Maybe", "", "","Yes","No",""]

# Create an empty list to store non-empty responses
clean_responses = []

# Iterate over each response in the responses list
for response in responses:
    # Check if the response is not empty
    if response != "":
        # Add non-empty response to the clean_responses list
        clean_responses.append(response)

# Print the cleaned responses
print(clean_responses)
```

```
['Yes', 'No', 'Maybe', 'Yes', 'No']
```

In [ ]: