

List Practice Questions:

1. How do you create an empty list?

⇒ To create empty list, we have different methods:

```
l1 = []  
l2 = list()  
l3 = [x for x in []]  
l4 = list(range(0))  
print(type(l1))  
print(type(l2))  
print(type(l3))  
print(type(l4))
```

Output:

```
<class 'list'>  
<class 'list'>  
<class 'list'>  
<class 'list'>
```

2. How do you add an element to a list using indexing?

⇒ To add elements in the list using indexing:

```
l = [10,20,30,40,50]  
l[2] = 100  
print(l)
```

But the index should be already present in the list.

3. How do you access the first element of a list?

⇒ Passing 0 as index:

```
l = [10,20,30,40,50]  
print(l[0])
```

⇒ 10

4. How can you replace an element at a specific index in a list?

⇒ `my_list[index] = new_value`

5. How can you access the last element of a list without knowing its length?

⇒ `last_element = my_list[-1]`

Example:

```
l = [10,20,30,40,50]
print(l[-1])
```

⇒ 50

6. How do you access a slice of a list (e.g., the first three elements)?

Slicing can be done through both positive index and negative index of elements.

Some Examples of List Slicing:

```
my_list = [10,20,"Kabin", "Shyam", 10.20, 5.5, [90,80,70]]
```

```
#positive slicing
```

```
sliced_list = my_list[1:4]
print(sliced_list)
```

```
print(my_list[6:])
```

```
#negative slicing
```

```
print(my_list[-3:])
```

```
#we can also reverse list using slicing
```

```
reversed_list = my_list[::-1]
```

```
print(reversed_list)

#using steps:
print(my_list[1:-1:2])
```

Output:

```
[20, 'Kabin', 'Shyam']
[[90, 80, 70]]
[10.2, 5.5, [90, 80, 70]]
[[90, 80, 70], 5.5, 10.2, 'Shyam', 'Kabin', 20, 10]
[20, 'Shyam', 5.5]
```

7. How can you concatenate two lists?

Lists can be concatenated using addition (+) operator:

```
l1 = [10,20,30,40]
l2 = [50,60,70,80]
l3 = l1 + l2
print(l3)

⇒ [10, 20, 30, 40, 50, 60, 70, 80]
```

Also we can use extend() method

```
l1 = [10,20,30,40]
l2 = [50,60,70,80]
# l3 = l1 + l2
l1.extend(l2)
print(l1)

⇒ [10, 20, 30, 40, 50, 60, 70, 80]
```

8. How can you check if an element is in a list?

To check if an element is in a list, we can use membership operator: in , if element is present it returns true, if not it returns false:

Example:

```
my_list = [10, 20, 30, 40, 50]
element = 30

if element in my_list:
    print(f"{element} is present in the list.")
else:
    print(f"{element} is not present in the list.")

⇒ 30 is present in the list.
```

9. How can you count the number of elements in a list?

Simply using len() function

```
my_list = [10,20,"Kabin", "Shyam", 10.20, 5.5, [90,80,70]]
print(len(my_list))
```

⇒ 7

10. How do you reverse a list using indexing?

Syntax: reversed_list = my_list[::-1]

```
my_list = [10,20,"Kabin", "Shyam", 10.20, 5.5, [90,80,70]]

reversed_list = my_list[::-1]
print(reversed_list)
```

⇒ [[90, 80, 70], 5.5, 10.2, 'Shyam', 'Kabin', 20, 10]

11. How can you remove an element from a list by index?

We can remove an element from a list by index by using del keyword and pop() method.

Example:

```
my_list = [100, 200, 300, 400, 500]
```

```
del my_list[2]
print("After deletion using del:", my_list)

removed_element = my_list.pop(1)
print("After deletion using pop:", my_list)
print("Removed element using pop:", removed_element)
```

Output:

```
After deletion using del: [100, 200, 400, 500]
After deletion using pop: [100, 400, 500]
Removed element using pop: 200
```

12. How do you find the index of a particular element in a list?

We can find the index of a particular element in a list using the `index()` method. This method returns the first occurrence of the element we are searching for:

```
my_list = [100, 200, -30, -40, -50]

element = 200

index = my_list.index(element)
print(f"The index of {element} is: {index}")
```

⇒The index of 200 is: 1

If the element is not present in the list, the `index()` method raises a `ValueError`.

13. How do you check if a list is empty?

We have 3 ways to check if the list is empty or not.

- Method 1: if not my_list: (Most Pythonic and recommended)
- Method 2: if len(my_list) == 0:
- Method 3: if my_list == []:

```
my_list = []
```

```

# 1 Using if statement directly
if not my_list:
    print("Method 1: The list is empty.")
else:
    print("Method 1: The list is not empty.")

# 2 Using len() function
if len(my_list) == 0:
    print("Method 2: The list is empty.")
else:
    print("Method 2: The list is not empty.")

# 3 Comparing with an empty list []
if my_list == []:
    print("Method 3: The list is empty.")
else:
    print("Method 3: The list is not empty.")

my_list = [10,20]

if not my_list:
    print(" The list is empty.")
else:
    print(" The list is not empty.")

```

Output:

Method 1: The list is empty.
Method 2: The list is empty.
Method 3: The list is empty.
Method 1: The list is not empty.

14. How can you access every second element in a list?

We can access every second element in a list using list slicing with a step. To access every second element, we can set the step to 2. Here's an example:

```

my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
second = my_list[::2]
print("Every second element:",second)

```

⇒ Every second element: `[10, 30, 50, 70, 90]`

15. How can you access a list in reverse order without modifying it?

We can access lists in reverse order by simply using list slicing.

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
reversed_list = my_list[::-1]
print("Reversed list:", reversed_list)
```

Output:

```
Reversed list: [100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
```

16. How would you remove all occurrences of a specific element from a list?

There are different methods to remove all occurrences of specific elements from a list but list comprehension and using `remove()` method in a while loop are more often used.

Example:

```
my_list = ["Kabin", "Dipesh", "Isha", "Kabin", "Nikita", "Aayush",
           "Kabin"]
element = "Kabin"

new_list = [x for x in my_list if x != element]

print("List after removing all occurrences of", element, ":",
      new_list)
```

Output:

```
List after removing all occurrences of Kabin : ['Dipesh', 'Isha',
        'Nikita', 'Aayush']
```

17. Given two lists, how would you find the common elements between them?

To find the common elements between two lists, we can use list comprehension and by converting them to sets.

Using List Comprehension

```
list1 = [1, 2, 3, 4, 5]
```

```
list2 = [4, 5, 6, 7, 8]
```

```
common_elements = [x for x in list1 if x in list2]  
print("Common elements using list comprehension:",  
      common_elements)
```

Output:

Common elements using list comprehension: [4, 5]

Using Set Intersection

```
list1 = [1, 2, 3, 4, 5]  
list2 = [4, 5, 6, 7, 8]
```

```
common_elements = list(set(list1) & set(list2))  
  
print("Common elements using set intersection:", common_elements)
```

⇒ Common elements using set intersection: [4, 5]

18. How would you create a new list that contains only the unique elements from an existing list?

We can create a new list that contains only the unique elements from an existing list using several methods, but mostly we can do it by using sets or append() method in for loop.

Example:

```
my_list = [10, 20, 20, 30, 40, 40, 50, 50, 50]
```

```
unique_list = list(set(my_list))  
print("Unique elements using set():", unique_list)
```

```
new_list = []
```

```
for item in my_list:  
    if item not in new_list:  
        new_list.append(item)  
print("Unique elements using for loop:", unique_list)
```


Output:

```
Unique elements using set(): [40, 10, 50, 20, 30]
```

```
Unique elements using for loop: [40, 10, 50, 20, 30]
```

19. How would you flatten a list of lists into a single list (without using any built-in functions)?

We can achieve this by manually iterating through the list of lists and appending each element to a new list as follows:

```
lists = [[1, "Kabin"], ["Shyam", 4], [5, "Hari", "Gita"]]
```

```
flat_list = []
```

```
for sublist in lists:
    for item in sublist:
        flat_list.append(item)
```

```
print("Flattened list:", flat_list)
```

Output:

```
Flattened list: [1, 'Kabin', 'Shyam', 4, 5, 'Hari', 'Gita']
```

20. How would you split a list into two parts at a given index?

We can split by using the slicing technique.

```
my_list = [10, 20, 30, 50, 60, 70, 80]
index = 4;
first_part = my_list[:index]
second_part = my_list[index:]
print(first_part)
print(second_part)
```

Output:

```
[10, 20, 30, 50]
```

```
[60, 70, 80]
```

21. How would you rotate a list by n positions to the left?

To rotate a list by n positions to the left, we can split the list into two parts: the first n elements and the remaining elements. Then, we concatenate these two parts by placing the remaining elements first, followed by the first n elements.

```
my_list = [10,20,30,50,60,70,80]
n = 4;
first_part = my_list[:n]
second_part = my_list[n:]
print(second_part + first_part )
```

Output:

```
[60, 70, 80, 10, 20, 30, 50]
```

22. How would you check if a list is a palindrome?

To check if a list is palindrome or not, we can first reverse a list and if the reversed list and original list are the same, then we can say it is palindrome. We can use the reverse() method or use slicing to reverse the list.

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

if list(reversed(my_list)) == my_list:
# if mylist == mylist[::-1]
    print("The list is a palindrome.")
else:
    print("The list is not a palindrome.")
```

⇒ The list **is** palindrome.

```
my_list = [10, 20, 30, 20, -10]

if list(reversed(my_list)) == my_list:
# if mylist == mylist[::-1]
    print("The list is a palindrome.")
else:
    print("The list is not a palindrome.")
```

⇒ The list `is not` a palindrome.

23. How would you sort a list without using any built-in sort functions (e.g., using a sorting algorithm like bubble sort)?

Using Bubble sort to sort the list

```
my_list = [10, 20, 30, 2, 100,3, -2]

for i in range(len(my_list)):
    for j in range(0, len(my_list)-i-1):
        if my_list[j] > my_list[j+1]:
            my_list[j], my_list[j+1] = my_list[j+1], my_list[j]

print(f"Sorted list is: ", my_list)
```

Sorted list `is:` `[-2, 2, 3, 10, 20, 30, 100]`

24. How would you find the second largest element in a list without using built-in sorting functions?

We can first do sorting like the above question using bubble sort, then we can reverse the list, remove the 0th index value and then print the first index value; there are other better methods as well but I find it simpler.

```
my_list = [10, 20, 30, 2, 100,3, -2,90]

for i in range(len(my_list)):
    for j in range(0, len(my_list)-i-1):
        if my_list[j] > my_list[j+1]:
            my_list[j], my_list[j+1] = my_list[j+1], my_list[j]

l2 = my_list[::-1]
del l2[0]
print(f"The second largest element is: ",l2[0])
```

⇒The second largest element `is:` `90`

25. How would you find pairs of elements in a list that sum up to a specific number?

```
lst = [1, 2, 3, 4, 5, 6]
target_sum = 9

pairs = []
for i in range(len(lst)):
    for j in range(i + 1, len(lst)):
        if lst[i] + lst[j] == target_sum:
            pairs.append((lst[i], lst[j]))
```

```
print("Pairs: ", pairs)
```

```
Pairs:  [(3, 6), (4, 5)]
```

26. How would you remove duplicates from a list while preserving the original order?

```
l = [1,1,2,3,3,4,5,6,6,8]
```

```
seen = set()
result = []
```

```
for item in l:
    if item not in seen:
        result.append(item)
        seen.add(item)
```

```
print(f"List without duplicates with original order: ", result)
```

Output:

```
List without duplicates with original order:  [1, 2, 3, 4, 5, 6, 8]
```

27. How would you merge two sorted lists into a single sorted list?

Program:

```
list1 = [10,20,30,40,50]
list2 = [60,70,80,90,100]
```

```
list3 = []
```

```

i=0
j=0

while i < len(list1) and j < len(list2):
    if list1[i] < list2[j]:
        list3.append(list1[i])
        i = i + 1
    else:
        list3.append(list2[j])
        j = j + 1

list3.extend(list1[i:])
list3.extend(list2[j:])

print(f"Merged Sorted List is: ", list3)

```

Output:

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

28. How would you check if one list is a subset of another list?

```

list1 = [10,20,30,40,50,60,70]
list2 = [10,20,30,40,50,60,70,60,70,80,90,100]

is_subset = True

for item in list1:
    if item not in list2:
        is_subset = False
        break
if is_subset:
    print("list1 is a subset of list2")
else:
    print("list1 is not a subset of list2")

```

29. How would you find the longest consecutive sequence in a list of numbers?

```

nums= [9, 4, 7, 1, 3, 2,3,4, 5, 6, 10, 8, 9,1000]

num_set = set(nums)

```

```

max_length = 0
longest_sequence = []

for num in num_set:
    if num - 1 not in num_set:
        current_num = num
        current_length = 1
        current_sequence = [num]

        while current_num + 1 in num_set:
            current_num += 1
            current_length += 1
            current_sequence.append(current_num)

        if current_length > max_length:
            max_length = current_length
            longest_sequence = current_sequence

print("Longest sequence length:", max_length)
print("Longest sequence:", longest_sequence)

```

Output:

```

Longest sequence length: 10
Longest sequence: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```

30. How would you find the majority element in a list (the element that appears more than half the time)?

Case1: No majority element

```

nums= [9, 4, 7, 1, 3, 2,3,4, 5, 6, 10, 8, 9,1000]

candidate = None

count = 0
for num in nums:

```

```

    if count == 0:
        candidate = num
    if num == candidate:
        count = count + 1
    else:
        count -= 1

if nums.count(candidate) > len(nums) // 2:
    print(f"The majority element is: {candidate}")
else:
    print("No majority element found.")

```

Case2: Majority element exists

```

nums= [9, 4,2,2,2,2,2, 7, 1, 3, 2,3,4, 5, 6, 10, 8, 9,1000,
2,2,2,2,2,2,2,2,2,2]

```

```

candidate = None

```

```

count = 0
for num in nums:
    if count == 0:
        candidate = num
    if num == candidate:
        count = count + 1
    else:
        count -= 1

if nums.count(candidate) > len(nums) // 2:
    print(f"The majority element is: {candidate}")
else:
    print("No majority element found.")

```

Output:

The majority element is: 2