# LISTS

Python lists are a versatile data structure that can hold a collection of items. They are defined using square brackets `[]` and can contain elements of different data types, such as integers, strings, or even other lists.

Here's an example of creating a list in Python:

```python
my_list = [1, 2, 3, "apple", "banana"]
```

In the above example, `my_list` is a list that contains integers (1, 2, 3) and strings ("apple", "banana").

Lists are mutable, which means you can modify their elements. You can access individual elements of a list using indexing, starting from 0. For example, to access the first element of `my_list`, you can use `my_list[0]`.

Lists also support various operations such as appending elements, removing elements, slicing, and concatenation. Here are a few examples:

```python
my_list.append(4)  # Adds 4 to the end of the list
my_list.remove("apple")  # Removes the element "apple" from the list
sliced_list = my_list[1:3]  # Creates a new list with elements from index 1 to 2
concatenated_list = my_list + [5, 6]  # Concatenates two lists
```

Lists are a fundamental data structure in Python and are widely used in various applications. They provide flexibility and convenience for storing and manipulating collections of data.

```python
arr = [10, 20, 20, 20, 4, 40, 22, 12, 34, 76, 123, 55]
arr2 = [20, 20, 20, 20, 50, 50, 22]

# 1. Write a Python program to find the second largest element in a
list.

def second_largest_element(arr):
    max_element = float('-inf')
    second_max_element = float('-inf')
    for num in arr:
        if num > max_element:
            second_max_element = max_element
            max_element = num
        elif num > second_max_element and num < max_element:
            second_max_element = num

    return second_max_element
print(second_largest_element(arr))  # 10

76
```

```python
# 2. Write a Python program to calculate the average of numbers in a
list.
average = sum(arr) / len(arr)
print(average)
```

```
36.333333333333336
```

```python
# 3. Write a Python program to check if two lists are identical
(contain the same elements in the same order).

def check_identical(arr1, arr2):
    return arr1 == arr2
print(check_identical([10, 20, 20, 20, 4, 40, 22, 12, 34, 76, 123,
55], [10, 20, 20, 20, 4, 40, 22, 12, 34, 76, 123, 55]))  # True
print(check_identical([10, 20, 20, 20, 4, 40, 22, 12, 34, 76, 123,
55], [10, 20, 20, 20, 4, 40, 22, 12, 34, 76, 123, 55, 66]))  # False
```

```
True
False
```

```python
# 4. Write a Python program to rotate a list by k elements to the
right.

def rotate_list(arr, k):
    # Rotate the list by k elements to the right
    k = k % len(arr)  # Ensure k is within the bounds of the list
length
    return arr[-k:] + arr[:-k]
# Example usage
k = int(input("Enter the value of k: "))  # number of elements to
rotate (here 5 is taken)
rotated_list = rotate_list(arr, k)
print("Rotated list:", rotated_list)
```

```
Rotated list: [12, 34, 76, 123, 55, 10, 20, 20, 20, 4, 40, 22]
```

```python
# 5. Write a Python program to find the most frequent element in a
list.

def most_frequent(arr):
    max_count = -1
    most_frequent_element = None
    for element in arr:
        count = arr.count(element)
        if count > max_count:
            max_count = count
            most_frequent_element = element
    return most_frequent_element
# Example usage
arr = [55, 66, 77, 88, 99, 99]
print("Most frequent element:", most_frequent(arr))
print("Most frequent element:", most_frequent(arr2))
```

```
Most frequent element: 99
Most frequent element: 20
```

```python
# 6. Write a Python program to find common elements in two lists.

def common_elements(list1, list2):
    return [element for element in list1 if element in list2]
# Example usage
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
print("Common elements:", common_elements(list1, list2))
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]
print("Common elements:", common_elements(list1, list2))
```

```
Common elements: [4, 5]
Common elements: []
```

```python
# 7. Write a Python program to find the intersection of multiple
lists.

def intersection_of_lists(lists):
    if not lists:
        return []
    intersection = lists[0]
    for lst in lists[1:]:
        intersection = [element for element in intersection if
element in lst]
    return intersection
# Example usage
lists = [[1, 2, 3], [2, 3, 4], [2, 3, 5]]
print("Intersection:", intersection_of_lists(lists))
lists = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print("Intersection:", intersection_of_lists(lists))
```

```
Intersection: [2, 3]
Intersection: []
```

```python
# 8. Write a Python program to remove duplicates from a list while
maintaining the original order.

def remove_duplicates(lst):
    seen = []
    result = []
    for element in lst:
        if element not in seen:
            seen.append(element)
            result.append(element)
    return result
# Example usage
lst = [1, 3, 3, 5, 7, 8, 8]
print("List after removing duplicates:", remove_duplicates(lst))
lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print("List after removing duplicates:", remove_duplicates(lst))
```

```
List after removing duplicates: [1, 3, 5, 7, 8]
List after removing duplicates: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```python
# 9. Write a Python Program to find the duplicate elements in a list

def find_duplicates(lst):
    counts = {}
    duplicates = []
    for element in lst:
        if element in counts:
            counts[element] += 1
            if counts[element] == 2:
                duplicates.append(element)
        else:
            counts[element] = 1
    return duplicates
# Example usage
lst = [1, 2, 2, 3, 3, 3, 4]
print("Duplicate elements:", find_duplicates(lst))
lst = [1, 2, 3, 4, 5, 6, 7]
print("Duplicate elements:", find_duplicates(lst))
```

```
Duplicate elements: [2, 3]
Duplicate elements: []
```

```python
# 10. Write a Python program to generate a list of squares of
numbers from 1 to n.

def generate_squares(n):
    return [i**2 for i in range(1, n+1)]
# Example usage
n = int(input("Enter the value of n: ")) # 5
print("List of squares:", generate_squares(n))
```

```
List of squares: [1, 4, 9, 16, 25]
```

```python
# 11. Write a Python program to find the sum of all even numbers in
a list.

def sum_of_even_numbers(lst):
    return sum([num for num in lst if num % 2 == 0])
# Example usage
lst = [1, 2, 3, 4, 5, 6]
print("Sum of even numbers:", sum_of_even_numbers(lst))
```

```
Sum of even numbers: 12
```

```python
# 12. Write a Python program to convert a list of characters into a
single string.

def list_to_string(lst):
    return ''.join(lst)

# Example usage
```

```python
input_string = list(input("Enter the string: ")) # hello
print("Converted string:", list_to_string(input_string))
```

```
Converted string: hello
```

```python
# 13. Write a Python program to find the smallest element in a list.

def smallest_element(lst):
    if not lst:
        return None
    smallest = lst[0]
    for element in lst:
        if element < smallest:
            smallest = element
    return smallest
# Example usage
lst = [3, 1, 4, 1, 5, 9, 2, 6]
print("Smallest element:", smallest_element(lst))
lst = [10, 20, 30, 40, 50]
print("Smallest element:", smallest_element(lst))
```

```
Smallest element: 1
Smallest element: 10
```

```python
# 14. Write a Python program to reverse a list.

def reverse_list(lst):
    return lst[::-1]
# Example usage
lst = [1, 2, 3, 4, 5]
print("Reversed list:", reverse_list(lst))
```

```
Reversed list: [5, 4, 3, 2, 1]
```

```python
# 15. Write a Python program to find the union of two lists.

def union_of_lists(list1, list2):
    union = list1[:]
    for element in list2:
        if element not in union:
            union.append(element)
    return union
# Example usage
list1 = [1, 2, 3]
list2 = [3, 4, 5]
print("Union of lists:", union_of_lists(list1, list2))
list1 = [1, 2, 3]
list2 = [4, 5, 6]
print("Union of lists:", union_of_lists(list1, list2))
```

```
Union of lists: [1, 2, 3, 4, 5]
Union of lists: [1, 2, 3, 4, 5, 6]
```

```python
# 16. Check if the element 'banana' exists in the list l = ['apple',
'banana', 'cherry'].
```

```python
def element_exists(lst, element):
    return element in lst
# Example usage
lst = ['apple', 'banana', 'cherry']
element = 'banana'
print(f"Element '{element}' exists in the list:",
element_exists(lst, element))
```

Element 'banana' exists in the list: True

```python
# 17. Sort the list l = [3, 1, 4, 1, 5, 9, 2, 6] in ascending order
and print the sorted list.

def sort_list_ascending(lst):
    return sorted(lst)

def sort_list_descending(lst):
    return sorted(lst, reverse=True)
# Example usage
lst = [3, 1, 4, 1, 5, 9, 2, 6]
print("Sorted list in ascending order:", sort_list_ascending(lst))
print("Sorted list in descending order:", sort_list_descending(lst))
```

Sorted list in ascending order: [1, 1, 2, 3, 4, 5, 6, 9]
Sorted list in descending order: [9, 6, 5, 4, 3, 2, 1, 1]

```python
# 18. Convert the list l = [1, 2, 3] into a tuple t and convert the
tuple t = (4, 5, 6) into a list. Print both.

def list_to_tuple(lst):
    return tuple(lst)

def tuple_to_list(tpl):
    return list(tpl)
# Example usage
lst = [1, 2, 3]
tpl = (4, 5, 6)
print("Converted tuple:", list_to_tuple(lst))
print("Converted list:", tuple_to_list(tpl))
```

Converted tuple: (1, 2, 3)
Converted list: [4, 5, 6]

```python
# 19. Create two lists l1 with elements [1, 2, 3] and l2 with
elements [4, 5, 6]. Concatenate them into a single list l3 and print
l3.

def concatenate_lists(l1, l2):
    return l1 + l2
# Example usage
l1 = [1, 2, 3]
l2 = [4, 5, 6]
print("Concatenated list:", concatenate_lists(l1, l2))
```

```
Concatenated list: [1, 2, 3, 4, 5, 6]

# 20. Create a list l = ['apple', 'banana', 'cherry']. Find the
index of the element 'banana'.

def find_index(lst, element):
    for i, e in enumerate(lst):
        if e == element:
            return i
    return -1
# Example usage
lst = ['apple', 'banana', 'cherry']
element = 'banana'
print(f"Index of element '{element}':", find_index(lst, element))

Index of element 'banana': 1

# 21. Write a Python program to split the string "hello world" into
a list of words.

def split_string(s):
    return s.split()
# Example usage
s = "hello world"
print("List of words:", split_string(s))
s = "python programming"
print("List of words:", split_string(s))

List of words: ['hello', 'world']
List of words: ['python', 'programming']

# 22. Write a Python program to join the list ['hello', 'world']
into a single string with spaces.

def join_list(lst):
    return ' '.join(lst)
# Example usage
lst = ['hello', 'world']
print("Joined string:", join_list(lst))
lst = ['apple', 'banana', 'cherry']
print("Joined string:", join_list(lst))

Joined string: hello world
Joined string: apple banana cherry
```