13 List Assessments in Python

This document provides 13 list assessments in Python, ranging from easy to hard, including their solutions and expected results.

Easy Assessments

1. Sum of List Elements

Write a function that calculates the sum of all elements in a list.

```python
def sum_list(lst):
    return sum(lst)

# Test
print(sum_list([1, 2, 3, 4, 5]))  # Output: 15
```

2. Find Maximum Element

Write a function to find the maximum element in a list.

```python
def max_element(lst):
    return max(lst)

# Test
print(max_element([1, 2, 3, 4, 5]))  # Output: 5
```

3. Find Minimum Element

Write a function to find the minimum element in a list.

```python
def min_element(lst):
    return min(lst)

# Test
print(min_element([1, 2, 3, 4, 5]))  # Output: 1
```

4. Average of List Elements

Write a function to calculate the average of all elements in a list.

```python
def average_list(lst):
    return sum(lst) / len(lst)

# Test
print(average_list([1, 2, 3, 4, 5]))  # Output: 3.0
```

5. Remove Duplicates

Write a function to remove duplicate elements from a list.

```
def remove_duplicates(lst):
    return list(set(lst))

# Test
print(remove_duplicates([1, 2, 2, 3, 4, 4, 5]))  # Output: [1, 2, 3, 4, 5]
```

Medium Assessments

6. Find Common Elements

Write a function to find common elements between two lists.

```
def common_elements(lst1, lst2):
    return list(set(lst1) & set(lst2))

# Test
print(common_elements([1, 2, 3], [2, 3, 4]))  # Output: [2, 3]
```

7. Merge and Sort Lists

Write a function to merge two lists and sort the result.

```
def merge_and_sort(lst1, lst2):
    return sorted(lst1 + lst2)

# Test
print(merge_and_sort([3, 1, 2], [6, 5, 4]))  # Output: [1, 2, 3, 4, 5, 6]
```

8. Second Largest Element

Write a function to find the second largest element in a list.

```
def second_largest(lst):
    lst = list(set(lst))
    lst.sort()
    return lst[-2]

# Test
print(second_largest([1, 2, 3, 4, 5]))  # Output: 4
```

9. Rotate List

Write a function to rotate a list by a given number of positions.

```
def rotate_list(lst, positions):
    return lst[positions:] + lst[:positions]

# Test
print(rotate_list([1, 2, 3, 4, 5], 2))  # Output: [3, 4, 5, 1, 2]
```

10. Flatten a Nested List

Write a function to flatten a nested list.

```
def flatten_list(nested_lst):
    flat_list = []
    for sublist in nested_lst:
        for item in sublist:
            flat_list.append(item)
    return flat_list

# Test
print(flatten_list([[1, 2, 3], [4, 5], [6]]))  # Output: [1, 2, 3, 4, 5, 6]
```

Hard Assessments

11. Find Duplicates

Write a function to find all duplicate elements in a list.

```
def find_duplicates(lst):
    return [item for item in set(lst) if lst.count(item) > 1]

# Test
print(find_duplicates([1, 2, 2, 3, 3, 3, 4]))  # Output: [2, 3]
```

12. Group by Frequency

Write a function to group elements by their frequency.

```
from collections import defaultdict

def group_by_frequency(lst):
    freq_dict = defaultdict(list)
    for item in lst:
        freq_dict[lst.count(item)].append(item)
    return dict(freq_dict)

# Test
print(group_by_frequency([1, 2, 2, 3, 3, 3, 4]))  # Output: {1: [1, 4], 2: [2, 2], 3: [3, 3
```

13. Sublist with Maximum Sum

Write a function to find the contiguous sublist with the maximum sum in a list.

```
def max_sublist_sum(lst):
    max_sum = current_sum = lst[0]
    for num in lst[1:]:
        current_sum = max(num, current_sum + num)
        max_sum = max(max_sum, current_sum)
    return max_sum

# Test
print(max_sublist_sum([1, -2, 3, 4, -1, 2, 1, -5, 4]))  # Output: 9
```

Stay Updated

Be sure to this repository to stay updated with new examples and enhancements!

License

This project is protected under the MIT License.

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with by Panagiotis Moschos (https://github.com/pmoschos)