

This notebook (list_ops_basics.ipynb) covers basic Python programs that perform essential operations on lists and strings, including finding maximum values, filtering, counting occurrences, cloning lists, and string manipulations. Ideal for Python beginners to strengthen foundational concepts.

Can be accessed here(https://github.com/amitvsuryavanshi04/SIC_programming_and_coding)

Program 51 Find the largest number in a list

```
# Sample list of the numbers
numbers = [30, 10, -45, 5, 20]

#initialize a variable to store the minimum value
minimum = numbers[0]

#iterate through the list and update the minimum number
for i in numbers:
    if i > minimum:
        minimum = i
# Print the minimum value
print("The largest number in the list is:", minimum)
```

➡ The largest number in the list is: 30

Program 52 code to find the second largest number in the list

```
# Sample list of numbers
numbers = [30, 10, 45, 5, 20]
# Sort the list in descending order
numbers.sort(reverse=True)
# Check if there are at least two elements in the list
if len(numbers) >= 2:
    second_largest = numbers[1]
    print("The second largest number in the list is:", second_largest)
else:
    print("The list does not contain a second largest number.")
```

➞ The second largest number in the list is: 30

Program 53 to find N Largest elements from a list.

```
def find_n_largest_elements(lst, n):  
    # Sort the list in descending order  
    sorted_lst = sorted(lst, reverse=True)  
    # Get the first N elements  
    largest_elements = sorted_lst[:n]  
    return largest_elements  
  
# Sample list of numbers  
numbers = [30, 10, 45, 5, 20, 50, 15, 3, 345, 54, 67, 87, 98, 100, 34]  
# Number of largest elements to find  
N = int(input("N = " ))  
# Find the N largest elements from the list  
result = find_n_largest_elements(numbers, N)  
# Print the N largest elements  
print(f"The {N} largest elements in the list are:", result)
```

➞ N = 8
The 8 largest elements in the list are: [345, 100, 98, 87, 67, 54, 50, 45]

Program 54 code to print even numbers in a list.

```
# Sample list of numbers  
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
# Using a list comprehension to filter even numbers  
even_numbers = [num for num in numbers if num % 2 == 0]  
# Print the even numbers  
print("Even numbers in the list:", even_numbers)
```

➞ Even numbers in the list: [2, 4, 6, 8, 10]

Program 55 python program to print odd numbers in a list.

```
# Sample list of numbers.  
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# Using a list comprehension to filter even numbers
even_numbers = [num for num in numbers if num % 2 != 0]
# Print the even numbers
print("Odd numbers in the list:", even_numbers)
```

➞ Odd numbers in the list: [1, 3, 5, 7, 9]

Program 56 to remove empty list from a list

```
# Sample list containing lists
list_of_lists = [[1, 2, 3], [], [4, 5], [], [6, 7, 8], []]
# Using a list comprehension to remove empty lists
filtered_list = [i for i in list_of_lists if i]
# Print the filtered list
print("List after removing empty lists:", filtered_list)
```

➞ List after removing empty lists: [[1, 2, 3], [4, 5], [6, 7, 8]]

Program 57 cloning or copying a list.

```
#Method - 01 Using the slice operator.
ori_list = [1,2,3,4,5]
cloned_li = ori_list[:]
print(cloned_li)
```

➞ [1, 2, 3, 4, 5]

```
#Method - 02 using the list()operator
ori_list = [1,2,3,4,5,6]
cloned_list = ori_list[:]
print(cloned_list)
```

➞ [1, 2, 3, 4, 5, 6]

```
#Method - 03 using the list comprehension
ori_list = [1,2,3,4,5,6]
cloned_list = [i for i in ori_list]
print(cloned_list)
```

➞ [1, 2, 3, 4, 5, 6]

Program 58 count occurrences of an element in a list

```
def count_occ(l,elem):
    count = l.count(elem)
    return count

#Example usage:
my_list = [1,2,3,4,5,2,3,4,6,5]
elem_to_count = 2

occ = count_occ(my_list,elem_to_count)
print(f"The element {elem_to_count} appears {occ} times in the list ")
```

➞ The element 2 appears 2 times in the list

Program 59 to find words which have greater length compared to a given 'k'

```
def find_words(words, k):
    # Create an empty list to store words greater than k
    result = []
    # Iterate through each word in the list
    for i in words:
        # Check if the length of the i is greater than k
        if len(i) > k:
            # If yes, append it to the result list
            result.append(i)
    return result

# Example usage
word_list = ["apple", "banana", "cherry", "date", "elderberry", "dragonfruit"]
k = 5
long_words = find_words(word_list, k)
print(f"Words longer than {k} characters: {long_words}")
```

➦ Words longer than 5 characters: ['banana', 'cherry', 'elderberry', 'dragonfruit']

Program 60 removing ith character from a string.

```
def remove_char(input_str, i):  
# Check if i is a valid index  
    if i < 0 or i >= len(input_str):  
        print(f"Invalid index {i}. The string remains unchanged.")  
        return input_str  
# Remove the i-th character using slicing  
    result_str = input_str[:i] + input_str[i + 1:]  
    return result_str  
# Input string  
input_str = "Hello, wWorld!"  
i = 7 # Index of the character to remove  
# Remove the i-th character  
new_str = remove_char(input_str, i)  
print(f"Original String: {input_str}")  
print(f"String after removing {i}th character : {new_str}")
```

➦ Original String: Hello, wWorld!
String after removing 7th character : Hello, World!