

PYTHON IN 10 VIDEOS

Assignment - 3

1. Method to remove element from list?

- a. `remove()`
- b. `pop()`
- c. `delete()`
- d. `discard()`

Ans:a

2. Syntax to access 1st list element:

- a. `List[1]`
- b. `list(0)`
- c. `List[0]`
- d. `list.first()`

Ans:c

3: Output: test_list[-1] ?

- a. First element of the list
- b. Last element of the list
- c. Second to last element of the list
- d. 0

Ans:b

4. Output: `test_list.count(element)` ?

- a. True if element is list else, False
- b. Index of the element
- c. no.of.occurences of element in the list
- d. None of the above

Ans:c

5. How to reverse a list?

- a. `list.reverse()`
- b. `reverse(list)`
- c. `List[::-1]`
- d. All of the above

Ans:d

6. Method to sort a list

- a. `list.sort()`
- b. `sorted(list)`
- c. Both A and B
- d. None of the above

Ans:c

7. How to extract sublist from a list?

- a. `list.extract()`
- b. `list.slice()`
- c. `List[start:end]`
- d. `sublist(list)`

Ans:c

8. How to initialize an empty list ?

- a. `list()`
- b. `[]`
- c. `empty_list()`
- d. Both a and b

Ans:d

9. Create a list of random numbers and sort it in ascending and descending order. Remove the duplicates from the list and print the modified list.

```
import random

random_numbers = [random.randint(1, 20) for _ in range(15)]
print(f"Original list: {random_numbers}")

sorted_numbers = sorted(random_numbers)
print(f"Sorted in ascending order: {sorted_numbers}")

sorted_numbers_desc = sorted(random_numbers, reverse=True)
print(f"Sorted in descending order: {sorted_numbers_desc}")

unique_numbers = list(set(random_numbers))
print(f"List with duplicates removed: {unique_numbers}")
```

10. Write a function that takes a 3x3 matrix (nested list) as input and returns its transpose. Print the original and transposed matrices.

```
def transpose_matrix(matrix):  
    transposed = [[matrix[j][i] for j in range(len(matrix))] for i in range(len(matrix[0]))]  
    return transposed  
  
matrix = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
transposed = transpose_matrix(matrix)  
print("Original matrix:")  
for row in matrix:  
    print(row)  
print("Transposed matrix:")  
for row in transposed:  
    print(row)
```

11. Create a list of the first 10 positive integers. Remove the elements at indices 2, 4, and 6, and insert the element '99' at index 5. Print the modified list.

[3]:

```
lst = list(range(1, 11))
print(f"Original list: {lst}")
del lst[6]
del lst[4]
del lst[2]
lst.insert(5, 99)
print(f"Modified list: {lst}")
```

Original list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Modified list: [1, 2, 4, 6, 8, 99, 9, 10]

12. Create a new list containing the squares of the first 10 positive integers using a list comprehension. Print the new list.

```
squares = [x**2 for x in range(1, 11)]  
print(squares)
```

13. Write a function that rotates a list by n positions. Print the original and rotated lists.

```
def rotate_list(lst, n):  
    return lst[n:] + lst[:n]  
  
original_list = [1, 2, 3, 4, 5]  
rotated_list = rotate_list(original_list, 2)  
print(f"Original list: {original_list}")  
print(f"Rotated list: {rotated_list}")  
  
Original list: [1, 2, 3, 4, 5]  
Rotated list: [3, 4, 5, 1, 2]
```

14. Write a function that takes two lists and returns a new list containing only the elements that are present in both lists. Print the intersected list.

```
def list_intersection(lst1, lst2):  
    return [x for x in lst1 if x in lst2]  
  
list1 = [1, 2, 3, 4, 5]  
list2 = [3, 4, 5, 6, 7]  
intersection = list_intersection(list1, list2)  
print(f"List 1: {list1}")  
print(f"List 2: {list2}")  
print(f"Intersection: {intersection}")
```