**Python Data Structure Exercise for Beginners**

### Exercise 1: Create a list by picking an odd-index items from the first list and even index items from the second

Given two lists, l1 and l2, write a program to create a third list l3 by picking an odd-index element from the list l1 and even index elements from the list l2.

**Given**:

l1 = [3, 6, 9, 12, 15, 18, 21]

l2 = [4, 8, 12, 16, 20, 24, 28]

**Expected Output**:

Element at odd-index positions from list one

[6, 12, 18]

Element at even-index positions from list two

[4, 12, 20, 28]

Printing Final third list

[6, 12, 18, 4, 12, 20, 28]

### Exercise 2: Remove and add item in a list

Write a program to remove the item present at index 4 and add it to the 2nd position and at the end of the list.

**Given**:

list1 = [54, 44, 27, 79, 91, 41]

**Expected Output**:

List After removing element at index 4 [34, 54, 67, 89, 43, 94]

List after Adding element at index 2 [34, 54, 11, 67, 89, 43, 94]

List after Adding element at last [34, 54, 11, 67, 89, 43, 94, 11]

### Exercise 3: Slice list into 3 equal chunks and reverse each chunk

**Given**:

sample\_list = [11, 45, 8, 23, 14, 12, 78, 45, 89]

**Expected Outcome**:

Chunk 1 [11, 45, 8]

After reversing it [8, 45, 11]

Chunk 2 [23, 14, 12]

After reversing it [12, 14, 23]

Chunk 3 [78, 45, 89]

After reversing it [89, 45, 78]

### Exercise 4: Count the occurrence of each element from a list

Write a program to iterate a given list and count the occurrence of each element and create a [dictionary](https://pynative.com/python-dictionaries/) to show the count of each element.

**Given**:

sample\_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]

**Expected Output**:

Printing count of each item {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}

### Exercise 5: Create a Python set such that it shows the element from both lists in a pair

**Given**:

first\_list = [2, 3, 4, 5, 6, 7, 8]

second\_list = [4, 9, 16, 25, 36, 49, 64]

**Expected Output**:

Result is {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}

### Exercise 6: Find the intersection (common) of two sets and remove those elements from the first set

**Given**:

first\_set = {23, 42, 65, 57, 78, 83, 29}

second\_set = {57, 83, 29, 67, 73, 43, 48}

**Expected Output**:

Intersection is {57, 83, 29}

First Set after removing common element {65, 42, 78, 23}

### Exercise 7: Checks if one set is a subset or superset of another set. If found, delete all elements from that set

**Given**:

first\_set = {27, 43, 34}

second\_set = {34, 93, 22, 27, 43, 53, 48}

**Expected Output**:

First set is subset of second set - True

Second set is subset of First set - False

First set is Super set of second set - False

Second set is Super set of First set - True

First Set set()

Second Set {67, 73, 43, 48, 83, 57, 29}

### Exercise 8: Iterate a given list and check if a given element exists as a key’s value in a dictionary. If not, delete it from the list

**Given**:

roll\_number = [47, 64, 69, 37, 76, 83, 95, 97]

sample\_dict = {'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}

**Expected Outcome**:

After removing unwanted elements from list [47, 69, 76, 97]

### Exercise 9: Get all values from the dictionary and add them to a list but don’t add duplicates

**Given**:

speed = {'jan': 47, 'feb': 52, 'march': 47, 'April': 44, 'May': 52, 'June': 53, 'july': 54, 'Aug': 44, 'Sept': 54}

**Expected Outcome**:

[47, 52, 44, 53, 54]

### Exercise 10: Remove duplicates from a list and create a tuple and find the minimum and maximum number

**Given**:

sample\_list = [87, 45, 41, 65, 94, 41, 99, 94]

**Expected Outcome**:

unique items [87, 45, 41, 65, 99]

tuple (87, 45, 41, 65, 99)

min: 41

max: 99