Home Work - Day 7

**Python List Datatype**

1. **Create a list of items using the list() Constructor and using [] of the given below:**
   1. Rose
   2. Peony
   3. Orchids
   4. Snowdrop
   5. Calendula
   6. Aster
   7. Tulips
   8. Bluebell
   9. Sunflower

Program:

L1=["Rose","Peony","Orchids","Snowdrop","Calendula","Aster","Tulips","Bluebell","Sunflower"]  
print(L1)  
print()  
L2 = list(["Rose", "Peony", "Orchids", "Snowdrop", "Calendula", "Aster", "Tulips", "Bluebell", "Sunflower"])  
print(L2)  
print()

Output:

['Rose', 'Peony', 'Orchids', 'Snowdrop', 'Calendula', 'Aster', 'Tulips', 'Bluebell', 'Sunflower']

['Rose', 'Peony', 'Orchids', 'Snowdrop', 'Calendula', 'Aster', 'Tulips', 'Bluebell', 'Sunflower']

1. **Create program to find the length of the given list of the above example?**

**Program:**

L1=["Rose","Peony","Orchids","Snowdrop","Calendula","Aster","Tulips","Bluebell","Sunflower"]  
print(L1)  
print()  
Length=len(L1)  
print(Length)

**Output:**

['Rose', 'Peony', 'Orchids', 'Snowdrop', 'Calendula', 'Aster', 'Tulips', 'Bluebell', 'Sunflower']

9

**3.Create program to find the list of student details in the mixed datatypes?**

**4.Reverse a list in Python using the above example?**

1. **Concatenate two lists in the following order: list1** = ["Hello ", "take "]

**list2** = ["Dear", "Sir"]

1. **Add new item to list after a specified item:**

Write a program to add item 7000 after 6000 in the following Python List:

Given:

list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]

Expected output:

[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]

1. **You have given a nested list. Write a program to extend it by adding the sublist ["h", "i", "j"] in such a way that it will look like the following list.**

Given List:

**list1** = ["a", "b", ["c", ["d", "e", ["f", "g"], "k"], "l"], "m", "n"]

**# sub list to add sub\_list** = ["h", "i", "j"]

Expected Output:

['a', 'b', ['c', ['d', 'e', ['f', 'g', 'h', 'i', 'j'], 'k'], 'l'], 'm', 'n']

1. **You have given a Python list. Write a program to find value 20 in the list, and if it is present, replace it with 200. Only update the first occurrence of an item.**

Given:

list1 = [5, 10, 15, 20, 25, 50, 20]

Expected output:

[5, 10, 15, 200, 25, 50, 20]

1. **Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.**

**Given List:** ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

**Expected Output:** ['Green', 'White', 'Black']

Program:

import time  
colors = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
  
# Remove by slicing — deletions won't affect the given indices  
del colors[5] # removes 'Yellow'  
del colors[4] # removes 'Pink'  
del colors[0] # removes 'Red'  
  
print(colors) # ['Green', 'White', 'Black']

output:

['Green', 'White', 'Black']

1. **Write a Python program to replace the last element in a list with another list.**

**Sample data:** [1, 3, 5, 7, 9, 10], [2, 4, 6, 8]

**Expected Output:** [1, 3, 5, 7, 9, 2, 4, 6, 8]

1. **Write a Python program to extend a list without append. Sample data:** [10, 20, 30] [40, 50, 60]

**Expected output:** [40, 50, 60, 10, 20, 30]

1. **Write a Python program to append a list to the second list. Sample data:**

**Number =** [10, 20, 30, 40]

**animal =** ["Cat", "Dog", "Lion", "Ponda"]

**Expected Output:** [10, 20, 30, 40, 'Cat', 'Dog', 'Lion', 'Ponda']

1. **What are the properties or characteristics of List in Python?**

* **Insertion is preserved.**
* **Duplicate values are allowed.**
* **Heterogenous obeject are allowed.**
* **List is mutable obj(start\_full\_obj).**
* **List is a dynamic or growable.**
* **Indexing and slicing operator is available.**

1. **How to create a list in python?**

**my\_list=[1,2,3,4,5]**

**my\_list.append(6)**

**print(my\_list)**

**output:**

**[1, 2, 3, 4, 5, 6]**

1. **How many types of modes do we have to Accessing items from a list?**

You can access items from a Python list in 4 main ways: indexing (by position), slicing (extracting a sub-list), iteration (looping over items), and advanced methods like list comprehensions

1. **How to Accessing single items from the list? animal =** ["Cat", "Dog", "Lion", "Ponda"]

import time  
 animal = ["Cat", "Dog", "Lion", "Ponda"]  
 for i in animal:  
 print(i)

output:

Cat

Dog

Lion

Ponda

1. **How to Accessing Multiple items from the list? animal =** ["Cat", "Dog", "Lion", "Ponda"]

**import time**

**animal = ["Cat", "Dog", "Lion", "Ponda"]**

**print(animal)**

**output:**

**['Cat', 'Dog', 'Lion', 'Ponda']**

1. **How to find length of the list?**

You can find the length of a list using the len() function, for example: length = len(my\_list).

1. **What is the list datatype in python?**

A list in Python is a built-in, dynamic container for organizing a collection of items.

1. **How many methods we do h Add elements to the list.**

Append(x)

Extend(iterable),insert(index,x)

**19.How to use the given list operators:**

For the following examples, we assume that **l1** and **l2** are lists, **x, i, j, k, n** are integers.

**l1** = [10, 20, 30, 40, 50] and **l2** = [60, 70, 80, 60]

|  |  |
| --- | --- |
| **Operation** | **Description** |
| **x in l1** | Check if the list **l1** contains item **x**. |
| **x not in l2** | Check if list **l1** does not contain item **x**. |
| **l1 + l2** | Concatenate the lists **l1** and **l2**. Creates a new list containing the items from **l1**  and **l2**. |
| **l1 \* 5** | Repeat the list **l1** 5 times. |
| **l1[i]** | Get the item at index **i**. Example **l1[2]** is **30**. |
| **l1[i:j]** | List slicing. Get the items from index **i** up to index **j** (excluding **j**) as a List. An example **l1[0:2] is [10, 20]** |
| **l1[i:j:k]** | List slicing with step. Returns a List with the items from index **i** up to index **j** taking every k-th item. An example **l1[0:4:2]** is **[10, 30]**. |
| **len(l1)** | Returns a count of total items in a list. |
| **l2.count(60)** | Returns the number of times a particular item (60) appears in a list. The answer is **2**. |
| **l1.index(30)** | Returns the index number of a particular item (30) in a list. The answer is **2**. |
| **l1.index(30, 2, 5)** | Returns the index number of a particular item (30) in a list. But search Returns the item with maximum value from a list.  The answer is **60** only from index number 2 to 5. |
| **min(l1)** | Returns the item with a minimum value from a list. The answer is **10**. |
| **max(l1)** | Returns the item with maximum value from a list. The answer is **60**. |
| **l1.append(100)** | Add item at the end of the list |
| **l1.append([2, 5, 7])** | Append the nested list at the end |
| **l1[2] = 40** | Modify the item present at index **2** |
| **l1.remove(40)** | Removes the first occurrence of item **40** from the list. |
| **pop(2)** | Removes and returns the item at index **2** from the list. |
| **l1.clear()** | Make list empty |
| **l3= l1.copy()** | Copy **l1** into **l2** |