**Day 4 – USER INPUTS,LISTS**

# Python User Input:

# **The input() function allows user input**

# **You can assign the user input to a variable**

# **Example:**       x = input('Enter your Favourite cricter:')       print(‘Your favourite cricketer is ' + x)

# Python Lists:

# **The list type is a container that holds a number of other objects, in a given order**

# **The list type allows you to add and remove objects from the sequence.**

# Building a list:

**A = [‘a’,’b’,’c’]**

**B = [1,2,3]**

**Accessing a list:**

* The number of items in the list, **L[i]** returns the item at index *i* (the first item has index 0), and **L[i:j]** returns a new list, containing the objects between *i* and *j*.
* Index of list starts from 0

**item = L[index]**

**Example: second\_element = A[1]**

**seq = L[start:stop]**

**Example: print(A[1:2])**

**>> [2,3]**

* If you pass in a negative index, Python adds the length of the list to the index. L[-1] can be used to access the last item in a list.
* For normal indexing, if the resulting index is outside the list, Python raises an **IndexError** exception.
* You can also add items to an existing sequence. The **append** method adds a single item to the end of the list, the **extend** method adds items from another list (or any sequence) to the end, and **insert** inserts an item at a given index, and move the remaining items to the right.

**L.append(item)**

**L.extend(sequence)  
L.insert(index, item)**

* You can also remove items. The **del** statement can be used to remove an individual item, or to remove all items identified by a slice. The **pop** method removes an individual item and returns it, while **remove** searches for an item, and removes the first matching item from the list.

**del L[i]**

**del L[i:j]**

**item = L.pop() # last item**

**item = L.pop(0) # first item**

**item = L.pop(index)**

**L.remove(item)**

* To get the smallest or largest item in a list, use the built-in **min** and **max** functions:

**lo = min(L)**

**hi = max(L)**

* The **sort** method sorts a list in place.

**L.sort()**

* To get a sorted copy, use the built-in **sorted** function

**out = sorted(L)**

# Python Tuples:

# **A tuple in Python is similar to a** [**list**](https://www.programiz.com/python-programming/list)**. The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas we can change the elements of a list.**

# **A tuple can have any number of items and they may be of different types (integer, float, list,** [**string**](https://www.programiz.com/python-programming/string)**, etc.).**

**my\_tuple = (1,2,3)**

**print(my\_tuple)**

* Tuple with mixed datatypes:

**my\_tuple = (1, "Hello", 3.4)**

**print(my\_tuple)**

* Accessing tuple elements using indexing

**my\_tuple = ('p','e','r','m','i','t')**

**print(my\_tuple[0])**

* Unlike lists, tuples are immutable ( They cannot be changed)

Difference between tuple and list

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **List** | **Tuple** |
| 1 | List is mutable. | Tuple is immutable. |
| 2 | List iteration is slower and is time consuming. | Tuple iteration is faster. |
| 3 | List is useful for insertion and deletion operations. | Tuple is useful for readonly operations like accessing elements. |
| 4 | List consumes more memory. | Tuples consumes less memory. |
| 5 | List provides many in-built methods. | Tuples have less in-built methods. |
| 6 | List operations are more error prone. | Tuples operations are safe. |

**Exercise:**

1. Write a program to create a list of n integer values and do the following

* Add an item in to the list (using function)
* Delete (using function)
* Store the largest number from the list to a variable
* Store the Smallest number from the list to a variable

     2) Create a tuple and print the reverse of the created tuple

     3) Create a tuple and convert tuple into list