**List**

**Definition:** A list is a mutable (changeable) ordered collection of items.

Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are tuple, set, and Dictionary, all with different qualities and usage.

Lists are created using square brackets:

Create a List:

thislist = ["apple", "banana", "cherry"]

print(thislist) o/p ["apple", "banana", "cherry"]

**List Items**

List items are ordered, changeable, and allow duplicate values.

List items are indexed, the first item has index [0], the second item has index [1] etc.

**Ordered**

When we say that lists are ordered, it means that the items have a defined order, and that order will not change.

If you add new items to a list, the new items will be placed at the end of the list.

**Changeable**

The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

**Allow Duplicates**

Since lists are indexed, lists can have items with the same value:

Example

Lists allow duplicate values:

list\_a = ["apple", "banana", "cherry", "apple", "cherry"]

print(list\_a)

**List Length**

To determine how many items a list has, use the len() function:

Example Print the number of items in the list:

thislist = ["apple", "banana", "cherry"]

print(len(thislist)) o/p 3

**List Items - Data Types**

List items can be of any data type:

**Example** String, int and boolean data types:

list1 = ["apple", "banana", "cherry"]

list2 = [1, 5, 7, 9, 3]

list3 = [True, False, False]

A list can contain different data types:

**Example** list1 = ["abc", 34, True, 40, "male"]

**type()**

From Python's perspective, lists are defined as objects with the data type 'list': <class 'list'>

Example mylist = ["apple", "banana", "cherry"]

print(type(mylist))

**Access Items**

List items are indexed and we can access them by referring to the index number:

Example Print the second item of the list:

thislist = ["apple", "banana", "cherry"]

print(thislist[1]) o/p [“banana”]

Note: The first item has index 0.

**Negative Indexing**

Negative indexing means start from the end

-1 refers to the last item, -2 refers to the second last item etc.

Example Print the last item of the list:

thislist = ["apple", "banana", "cherry"]

print(thislist[-1]) o/p [‘cherry’]

**Range of Indexes (Slicing)**

You can specify a range of indexes by specifying where to start and where to end the range.

When specifying a range, the return value will be a new list with the specified items.

Example

Return the third, fourth, and fifth item:

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:5])

o/p [‘cherry’, ’orange’, ’kiwi’]

print(thislist[ :4])

o/p ["apple", "banana", "cherry", "orange"]

**Adding Items to List**

**Append Items:** To add an item to the end of the list, we use the append() method and it takes only 1 argument

**Example** thislist = ["apple", "banana", "cherry"]

thislist.append("orange")

print(thislist) o/p ["apple", "banana", "cherry", ”orange”]

**Insert Items**

The insert() method inserts an item at the specified index. It requires two arguments- the index where you want to insert an item, and the item itself

Example thislist = ["apple", "banana", "cherry"]

thislist.insert(1, "orange")

print(thislist) o/p ["apple", ”orange”, "banana", "cherry"]

**Extend List:** To append elements from another list or any iterable object to the current list, use the extend() method.

**Example** thislist = ["apple", "banana", "cherry"]

tropical = ["mango", "pineapple", "papaya"]

thislist.extend(tropical)

print(thislist)

o/p ["apple", "banana", "cherry", "mango", "pineapple", "papaya"]

**Remove Item from List**

**Remove Specified Item**

The remove() method removes the specified item.If there are more than one item with the specified value, it removes the first occurrence

**Example** thislist = ["apple", "banana", "cherry"]

thislist.remove("banana")

print(thislist)

Example

Remove the first occurrence of "banana":

thislist = ["apple", "banana", "cherry", "banana", "kiwi"]

thislist.remove("banana")

print(thislist)

**Remove Item at Specified Index**

The pop() method removes the specified index.

Example thislist = ["apple", "banana", "cherry"]

thislist.pop(1)

print(thislist) o/p ["apple", "cherry"]

If you do not specify the index, the pop() method removes the last item.

Example

thislist = ["apple", "banana", "cherry"]

thislist.pop()

print(thislist) o/p ["apple", "banana"]

**Sorting List**

List objects have a sort() method that will sort the list alphanumerically.

**Example**  thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]

thislist.sort()

print(thislist) o/p [“banana”, ”kiwi”, ”mango”, ”pineapple”, ”orange”]

**Example** thislist = [100, 50, 65, 82, 23]

thislist.sort()

print(thislist) o/p [23,50,65,82,100]

**Example** thislist = [100, 50, 65, 82, 3]

thislist.sort(reverse = True)

print(thislist) o/p [100,82,65,50,3]

Reverse Order

What if you want to reverse the order of a list, regardless of the alphabet?

The reverse() method reverses the current sorting order of the elements.

Example

Reverse the order of the list items:

thislist = ["banana", "Orange", "Kiwi", "cherry"]

thislist.reverse()

print(thislist) o/p [‘cherry’,’kiwi’,’Orange’,’banana’]

Tuple

Tuples are used to store multiple items in a single variable.

A tuple is a collection which is ordered and unchangeable. They allow duplicate values.

Tuples are written with round brackets or parenthesis.

We use Tuples when we know the items in tuples are not going to change.

Example

Create a Tuple:

thistuple = ("apple", "banana", "cherry")

print(thistuple) o/p ("apple", "banana", "cherry")

**Tuple Items**

Tuple items are ordered, unchangeable, and allow duplicate values.

Tuple items can be of any data type.

Tuple items are indexed, the first item has index [0], the second item has index [1] etc.

**Ordered**

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.x

**Unchangeable**

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

**Allow Duplicates**

Since tuples are indexed, they can have items with the same value:

Example thistuple = ("apple", "banana", "cherry", "apple", "cherry")

print(thistuple) o/p ("apple", "banana", "cherry", "apple", "cherry")

**Access Tuple Items**

You can access tuple items by referring to the index number, inside square brackets

**Change Tuple Values**

Once a tuple is created, you cannot change its values. Tuples are **unchangeable**, or **immutable** .

But we can convert the tuple into a list, change the list, and convert the list back into a tuple.

Example:

x = ("apple", "banana", "cherry")  
y = list(x)  
y[1] = "kiwi"  
x = tuple(y)  
print(x)

**Add tuple to a tuple**. You are allowed to add tuples to tuples, so if you want to add one item, (or many), create a new tuple with the item(s), and add it to the existing tuple:

Example thistuple = ("apple", "banana", "cherry")  
y = ("orange",)  
thistuple += y  
  
print(thistuple) o/p ("apple", "banana", "cherry",”orange”)

**Tuple Methods**

Python has two built-in methods that you can use on tuples.

Count() Returns the number of times a specified value occurs in a tuple

Example :

thistuple=(“apple”,”banana”,”apple”)

y=thistuple.count(“apple”)

print(y) o/p 2

Index() Searches the tuple for a specified value and returns the position of where it was first found.

thistuple=(“apple”,”banana”,”apple”)

y=thistuple.index(“apple”)

print(y) o/p 0