**LIST IN PYTHON WITH EXAMPLE :**

**What is Python List?**

* A list is a data structure in Python that is a **mutable**, **ordered sequence** of elements.
* Mutable means that you can change the items inside, while ordered sequence is in reference to index location.
* The first element in a list will always be located at index 0. Each element or value that is inside of a list is called an item.
* Just as strings are defined as characters between quotes, lists are defined by having different data types between square brackets [ ].
* Also, like strings, each item within a list is assigned an **index**, or location, for where that item is saved in memory.
* Lists are also known as a **data collection**. Data collections are simply data types that can store multiple datatypes items.

## Creating a list in python

### Method-1: Create a list from tuple and string data type Using list()

|  |
| --- |
| **#cretae a list from tuple and string using list() method**  yl1= 'welcome' #string  myl2=("tuple")# tuple  myl3=('1234') #string  #print as list type  print ('my first list ',list(myl1))  k =print ('my second list ',list(myl2))  print ('my third list ',list(myl3))  print(type(myl1)) # string type but  print( type(list(myl1))) |

### Method-2: Create a list using square brackets

This is the **traditional method** if you are defining a new list in your python code. You can place your list of items **separated by comma character within square brackets**. In this example I am creating different types of list enclosed within square brackets:

|  |
| --- |
| cars\_list=['maruti','honda',['blue','red'],'hundai',]  print("first element : ", cars\_list[0])  print("last element : ", cars\_list[-1])  print("third element : ", cars\_list[2]) # inside list index is 2  print("second element inside list : ", cars\_list[2][1])# print inside list red  print("first element inside list : ", cars\_list[2][0])# print inside list blue  **# access 4 th string 2nd letter access**  print("accsing value of 4 th string element with single characterr is :" , cars\_list[3][1])  print("accsing value of 4 th string element with single characterr is :" , cars\_list[3][0])  #EMPTY LIST  #CREATING LIST USING NO, FLOAT,STRING  empty\_list=[]  print(empty\_list)  **#list containg no.s**  l1=[1,2,3,4,6,8]  print(l1)  l2=[1.1,2.2,3.4,8.0]  print(l2)  print(type(l2))  l3=['a','b','a','c','d']  print(l3)  l4=['car','bus','truck'] #string list  print(l4)  print(type(l4))  l5=["abc",'car',0,2,3.5,6.7,2] #mixed data non homogeneous data  print(l5)  type(l5) |

**3. Accessing elements within a list**

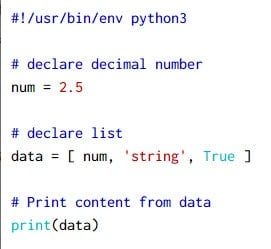
In order to access a specific element within a list, you use an **index**. When we declare our list variable, each item is given an index. Remember that indexing in Python starts at zero and is used with brackets.

You can use both positive and negative index value to access the elements of a list

* A **positive index** value 0 will start the iteration from the first element, index 1 to access the second element and so on
* A **negative index** value -1 will represent the last element in the list, -2 to access the second last and so on

|  |
| --- |
| **l1= [1,2,3.5,0.4,"tip"] # accasing element from list forward and reverse indexing**  print("first element :", l1[0])  print("second element :", l1[1])  print("third element :", l1[2])  print("fourth element :", l1[3])  print("last element :", l1[-1])  print("second last element :", l1[-2]) |

## 4. Declare a list with mixed data types

Lists can hold any data type, even other lists. Let’s check out an example of several data types:  


## 5. Define and access lists within another list

We also have the flexibility to define a list inside another list. In this example I have defined a parent list cars with some of the car manufacturers, in the second element of cars I have created another list with some colors. So I have created a list within another list:

|  |
| --- |
| cars\_list=['maruti','honda',['blue','red'],'hundai',]  print("first element : ", cars\_list[0])  print("last element : ", cars\_list[-1])  print("third element : ", cars\_list[2]) # inside list index is 2  print("second element inside list : ", cars\_list[2][1])# print inside list red  print("first element inside list : ", cars\_list[2][0])# print inside list blue  **# access 4 th string 2nd letter access**  print("accsing value of 4 th string element with single characterr is :" , cars\_list[3][1])  print("accsing value of 4 th string element with single characterr is :" , cars\_list[3][0]) |

**6. Iterate over list items using for loop**

Since we know that Python list data type is iterable, we can use for loop to iterate over individual elements of the list:

**7. Add elements to python list**

**append(item), list.extend(iterable), list.insert(index, item)**  to add elements to an existing python list,

### Method-1: Append item to existing list using - list.append(item)

We will use list.append(item) to add a single item to the end of a list. This **doesn't** return a new list instead it **only modifies the original list**.

|  |
| --- |
| # using append(item) method 1  num = [1, 2, 3, 4] #listname.append(value)  num.append(5)  print(num)  num.append(6)  print(num)  num.append(6.5)  print(num)  # If I try to append more than one item using num.append(5, 6) then I get TypeError as only one element can be added at a time. |

### Method-2: Add element to a list using list.extend(iterable)

The list.extend(iterable) method takes one argument, which should be an iterable data type. It then extends the list by appending all of the items from the iterable to the list.  
We can take the existing example where wo got the TypeError, now I will add 5 and 6 into our existing list using extend(iterable)

|  |
| --- |
| **# add elements more than one we use extend([elements])**  num = [1, 2, 3, 4]  num.extend([5, 6]) ## Add elements in the list format more than one  print(num)  #We **can also create another list and extend the existing one:**  extra=[7, 8]  num.extend(extra)  print(num) |

### Method-3: Add item at a certain position in a list

The **list.insert(index, item)** method (as the name suggests) inserts an **item at a given position** in a list. The method takes two arguments, index and item. The index is the position in the list before which to insert the item defined in the second argument. Both arguments are required.

**8. Remove item from a list ;**

We can use **list.remove(item), list.pop([index]) to remove** an element from a list or  **list.clear() to remove all the elements from the list.**