List – Unit 4

By : Bhavika Vaghela

Asst. Prof.

PICA – BCA – PU

What is List?

- Like a string, a **list** is a sequence of values. In a string, the values are characters.
- in a list, they can be any type. The values in list are called **elements** or sometimes **items**.
- The items in the list are separated with the comma (,) and enclosed with the square brackets [].

How to Define or Declare List?

- A list is created by placing all the items (elements) inside a square bracket [ ], separated by commas.
- It can have any number of items and they may be of different types (integer, float, string etc.).

E.g. mylist = [“xyz”, ”hello”, 10, 30.4, 45, ‘a’, ‘b’]

- One can also declare list using inbuilt method list() like,

Mylist = list(“xyz”,1,2,5,6,7,70.4) #it gives an error

Declare empty list

- As list is mutable so we can declare empty list and later on one can add or remove element from list.
- A list that contains no elements is called an empty list.

E.g. list1=[] #empty [ ] bracket

List1= list() # by using inbuilt method

Nested list

- List within list is known as nested list.

```
my_list = ["mouse", [8, 4, 6], ['a']]
```

Lists are mutable

- Unlike string list is mutable so one can change or delete the element of list by passing index value inside [] bracket.
- Again indices must be integer else it fire an type error. It must not be any float value.

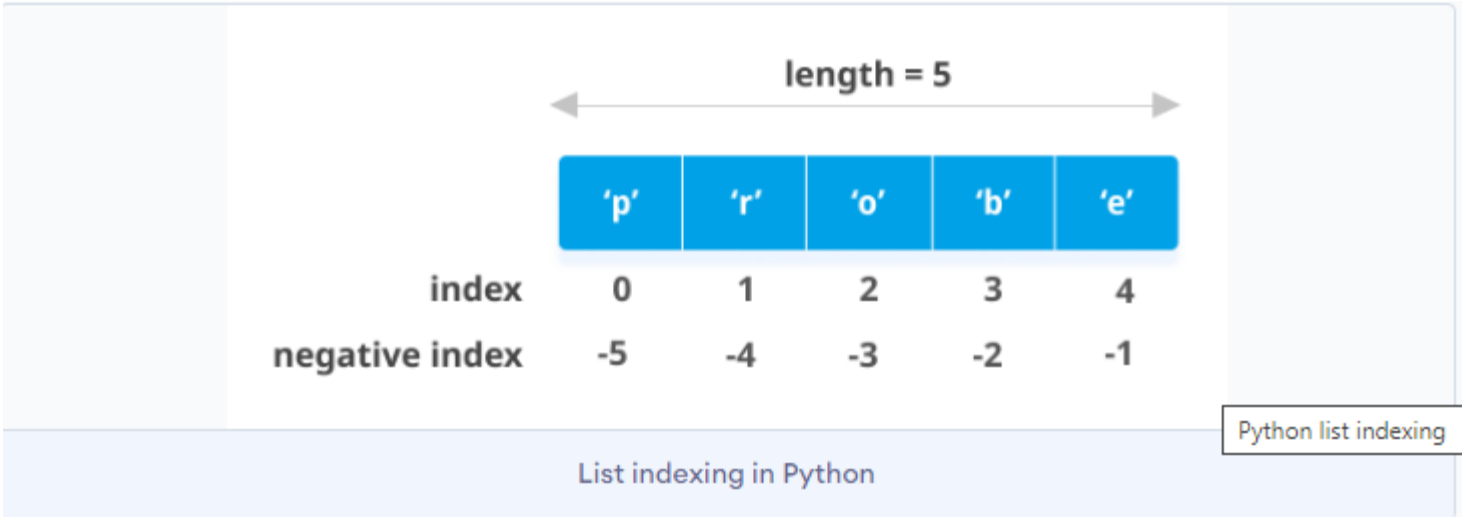
```
>>> list1=[1,2,3,[10,20,40],['hello','how'],30.4,56,0]
>>> list2=[30,20,50,60]
>>> list2[2]="hello"
>>> print(list2)
[30, 20, 'hello', 60]
>>> list2[2][0]='H'
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    list2[2][0]='H'
TypeError: 'str' object does not support item assignment
```

Deleting element from list

```
>>> print(list1)
[1, 2, 3, [10, 20, 40], ['hello', 'how'], 30.4, 56, 0]
>>> del list1[2]
>>> print(list1)
[1, 2, [10, 20, 40], ['hello', 'how'], 30.4, 56, 0]
>>> del list1[2:4]
>>> print(list1)
[1, 2, 30.4, 56, 0]
```

** you can also declare list by passing list inside [] bracket.

Indexing in List



- Indexing start by 0 for first element and so on.
- Element at -1 position is the last element of list.
- You can also perform slicing operation like string on list also by passing start index and end index value inside [] bracket.
- For the slicing you can also pass negative index like string.

E.g. mylist[2:4], mylist[2:], mylist[::-1], mylist[-1:]

Traversing a list

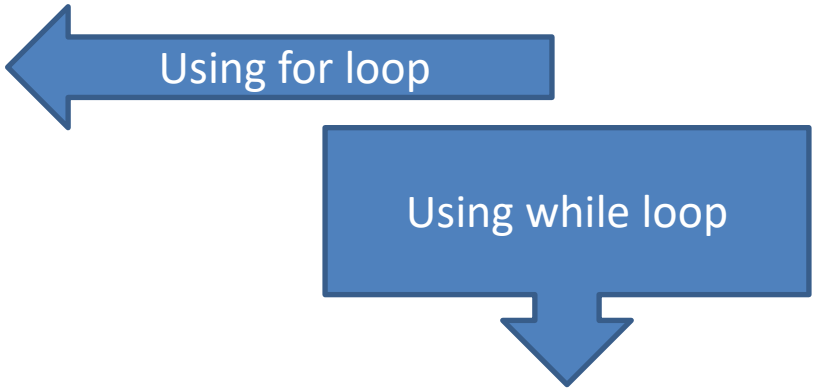
- The most common way to traverse the elements of a list is with a for loop. The syntax is the same as for strings.
- a list can contain another list, the nested list still counts as a single element. The length of this list is four:

e.g. ['spam', 1, ['Brie', 'Roquefort', 'Pol le Veq'], [1, 2, 3]]

```
list1=[1,2,3,4,5]
for i in list1:
      print(i)

** another way

for i in range(len(list1)):
      print(list1[i])
```




```
List1=[1,2,3,45,6]
index=0
while index<len(list1):
      print(list1[index])
       index=index+1
```

- A for loop over an empty list never executes the body.

```
list2=list()
for i in list2:                          ** it will not execute the block
      print("how are you")
```

List operation

- Same as string you can perform concatenates and repeat operation on list by using + an * sign respectively.

```
>>> A=[1,2,3]
>>> B=[5,6]
>>> print(A+B)
[1, 2, 3, 5, 6]
>>> B=B+[10,30]
>>> print(B)
[5, 6, 10, 30]
>>> C=[A+B]
>>> print(C)
[[1, 2, 3, 5, 6, 10, 30]]
```

concatenates

Repetition

```
>>> print(A*3)
[1, 2, 3, 1, 2, 3, 1, 2, 3]
>>> D=[A*3]
>>> print(D)
[[1, 2, 3, 1, 2, 3, 1, 2, 3]]
>>> D=A*3
>>> print(D)
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Update list using slice operation

- A slice operator on the left side of an assignment can update multiple elements:

```
>>> t = ['a', 'b', 'c', 'd', 'e', 'f']
>>> t[1:3] = ['x', 'y']
>>> print t
['a', 'x', 'y', 'd', 'e', 'f']
```

List method

- List provide various inbuilt method but for now we discuss on append(), extend(), sort() and insert() method.
- append() is use to add single element or list at the end of existing list.

```
>>> A=[1,2,3]
>>> A.append('hello')
>>> print(A)
[1, 2, 3, 'hello']
>>> B=[11,12,13]
>>> A.append(B)
>>> print(A)
[1, 2, 3, 'hello', [11, 12, 13]]
```

- **extend()** method takes a list as an argument and appends all of the elements.

```
>>> A=[1,2,3]
>>> B=[11,12,13]
>>> A.extend(B)
>>> print(A)
[1, 2, 3, 11, 12, 13]
```

- **sort()** method sort string in ascending order.

```
>>> C=[56,3,23,1,12,30]
>>> C.sort()
>>> print(C)
[1, 3, 12, 23, 30, 56]
>>> mystr=["who", "hello", "cat", "apple"]
>>> mystr.sort()
>>> print(mystr)
['apple', 'cat', 'hello', 'who']
```

- **insert()** method is use to insert element at given location for that we need pass index value where want to insert new element and the element which want to insert.

```
>>> mystr.insert(1,"ball")
>>> print(mystr)
['apple', 'ball', 'cat', 'hello', 'who']
```

- **pop()** is use to delete element from list if you know the index of it.
- **remove()** is use to remove a element if you don’t know its index but if you know element.

```
>>> mystr.pop(1)
'ball'
>>> mystr.remove('cat')
>>> print(mystr)
['apple', 'hello', 'who']
```

** we have many more inbuilt method of list

List and String

- A string is a sequence of characters and a list is a sequence of values, but a list of characters is not the same as a string.
- So to convert string into list we have inbuilt method list().
- The list function breaks a string into individual letters. If you want to break a string into words, you can use the split method.

```
>>> list1=list(mystr)
>>> print(list1)
['p', 'y', 't', 'h', 'o', 'n']
>>> mystr="wel come to parul university"
>>> list1=mystr.split()
>>> print(list1)
['wel', 'come', 'to', 'parul', 'university']
```

- An optional argument called a delimiter specifies which characters to use as word boundaries.
- The following example uses a hyphen as a delimiter:

```
>>> s = 'spam-spam-spam'
>>> delimiter = '-'
>>> s.split(delimiter)
['spam', 'spam', 'spam']
```

- join is the inverse of split. It takes a list of strings and concatenates the elements.
- join is a string method, so you have to invoke it on the delimiter and pass the list as a parameter.

```
>>> t = ['pining', 'for', 'the', 'fjords']
>>> delimiter = ' '
>>> delimiter.join(t)
'pining for the fjords'
```

Object and value

- When you create two list with same at that time you created two different object but when you create two string with same value there may be two possibility.
- In one case, str1 and str2 refer to two different objects that have the same value. In the second case, they refer to the same object.

```
>>> str1="apple"
>>> str2="apple"
>>> str1 is str2
True
>>> del str1
>>> print(str2)
apple
>>> str1=["apple"]
>>> str2=["apple"]
>>> str1 is str2
False
```

Aliasing

- If list1 refers to an object and you assign list2 = list1, then both variables refer to the same object.
- The association of a variable with an object is called a reference. In this example, there are two references to the same object.
- An object with more than one reference has more than one name, so we say that the object is aliased.
- If the aliased object is mutable, changes made with one alias affect the other.

```
>>> list1=[12,45,30]
>>> list2=list1
>>> list2.append(56)
>>> print(list1)
[12, 45, 30, 56]
>>> list1.pop(1)
45
>>> print(list2)
[12, 30, 56]
```

Inbuilt methods of list

Function	Description
list.append(obj)	The element represented by the object obj is added to the list.
list.clear()	It removes all the elements from the list.
List.copy()	It returns a shallow copy of the list.
list.count(obj)	It returns the number of occurrences of the specified object in the list.
list.extend(seq)	The sequence represented by the object seq is extended to the list.
list.index(obj)	It returns the lowest index in the list that object appears.
list.insert(index, obj)	The object is inserted into the list at the specified index.
list.pop(obj=list[-1])	It removes and returns the last object of the list.
list.remove(obj)	It removes the specified object from the list.
list.reverse()	It reverses the list.
list.sort([func])	It sorts the list by using the specified compare function if given.

Perform below listed programs

- Write a python program to Remove duplicate element from list using single list.
- Write a python program to Copy only upper case element into second list.
- Write a python program to count occurrence of word into list of string.
- Take 10 integer inputs from user and store them in a list. Now, copy all the elements in another list but in reverse order.
- Write a Python function that takes two lists and returns True if they have at least one common member.
- Write a Python program to convert a list of characters into a string.
- Write a Python program to get the frequency of the elements in a list.

- https://www.programiz.com/python-programming/list
- https://www.geeksforgeeks.org/python-list/
- https://beginnersbook.com/2018/02/python-list/
- https://data-flair.training/blogs/python-list-examples/
- https://developers.google.com/edu/python/lists
- https://realpython.com/python-lists-tuples/
- https://www.javatpoint.com/python-lists