**Introduction to pop() in Python**

pop() is a method of the complex datatype called list. The list is among the most commonly used complex datatype in python, and the pop() method is responsible for popping an item from the python list. The pop method will remove an item from a given index of the list and returns the removed item. So, in the end, the pop() method will be like a list without the element that was popped.

**Syntax:**

list.pop(index)

The index placed in the method is used to represent the specific element that needs to be popped out of the list. When some element is placed in the list’s argument in the case where that specific element is removed, considering a case where no specific index is specified, it pops the last element in the list. So the understanding is that the index value which is appended when no argument is passed on will be -1, which represents the last element in the list.

**How does pop() Function work?**

* The pop() will be accepting only one argument for its execution. This argument will be representing the index of the element, which is expected to be popped out from the specified list.
* The process of passing an argument to the list is very much optional; when no argument has been passed, then it represents the value of ‘-1’ as the default. Here basically, the default value of -1 represents the last position in the list.
* The outcome of the pop() method is the item placed at that specific position and along with the item being removed from that specific list.
* When the index value which has been passed is out of the range of the list index, then the ” IndexError: pop index out of rangeexception” will be thrown by the pop() method. So this means the pop() method will not work when the index is out of the index range.
* The key process performed in the backend during this index removal process is as listed, updating the index of the list, removing an element from the index, updating the remaining elements in the index, and removing elements from the index at the reverse of the list.

**Examples of pop() in Python**

Given below are the examples:

**Example #1**

**Code:**

# List of countries  
Country = ['India','Australia','China','Greece','Italy'] # Return the 3rd item along with a removal  
returned\_Country\_value = Country.pop(2)  
print('Returned country Value:', returned\_Country\_value)  
# Updated List  
print('Updated country List:',Country)

**Output:**

**Explanation:**

* A couple of country names are declared as items of a list.
* The third item, namely the county china, is expected to be removed from the list from this list of items.
* This item is achieved by the use of the pop() function by holding the index value of 3 in it.
* The returned value and the updated values are printed in the console.

**Example #2**

**Code:**

# List provided  
List = [11,3,211,6444,12,26] # Before Pop operation of the list  
print("List Before applying pop() method:", List)  
# Poping the third item from the given list  
List.pop(2)  
# Poping the fifth item from the given list  
List.pop(4)  
# After Pop operation of the list  
print("List After applying pop() method:", List)

**Output:**

**Explanation:**

* Here it is an integer list that holds a list of all integer values in it.
* The list has initially been printed in the console to display the original list, which is without any pop operation being performed.
* Then third and fourth items from the list are popped out, and the resulting list is again displayed in the console after the pop operation is performed.

**Example #3**

**Code:**

# List provided  
List = [11,'c#',3,211,'python','java',6444,12,26] # Before Pop operation of the list  
print("List Before applying pop() method:\n", List)  
# Poping the third item from the given list  
List.pop(2)  
# Poping the fifth item from the given list  
List.pop(4)  
# After Pop operation of the list  
print("List After applying pop() method:\n", List)

**Output:**

**Explanation:**

* This example is again very similar to the above-given example; the only difference is here the list is a combination of both the integer and string values.
* The list has initially been printed in the console to display the original list, which is without any pop operation being performed.
* Then third and fourth items from the list are popped out, and the resulting list is again displayed in the console after the pop operation is performed.

**Example #4**

**Code:**

# List of countries  
Country = ['India','Australia','China','Greece','Italy'] # Return the 3rd item along with a removal  
returned\_Country\_value = Country.pop(7)  
print('Returned country Value:', returned\_Country\_value)  
# Updated List  
print('Updated country List:',Country)

**Python List pop() - Return and Remove Element**

The pop() method returns an item from the specified position in the list and removes it. If no index is specified, the pop() method removes and returns the last item in the list.

**Syntax:**

list.pop(index)

**Parameters:**

index: (Optional) The element at the specified index is removed. If the index is not specified, the last element is removed.

**Return Value:**

A list item.

The following example demonstrates the pop() method.

Example: pop()

cities = ['Mumbai', 'London', 'Paris', 'New York']

print(cities.pop()) # returns and removes 'New York'

print('List Elements: ', cities)

print(cities.pop()) # returns and removes 'Paris'

print(cities)

print(cities.pop()) # returns and removes 'London'

print(cities)

print(cities.pop()) # returns and removes'Mumbai'

print(cities)

# cities.pop() # raise an error on an empty list

Output

New York

List Elements: ['Mumbai', 'London', 'Paris']

Paris

List Elements: ['Mumbai', 'London']

London

List Elements: ['Mumbai']

Mumbai

List Elements: []

In the above example, each call of cities.pop() will return the last element in the list and also removes it. Calling the pop() method on an empty list will raise an error.

You can specify an index of an item to be returned and removed from a list.

Example: pop(index)

cities = ['Mumbai', 'London', 'Paris', 'New York']

city = cities.pop(0)

print(city)

print("List Elements: ",cities)

city = cities.pop(2)

print(city)

print("List Elements: ",cities)

Output

Mumbai

List Elements: ['London', 'Paris', 'New York']

New York

List Elements: ['London', 'Paris']

In the above example, cities.pop(0) returns the first element and removes it. cities.pop(2) returns and removes element from the 2nd index.

The pop()method will throw an IndexError if the specified index is not found.

Example: pop()

cities = ['Mumbai', 'London', 'Paris', 'New York']

city = cities.pop(5)

Output

Traceback (most recent call last):

cities.pop(5)

IndexError: pop index out of range