



Inspiring Excellence

Course Title: Programming Language II

Course Code: CSE 111

Semester: Summer 2020

Topic: List

Table of Contents

List.....	1
Creating a List.....	1
Indexing of List	2
Access items from a list.....	2
List Mutability.....	3
List Slicing	4
Add items to a list.....	5
Remove items from a list	6
Copy a list	8
Joining two list.....	8
List methods	9

List

List is a collection of values which are called elements or items. The elements in the list are ordered and changeable. A list can contain any data types such as integers, strings or objects. All of the elements have an index. If there is a list of length n , then the range of the index would be 0 to $n-1$.

Creating a List

List can be created in python by writing the elements that are comma-separated within a square brackets []. The elements can be of any data type. Also, the values need not to be of the same data type, they can be of mixed data types such as string, float, integer etc.

Creating without constructor:

```
# empty list
my_list = []

# list of strings
my_list = ['harry', 'tom', 'ron']
```

```
# lists with mixed data types
my_list = [10.5, 'chocolate', 100]
my_list2 = [15.43, 20, 5.23, 'hola', 450, 'amigo']

# nested lists
my_list = ['Bonjour', ['frodo', 'bilbo', 'gollum']]
my_list2 = ['belgium', [64, 75, 57], ['chocolate']]
```

Using list constructor: You can also use `list()` constructor to create a list.

```
# empty list
my_list = list()

# list of strings
my_list = list(['harry', 'tom', 'ron'])

# list of floats
my_list2 = list([65.88, 100.67, 78.38, 25.92])
```

List length: To determine how many items a list has we use function `len()`.

```
>>> number = [11.1,22.1,33.3,44.4,55.5]
>>> print(len(number))

#Output:
5

>>> rainbow = ['red', 'orange', 'yellow', 'green', 'blue',
'indigo', 'violet']
>>> print(len(rainbow))

#Output:
7
```

Indexing of List

We can access list by using indexing method.

- The range of an index is 0 to n-1 where n is the length of the list
- There are two types of indexing: Positive index and negative index
- The type of the index is integer
- If index is out of range, this will give an **IndexError**

Example:

```
sports=['football', 'cricket', 'hockey', 'tennis', 'golf']
```

pos. index	0	1	2	3	4
elements	football	cricket	hockey	tennis	golf
neg. index	-5	-4	-3	-2	-1

Access items from a list

If we want to access an element from a list, we can do so by using the index operator [].

Remember, index starts from 0 and ends at n-1. We can also use negative subscript, where -1 is the last element (rightmost) and -n is the first element (leftmost).

```
>>> sports=['football', 'cricket', 'hockey', 'tennis', 'golf']
>>> print(sports[3])
#Output: tennis
>>> print(sports[-3])
#Output: hockey

>>> music = ['rock', 'pop', 'jazz', 'classical', 'heavy
metal', 'hip hop', 'folk']
>>> print(music[5])
#Output: hip hop
>>> print(music[-4])
#Output: classical
```

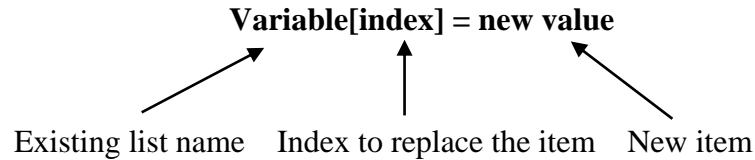
```
# This is acceptable
>>> fibonacci = [0, 1, 1, 2, 3, 5, 8, 13]
>>> print(fibonacci[-4])
#Output: 3
>>> print(fibonacci[7])
#Output: 13
>>> print(fibonacci[-8])
#Output: 0

>>> fruit_list = ['mango', 'orange', 'banana', 'guava']
>>> print(fruit_list[2]) #Positive Indexing
#Output: banana
>>> print(fruit_list[0]) #Positive Indexing
#Output: mango
>>> print(fruit_list[-1]) #Negative Indexing
#Output: guava
```

```
# This is not acceptable
>>> print(fruit_list[6])
#Output: IndexError: list index out of range
>>> print(fibonacci[5.0])
#Output: TypeError: list integer must be integer or slices
```

List Mutability

Lists are mutable or changeable, we can change the values in the list. We can change an existing item. The structure is:



```
# Example:
>>> prime = [2, 3, 5, 7, 11, 14, 17, 19, 23]
>>> print(prime)
#Output: [2, 3, 5, 7, 11, 14, 17, 19, 23]
>>> print(prime[5])
#Output: 14
>>> prime[5]= 13 #14 is replaced by 13
>>> print(prime)
#Output: [2, 3, 5, 7, 11, 13, 17, 19, 23]
#See 14 is no longer in the list

>>> student_list = ['harry', 'ron', 'fred', 'dean']
>>> print(student_list)
#Output: ['harry', 'ron', 'fred', 'dean']
>>> print(student_list[3])
#Output: dean
>>> student_list[3] = 'hermione' #dean is replaced by hermione
>>> print(student_list)
#Output: ['harry', 'ron', 'fred', 'hermione']
#See dean is no longer in the list
```

We can also add/append and delete/remove items.

List Slicing

We can print a specific range of elements from a list, by using Slice operation. The main structure of list slicing is:

list[start:end:step]

Here,

- start(inclusive): If this is provided, then starts from that specific index, otherwise starts from the beginning.
- end(exclusive): If this is provided, then ends at that specific index, otherwise ends at last index.
- step(optional): It specifies the increment

Additionally,

- For printing the whole list with list slicing, we use [:]

```
>>> squares = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121]
>>> print(squares[1:7])
#Output: [4, 9, 16, 25, 36, 49]
>>> print(squares[-6:-1])
#Output: [36, 49, 64, 81, 100]
>>> print(squares[:10])
#Output: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

>>> genre=['sci-fi', 'action', 'thriller', 'mystery', 'history',
'fantasy', 'mythology']
>>> gn = genre[1:4]
>>> print(gn)
#Output: ['action', 'thriller', 'mystery']
>>> print(genre[:5])
#Output: ['sci-fi', 'action', 'thriller', 'mystery', 'history']
>>> print(genre[3:])
#Output: ['mystery', 'history', 'fantasy', 'mythology']
>>> print(genre[-5:-1])
#Output: ['thriller', 'mystery', 'history', 'fantasy']
>>> print(genre[:])
#Output: ['sci-fi', 'action', 'thriller', 'mystery', 'history',
'fantasy', 'mythology']
```

Add items to a list

If you want to add an item to the end of the list, then you can do so by using the **append()** method.

```
>>> price=[455.67,673.56,1000.75,245.98]
>>> price.append(642.52)
>>> print(price)
#Output: [455.67, 673.56, 1000.75, 245.98, 642.52]

>>> closet = ['shirt', 'pant', 'scarf', 'skirt']
>>> closet.append('top')
>>> print(closet)
#Output: ['shirt', 'pant', 'scarf', 'skirt', 'top']
```

You can also use **insert(index,value)** method.

```
>>> price=[455.67,673.56,1000.75,245.98]
>>> price.insert(3, 642.52)
>>> print(price)
#Output: [455.67, 673.56, 1000.75, 642.52, 245.98]

>>> closet = ['shirt', 'pant', 'scarf', 'skirt']
>>> closet.insert(2, 'top')
>>> print(closet)
#Output: ['shirt', 'pant', 'top', 'scarf', 'skirt']
```

Remove items from a list

To remove the item of specific index, we use **pop(index)** method. Or when you use **pop()** method it removes the last item as index is not mentioned.

```
>>> marks = [64, 89, 91, 45, 77, 94]
>>> marks.pop(3)
>>> print(marks)
#Output: [64, 89, 91, 77, 94]

>>> purse = ['money', 'credit card', 'keys']
>>> closet.pop(1)
>>> print(closet)
#Output: ['money', 'keys']
```

```
>>> marks = [64, 89, 91, 45, 77, 94]
>>> marks.pop()
>>> print(marks)
#Output: [64, 89, 91, 45, 77]

>>> purse = ['money', 'credit card', 'keys']
>>> purse.pop()
>>> print(purse)
#Output: ['money', 'credit card']
```


If we want to remove a specified item, we can use **remove()** method:

```
>>> price= [455.67, 673.56, 1000.75, 642.52, 245.98]
>>> price.remove(1000.75)
>>> print(price)
#Output: [455.67, 673.56, 642.52, 245.98]

>>> purse = ['money', 'credit card', 'keys']
>>> purse.remove('credit card')
>>> print(purse)
#Output: ['money', 'keys']
```

Also, by using **del** keyword we can remove a specific index.

```
>>> fibonacci = [0, 1, 1, 2, 3, 4, 5, 8, 13]
>>> del fibonacci[5]
>>> print(fibonacci)
#Output: [0, 1, 1, 2, 3, 5, 8, 13]

>>> purse = ['money', 'credit card', 'keys']
>>> del purse[1]
>>> print(purse)
#Output: ['money', 'keys']
```

The **del** keyword can also remove the whole list.

```
>>> fibonacci = [0, 1, 1, 2, 3, 4, 5, 8, 13]
>>> del fibonacci
>>> print(fibonacci)
#Output: name 'fibonacci' is not defined

>>> purse = ['money', 'credit card', 'keys']
>>> del purse
>>> print(purse)
#Output: name 'purse' is not defined
```

Also, by using **clear()** method, we can empty a list.

```
>>> prime = [2, 3, 5, 7, 11, 14, 17, 19, 23]
>>> prime.clear()
>>> print(prime)
#Output: []

>>> purse = ['money', 'credit card', 'keys']
>>> purse.clear()
>>> print(purse)
#Output: []
```

Copy a list

If we want to create a copy of a list, we can use the built-in method **copy()**.

```
>>> quiz = [15,20,19,18,9]
>>> best = quiz.copy()
>>> print(best)
#Output: [15, 20, 19, 18, 9]

>>> instrument = ['flute', 'piano', 'violine','guitar']
>>> instrument2 = instrument.copy()
>>> print(instrument2)
#Output: ['flute', 'piano', 'violine', 'guitar']
```

Joining two list

By using the **+** operator, we can join two or more lists.

```
>>> girls_mark = [95,80,79,88,90]
>>> boys_mark = [65,84,59,78,83,92]
>>> marks= girls_mark + boys_mark
>>> print(marks)
#Output: [95, 80, 79, 88, 90, 65, 84, 59, 78, 83, 92]

>>> fiction = ['Mysteries', 'romance', 'fantasy']
>>> non_fiction = ['biography', 'history', 'travel']
>>> books = fiction + non_fiction
>>> print(books)
#Output: ['Mysteries', 'romance', 'fantasy', 'biography',
'history', 'travel']
```

Also, you can use **extend()** method for adding items from one list to another list.

```
>>> girls_mark = [95,80,79,88,90]
>>> boys_mark = [65,84,59,78,83,92]
>>> girls_mark.extend(boys_mark)
>>> print(girls_mark)
#Output: [95, 80, 79, 88, 90, 65, 84, 59, 78, 83, 92]

>>> fiction = ['Mysteries', 'romance', 'fantasy']
>>> non_fiction = ['biography', 'history', 'travel']
>>> fiction.extend(non_fiction)
>>> print(fiction)
#Output: ['Mysteries', 'romance', 'fantasy', 'biography',
'history', 'travel']
```

Also, you can use **extend()** method for adding items from one list to another list.

List methods

There are several built-in methods in python that you can also use.

Method	Description
list.count(a)	Returns the integer count of how many times a has occurred in the list
list.index(a)	Returns the index of the first occurrence of a
list.reverse()	Reverses order of the elements of the list
list.sort()	Sorts the list but the elements have to be commensurate