LIST

Module 3

Module Outline:

- List Overview
- List Representation in Python
- Array versus List
- Basic Operations
  - Traverse
  - Accessing
  - Insertion
  - Deletion
  - Search
  - Update
- Python List Methods

What is a List?

Overview

- The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

List in Python

Creating a list is as simple as putting different comma-separated values between square brackets. For example –

list1 = [**“computer”**, **“engineering”**, 2019, 2020]
list2 = [1, 2, 3, 4, 5]
list3 = [**‘a’**, **‘b’**, **‘c’**, **‘d’**]

Similar to string indices, list indices start at 0, and lists can be sliced, concatenated and so on.

Array versus List

Key Differences of Array and List

Similarities

- Both are used for storing data
- Both are mutable
- Both can be indexed and iterated through
- Both can be sliced

Differences

- Arrays are specially optimized for arithmetic computations
- Lists are containers for elements having differing data types but arrays are used as containers for elements of the same data type.

List Operations

Accessing Values in Lists

To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example –

```
list1 = [“computer”, “engineering”, 2019, 2020]
list2 = [1, 2, 3, 4, 5 ]
print "list1[0]: ", list1[0]
print "list2[1:5]: ", list2[1:5]
```

What is the output?

Traversing List

You can print a list using the print function. For example –

```
list = ['computer', 'engineering', 2019, 2020]
print list
```

What is the output?

Inserting Elements in Lists

- Insert operation is to insert one or more data elements into an array. Based on the requirement, a new element can be added at the beginning, end, or any given index of list.

- Here, we add a data element at the middle of the array using the python in-built **insert()** method.

Inserting Elements in Lists

```
list = ['computer', 'engineering', 2019, 2020]
print list
print "After adding element at index 4 : "
list.insert(2, 2025)
print list
```

What is the output?

Updating Lists

You can update single or multiple elements of lists by giving the slice on the left-hand side of the assignment operator, and you can add to elements in a list with the **append()** method. For example –

```
list = ['computer', 'engineering', 2019, 2020]
print "Value available at index 2 : "
print list[2]
list[2] = 2025
print "New value available at index 2 : "
print list[2]
```

What is the output?

Deleting List Elements

To remove a list element, you can use either the del statement if you know exactly which element(s) you are deleting or the **remove()** method if you do not know. For example –

```
list = ['computer', 'engineering', 2019, 2020]
print list
del list[2]
print "After deleting value at index 2 : "
print list
```

What is the output?

Basic List Operations

- Lists respond to the **+** and ***** operators much like strings; they mean concatenation and repetition here too, except that the result is a new list, not a string.

Python Expression	Results	Description
len([1, 2, 3])	3	Length
[1, 2, 3] + [4, 5, 6]	[1, 2, 3, 4, 5, 6]	Concatenation
['Hi!'] * 4	['Hi!', 'Hi!', 'Hi!', 'Hi!']	Repetition
3 in [1, 2, 3]	True	Membership
for x in [1, 2, 3]: print x,	1 2 3	Iteration

Python Array Methods

Array Methods on Python

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value

Array Methods on Python

Method	Description
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list
insert()	Adds an element at the specified position

Reference:

The instructor does not take the credits on the contents of this presentation.

https://www.tutorialspoint.com/python_data_structure/