GSET - Intro to Programming with Python

Tristan Hill

Tennessee Technological University

Summer 2024

Module 4 - Lists

Module 4 - Lists

- Data Stuctures and Programming
- Using Lists in Python
- Common Methods
- Advanced Methods
- Example Python Code

Data Stuctures and Programming

There are many ways to store data in the computer.

Q: We have practiced using single values, but how do we store multiple values?

A:

reference: docs.python.org

Data Stuctures and Programming

Common Data Stuctures

Name	Description
Array	
*List	
Stack	
Queue	
*Tuple and Sequence	
*Set	
*Dictionary	

* Python Data Structures

Data Stuctures and Programming

Python Data Structures

- List
- Tuple
- Set
- Dictionary

Note: C++ and MATLAB use arrays, but Python does not.

reference: docs.python.org

Using Lists in Python

Initializing a List

```
buildings = ['Brown', 'Clement', 'Prescott', 'Bruner']
```

The position of an items in list is the index. Duplicate items are allowed at different indicies.

reference: docs.python.org

Using Lists in Python

Accessing Items in a List

```
buildings = ['Brown', 'Clement', 'Prescott', 'Bruner']

print('The north building is', buildings[3])

print('The south building is', buildings[0])
```

The indicies of the list are used to access items. A *slice* of the list can be accessed as well.

ref: some text

Using Lists in Python

Redefining Items in a List

```
buildings[3] = 'New Bruner'

print('The CSC department is in', buildings[3])
```

Remember the data in the list is *mutable*, meaning it can be changed after it has been defined.

ref: python.org

Common Methods

Built-in Functions

- `len(list)` - get the length of list
- `del(a)` - del the variable a

List Object Methods

- `list.append( x )` - Add item x to end of list
- `list.insert( x )` - Insert item x at position
- `list.pop( i )` - Remove and item in list at position i and return it
- `list.clear( )` - Remove all items from the list

See the full list in the official python tutorial by clicking the link below.
ref: python.org

Common Methods

Advanced List Object Methods

- `list.sort(*, key=None, reverse=False)` - Sort the items of the list in place (the arguments can be used for sort customization, see sorted() for their explanation).
- `list.reverse()` - Reverse the elements of the list in place.
- `list.copy()` - Return a shallow copy of the list. Equivalent to `a[:]`.

List Comprehensions are a very powerful way to iterate through the items in a list.

reference: more info

Example Python Code

See **lecture5_lists.py** for example Python 3 code.