DATA STRUCTURES

OUTLINE

- Tuple
- List
- Dictionary
- Set
- Class

TUPLE

TUPLE

- A tuple is a fixed-length immutable sequence of Python objects.
- Tuple can be created using parentheses or invoking tuple().
- That said, the parentheses for creating a tuple is a norm, not a requirement.
- Elements of a tuple can be accessed using square brackets
- Tuples do not have too many methods. A particularly useful method is count which counts occurrences of a value.

LIST

LIST

- A list is a data structure that holds an ordered collection of values.
- Unlike tuples, lists are mutable and have variable length.
- Although a list can contain elements of different types, lists generally have elements of the same type.
- We often use the term element to describe an item in a list.
- The length of a list is its number of elements.
- A list may be instantiated using [ ] or list( ).

METHODS

- Lists have a number of methods that can be used to mutate or change the list object.
- Since lists are mutable, after the method is invoked, the result does not have to be assigned to an object.
- Here are a set of common methods for a list called lst:
  - lst.append(): Add an elements to the end
  - lst.remove(): Remove a specific element
  - in: Check if an element is present in the list
  - not in: Check if an element is not in the list
  - lst.extend(): add elements to the end of the list
  - lst.pop(): removes last element in the list
  - lst.sort(): arrange elements

SLICING

- Since a list holds an ordered collection of values, elements can be identified by their position.
- Each element is assigned an index position.
- Indices are consecutive and begin in 0.
- Square brackets are used for extracting an element by index position.
- A negative sign before an index means values are extracted relative to the end of the list.
- A start:stop notation can be passed to a list to subset or slice multiple elements. In this notation, the first element is inclusive but the last element is not included. So, lst[1:3] includes the second, and third elements in the list but not the fourth.
- A missing index on either side of the : in a slicing operation is a shortcut to indicate all elements before or after.

DICTIONARY

DICTIONARY

- A dictionary is a data structure that stores data as a collection of key-value pairs, where the key and value are Python objects.
- The key can be used to identify a value to be retrieved, inserted, modified or deleted. A key is a unique identifier for a value.
- A value corresponds to the key.
- The values can contain duplicates, the keys should not.
- Dictionary is mutable

DICTIONARY

- One way to distinguish a dictionary from a list is to recognize that a dictionary is used to create an association, while a list is defined to place elements in order.
- A Python dictionary is similar to the Webster Dictionary where words are associated ot their meaning. One can lookup a word in the dictionary and find its corresponding meaning. There are many real-world situations where data includes a set of key-value pairs such as menu item and price on a restaurant menu, customer and dollars spent in a customer transaction file, etc.
- One way to construct a dictionary is using curly braces.

SET

SET

- A set is an unordered collection of unique elements.
- A set can be created using the set() or curly braces

CLASS

CLASS

- A class is a blue print for creating objects.
- A class defines the methods/functionalities that objects made from it will have
- An object is called an instance of the class it is made from
- The act of creating an object is called instantiation
- Every time we work with an Python object, it has been "instantiated" from a Class.
- Syntax options like "" or [] or {} are shortcuts for instantiation
- For other classes, we will need to use class() syntax to instantiate an object

CONCLUSION

- In this section, we examined built-in data structures in Python.
  - Tuple
  - List
  - Dictionary
  - Set
  - Class