Python Sequences

Vivek K. S., Deepak G.

Information Systems Decision Sciences (ISDS)
MUMA College of Business
University of South Florida
Tampa, Florida

2017

Objectives

- To understand built-in data sequence types such as Lists, Sets, Tuples and Dictionaries.

- To understand their applications and advantages.

- To understand iterators and constructs that support sequences.

- To learn the different operations that could be performed on these types.

Lists

- A list is a collection of data elements arranged sequentially.
- They are mutable in nature, in that they can be appended to, and data removed from.
- Lists are enclosed by a pair of squared brackets '[]'
- Lists in Python are heterogeneous. A list can contain any different type of data.
- Lists are especially good for keeping track of things that could change in the course of the program execution.
- A list could be created by a pair of squared-brackets [] or list().

Creating a List

A list could be created as follows:

```
cities = ['Chicago','San Francisco','New York']
# A list could also become an
# element of another list
metros = ['Dallas','Las Vegas', cities]
print(cities)
['Chicago', 'San Francisco', 'New York']
print(metros)
['Dallas', 'Las Vegas', ['Chicago',
'San Francisco', 'New York']]
```

- Changing the cities list here modifies the metros list as well.
- This is because, cities is just a reference to the block of memory containing "cities list" and changes to it is being reflected in the metros list.

Data Type Conversions in Lists

Converting a string to a list is easy as follows:

```python
language = list('Python')
print(language)
['P', 'y', 't', 'h', 'o', 'n']

# Converting String to List using split()
birthday = '8/11/2017'
dob = birthday.split('/')
print(dob)
['8', '11', '2017']

# List can be converted to a String using join()
statement = ['I','love','Python']
sentence = ' '.join(statement)
print(sentence)
'I love Python'
```

Common List Operations

The following are the common operations in List.

- Adding elements to the list using add, append and insert.
- Indexing and Slicing.
- Modifying the list using indexing and Slicing techniques.
- Concatenating Lists.
- Removing items from the list using "del" remove and pop.
- Determining the membership of an item and counting the number of items.
- Searching and Sorting the list.

Tuples

Similar to lists, tuples are sequences of arbitrary items.

- Tuples are is the immutable collection sequence type in Python.
- Tuples can't be modified using add, delete, or changing of items after the tuple is defined. A tuple is this a constant list.
- A tuple can be created using ().
- Notice that a list can be created as follows "list()".

Tuple Unpacking

- Multiple variables can be assigned at one go using Tuples in Python
- This underlying nature has a bigger advantage in Python is a few operations such as variable swapping.
- In other programming languages, particularly statically-typed language, it is more complicated than how it's done in Python.
- Without the use of a temporary variable, unlike in other languages, in Python, two variables can be swapped as follows: a,b = b,a

Summary

- We understood what sequences are in Python.
- We extensively experimented with Python's sequential data types such as Lists and Tuples.
- We learned the difference between these two sequence types and how they help us in different situations.
- We understood and worked with the different methods that these two structures provide.