GENERAL ASSEMBLY

Other Useful Python

Patrick D. Smith

Lead Instructor, General Assembly DSI

LEARNING OBJECTIVES

- Insert Objective 1
- Insert Objective 2

Opening

Principles of Data Visualization

Visualizing your data is extremely important to be proficient at as a data scientist. Why?

2. You will be always be required to report on your findings working as a data scientist.

- Technical coworkers such as other data scientists or analysts will want to get an intuition for the data too
- Visualization will make your findings compelling and intuitive to non-technical coworkers.

Lambda Functions

Lambda Functions

- Python supports a style of programming called ***functional programming***, where you can easily pass functions to other functions
- A **lambda function** is a function that takes any number of arguments and returns the value of a single expression
- Lambda functions can not contain commands, and they can not contain more than one expression
- Lambda functions are Python's name for ***anonymous functions***, IE functions without a formal definition (def)

Lambda Functions

- At a base level, the syntax of lambda functions is:

lambda “arguments”: “expression”

Lambda Functions

- At a base level, the syntax of lambda functions is:

lambda “arguments”: “expression”

- Lambda functions are good as functions that are going to be used only once

Lambda Functions

Typically, functions are created for one of two purposes:

1. To reduce code duplication
2. To modularize code.

Lambda Functions

1. If your program contains duplicate code in various places, you can instead turn one chunk of that code into a functions, and then call it from various places in your code
2. If you have a chunk of code that performs one well-defined operation, but is quite long and ugly, you can replace that chunk of code with a functions

So why create a one lined function, that we only plan to use once?

Let’s take a look at an example

Modules

Modules

- We've talked about how as good data scientists and developers we want to make functions reusable

Modules

- We've talked about how as good data scientists and developers we want to make functions reusable
- However…..What if we want to reuse functions in *other programs* that we write?

Modules

- We've talked about how as good data scientists and developers we want to make functions reusable
- However…..What if we want to reuse functions in *other programs* that we write?
- The answer - ***modules!***

Modules

- There are various methods of writing modules, but the simplest way is to create a file with a .py extension that contains functions and variables
- A module can be imported by another program to make use of its functionality. This is how we can use the Python standard library as well
- Importing entire modules can be expensive for memory - in place, we can use the syntax:

from *module* import *variable*

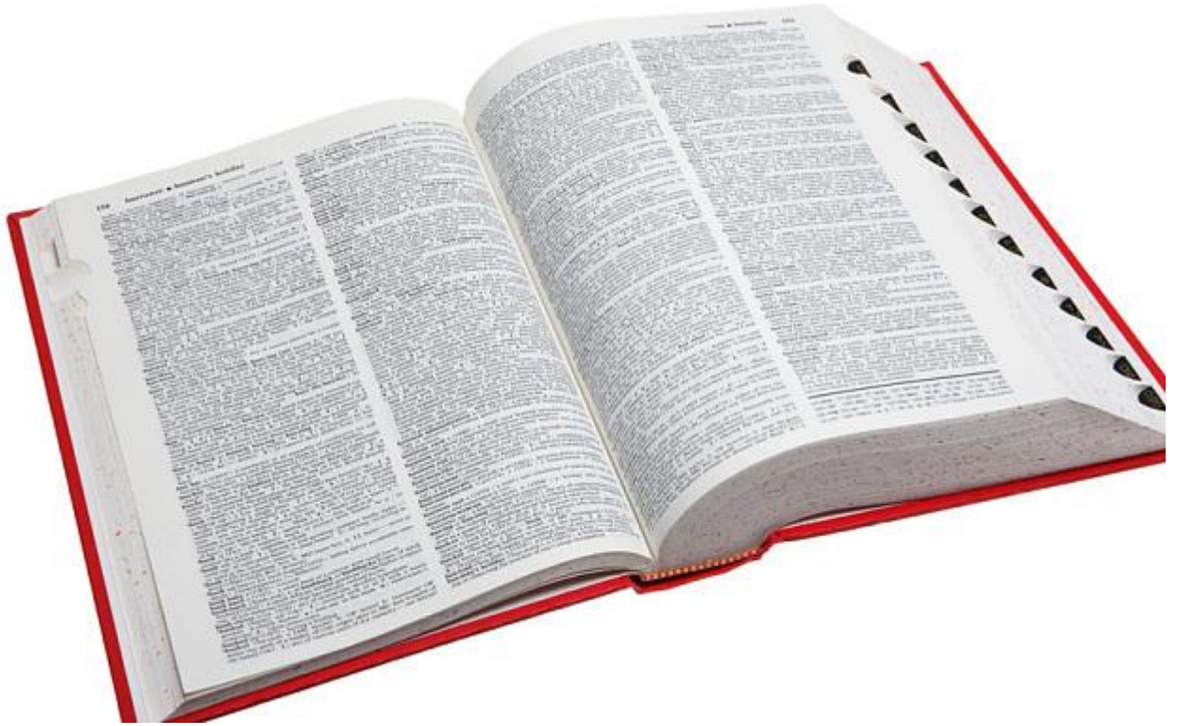
Data Structures

Other Types of Data Structures

- Data structures are basically just that - they are structures which can hold some data together. In other words, they are used to store a collection of related data.
- There are four built-in data structures in Python (we've talked about two) - list, tuple, dictionary and set.

Dictionaries

- Dictionaries are just like real dictionaries or phone books - we associate keys (name) with values (details)
- **Note:** Key must be unique, just like you cannot find out the correct information if you have two people with the exact same name.



Dictionaries

- Pairs of keys and values are specified in a dictionary by using the notation

d = {key1 : value1, key2 : value2 }

- ***Remember***: key-value pairs in a dictionary are not ordered in any manner. If you want a particular order, then you will have to sort them yourself before using it.
- Dictionaries are stored in **Dict** objects

Let’s take a look at an example

Sequences

- Lists, tuples and strings are examples of sequences, but what are sequences and what is so special about them?

Sequences

- Lists, tuples and strings are examples of sequences, but what are sequences and what is so special about them?
- The major features are membership tests, (i.e. the in and not in expressions) and indexing operations, which allow us to fetch a particular item in the sequence directly.

Sequences

- Lists, tuples and strings are examples of sequences, but what are sequences and what is so special about them?
- The major features are membership tests, (i.e. the in and not in expressions) and indexing operations, which allow us to fetch a particular item in the sequence directly.
- The three types of sequences mentioned above - lists, tuples and strings, also have a slicing operation which allows us to retrieve a slice of the sequence i.e. a part of the sequence

Let’s take a look at an example

Sets

- Sets are unordered collections of simple objects. These are used when the existence of an object in a collection is more important than the order or how many times it occurs.
- Using sets, you can test for membership, whether it is a subset of another set, find the intersection between two sets, and so on.

Let’s take a look at an example

References

- When you create an object and assign it to a variable, the variable only refers to the object and does not represent the object itself.
- Thus, the variable name points to that part of your computer's memory where the object is stored. This is called ***binding***.

Let’s take a look at an example

Some Extra thoughts on Strings

- .

Let’s take a look at an example

Conclusion