An Introduction to

Python For Data Science

October, 2017

What Is Python?

Python is:

- A High Level
- General Purpose Language
- Object Oriented (with full support for other paradigms)
- **Interpreted**
- Created in the early 1990’s
- Python 3.6 is the current version

Code Along 1

Jump right in – Code Along 1

Simple Hello World Program

Code Along 1

Takeaways - Code Along 1:

- Python Interpreter Vs Python modules
  - ➢ Use Interpreter for testing or quick experiments
  - ➢.py files – modules that can be rerun, need a Python interpreter to be executed

- Variables
  - ➢Get assigned values
  - ➢Can be reused, manipulated, reassigned … etc.

- Printing
  - ➢Visual output of your code – is It working as it should be?

…You just wrote your first python code!

Code Along 2

Code Along 2

Variables and Operators (and comments)

Code Along 2

Takeaways - Code Along 2:

- Variables
  - ➢ No type declaration necessary (Python figures out the type)
  - ➢ first assignment created the variable
  - ➢ Assignment is done using “=“
- Operations
  - ➢ Carried out on variables (operands)
  - ➢ Operator can behave differently based on the data type
    - + Adds Integers, concatenates Strings
  - ➢ Strongly-Typed is the way! (No implicit type conversions)
- Comments Keep code clean and readable
  - ➢ Whoever reads your code will thank you!
  - ➢ # used to comment a single line

Code Along 3

More on Variables

Code Along 3

Takeaways - Code Along 3:

- Multiple Assignment
  - ➢ x,y=4,6
- Basic Data Types:
  - ➢Integers (Default for Numbers) 5, 17 , 3000
  - ➢Floats :5.3, 7.324, -34.11, 5/2
  - ➢Strings: “Bob”, ‘John’, “Kevin’s”, “””Mark’s car is “black”””””
- Variable names are :
  - ➢**Case Sensitive!**
  - ➢Can **NOT** start with a number
  - ➢Can contain underscores, letters, numbers
  - ➢Can NOT be a reserved word (if, elif, global, return, pass, import …..etc.)

Code Along 3

Additional Notes:

- Python binds variables to **object references**
  - ➢Assigning a variables created a reference to an object, NOT a copy of the object

- A variable name does not imply the object type, the object referenced does
  - ➢X=7 , X="Bob" is completely fine.
- Some datatypes are **mutable**, some are **immutable**

More on that in later courses

Code Along 4

More Data Types

Code Along 4

Takeaways - Code Along 4:

- Tuples
  - ➢A collection of “Elements”
  - ➢Can be sliced
    - Elements Accessible individually using [n] or [-n]
    - Ranges [1:2], [:2], [2:], [1:-1], [:]
  - ➢Elements cannot be changed (**immutable**)
  - ➢Check for element presence using “in” clause
- Lists
  - ➢Like a Tuple, but with added functionalities
  - ➢Slower but more useful
  - ➢Elements can be inserted, appended, removed, deleted, “popped” and changed
- Use **len(x)** to find length, **x.index(n)** on lists and tuples to “know your way”
- A string is also a sequence type, closer to a tuple (immutable)!

Code Along 4

Takeaways - Code Along 4 (Continued):

- You can “**Add**” sequences:
  - ➢[1,2,3]+[4,5,6]
  - ➢(1,2,3)+(4,5,6)
  - ➢"Hello"+" "+"World! " (Look familiar?)

- You can “**Multiply**” a sequence and an integer:
  - ➢[1,2,3]*3
  - ➢(1,2,3)*2
  - ➢"Hello"*3

Code Along 5

Code Along 5

Conditionals and loops

Code Along 5

Takeaways - Code Along 5 (Continued):

- Code blocks are identified using **Indentation** **(no { } here!)**
  - Standard is 4 white spaces – tabs not recommended

- Conditions can be evaluated using **if, elif, else** statements
- = used for assignment, == used for comparison
- != is the opposite of ==

- Loops allow you to execute a block of code several times using **while** or **for..in**
- Else condition in loops are executed when condition is false
- Stop a loop using break
- **Watch out for infinite loops!**

Code Along 6

Code Along 6

Functions

Code Along 6

Takeaways - Code Along 6:

- Functions are defined using the keyword **def**
  - ➢def addition_function(x,y):
- Values are returned using the **return** keyword (even if not present!)
  - ➢None value
- Functions take arguments
- A function can be an argument to another function
  - ➢addition_function(3,addition_function(3,5))
- No types are defined for arguments or return types
- Functions can call other functions
- Objects have scopes

Code Along 7

Scopes

Code Along 7

Takeaways - Code Along 7:

- Objects have scopes
- Be careful of what you are trying to reference
- Use of **return** to make an object available

Code Along 8

Modules

Code Along 8

Takeaways - Code Along 8:

- A module gains access to code in another module by importing it
- Modules provide a way of code reuse
- Python comes with a library of standard modules
  - Such as the datetime module
  - …or the statistics module
    - Import statistics
    - print(statistics.mean([1,2,3,4,5,6]))

Questions?

GSA

D2D

DATA TO DECISIONS

Thank You

Get to coding!

GSA

D2D
DATA TO DECISIONS