CRMDA Python Workgroup (2016-2017)

Boryana Koseva

August 19, 2016

Contents

I Workgroup Meeting Format **2**

II Core Topics **2**

1 Why learn to program? **2**

2 Introduction to Programming in Python **2**

3 Lists, Dictionaries, and other Data Structures **3**

4 Manipulating Strings and Regular Expressions **4**

5 Functions **4**

6 Flow Control and Top-Down Programming **4**

7 Reading/Writing Files and Organizing Project Directories **4**

8 Plotting in Python **5**

III Possible Advanced Topics **5**

Part I
Workgroup Meeting Format

My initial thought on the overall format is that we should focus on a few "core" topics during the first half of the semester. I envision that this part would be somewhat fast-paced. During the second half of the semester, we can explore more advanced topics. These topics can be tailored specifically to the participants. We could ask them to fill out a survey and tell us what their specific interests are (if any), or provide them with a numbered of predefined topics and ask for their preference. It is my experience that if using applications related to the participants' interests will help retain more people until the end of the semester.

Other thoughts on the workgroup format:

- ask participants to have a sample of data file that they normally work with?
- bi-weekly meetings; the in-between week could serve as a time for participants to come ask questions about the material we have covered?
- instead of having different people present every week, we can have one person do a short overview of the material, and then we turn it around to the participants to practice with guided examples?

Part II
Core Topics

1 Why learn to program?

- What is a program?
- The advantages of learning to program?
- Real life examples of projects one could do in Python

2 Introduction to Programming in Python

- What is Python and why should you use Python?
- Interacting with Python
  - IDLE/Spyder
  - Command Line

  - IPython Notebook *

    I strongly recommend the last option

- How to run a Python program?
- Data types and data structures in Python
  - Variables
  - Data types (including *None*)
  - Expressions
  - Data Structures
- Getting help and debugging your code
  - Understanding error messages
  - Modules (*import*)
  - Logging module
  - IDLE's Debugger

3 Lists, Dictionaries, and other Data Structures

- Slicing
- *Len*
- *Index*
- *Append*
- *Ins*ert
- *Remove*
- *Sort*
- *Keys*
- *Values*
- Nesting

4 Manipulating Strings and Regular Expressions

- *Slicing*
- *Indexing*
- *Upper/lower*
- *isX*
- *Startswith/endswith*
- *Join/split*
- *Strip/rstrip/lsplit*
- Regex (*search*, *groups*, *findall*)

5 Functions

- Defining functions
- Local/global variables
- Error handling (including *sys.exit*)

6 Flow Control and Top-Down Programming

- Boolean velues and operators
- Comparison operators
- Conditional statements (*if*, *elif*, *else*)
- Loops (*for*, *while*)
  - *break*, *continue*, *range*
- Top-down programming principles

7 Reading/Writing Files and Organizing Project Directories

- File/directory paths (absolute vs relative paths, OSX vs Windows)
- Working directory
- Directory contents
- *Open (r, w, a)*
- Shutil

8 Plotting in Python

- Matplotlib

Part III
Possible Advanced Topics

- Data manipulation with Pandas (I think this could easily be a core topic depending on the audience)
- Web scrapping
- Working with Excel spreadsheets, PDF, Word files, CSV files, JSON Data
- Keeping time, scheduling tasks, and launching programs
- Sending email and text messages
- Manipulating images
- Controlling the keyboard and mouse with GUI automation
- More...