SOCS Learning Module:
Introductory Programming with Python
(Approved by SoCS Council on June 27, 2017)

Intended Audience

This learning module is intended for Ph.D. students in the School of Computer Science who have little to no previous experience in writing code and specifically no experience with coding in the Python programming language.

Learning Objectives

Core Objectives

After completing this learning module, the student should be able to:

1. Run Python code from within the Python interpreter
2. Run Python code from the terminal in "batch mode"
3. Read high-level imperative-style Python code
4. Write simple to medium-level imperative-style Python code, including
   a. conditional expressions and branching statements
   b. iterative and looping constructs
   c. using predefined functions and methods from the Python standard library
   d. organizing code into procedures through user defined functions
5. Read and write simple Object-Oriented style Python code
6. Debug simple to medium-level Python code
7. Use simple data types such as lists and dictionaries (hash-tables)
8. Process user input, and read and write to a text file
9. Include and use functions/methods from external Python libraries
10. Write code using proper Python-style formatting and programmatic choices

Additional Objectives

As provided by the student's Advisory Committee

- Should learning cover Python 2 only, Python 3 only, or both?

Learning Resources

Recommended Textbook

Python Crash Course: A Hands-On, Project-Based Introduction to Programming
by Eric Matthes, No Starch Press (2015)

Chapters 1-12 inclusive (Chapters 13 & 14 also recommended)

This textbook is designed around exercises.
The student is encouraged to attempt most exercises of Chapters 1-11 and especially the extended project of Chapter 12.

Other Recommended Resources

Python tutorial from the Python language home page

- **https://docs.python.org/3/tutorial/**

A step-by-step tutorial that allows you to learn Python by writing Python

- **https://www.codecademy.com/learn/python**

Another interactive tutorial, including video instructions

- **http://www.learnpython.org/**

A list of Python resources (including some of the above tutorials)

- **http://python.berkeley.edu/resources/**

Additional Resources

As recommended by the student's Advisory Committee

Additional Exercises

As recommended by the student's Advisory Committee

Appendix: Learning Objectives as Covered in the Recommended Textbook

Chapter 1: Getting Started

- 1 Run Python code from within the Python interpreter
- 2 Run Python code from the terminal in "batch mode"

Chapter 2: Variables and Simple Data Types

- 3, 4 Read high-level and write simple to medium-level imperative style Python code
- 6 Debug simple to medium-level Python code

Chapter 3: Introducing Lists

- 7 Use simple data types such as lists and dictionaries (hash-tables)
- 3, 4 Read high-level and write simple to medium-level imperative style Python code
    - 4c using predefined functions and methods from the Python standard library
- 6 Debug simple to medium-level Python code

Chapter 4: Working with Lists

- 7 Use simple data types such as lists and dictionaries (hash-tables)
- 3, 4 Read high-level and write simple to medium-level imperative style Python code:
    - 4b iterative and looping constructs

Chapter 5: `if` Statements

- 3, 4 Read high-level and write simple to medium-level imperative style Python code:
    - 4a conditional expressions and branching statements

Chapter 6: Dictionaries

- 7 Use simple data types such as lists and dictionaries (hash-tables)
- 3, 4 Read high-level and write simple to medium-level imperative style Python code:
    - 4b iterative and looping constructs

Chapter 7: User input, and while-loops

- 8 Processing user input, and reading and writing to a text file
- 3, 4 Read high-level and write simple to medium-level imperative style Python code:
    - 4b iterative and looping constructs

Chapter 8: Functions

- 3, 4 Read high-level and write simple to medium-level imperative style Python code:
    - 4d organizing code into procedures through user defined functions

Chapter 9: Classes

- 3 Read high-level imperative style Python code
- 5 Read and write simple Object-Oriented style Python code

Chapter 10: Files and Exceptions

- 3 Read high-level imperative style Python code
- 8 Process user input, and read and write to a text file

Chapter 11: Testing your Code

- 3 Read high-level imperative style Python code

Chapters 12, {13, 14}: Project 1 (Aliens)

- 3 Read high-level imperative style Python code
- 5 Read and write simple Object-Oriented style Python code
- 9 Include and use functions/methods from external Python libraries