Python review

The sections below indicate Python material, the degree to which it will be used in the course, and various resources you can use to review the material. You are not expected to use all the resources, just the ones that suit your needs most. The course website lists additional Python resources, should you find them a preferable way to review the material.

One possible way of reviewing is to try the exercises in Python from scratch, and then to use other resources to review any material that requires a bit more effort.

If your previous exposure to Python was version 2, please see the course website for a link to a summary of differences between Python 2 and Python 3.

Note: The sections below follow the order in which the material should be mastered for CS 234, not necessarily the most logical order in which to master the material if you are studying Python for the first time or are in need of extensive review. In such a case, it might be better to follow Python from scratch in order, reading the material on files as needed to ensure that you master the material by the dates given.

1 Python basics

You will be expected to know how to create and use variables and functions, and to import modules such as the math module and modules provided for assignments.

Although you will not be expected to create objects until later, at this point you should be familiar with dot notation so that you can use methods on objects.

Relevant information includes:

- def
- return
- import (use from xx import * for a user-defined module xx.py)
- None

References

CS 116 notes

- Module 01: Introduction to Programming in Python
  - Basic types and mathematical operations
  - Calling functions
  - Defining functions
– Mutation
– Changes in design recipe

Python from scratch

• Module 1: First steps
• Module 2: Built-in functions
• Module 3: Storing and using information
• Module 4: Creating functions

Necaise

• Appendix A.1 The Python Interpreter
• Appendix A.2 The Basics of Python
• Appendix A.7 User-Defined Functions

2 Program structure, input, and output

2.1 Booleans, conditionals, branching

Be sure that you can write conditional expressions to allow branching in your programs. Relevant information includes:

• True, False
• ==, !=, <, >, <=, >=,
• if, elif, else

References

CS 116 notes

• Module 02: Making Decisions in Python
  – Boolean expressions
  – Conditional statements

Python from scratch

• Module 5: Booleans
• Module 6: Branching

Necaise

• Appendix A.4.1 Selection Constructs
2.2 Iteration

Iteration (while and for loops) will be used extensively in the course; be sure that you are comfortable with both concepts.

Relevant information includes:

- while
- for
- range
- break
- continue

References

CS 116 notes

- Module 06: Iterative Structure in Python
  - while loops
  - for loops

Python from scratch

- Module 8: Iteration using while
- Module 10: Iteration using for

Necaise

- Appendix A.4.2 Repetition Constructs

2.3 Strings and user interaction

We will use only basic string methods. You can look up string methods as you need them rather than investing time in reviewing them all at the beginning of the course.

User interaction can come in handy in testing out your implementations, allowing you to test your code by executing various operations. Make sure to use int if needed to convert the string input into an integer.

Relevant information includes:

- input
- int
References

CS 116 notes

- Module 03: Strings and Input/Output
  - Printing to standard output
  - Strings and their methods
  - Reading from standard input

Necaise

- Appendix A.5.1 Strings
- Appendix A.3 User Interaction

2.4 Files

At times we will use files to store data items that we will add to an ADT. Be sure you can open and close a file, read and write lines, strip blank spaces, and divide an input string into a list of items (based on separation by blank spaces).

Relevant information includes:

- open
- close
-.readlines
- strip (string method)
- split (string method)

References

CS 116 notes

- Module 10: File Input and Output

Necaise

- Appendix A.6 Text Files
3 Classes

We will use classes extensively in the course. You should be able to create a class, create an object, and write and use methods.

Relevant information includes:

- `class`
- `__init__`
- `self`

References

CS 116 notes

- Module 09: Additional Options for Organizing Data
  - Classes

Python from scratch

- Module 11: Building information into objects

Necaise

- Appendix D Classes

4 Recursion and more data types

4.1 Recursion

We will use recursion, though it will not be a major emphasis of the course. Much of Module 05 of CS 116 is beyond what is needed in this course.

References

CS 116 notes

- Module 02: Making Decisions in Python
  - Recursion
- Module 05: Types of Recursion
  - Purely structural recursion
  - Accumulative recursion
Generative recursion

Python from scratch

- Module 13: Recursion

Necaise

- Appendix A.4.1 Selection Constructs

4.2 Lists

Although the types of operations used in lists will be a focus of the course, we will not use lists as extensively as they were used in CS 116. You should be comfortable with the idea of methods that mutate the input and methods that do not, but you do not need to put time into extensive review of list methods.

Relevant information includes:

- []
- accessing a particular list item by index
- +
- append

References

CS 116 notes

- Module 04: Lists
  - Lists and their methods
  - Mutating lists
  - Abstract list functions

Python from scratch

- Module 9: Storing elements in a sequence

Necaise

- Appendix A.5.2 Lists
4.3 Tuples and dictionaries

Neither tuples nor dictionaries will be used extensively in the course. Relevant information includes:

- \[
- \]
- extracting a particular list item
- +
- append

References

CS 116 notes

- Module 09: Additional Options for Organizing Data
  - Dictionaries

Python from scratch

- Module 12: Structuring data

Necaise

- Appendix A.5.3 Tuples
- Appendix A.5.4 Dictionaries