CS 234
Naomi Nishimura

Python guide 1

The sections below (and those in Python guide 2) indicate Python material, the degree to which it will be used in the course, and various resources you can use for review. 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.

If you feel that you could benefit from review of basic programming concepts and basic Python, you might consider a book such as "Murach's Python programming". It does not cover more advanced concepts such as `map`, `filter`, and `reduce`, but slowly and carefully covers the basics.

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 and then pick up the more advanced material, ensuring you do so by the deadlines specified for mastering material for assignments.

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. You will also import the module `check` to create tests for your functions.

Although the creation of objects will not be discussed until a later section, at this point you should be familiar with dot notation so that you can use methods and extract attributes.

Relevant information includes:

- `def`
- `return`
- `import xx` to import module `xx.py` and use the name of the module and dot notation to use the functions in it
- `from xx import *` to import from module `xx.py` and not need to use the name of the module to use the functions in it
- `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 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`
- use of `is` instead of `==` to compare `0` and `False`

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

3 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

4 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

5 Module `random`

The module `random` can be used to generate random integers in the range from `lower` to `upper` (inclusive), as `random.randint(lower, upper)`.
Relevant information includes:

- `random.randint`

6 Python data types

6.1 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

6.2 Tuples and dictionaries

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

- {}
- `[]`
- extracting a particular 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