CS 231 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 231, 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 both built-in modules (like the math module) and modules provided on the course website (like the grids module). 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

2 Booleans, conditionals, branching

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

- True, False, and, or, not
- $\bullet \ ==, \ !=, \, <, \, >, \, <=, \, >=,$
- 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

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

4 Lists

Although we will use lists in the course, we will be judicious in our choice of operations to use, as we wish to ensure that we are able to analyze the running times of our algorithms. Accordingly, please make sure that you choose operations mentioned explicitly in the assignment or exam or in the list of costs of functions available on the course website.

In reviewing list methods, make sure that you know whether or not a method mutates the input. If you produce None, you are probably mistaking a method that mutates the input for one that produces an output. If you do not wish to have your function mutate the input, you might use copy.deepcopy from the copy module to ensure that you are making changes to a copy of the data rather than the original input.

We will use the module itertools to form permutations and combinations of elements of a list. The function itertools.permutations takes two inputs, where the first is a list or a range, and the second is an integer length. The function will produce all permutations of the given length, taken from the list or range, with no item appearing more than once. This creates a permutations object; use the function list to convert it to a list as well as to convert each permutation into a list. For example, you can obtain a list of lists of all permutations of the numbers 1, 2, and 3 of length two using list(itertools.permutations([1,2,3],2)).

To create all combinations of a certain size (distinct sets, where the order doesn't matter), you can use itertools.combinations, which again takes as input a list or a range and an integer length.

Here are more examples of the use of the functions:

```
import itertools
print(list(itertools.permutations(range(4),2)))
print(list(itertools.combinations([1,2,3],2)))
print(list(itertools.combinations(range(4),2)))
```

Additional list methods will be discussed in Python guide 2. Relevant information includes:

- []
- accessing a particular list item by index
- list
- in
- slices
- +
- len
- append
- insert
- pop
- remove
- itertools.permutations
- itertools.combinations
- isinstance is used to determine if a variable is a list, producing True or False. An example of its use is isinstance(my_data, list); it can also be used with another type instead of list.

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

5 Strings

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. *Relevant information includes:*

- accessing a particular character by position
- slices
- int

References

CS 116 notes

- Module 03: Strings and Input/Output
 - Strings and their methods