CS 231
Naomi Nishimura

Python guide 2

1 Recursion

We will use recursion in designing algorithms.

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

2 Structuring information, random, and files

2.1 Classes

You may be required to create simple classes in your assignments. In this course, we will not be concerned with being able to hide attributes and methods.
Relevant information includes:

- `class`
- `__init__` to specify a function that creates an object
- `__repr__` to specify how an object is represented as a string, such as when it is printed
- `__eq__` to define = for objects in this class
- `self`
- `isinstance` to determine if an object is in a class

References

CS 116 notes

- Module 09: Additional Options for Organizing Data
  - Classes

Python from scratch

- Module 11: Building information into objects

2.2 Lists (advanced)

After using `map` or `filter`, use the function `list` to convert the output to a list.

When sorting a list of objects, it is possible to sort by a particular attribute. For example, to sort a list of grids by the number of rows in each grid, one can create a helper function that extracts the number of rows and then sort using the name of the helper function as a `key`, as follows:

```
def extract_rows(item):
   return item.rows

sorted(grid_list, key=extract_rows)
```

A `key` can also be specified for the method `sort`, and for either `sorted` or `sort` it is possible to sort in nonincreasing order, using `reverse=True`, as in the following:

```
sorted(grid_list, key=extract_rows, reverse=True)
```

Relevant information includes:

- `list`
- `map`
- `filter`
- `sort`, where `a_list.sort()` mutates `a_list`
- `sorted`, where `sorted(a_list)` produces a new list
- `key`
- `reverse`

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

2.3 Tuples and Dictionaries

The use of tuples and dictionaries in this course is limited. Be careful that you are able to determine the costs of any function or method you use.

Relevant information includes:

- {}
- `keys()`
- `append`

References

CS 116 notes

- Module 09: Additional Options for Organizing Data
  - Dictionaries

Python from scratch

- Module 12: Structuring data

2.4 Module random

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

Relevant information includes:

- `random.randint`

2.5 Files

Although in the course you will be using input stored in files, the functions used to read input and store it as an object will be provided for you. The material in this section is optional. If you are interested, you can learn how to 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