Gabriela Tedeschi

September 05, 2021

IT FDN 110

Final Assignment

# Lessons Learned

## Introduction

This class was my first exposure to Python and in general, my programming experience was limited. Overall, I think I learned quite a bit about programming and I enjoyed the process. Below, I’ll discuss some of the things I learned throughout the course.

## Reference vs. Value Types

I learned that most programming languages support two types of variable data storage. Value types store their data in the part of memory where the program is running (called the program’s stack) and reference types store their data separately from the application in the runtime area (called the heap). Though in Python all variables are reference types, some behave like value types and some behave like reference types. It’s important to be aware of this and test that variables are behaving as you expected.

## Separation of Concerns

I learned that the separation of concerns is a design principle that tells us to separate the code in our program into distinct sections that each address a separate concern. This makes scripts easier to work with because you know where to identify issues that deal with a given concern and when making changes that deal with a given concern, you minimize the impact it will have on the rest of your script. It is also helpful if you want to copy functions that deal with a given concern to create a new script. Most programs can be divided into Data, Processing, and Presentation (Input/Output) sections.

## Classes

Classes, which are used to group data and functions, allow us to separate concerns. Most classes are designed to focus on either storing data or processing data. You can either use the code in a class directly or indirectly. When you want to use a function from a class directly, you use the @staticmethod command and simply call the function in the body of your script. When you want to use a function from a class indirectly, you create an object instance of the class. One benefit of using a class indirectly is that you can create multiple object instances. Generally, you use classes indirectly when your purpose is storing data and object instances, like rows in a table, store each data entry. You generally use classes directly when your purpose is processing.

## Specific Features/Tasks

I learned how to:

* read data from a file and write to a file
* capture user input and compile data in lists, dictionaries, and tuples
* use while loops, for loops, and conditional statements
* build error handling into my programs so the program doesn’t break if the user enters invalid data

## Conclusion

After taking this course and Randal’s SQL course, I am very interested in building my SQL and Python skills as well as exploring other programming languages. I believe the skills I gained this quarter will serve as a good foundation for learning other languages. Thank you, Randal and Sophia, for the great course!