Exercise 1.3: Functions and Other Operations in Python

Learning Goals

- Implement conditional statements in Python to determine program flow
- Use loops to reduce time and effort in Python programming
- Write functions to organize Python code

Reflection Questions

1. In this Exercise, you learned how to use **if-elif-else** statements to run different tasks based on conditions that you define. Now practice that skill by writing a script for a simple travel app using an **if-elif-else** statement for the following situation:

    - The script should ask the user where they want to travel.
    - The user's input should be checked for 3 different travel destinations that you define.
    - If the user's input is one of those 3 destinations, the following statement should be printed: "Enjoy your stay in ______!"
    - If the user's input is something other than the defined destinations, the following statement should be printed: "Oops, that destination is not currently available."

Write your script here. *(Hint: remember what you learned about indents!)*

```
destination = str(input("What country would you like to visit?: "))

destination.capitalize()

if destination == "Greece" or destination == "Italy" or destination == "Australia":
  print("Enjoy your stay in", destination)
else:
  print("Oops, that destination is not currently available.")
```

2. Imagine you're at a job interview for a Python developer role. The interviewer says "Explain logical operators in Python". Draft how you would respond.

The logical operators in Python are AND, OR, and NOT. They are used for conditional statements and unlike some other languages, the syntax for python is to actually write out the word. When using and/or a condition is required on both sides of the operator, the not operator is different. As it is used to reverse the result of a logical expression that comes after it, it is written before the condition.

3. What are functions in Python? When and why are they useful?

Python has many built in functions that can be used, but it is also possible to define a custom function that you can use later on/repeatedly in a script. They are sets of instructions that process or manipulate your code in order to achieve certain things. Functions are useful because they allow for multiple operations to be conducted at once, and they make it easy to reuse the code as many times as you might want within the script.

4. In the section for Exercise 1 in this Learning Journal, you were asked in question 3 to set some goals for yourself while you complete this course. In preparation for your next mentor call, make some notes on how you've progressed towards your goals so far.

My two main goals that were listed in exercise 1 were to better understand the capabilities of Python, and to be able to create a Python application from scratch. As I continue to work with Python I am getting a clearer understanding of its capabilities. I now see how a script can be written to automate a specific, repetitive task that someone in virtually any professional field could take advantage of. I know I am still learning Python, but after this most recent exercise I am feeling far more confident in writing and testing scripts, which puts me well on my way to creating an application from scratch

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?

2. In this Exercise you learned about the pickling process with the **pickle.dump()** method. What are pickles? In which situations would you choose to use pickles and why?

3. In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?

4. Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?