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 = input("Where do you want to travel?: ")

if destination == "Greece" or destination == "Italy" or destination == "Spain":
 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.

    a. Logical operators in Python are intended to validate multiple conditions that return True or False as an overall result. The operators are "and", "or", and "not".

        i. With the "and" operator all conditions must be True so that the overall result is True.

        ii. With the "or" operator, at least one of the conditions must be True so that the overall result is True, otherwise it's False.

        iii. The "not" operator is used to reverse the result of a logical expression that comes after it, such as not time > 10 (if time is not greater than 10, it will return True).

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

    a. The goal of functions in Python is to define steps that can be reused multiple times in multiple places in the code without defining those steps again. Therefore, it helps to create more concise and reusable code.

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.

    a. I think so far the course has been focused on the fundamental concepts of Python that I've been more less familiar with therefore I haven't progressed much towards any advanced knowledge. But I am certainly progressing in terms of building a functional app, that now has some basic functions to receive recipes and format responses.