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!)*

```
user_destination = str(input("Where do you want to travel?"))

if user_destination == "Barcelona":
    print(f"Enjoy your stay in {user_destination}!")
elif user_destination == "Antalya":
    print(f"Enjoy your stay in {user_destination}!")
elif user_destination == "Montevideo":
    print(f"Enjoy your stay in {user_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.*

Logical operators in Python are used to perform logical operations on Boolean expressions. There are three logical operators: and, or, and not. The *and* operator returns True if both operands are True, otherwise it returns False. The *or* operator returns True if at least one operand is True, otherwise it returns False. The *not* operator returns the opposite of the Boolean value of the operand.

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

Functions in Python are reusable blocks of code that perform specific tasks. They help break down complex problems, improve code organization, and promote reusability. Functions allow you to encapsulate logic, making it easier to debug and maintain your 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.*

I have made progress towards my goals by learning the basics of Python. I am excited to continue learning and exploring more complex concepts in Python.