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? ')
destination = destination.strip().capitalize()
if destination == 'Berlin' or destination == 'London' or destination == 'Paris':
  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.**

    Logical operators in Python are used to check for multiple conditions at the same time (operators **and** and **or**) or to reverse the result of a logical expression (operator **not**).
    The **and** operator checks if both of the request conditions are met (if either of the conditions isn't met the output will be False; if both of the conditions are met the output will be True).
    The **or** operator checks if either, not both, of the requested conditions are met (if one of the conditions is True, and the other is False, the output will still be True).

The **not** operator doesn’t require a condition on the left side because it’s not comparing two conditions and how they relate to each other; it flips whether or not a conditional statement was True or False.

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

   Functions in Python are sets of instructions that process or manipulate code to achieve certain things. They are useful when we have steps for an operation in our code that we need to repeat several times. Functions save time and keep our code clean and concise.

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. Gain more solid programming and coding concepts.
      i. Although Python has its specificities many of the concepts presented in Exercises 1.1, 1.2 and 1.3 are already familiar to me from JavaScript.

   b. Learn Python fundamentals.
      i. Prepared developer environment for programming with Python (installation, virtual environments, Python’s interactive shell, package management, iPython shell)
      ii. Learned about data types (variables, data types, objects – scalar and non-scalar).
      iii. Learned about operators and functions (conditional statements, loops, functions).

   c. Get a comprehensive understanding of the benefits of developing with Python.
      i. In Exercise 1.1 I learned about the uses and benefits of Python for web development.