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 would you like to travel to?')

if destination ==  'San Diego':
  print('Enjoy your stay in ' + destination)

elif destination == 'Mexico City':
  print('Enjoy your stay in ' + destination)

elif destination == 'Manchester':
  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 are used to combine conditional statements allowing you to perform operations based on different conditions. The logical operators are AND, OR, and NOT. AND returns true if both the operands are true. OR returns true if either of the operands are true. NOT returns true if the operand is false.

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

   Functions are reusable code used to perform a specific task. They are defined using the ‘def’ keyword followed by the <function name> and the () containing parameters for the function. Functions are useful for reducing redundant code, organization, and improved code readability. They allow you to take complex tasks and break them down into smaller tasks, improving code maintainability. Functions can also be called multiple times reducing time and improving efficiency.

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.

   1. Improving my understanding of python basics and fundamentals.
   2. Understanding of how to use prompts and write effective and performant code.
   3. Learning the differences between python and JS and using python formatting correctly.