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!)*

```
location = input('Choose to go to Alaska, Turks and Caicos, or Joshua Tree: ')
if location == 'Alaska' or 'Turks and Caicos' or 'Joshua Tree':
    print('Have a fantastic trip in' + location)
else:
    print('Sorry, that's not a travel option')
```

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 operate fairly simply. "Or", "Not" and "And" all function as booleans, returning either true or false responses. "And" returns true if all conditions are met, "Or" returns true if even one condition is met, and "Not" returns the opposite of what the output would be if the "Not" operator were not in place.

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

Functions in Python are similar to functions in many other languages. They are defined steps in a piece of code. They are useful particularly in their ability to be called over again without having to rewrite the original block of code over and over. They can serve to keep your code condensed and tidy.

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 feel I'm continuing to make progress on these goals, particularly in my comfort with using Python and even being able to explain why I may or may not prefer using it for any given project.