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!)*

```
want =  str(input('Where do you want to travel: ')
if want == 'Hawaii':
   print('Enjoy your stay in Hawaii!')
elif want == 'Paris':
   print('Enjoy your stay in Paris')
elif want == 'Niagra Falls':
   print('Enjoy your stay in Niagra Falls')
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 include and, or and not. These check for multiple conditions at one time. The 'and' operator checks if condition a and condition b are true and if so it returns the Boolean 'true.'

The ‘or’ operator checks if either condition a or b are true and if so it returns the Boolean ‘true.’ For the ‘and’ operator to be true both conditions must be met, for ‘or’ one or both of the conditions can be met to be true. The ‘not’ operator reverses the condition that is present.

3. What are functions in Python? When and why are they useful?
   There are a few examples of functions in Python. Some are built in such as len(), append() and print(). Some can be defined by user by putting def before the function. Functions are defined and call be called whenever required. This is useful when on block of code is needed in multiple places.

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 been learning more about Python and how to code with Python so far in this achievement. With this knowledge I am making strides to build an app with Python as well. I want to learn how to use Python to code and to be able to use it in the field.