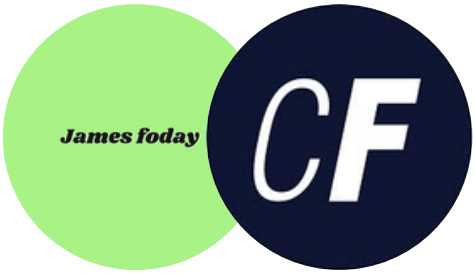
JAMES FODAY



REFLECTION QUESTION

Learning Journal - Exercise 1.3 Reflection Questions

1. Write a simple travel app script using if-elif-else statements:

The program asks the user where they want to travel and compares the input against three specific cities I chose. If the input matches any of these destinations, it prints a welcoming message. If not, it informs the user that the destination isn't available.

```
destinations = ["Paris", "Tokyo", "New York"]

destination = input("Where do you want to travel? ")



if destination == destinations[0]:

    print("Enjoy your stay in Paris!")

elif destination == destinations[1]:

    print("Enjoy your stay in Tokyo!")

elif destination == destinations[2]:

    print("Enjoy your stay in New York!")

else:

    print("Sorry, we don't offer trips to that destination right now.")
```

2. Explain logical operators in Python as if in an interview:

Logical operators help combine multiple conditions and control the flow of decisions in Python programs.

- The **and** operator returns True only if *both* conditions are true. For example, x > 10 and x < 20 checks if x is between 10 and 20.
- The **or** operator returns True if *at least one* condition is true, like x < 5 or x > 15.
- The **not** operator reverses a condition, so not(x == 5) is True if x is not 5.

These operators are essential for complex condition checks within if statements and loops.

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

Functions are reusable blocks of code that perform a specific task. They allow programmers to avoid repeating code by writing it once and calling it whenever needed.

Functions improve readability and maintainability by isolating logic into named blocks. For example:

```
def add_numbers(a, b):
    return a + b
```

Using functions helps organize programs into smaller, testable parts, which is important for larger projects and collaborative development.

4. Progress towards my course goals so far:

I have strengthened my understanding of Python basics such as control flow and data handling. I'm getting comfortable with writing clean, modular code and applying concepts like conditionals and functions effectively. My next steps include learning about error handling and object-oriented programming in more depth, which will prepare me for building more complex applications like web apps with Django.