

February 1st, 2025

1. What experiences have you had with coding and/or programming so far? What other experiences (programming-related or not) have you had that may help you as you progress through this course?

Answer: I have completed the Full-Stack Immersion part of the Web Development program with Career Foundry. Prior to that, I did not have much experience at all. I had done a month of the FreeCodeCamp program which is what sparked my interest in programming and led me consequently to CareerFoundry.

2. What do you know about Python already? What do you want to know?

Answer: Besides the fact that it is a programming language and a very popular one, I don't know anything about it.

3. What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.

Answer: I believe keeping myself motivated may be a challenge as the content of the course can sometimes present us with very complex ideas. I try to keep myself motivated by thinking that in a few years, I will be glad I started right now. I also try to think of the possibilities that completing this course will bring me and what possible doors might open when I can finally and confidently say that I am a web developer. But most of all, I know the importance of breaks and coming back to the course with a clear mind.

EXERCISE 1.1

1. In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working on?

Answer: To put it shortly: Frontend is what users interact with on an application or web-page, while the backend is what happens backstage and makes the program work. In other words, all of the work that users don't need to see in order to interact with the platform.

2. Imagine you're working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?

Answer: The similarity between JavaScript and Python is that both are object oriented (which means they both work with sets of code and data). Also, both work with variables and functions and give the developer the ability to work with classes and properties. On the other hand, the main differences lie on:

- Speed: where JavaScript tends to perform faster,,
- Syntax: with Python offering a simpler syntax than JavaScript.

When it comes to convincing my team to use Python, I might argue that Python tends to be more suitable for large-scale projects and that working with Python for backend allows us to have versatility due to its very simple language in terms of syntax. I might also add that there are a lot of different frameworks available and resources on how to work with python; besides a strong and collaborative community online.

3. Now that you've had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can react on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?

Answer:

1. I want to learn enough about Python so that I can work with it and discuss it confidently with my future co-workers. I want to be able to understand and follow guidelines on possible Python documentations I may come across.
2. By working on this achievement I hope to get a good introduction about Python so that I can grow my confidence in web development and reach my goal in point number one.
3. After the completion of this achievement, I see myself applying for Technical Writing jobs that will allow me to grow my knowledge even more. I also hope to work on smaller projects for myself and my portfolio using the skills that I have acquired throughout the course.

EXERCISE 1.2

1. Imagine you're having a conversation with a future colleague about whether to use the iPython Shell instead of Python's default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

Answer: To put it shortly, the iPython Shell is more practical and user-friendly than the default one. IPython's syntax highlighting makes it easier to read the code and it indents the code automatically. IPython Shell is also faster when it comes to testing small pieces of code.

2. Python has a host of different data types that allow you to store and organize information. List 4 examples of data types that Python recognizes, briefly define them, and indicate whether they are scalar or non-scalar.

Data Type	Definition	Scalar or Non-Scalar?
int	Represents integers, includes both negative and non-negative numbers.	Scalar
float	Holds decimal numbers and includes both negative and non-negative decimal numbers.	Scalar
bool	Represents boolean where data can only be either True or False.	Scalar
dictionary	Unordered set of items that requires a key-value pair.	Non-Scalar

3. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.

Answer: The difference between Lists and Tuples lies in the fact that lists are mutable and tuples are not. Elements within a list can be modified, deleted or rearranged.

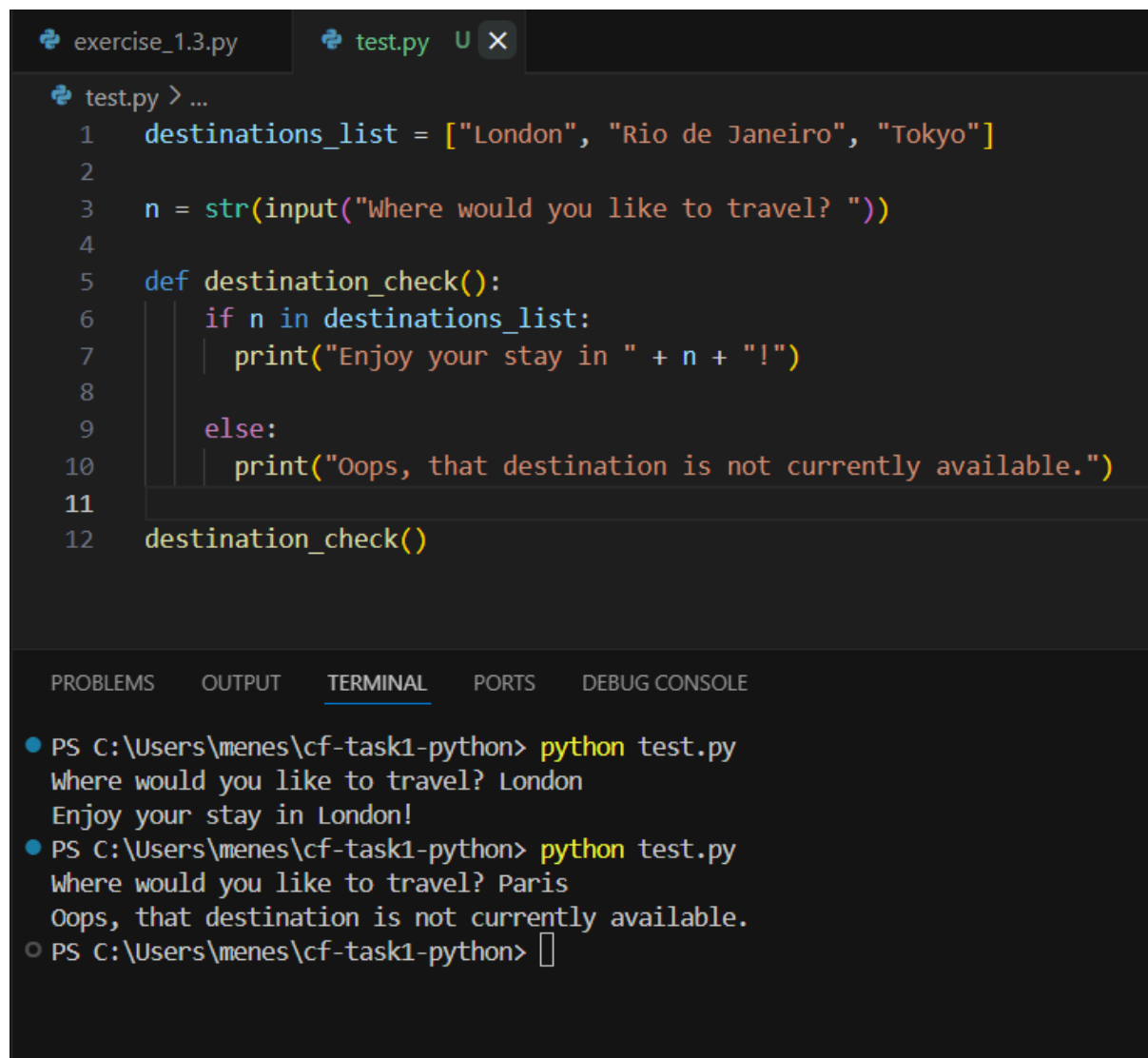
4. In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language-learning app beyond vocabulary memorization.

Answer: I would choose to go with dictionaries as they don't require data to be sequential. This particular feature of dictionaries may come in handy as we continue to develop and grow the app's data.

EXERCISE 1.3

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."



The screenshot shows a code editor with two tabs: `exercise_1.3.py` and `test.py`. The `test.py` tab is active, showing the following Python code:

```
test.py > ...
1 destinations_list = ["London", "Rio de Janeiro", "Tokyo"]
2
3 n = str(input("Where would you like to travel? "))
4
5 def destination_check():
6     if n in destinations_list:
7         print("Enjoy your stay in " + n + "!")
8
9     else:
10        print("Oops, that destination is not currently available.")
11
12 destination_check()
```

Below the code editor, the **TERMINAL** tab is selected, showing the execution of the script:

```
PS C:\Users\menes\cf-task1-python> python test.py
Where would you like to travel? London
Enjoy your stay in London!
PS C:\Users\menes\cf-task1-python> python test.py
Where would you like to travel? Paris
Oops, that destination is not currently available.
PS C:\Users\menes\cf-task1-python> 
```

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.

Answer: Logical operators in Python help you compare conditions in data. The operators are "and", "or" and "not", each having its own function when checking conditions that result in either a True or False response.

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

Answer: Functions in Python work as a set of instructions for the code to perform a task. Python has a set of predefined functions available, but developers can also write their own function. They are useful because they help maintain a clean and concise code, which is easier and faster to work with. They should be used when the code repeats itself, so by using function, your code will be shorter and more condensed.

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.

Answer: I would say I have made good progress towards my goals. I have learned a lot about Python so far and I am slowly feeling more comfortable with this new programming language. I look forward to learning more about this language and how it compares to other languages I know, such as JavaScript.

EXERCISE 1.4

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?

Answer: File storage is important because it allows us to permanently store data in our machines, as well as keeping track of values. If you didn't store local files, the data would be lost when the script stops running and couldn't be retrieved again.

2. In this Exercise you learned about the pickling process with the pickle.dump() method. What are pickles? In which situations would you choose to use pickles and why?

Answer: Pickles "convert complex data into a packaged stream of bytes". You would use pickles when you have a text file with complex structures (such as dictionaries), which is difficult to store in regular text - pickles convert this complex data and store it in a binary file for machines to read.

3. In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?

Answer: To find out which directory you're currently working in, you can use the `os.getcwd()` command. To change the directory in which you're currently working in, you can use the `os.chdir()` command.

4. Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?

Answer: I would use the `try-except` blocks to prevent my entire script from terminating.

5. You're now more than halfway through Achievement 1! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? Feel free to use these notes to guide your next mentor call.

Answer: I think the course is going well. I can tell it is getting more complex and I can potentially say I am moving past the basics with this programming language. I'm proud that I

have been able to understand the course and have managed to get multiple tasks approved. I believe I struggle with getting tasks started, but once I go back into the course and revise what I have just read, the instructions become clearer. I also believe I need more practice with organizing my initial thoughts on how to approach a task from the beginning.

EXERCISE 1.5

1. In your own words, what is object-oriented programming? What are the benefits of OOP?

Answer: Object-oriented programming (OOP) is a concept within web development where code is organized in chunks (called objects) within “classes” that can be reused; in other words, code is inherited from previously written code. The main benefit of OOP is that it avoids repetition of code and it therefore allows for a cleaner, more concise coding structure - as well as allowing a more flexible maintenance of such code (aka debugging).

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.

Answer: As mentioned in the course material: “Everything in Python is an object”. Objects contain data that is stored in attributes, while classes serve as templates with instructions on how to create objects. Classes define the structure of objects and contain methods and functions that dictate the object. A real-world example could be a bouquet of flowers, where classes would specify which flowers to use and how to arrange them, while the object would be the bouquet itself.

3. In your own words, write brief explanations of the following OOP concepts; 100 to 200 words per method is fine.

METHOD	DESCRIPTION
Inheritance	Inheritance in programming is when methods of a certain class can be reused in another without the need to re-write such methods. This avoids code repetition and makes for a cleaner and more concise coding structure. When dealing with inheritance, there is always a “subclass” or “inherited class” that uses data from the “parent class” or “base class”.
Polymorphism	Polymorphism is when different objects have the same name but their performance is different according to where this method is defined within your code. The perfect example to illustrate this concept in polymorphism is the len() method, which returns different results depending on the data type it is used with but possesses the same name at the end of the day.
Operator Overloading	Operator overloading is the process of defining methods for already existing operators such as “+” or “-”. Much like the len() method mentioned above, operators can perform differently depending on which data type they are manipulating. When working with operator overloading, you need to “define a function that Python already reserves for an operator” - these function definitions are then surrounded by double underscores (e.g. __add__).