CAREERFOUNDRY

Python for Web Developers Learning Journal

Objective

We find that the students who do particularly well in our courses are those who practice metacognition. Metacognition is the art of thinking about thinking; developing a deeper understanding of your own thought processes. With the help of this Learning Journal, you'll broaden your metacognitive knowledge and skills by reflecting on what you learn in this course.

Thanks to this Learning Journal, when you finish the course you'll have a complete and detailed record of your learning journey and progress over time. We really recommend that you take the time to complete this Journal; students do better in CF courses and in the working world as a result!

Directions

First complete the pre-work section before you start your course. Then, once you've begun learning, take time after each Exercise to return to this Journal and respond to the prompts.

There will be 3 to 5 prompts per Exercise, and we recommend spending about 10 to 15 minutes in total answering them. Don't overthink it—just write whatever comes to mind!

Also make sure that, once you've started filling this document in, you upload it as a deliverable on the platform. This is so that your mentor can also see your Journal and how you're progressing over time. Don't worry though—what you write here won't affect how you're graded for the Exercise tasks. The learning journal is mostly for you and your self-evaluation!

Pre-Work: Before You Start the Course

Reflection questions (to complete before your first mentor call)

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?

    a. **My recent experience with programming so far is in the Career Foundry bootcamp for JavaScript. I believe that my previous certification in web design from Design Labs might be beneficial for me to work through this course when it comes down to designing.**
2. What do you know about Python already? What do you want to know?
    a. **I honestly don't know too much about Python other than a lot of people say it's easier to understand once you've understood another language. Based on some opinions from my colleagues, they claim it's easier to comprehend compared to JavaScript. What I would like to know is how much of the world is running Python compared to JavaScript.**
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.
    a. **I think a big challenge for me is that since I'm using Python 3.12 and the achievement is using 3.8. I'm concerned about things that might be obsolete or may have changed. I believe a good way to combat this challenge is to probably read up on the documentation/release notes to see if there are any changes, and using Stack Overflow to see if there are any workarounds if something might not work. Similar to how I was working in the JavaScript course; I ran into an issue where certain operations in Angular 10 had changed in Angular 18, and I had to use Stack Overflow as a resource.**

Remember, you can always refer to Exercise 1.4 of the Orientation course if you're not sure whom to reach out to for help and support.

Exercise 1.1: Getting Started with Python

Learning Goals

- Summarize the uses and benefits of Python for web development
- Prepare your developer environment for programming with Python

Reflection Questions

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?

    a. **Front-end development focuses on the visible aspects of a site, including design and interactivity, using technologies like HTML, CSS, and JavaScript. Conversely, backend development handles the server-side processes, managing databases, servers, and application logic. Backend programmers engage in tasks such as database management, server-side coding, API development, security implementation, performance optimization, external service integration, and error handling.**

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?
   (Hint: refer to the Exercise section "The Benefits of Developing with Python")
    a. **Python stands out over JavaScript for web development due to its readability, dynamic typing, extensive package ecosystem, built-in essentials, and strong community support. With Python, developers find code easier to understand, variables more flexible, and access to a wealth of pre-built solutions for various tasks like mathematical calculations and database connections. Additionally, Python frameworks streamline common web operations, reducing development time, while its supportive community offers resources for troubleshooting and continuous improvement.**

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 reflect 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?
    a. **I want to build some real world projects with Python and compare it to JavaScript to see what might be my preferred language.**
    b. **I want to gain a solid understanding of the fundamentals concepts of Python so that I can write clean, efficient and maintainable code.**
    c. **I also want to explore possible specialization in the language itself.**

Exercise 1.2: Data Types in Python

Learning Goals

- Explain variables and data types in Python
- Summarize the use of objects in Python
- Create a data structure for your Recipe app

Reflection Questions

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?
    a. **IPython improves code readability with syntax highlighting, allowing for easier distinction between keywords and lines of code. It also automatically manages indentation for nested statements, which helps keep the code well-organized. In general, IPython offers a more user-friendly environment compared to Python's default shell.**

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?
Dictionary	**An unordered set of items that stores values and objects within itself indexed by identifiers, or keys.**	**Non-Scalar**
Tuple	**Linear arrays that can store multiple values of any type.**	**Non-Scalar**
List	**Similar to Tuple, but they are mutable; internal elements of a list can be modified or deleted.**	**Non-Scalar**
Bool	**Data type that represents one of two values: "True" or "False"**	**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.
    a. **Tuple and lists both store multiple values, however tuples are immutable while lists are mutable. This means that when tuples are created, they cannot be changed, whereas lists can still be modified by adding, removing, or updating after creating them.**
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.
    a. **For the language-learning app, dictionaries would be the best choice. They allow organizing vocabulary words with their definitions and categories efficiently, making it easy to access information when flipping through flashcards during quizzes. Dictionaries also offer flexibility for future enhancements like adding example sentences or user notes without needing to reorganize existing data. This adaptability**

and efficient data retrieval make dictionaries the optimal data structure for both current needs and potential app expansions.

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!)*

```
travel_destinations = ["california", "utah", "washington"]
state = input("Enter where you would like to travel: ")
if state.lower() in travel_destinations:
   print("Enjoy your stay in", state + "!")
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 when evaluating conditional statements. The `and` operator returns `True` if both operands are `True`. The `or` operator returns `True` if at least one operand is `True`. The `not` operator reverses the boolean value of its operand.

3. What are functions in Python? When and why are they useful?
    Functions in Python are defined using def, followed by a name and optional parameters in

parentheses. They encapsulate specific tasks, promoting code organization and reusability. They help facilitate by dividing complex tasks into manageable parts.

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.
    a. **Currently still working on understanding Python more before building a real-world app.**
    b. **Currently in progress with this course by doing the assignments and reading.**
    c. **Again, a work in progress from above. Also looking at examples of others that made Python apps.**

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?
    a. **It's crucial because without file storage, any data assigned to variables during script execution would vanish once the script concludes, rendering it inaccessible for future use. Employing file handling methods in Python allows this data to be saved into files, ensuring it remains available beyond the script's runtime.**

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?
    a. **Pickles are serialized byte streams used to store data. Through the pickle.dump() method, data is serialized into these pickles and written into binary files. They are particularly useful for handling complex data structures like dictionaries, allowing their storage in files. Pickles facilitate the retrieval and access of this data while maintaining its original structure.**

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?
    a. **I would utilize functions provided by the os module. To determine the current directory, os.getcwd() can be employed, while os.chdir() can be utilized to switch to a different directory.**

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?

    a. **I would attempt using try-except blocks. Within the try block, I'd place code that might encounter an error, while in the except block, I'd handle what to do if an error occurs. If no error arises in the try block, the remaining code continues execution normally without triggering the except block. However, if an error occurs, the code within the except block executes, preventing the 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.
    a. **The course is going smoothly so far. I feel like I understand the functions a lot more compared to Javascript, but still struggle with syntax a bit. However, I believe a bit more practice on my end when it comes to actually programming will make it more clear to myself.**

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

Reflection Questions

1. In your own words, what is object-oriented programming? What are the benefits of OOP?
    a. **Object-oriented programming organizes code into objects, each with its own attributes and methods. The benefits of OOP include making code reusable and easier to maintain and debug.**
2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
    a. **In Python, a class is a blueprint for creating objects. It defines the attributes and methods of objects. A real-world example would be a car, where a "car" class can be defined with attributes such as the `make`, `model`, `year`, and `color`, and methods would be `start`, `stop`, `accelerate`, and `brake`. The objects would be specifying if it is my car, or someone else's car.**

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

Method	Description
Inheritance	**A feature that enables a new class, known as a subclass or inherited class, to inherit attributes and methods from an existing class, referred to as a parent class or base class.**

	This allows the subclass to reuse code from the parent class and expand upon its functionality. By inheriting all attributes and methods of the parent class, the subclass can also introduce its own additional attributes and methods. This concept encourages the reuse of code and facilitates the extension of software capabilities.
Polymorphism	**Refers to the ability of methods or attributes to share the same name across different classes while performing different actions based on where they are implemented. Because these definitions are independent and distinct, there is no conflict between them.**
Operator Overloading	**Allows you to redefine how operators such as +, -, and others behave for custom classes. This means you can customize these operators to work with user-defined objects, in addition to their default functionality.**

Exercise 1.6: Connecting to Databases in Python

Learning Goals

- Create a MySQL database for your Recipe app

Reflection Questions

1. What are databases and what are the advantages of using them?

2. List 3 data types that can be used in MySQL and describe them briefly:

Data type	Definition

3. In what situations would SQLite be a better choice than MySQL?

4. Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?