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?

I completed the Career Foundry Intro and Immersion course in Full Stack Development. During the course, I gained experience with HTML, CSS, JavaScript, React, Angular, NoSQL (specifically mongoose), and SQL. Before that, I studied business administration, which included an introductory module in business informatics where I learned basic Python. Additionally, I attended a short university course focused on PHP and SQL.

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

I'm familiar with basic Python concepts like if-then statements and loops. I want to learn how to apply Python in real-world projects and how to integrate it with databases.

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.

I anticipate challenges with exercises that involve more math. To tackle them, I plan to work on them during late morning hours (9-12), when I feel most energized and focused.

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?

Frontend development focuses on creating the user interface that users interact with directly, while backend development handles the hidden operations that support the frontend, like interacting with servers and databases, processing data, and managing requests. If hired for backend programming, I'd focus on tasks such as database management, server-side logic, API integration, and ensuring smooth data processing and communication.

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")

Python and JavaScript both emphasize readability with clear syntax and keywords. Python, however, offers an extensive library of built-in packages, simplifying complex tasks like math operations and data manipulation. Its frameworks come with ready-to-use features for web development, like setting up web pages and databases. Plus, Python has a big community that helps with learning and fixing problems.

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?

- *I plan to apply Python to real-world projects, improving my web development abilities*
- *I'm interested in learning frameworks like Django to build dynamic web applications.*
- *Master Python essential concepts like variables, data types, loops, and functions to build a strong foundation.*