Python for Web Developers 

Learning Journal: Achievement 1

# 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.

The Learning Journal begins with some pre-course work and will then prompt you to step back and reflect once you’ve completed each Exercise. You’ll be asked questions relating to your learning process and be prompted to apply your knowledge to real-life professional scenarios.

Through the metacognitive prompts, the Learning Journal will not only help you understand how you learn and how you could improve over the course of the Achievement; it will also prepare you for future professional situations such as job interviews, in-job challenges, and more. You can also use the Journal to identify what you want to discuss with your mentor during your calls.

Thanks to this Learning Journal, when you finish the Achievement 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?
2. What do you know about Python already? What do you want to know?
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.

Remember, you can always refer to [Exercise 1.4](https://careerfoundry.com/en/steps/your-cf-team#receiving-support) of the Orientation course if you’re not sure whom to reach out to for help and support.

### Exercise 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?
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”)*

1. 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?

### Exercise 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?
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?** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. 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.
2. 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.

### Exercise 3: Functions and Other Operations in Python

**Learning Goals**

* Use conditional statements in Python to determine program flow
* Use loops to reduce time and effort in Python programming

**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!)*

|  |
| --- |

1. 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.
2. What are functions in Python? When and why are they useful?
3. 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.

### Exercise 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?
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?
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?
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 terminating due to an error?
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.

### Exercise 5: Object-Oriented Programming in Python

**Learning Goals**

* Discuss object-oriented programming concepts in Python
* 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?
2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
3. In your own words, write brief explanations of the following OOP concepts; 100 to 200 words per method is fine.

| **Method** | **Description** |
| --- | --- |
| Inheritance |  |
| Polymorphism |  |
| Operator Overloading |  |

### Exercise 6: Connecting to Databases in Python

**Learning Goals**

* Create a MySQL database for your Recipe app using SQL

**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** |
| --- | --- |
|  |  |
|  |  |
|  |  |

1. In what situations would SQLite be a better choice than MySQL?
2. Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?
3. Now that you’re nearly at the end of Achievement 1, consider what you know about Python so far. What would you say are the limitations of Python as a programming language?

### Exercise 7: Finalizing Your Python Program

**Learning Goals**

* Interact with a database using an Object Relational Mapper
* Build your final command-line Recipe app

**Reflection Questions**

1. What is an Object Relational Mapper and what are the advantages of using one?
2. By this point, you’ve finished creating your Recipe app. How did it go? What’s something in the app that you did well with? If you were to start over, what’s something about your app that you would change or improve?
3. Imagine you’re at a job interview. You’re asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.

1. You’ve finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:
   1. What went well during this Achievement?
   2. What’s something you’re proud of?
   3. What was the most challenging aspect of this Achievement?
   4. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
   5. What’s something you want to keep in mind to help you do your best in Achievement 2?

Well done—you’ve now completed the Learning Journal for Achievement 1. As you’ll have seen, a little metacognition can go a long way!