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?

Prior to the Python specialization course I've recently completed the CF immersion into Full Stack development. In this immersion course I focused on the MERN (MongoDB, Express, React, Node.js) framework. I also was introduced to Angular to develop a client-side interface for an API database app I created.

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

I didn't know much prior to taking this first exercise. I knew it was a popular framework and given the choice between the Python specialization and the Cloud Computing specialization I chose this course as it seemed imperative to understand prior to entering the job market for web development.
I want to understand the basics so that I can easily recognize Python and know when/how to use 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.

My main challenge is time. I am already behind in this program and I really want to finish without having to pay for an extension. However, I also want to take my time so that I truly understand the topics in this course. My plan right now to is complete the program on time and come back to review as much as needed once I have finished my projects.

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 focuses on what clients/users will see and interact with. Backend deals with the server-side and focuses logic, function, and performance of an app or site. If working on backend programming the

operations I would work on would include: ensuring security, managing databases and servers, and performance enhancements.

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")

The number one reason for using Python is the readability. Python uses easy to understand keywords to make commands so that those with basic coding experience can grasp the code. Another benefit is efficiency. Python comes with its own pre-installed operations such as URL routing, form handling and validation, template engines, database connections, web security, and session handling. No need to install another framework/library for this!

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?

Top 3 goals for this Achievement:

- Gain an excellent understanding of the basics and know where to look for help
- Know when/how to apply Python to future projects
- Be able to jump into an existing project utilizing Python and contribute

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?

The iPython shell is much more user friendly as it provides syntax highlighting to show you different features of your code in contrasting fonts and colors, and it automatically indents code for you unlike the default Python 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?
Integers (int)	Integers which negative and non-negative numbers (from zero to infinity)	Scalar
Tuples	linear arrays that can store multiple values of any type	Non-Scalar
Strings (str)	array of characters	Non-Scalar
Bool	Boolean: This data type stores either 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.

While they are similar, the difference between lists and tuples are that lists are mutable, meaning they can be modified or deleted. Tuples are immutable and cannot be modified. While the flexibility of a list may be better suited for certain data, tuples are much faster to read and access, especially with large sets of data.

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.

For this app I would use lists for the vocabulary words so that it's easy for users to search, new words can be added and/or modified to the list of vocabulary words as well. Within the vocabulary lists would be dictionaries containing more information about each word such as the definitions and categories. If I were to continue to develop the app I could then add data such as a rating system (Easy/Hard) for the words, conjugations for verbs, categories for short phrases (i.e. Travel, Greetings, Family, etc.). These sets of data could be added to the dictionaries of specific words where I could potentially use Booleans for the rating system, strings for the category of phrases, or tuples for the conjugations. The possibilities are endless!

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!)*

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

if destination == "Los Angeles":
   print("Enjoy your stay in Los Angeles!")
elif destination == "Rio de Janeiro":
   print("Enjoy your stay in Janeiro!)
elif destination == "Singapore":
   print(Enjoy your stay in Singapore!)
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 "not", "and", and "or" conditional statements. These statements check for multiple conditions at the same time so that you do not have to check whether a single condition is present one condition at a time. Thus, logical operators create more efficient code.

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

Python functions are just like JavaScript functions in that they are instruction to help process or modify your code to do something. In Python there are built-in ones like print(), len(), and append() but you can also create your own functions. These custom functions are useful because you can condense the built in functions in cases where would have to repeat the code multiple times. The custom functions will save you time!

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.

Top 3 goals for this Achievement (update):

- Gain an excellent understanding of the basics and know where to look for help
    - As I am going through these exercises I feel that I have a better understanding not only of Python but of JavaScript as well. I think I will need a little more practice on this particular exercise but it's starting to connect and sink in.
- Know when/how to apply Python to future projects
    - So far I know how to use Python for this project. With these examples I see how to use Python in different scenarios but I don't quite have a handle on when best to use Python vs. JavaScipt
- Be able to jump into an existing project utilizing Python and contribute
    - I am not here yet! But I feel like I would at least be able to recognize Python code in a given project.

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?

File storage is important for saving and retrieving data. If you didn't store in local files your data would no longer exist after running the script stops running. File storage allows you to share data and maintain it between sessions of running a program.

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?

Pickles are bytes that convert complex data into a binary file. The binary file stores the complex information in a way that can be read by a machine. Pickling is usefull when you need to preserve the

state of complex/large data objects or structures. It enables easy storage and retrieval, simplifies data sharing between different python programs, and facilitates communication between processes.

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?

You would use os.getcwd(). If you wanted to change your current working directory you would need to use os.chdir().

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?

I would implement the try-except block into my code to prevent the script from terminating because of an error. By placing my code within the try block I can include multiple except block to handle any specific exceptions that may occur or a general exception. If an error doe occur the script will continue to run, allowing me to handle errors with ease.

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.

This was tough! I definitely needed help from the examples of my peers' work. I try to start out on my own but my previous mentor stated that if it's taking me more than 20mins to figure it out, look for help; and so I did! I am proud that I am managing to get through here fairly quickly considering I am far behind on my timeline but some much needed review will be done once this is complete!

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?

Since everything in Python can be considered an object, object-oriented programming is taking the data and methods used to manipulate those objects, and placing them into classes. OOP allows the developer to avoid repeating code – reducing redundancy – which all helps to keep code efficient.

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

As stated, everything in Python is an object. Classes are used to organize those objects. It functions as a template that houses the structure of an object.

Example:
If Recipe is the class you can set the objects different objects that would create the class. The objects could be set to ingredients, cook time, and difficulty.

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 is used when you want to you have properties or methods of a class that would be useful in another class. So instead of copying all the code over, you can use the inheritance method to utilize the properties of one class in another. The class being inherited from is called the parent class and the class to which all data attributes and procedures are copied over is known as the subclass.
Polymorphism	This method is used when you want to use the same name for an attribute or method. The caveat here is that the attribute performs different operations depending on where it is defined. For example, you can have a class called Dog and a speak method that prints “Woof!” Using the same speak method under a class named Cat you can use the speak method to print “Meow!”
Operator Overloading	Typical operators such as “+” and “-“ cannot be used in custom classes. To use a typical operators in custom classes, you need to define your own methods for them. To do this you would need to define a function with a name that Python already reserves for your operator like “__add__”. Once defined you can then use the operator overload method within your custom class.

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?

5. 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 1.7: Finalizing Your Python Program

Learning Goals

- Interact with a database using an object-relational mapper
- Build your final command-line Recipe application

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.

4. You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:
    a. What went well during this Achievement?
    b. What's something you're proud of?
    c. What was the most challenging aspect of this Achievement?

d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
e. 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!

Pre-Work: Before You Start Achievement 2

In the final part of the learning journal for Achievement 1, you were asked if there's anything—on reflection—that you'd keep in mind and do similarly or differently during Achievement 2. Think about these questions again:

- Was your study routine effective during Achievement 1? If not, what will you do differently during Achievement 2?
- Reflect on your learning and project work for Achievement 1. What were you most proud of? How will you repeat or build on this in Achievement 2?
- What difficulties did you encounter in the last Achievement? How did you deal with them? How could this experience prepare you for difficulties in Achievement 2?

Note down your answers and discuss them with your mentor in a call if you like.

Remember that 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 2.1: Getting Started with Django

Learning Goals

- Explain MVT architecture and compare it with MVC
- Summarize Django's benefits and drawbacks
- Install and get started with Django

Reflection Questions

1. Suppose you're a web developer in a company and need to decide if you'll use vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?

3. Now that you've had an introduction to the Django framework, write down three goals you have for yourself and your learning process during this Achievement. You can reflect on the following questions if it helps:
    - What do you want to learn about Django?
    - 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 2.2: Django Project Set Up

Learning Goals

- Describe the basic structure of a Django project
- Summarize the difference between projects and apps
- Create a Django project and run it locally
- Create a superuser for a Django web application

Reflection Questions

1. Suppose you're in an interview. The interviewer gives you their company's website as an example, asking you to convert the website and its different parts into Django terms. How would you proceed? For this question, you can think about your dream company and look at their website for reference.
(*Hint: In the Exercise, you saw the example of the CareerFoundry website in the Project and Apps section.*)

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.

3. Do some research about the Django admin site and write down how you'd use it during your web application development.

Exercise 2.3: Django Models

Learning Goals

- Discuss Django models, the “M” part of Django’s MVT architecture
- Create apps and models representing different parts of your web application
- Write and run automated tests

Reflection Questions

1. Do some research on Django models. In your own words, write down how Django models work and what their benefits are.

2. In your own words, explain why it is crucial to write test cases from the beginning of a project. You can take an example project to explain your answer.

Exercise 2.4: Django Views and Templates

Learning Goals

- Summarize the process of creating views, templates, and URLs
- Explain how the “V” and “T” parts of MVT architecture work
- Create a frontend page for your web application

Reflection Questions

1. Do some research on Django views. In your own words, use an example to explain how Django views work.

2. Imagine you’re working on a Django web development project, and you anticipate that you’ll have to reuse lots of code in various parts of the project. In this scenario, will you use Django function-based views or class-based views, and why?

3. Read Django’s documentation on the Django template language and make some notes on its basics.

Exercise 2.5: Django MVT Revisited

Learning Goals

- Add images to the model and display them on the frontend of your application
- Create complex views with access to the model
- Display records with views and templates

Reflection Questions

1. In your own words, explain Django static files and how Django handles them.

2. Look up the following two Django packages on Django's official documentation and/or other trusted sources. Write a brief description of each.

Package	Description
ListView	
DetailView	

3. You're now more than halfway through Achievement 2! 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? You can use these notes to guide your next mentor call.

Exercise 2.6: User Authentication in Django

Learning Goals

- Create authentication for your web application
- Use GET and POST methods
- Password protect your web application's views

Reflection Questions

1. In your own words, write down the importance of incorporating authentication into an application. You can take an example application to explain your answer.

2. In your own words, explain the steps you should take to create a login for your Django web application.

3. Look up the following three Django functions on Django's official documentation and/or other trusted sources and write a brief description of each.

Function	Description
authenticate()	
redirect()	
include()	

Exercise 2.7: Data Analysis and Visualization in Django

Learning Goals

- Work on elements of two-way communication like creating forms and buttons
- Implement search and visualization (reports/charts) features
- Use QuerySet API, DataFrames (with pandas), and plotting libraries (with matplotlib)

Reflection Questions

1. Consider your favorite website/application (you can also take CareerFoundry). Think about the various data that your favorite website/application collects. Write down how analyzing the collected data could help the website/application.

2. Read the Django official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.

3. In the Exercise, you converted your QuerySet to DataFrame. Now do some research on the advantages and disadvantages of QuerySet and DataFrame, and explain the ways in which DataFrame is better for data processing.

Exercise 2.8: Deploying a Django Project

Learning Goals

- Enhance user experience and look and feel of your web application using CSS and JS
- Deploy your Django web application on a web server
- Curate project deliverables for your portfolio

Reflection Questions

1. Explain how you can use CSS and JavaScript in your Django web application.

2. In your own words, explain the steps you'd need to take to deploy your Django web application.

3. (Optional) Connect with a few Django web developers through LinkedIn or any other network. Ask them for their tips on creating a portfolio to showcase Python programming and Django skills. Think about which tips could help you improve your portfolio.

4. You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:
    a. What went well during this Achievement?
    b. What's something you're proud of?
    c. What was the most challenging aspect of this Achievement?
    d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

Well done—you've now completed the Learning Journal for the whole course.