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)

- 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 just finished the Full-Stack Immersion course and I am familiar with a JavaScript and a few of its frameworks in addition to HTML and CSS. After having completed the first two courses I have learned how I best absorb information and I think that will help me to progress through this course.**

- What do you know about Python already? What do you want to know?
- **I know that it is a scripting language similar to JavaScript and that it is easy to read and understand. I want to know how to use python for backend development.**

- 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 think the biggest challenge will be the actual learning of the language. I know that I can always reach out to my mentor with any questions if I get stuck.**

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

- 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 web development you are building the client side which is what the user sees and interacts with, and with backend web development you are working on things like the database, server side logic, API's, etc. If I were hired to work on the backend I would most likely be working on their API or database.**

- 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?
- **They are similar in that they are both dynamically typed languages and are both quite easy to read and understand. Python differs in that it comes built-in with a lot of useful libraries such as routing, form handling and validation, template engines, database connections, web security, and session handling which will save a lot of time during development as well as Python is built for quick deployment.**

- 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 want to know enough about python to begin working on and implementing it into my own projects**.
- **I want to learn the fundamentals of python during this achievement.**
- **I hope to be working as a backend developer at a company.**

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

- 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?
- **IPython is much more user friendly compared to Python's default shell. IPython has syntax highlighting which displays your features in contrasting fonts and colors, iPython also automatically indents nested statements for you. Another benefit of the iPython shell is that it lets you test out small chunks of code quickly and easily as each command is executed immediately after you type it, avoiding having to create separate files to test out small pieces of code.**
- 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	**This data type represents integers including negative and non-negative numbers from 0 to infinity.**	**Scalar**
float	**This data type holds decimal numbers and can store both negative and positive numbers.**	**Scalar**
bool	**This data type stores either of two values, True or False.**	**Scalar**
Dictionary	**This data type stores data in key-value pairings.**	**Non-Scalar**

- 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.
- **The difference is that lists are multiple whereas tuples are immutable.**
- 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.
- **I would use a dictionary as it would make the most sense in my opinion to store the data as key value pairs.**

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

- 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!)*

```python
travel_destination = input("Enter a city in the united states you
would you like to travel to: ")

if travel_destination in ["Florida", "Las Vegas", "Los Angeles"]:
  print("Enjoy your stay in", travel_destination)
else:
  print("Oops, that destination is not currently available.")
```

- 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 to combine conditional statements and evaluate boolean expressions, the three operators are <u>and</u>, <u>not</u>, <u>or</u>.**

- What are functions in Python? When and why are they useful?
- **Functions are reusable blocks of code containing instructions on what task or tasks to perform. They are useful for when you need to repeat a task multiple times such as taking user input for the details of several recipes.**

- 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.
- **I feel like I am really starting to understand the fundamentals of python, I am looking forward to learning more about how to use it on the backend.**

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

- Why is file storage important when you're using Python? What would happen if you didn't store local files?
- **File storage is important because without it data would wipe after the program ends.**

- 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 convert complex data into a packaged stream of bytes and then writes them into binary. They are best used when you need to store data locally.**

- 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?
- **The os.getcwd() command is used to find out which file directory you're currently in, and os.chdir() is used to change the working directory.**
-
- 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 use a try block for the code to run in so if there are any errors found by python the program won't terminate but will notify you of any errors.**

- 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.
- **I would say that it is going very well, I feel very proud to be able to put together a program using what I know so far.**

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

Reflection Questions

- In your own words, what is object-oriented programming? What are the benefits of OOP?
- **Object-oriented programming is a programming paradigm that organizes code around objects. The benefits of OOP are that code is very reusable through inheritance and it is well structured and organized making it easy to read and work with.**

- What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
- **Objects are an instance of a class that contains both data and the methods that can be performed on the data. A class is a template for creating objects, it defines the structure and behavior of the objects. An example of this could be a collection of movies. The movies class would contain the blueprint for the movies (title, year of release, director, genre, etc). And each individual movie object would contain its own specific details while still following the blueprint of the movies class.**

- 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 a fundamental concept in object-oriented programming where a new class, typically a child class, can inherit properties and methods from an existing class parent or base class.
Polymorphism	Polymorphism is when a data attribute has the same name across different classes, but performs different operations depending on where it is defined.
Operator Overloading	Operator overloading is a feature that allows you to redefine the behavior of standard operators such as +, -, *, /, ==, to work with custom classes.

Exercise 1.6: Connecting to Databases in Python

Learning Goals
- Create a MySQL database for your Recipe app

Reflection Questions

- What are databases and what are the advantages of using them?
- **A database is an organized collection of data that is structured for easy storage, retrieval, and management. The advantages of using databases include efficient data management, enhanced data integrity and security, and scalability.**

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

Data type	Definition
Int	This represents an integer
Date	This represents a date stored as yyyy-mm-dd
VarChar	This represents a string

- In what situations would SQLite be a better choice than MySQL?
- **SQLite would be better in situations where serverload would not be a concern, so I would say smaller projects are where it would most likely be preferred over MySQL.**

- Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?
- **I feel like Python is easier to work with compared to JavaScript at least in my opinion, at the same time JavaScript is where I learned the fundamentals of programming whereas in Python I am just learning the syntax of Python rather than the fundamentals. I also really like the readability of the syntax and the built in libraries and how easy it is to install libraries.**

- 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?
- **I would say with what I know now python is best used for backend and scripting, it doesn't really seem like you would be able to use it very easily on the frontend compared to JavaScript.**

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

- What is an Object Relational Mapper and what are the advantages of using one?
- **An Object Relational Mapper converts the contents and structure of a database into classes and objects that can be interacted with directly making it much easier to read and work with compared to SQL.**

- 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?
- **I would say it went well, I feel like I made it very responsive and user-friendly. If I were to start over I would try to create slightly better error handling in the event that a user messes up their input.**

- 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.
- **I have experience using python to build an application where a user can create, store, view, edit, and delete recipes stored in a SQL database all through the CLI.**

- You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:
    - What went well during this Achievement?
    - **I think I learned it pretty quickly because I am already familiar with the fundamentals of programming.**
    - What's something you're proud of?
    - **I am proud of how quickly I finished the first achievement.**
    - What was the most challenging aspect of this Achievement?
    - **I would say the object oriented part of the course as we only briefly learned it in the full-stack immersion course.**
    - Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
    - **It most certainly did give me the confidence to start building my own applications with python, it met my expectations.**
    - What's something you want to keep in mind to help you do your best in Achievement 2?
    - **Keep working through at the pace I am going because I am really enjoying just how much and what I am learning.**

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?
- **I would say I didn't really have a study routine, rather I just focused and carefully read the material to absorb it as best as possible, there isn't anything I will change.**

- 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?
- **I was proud of how quickly I got through it, I will try to carry that momentum into achievement 2 to learn and get better.**
- 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?
- **I struggled a little bit with OOP, but after reviewing I feel I understand it better now. I will make sure not to let any challenge prevent me from improving and learning more.**

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

- 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?
- **It would mainly depend on how big the project will be, if it is going to be a simple web app then perhaps look into a different framework but otherwise Django could be a perfect fit because of its scalability and built in features.**

- In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?
- **From what I can tell the most significant advantage MVT has over MVC would be its simplified handling of the controller logic, leading to faster and more streamlined web development.**

- 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?
- **I want to learn how to build my own projects using Django. After this achievement I hope to continue building my own projects and afterwards land my first developer job.**

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

- 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.
- **For this question I will use youtube, I would break it down into individual apps depending on their intended functionality, accounts for user login/signout, viewing profile, permissions. homepage showing the user videos, allowing them to upload videos, delete their own videos, etc.**

- In your own words, describe the steps you would take to deploy a basic Django application locally on your system.
- **First I would create a local environment for it and ensure that it is active, I would then install Django and start a new project using "django-admin startproject new_project". This would generate the basic file structure for the app I am working on. I would then run migrations to create the database. Then I would run it to ensure that it works, then create a superuser for an admin account.**

- Do some research about the Django admin site and write down how you'd use it during your web application development.
- **The Django admin panel is an interface where a user can read, update, and create on the model directly. It is an instant interface where users can manage the contents of their site using CRUD operations**

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

- Do some research on Django models. In your own words, write down how Django models work and what their benefits are.
- **From what I have read Django models are objects that are used by the application to interact with the database. They determine the structure of the data and its details.**

- 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.
- **Test cases help you really understand how your application works. It also helps to identify any problems early on during development which in turn prevents any bugs from appearing on the production build.**

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

- Do some research on Django views. In your own words, use an example to explain how Django views work.
- **Views in Django are where you set up how the code will look, basically how the models and classes will interact with the web page.**

- 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?
- **I would use function-based views in this scenario because wherever they are setup I would be able to reference them very easily.**

- Read Django's documentation on the [Django template language](#) and make some notes on its basics.
- **From what I have read it seems to be about how to setup dynamic html files, and how to use Django to access parts of your app easier.**

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

- In your own words, explain Django static files and how Django handles them.
- **Static files in Django are how Django interacts with html files which makes it easier to write clean and concise code.**

- 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	**Displays items as a list in a cyclical way to display each item.**
DetailView	**Displays items in a large view so you can see everything about an object.**

- 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.
- **I would say it's going very well and I am looking forward to the next exercises.**

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

- In your own words, write down the importance of incorporating authentication into an application. You can take an example application to explain your answer.
- **Incorporating authentication is necessary in my opinion because you don't want people getting into parts of your app that you don't intend them to.**

- In your own words, explain the steps you should take to create a login for your Django web application.
- **First create the login and logout views, then create the paths in 'urls.py' then link them to the html pages so that they are usable throughout the app.**

- 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()	**This will verify that the username and password both exist and match what is in the database.**
redirect()	**This wlll redirect you to another webpage once authentication has been verified.**
include()	**Includes a link to a specific path.**

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

- 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.
- **Thinking about an application that displays basketball statistics, it would be important to collect and analyze data because then the user would be able to focus on say how a team is doing instead of having to read and analyze the data themselves.**

- Read the Django HYPERLINK "https://docs.djangoproject.com/en/3.2/ref/models/querysets/"official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.
- **It can be used to evaluate data and display it in different ways to get the project to display exactly which type of data you want and exactly how you want it displayed.**

- 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.
- **The disadvantages to Queryset lies in that the queries are not executed until they are needed, meaning this can make it slower when dealing with large amounts of data. One of its advantages is that it is very easy to integrate with django models.**

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

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

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

- (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.

- You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:
    - What went well during this Achievement?
    - What's something you're proud of?
    - What was the most challenging aspect of this Achievement?
    - 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.