# 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? So far, I have gained experience in developing web and mobile applications through the MEAN and MERN stacks during the CareerFoundry full-stack web development course. Currently,

I am working on a personal project where I am developing a 2D video game with a small group of friends. In this project, I am taking on the role of project manager, which has significantly enhanced my skills in project coordination and team collaboration. This experience has not only improved my technical abilities but also provided me with valuable insights into managing projects and leading a team.

2. What do you know about Python already? What do you want to know? I know that the indentation syntax in Python works differently than JavaScript, the indentation in Python defines the structure of the code. Python makes for a versatile programming language like JavaScript and can be used for scripting. I am looking forward to expanding my knowledge around Python and using it for projects like machine learning, web, software, and game development.

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 think the biggest challenge in this course is the extra work that I will have to do outside of the course to have a better understanding of Python. Although the challenge may be time consuming, it will teach me how to learn. Which is a simple concept but hard to master.

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? Front end web development involves building the user interface and visual structure of the website, this includes designing the buttons and other interactive elements. The backend holds the backbone of the website, it involves database management, authentication, API development, integration and performance optimization.

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")*

   I would recommend Python primarily for its readability and simplicity. Its clean and straightforward syntax makes the code easier to write and understand, which is beneficial for both individual developers and teams. Python's dynamic typing allows variables to hold values of different types without causing errors, which adds flexibility to the development process. Additionally, Python's package manager, pip, simplifies the integration of external libraries and resources, streamlining the development workflow. Python is also highly regarded within the developer community, which enhances its reliability and makes it easier to collaborate with others in a team setting.

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 am hoping to learn the nuances of Python and other topics outside of web development in order to help me grow as an individual programmer. I believe this would highly benefit me in the job market, help me become a flexible programmer, and allow me a foundation to pursue my career in machine learning, game development, web development, and/or automation/scripting.

# 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? iPython provides a handful of features when being compared to the default Python shell. Its readability is enhanced through its color coordination and indentation. The default Python shell is displayed in one color and indentation can only occur when it is manually added by the user.

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? |
|---|---|---|
| bool (boolean) | Represents logical values, true or false. | Scalar |
| int (integer) | Represents whole numbers | Scalar |
| dictionaries | Collections of value-key pairs | Non-Scalar |
| tuples | immutable , ordered collections of elements | Non-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. Lists are mutable, you are able to modify the list at any time. They're typically used when you need a collection of items that can change. Tuples are immutable so once they are initialized, they cannot change, an attempt to modify a tuple will result in an error. A list uses brackets [ ] while a tuple uses parenthesis ( ).

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. I would use dictionaries as the data type structure for this flashcard system. The structure provides clarity, each string would be linked to its corresponding information, making the data easy to access and update. Unlike a tuple, there would be flexibility in the data structure: more information/flashcards can be added to a deck if needed. Although a list could be a viable option, it is harder to access information in the data structure because you would have to remember the positional index in order to modify the data.

# 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!)*

```python
destinations = ["Japan", "United States", "Germany"]
user_destination = input("Type your destination: ")
    if user_destination in destinations:
        print("Enjoy your stay in", user_destination, "!")
    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. Python has three main logical operators, **and**, **or** and **not**. The and operator returns **True** if both operands are true, the or operator returns **True** if at least one of the operands are true, the not operator returns **True** if the operand is false and **False** if the operand is true

3. What are functions in Python? When and why are they useful? Functions act as reusable blocks of code in Python. Each function is designed to perform a specific task. They're sets of instructions that process or manipulate your code in order to achieve tasks. They are great with breaking down programs into smaller and manageable chunks. The indexing in Python improves code readability which adds onto its usefulness

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.
   I have learned some fundamentals about Python which will go a long way for me. I am understanding the benefits of working with Python and I am excited to build an app with Python to further understand the nuances of the language. I would like to see if there are other technologies that go hand in hand with Python in order to build a fully functioning application.

# 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  If you don't store the files when you're using Python? What would happen if you didn't store local files? When you don't store files while using Python, you will lose data from the user's input. You can store data permanently rather than losing the data after the script runs. When it comes to handling larger projects data, it is useful to be able to transfer information in order to utilize it for other use cases. Unlocking the ability to reuse data means you won't have to restart the building process and it will save you time in the long run.

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?  Like in the previous question of storing local files, pickles are useful in scenarios where you'd want to save the state of Python objects in order to use them later. They help store data no matter the complexity and allow for  files to communicate with other files seamlessly. The only time I would not recommend using pickles is when you are handling sensitive data because of how easily accessible it can be.

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? The os module is used for navigating and indicating files that are active. To use the os module, you would first need to import it in your terminal using **import os**. To find out what directory you are currently working in you would use the **os.getcwd()** command which will pull up the directory that is current (EX: **>>> print(os.getcwd())** \n **/home/careerfoundry/Documents/example_folder/scripts** ). Another addition to the module that comes is using the tree command which displays the rest of what is in your file to the terminal. To change your current working directory, you would need to use the **or.chdir()** command (EX: **os.chdir(<path to desired folder>)**).

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? If you are expecting an error, there are a variety of except blocks to try (ZeroDivisionError, FileNotFoundError, ValueError, IndexError, etc.). In a scenario of uncertainty a good way of preventing the entire script from terminating is to use except by itself in a try-except block. To run a block regardless of the script passing successfully or not, use the finally block to finalize the block.

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.  Sometimes I do find that solidifying some of the information based

# 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?
   OOP in Python promotes modular coding that is reusable. You are able to reuse data attributes and methods to create specific functionalities in different parts of your code. The classes prevent coding repetitiveness since you can create them to inherit other attributes from a parent class. The debugging process feels a lot more optimized since the classes provide structure and clarity.

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work. Objects are like the characteristics of a class in Python. Objects can be a name, model, part, etc. and the class can be the entirety of all those characteristics (animal, car, house, etc.). The objects are what makes the classes have value, without the objects, the class would be like a silhouette but without substance. In a real-world situation a cat has stripes, colors, and a name if they have an owner. To write this you would make the Cat a class, and stripes, colors, and name as the object. With this structure, you could use it to compare the class with other subclasses(types of cats).

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

| Method | Description |
| --- | --- |
| Inheritance | Reusing the attributes in a parent class into a subclass that has other attributes |
| Polymorphism | Creating a method so that it can be reused in other parts of your code and provide the same functionalities regardless of the values of the objects being different. |
| Operator Overloading | Defines the behavior of operators so that you would be able to use them with the different values but same attributes of a class. It's like making an equation that is waiting for input data to make an output. |

# 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?
   Databases are organized in a collection of data that can be accessed and managed. One of the advantages to using databases are the ability to query in complex ways. You are able to CREATE, READ, UPDATE and DELETE objects depending on your objective. A common and effective way to organize your data is through incrementing each row with an id. Accessing becomes a lot easier once you are able to organize your objects through incrementation. Columns are treated similar to attributes of an object, so when you are querying a database, the outcome is uniform and predictable.
2. List 3 data types that can be used in MySQL and describe them briefly:

| Data type | Definition |
|-----------|------------|
| int | Standard integers |
| varchar() | Variable length within (n) presenting the maximum number of characters |
| float | Floating-point decimal numbers |

3. In what situations would SQLite be a better choice than MySQL?
   SQLite would be most optimal in small-medium projects, where the development process is done within your local system like a personal project. Using it becomes quite efficient since you wouldn't need servers to run your database with your code.
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?
   I think that Python feels extremely flexible in comparison to JavaScript. There are minimal package downloads and you can get straight to coding right out of the box when it comes down to scripting. JavaScript is better out of the box for web development. But I can see Python being more efficient in other sorts of complex development because of its more forgiving syntax and debugging system.
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?  Based off of what

I learned so far, I can see the limitations residing in visual representation, in other words, making things come to life like you would with using React. Having the physical representation of what you are working on can be challenging so I would like to learn how to make that possible using Python.

# 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?
   An ORM is a tool used to work with databases. It is used to make conversion easier for the developer. Because of this tool, you are able to use OOP and organize your code a lot better than if you were to use the MySQL connector alone. The ORM provides a bridge which is your engine and the vehicle that uses that bridge is called the session. The session brings your python model class over to the SQL using the engine. Then, the SQL Query is able to take in that model class as if it were a table.

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? I think the thing that I did well the most in this Recipe App was the formatting overall, prompting the user and being able to display objects the way I intended to. If there was one thing I would have to change about the app in order to improve user experience, I would change the prompts slightly to make it user friendly. If I could learn to  program a way to use arrow keys to navigate and hit enter to return results, I would do so instead of making the user type to select specific indexes (for example: main menu, edit a recipe and delete a recipe).

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. I've experienced object oriented programming utilizing data structures using MySQL with and without ORM. In order to get a deeper understanding of how relational databases are supposed to work with Python.  I've made the app with the ability to prompt users to perform CRUD operations on their recipes and provided error catching inside my methods to make sure there was a scenario for anytime an input was not valid. Additionally, the user is informed when there was an error and prompt in order to continue or exit within the application.

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

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? Using vanilla Python would take lots of time but I would have a lot of control over the entire project. In a large scale project, this would not be the most optimal. If I used Django in a big project, I would be able to utilize the entirety of the framework and the flow of the development process would be a lot more efficient

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture? With MVT the visuals reside within the template of the project. You wouldn't have to worry much about the output document since the structure is already specified when Django takes the URL from the user who initiates it. The URL maps the corresponding View by interacting with the model and templates of the project.

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.*) Since TikTok has a huge audience, I would use Django to handle a massive amount of

user-generated content and interactions. The Models would hold data of the user (username, email, password, bio, profile_picture, etc.) and data of each video (video_file, caption, date_posted, likes, comments, shares, etc). The Views would fetch something like a list of videos to display to the users feed. There would be the main display/UI for the user (user profile, homepage, video player, etc.). The URLs will point to those existing main displays.

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system. To deploy a basic application locally on my system, I would start off with a folder as a placeholder and create a virtual environment dedicated to the application. Make sure that I have django-admin installed inside of my virtual environment. Using the django admin, I will then create a project under the same folder I made earlier. To avoid confusion, rename the folder from its sub folders. The next step would be to run migrations in order to run the server. Once the server is running I'd create a superuser in order to get access to the admin dashboard of my project.

3. Do some research about the Django admin site and write down how you'd use it during your web application development. The first thing that I've noticed off that bat about Django's admin dashboard is that it makes seeing data relationships a lot easier. Research tells me that the admin site also assists with debugging during development. If something is wrong with the model data or validation, the admin site can help you inspect what is happening.

# 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. Django models are used to define your 'table' and the objects that reside within that table. It helps keep your objects organized and set expectations for how you'd want for them to work in your application. Models are great for testing for an expected outcome. A schema is used to make your models and is a great point of reference to building your data structures in Django. That way, you have multiple sources to build off of.

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. Test cases are crucial in the beginning of a project because it handles unexpected and expected outcomes. They define the course of

# 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

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.