

# **Python for Web Developers**

## **Learning Journal**

## Pre-Work: Before You Start the Course

### Reflection questions (to complete before your first mentor call)

1. What experiences have you had with coding and/or programming so far? What other experiences (programming-related or not) have you had that may help you as you progress through this course?

I have completed the Full-Stack Web Development Immersion course at CareerFoundry. I also have a minor in Computer Science from the University of Nevada, Reno. Finally, I have a summer internship as a software engineer at a startup company.

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

I know that python is a very high-level language and that it is relatively easy to learn because the syntax is so simple. I want to know everything about Python.

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 managing this course while working, applying for jobs, and improving my resume, portfolio, etc. will be the biggest challenge of this course. Having encouragement and motivation from my mentor will be a big help. Also, scheduling a reasonable amount of time dedicated to working on this course every week will help me stay focused.

Remember, you can always refer to Exercise 1.4 of the Orientation course if you're not sure whom to reach out to for help and support.

## Exercise 1.1: Getting Started with Python

### Learning Goals

- Summarize the uses and benefits of Python for web development
- Prepare your developer environment for programming with Python

### Reflection Questions

1. In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working?

Frontend web development is building the software that the user actually sees and interacts with. This would be the website or user interface of a program. On the other hand, backend development is building the software that controls the data flow and business logic behind the scene. This would be pulling data from the database, doing whatever calculations that need to be done and then serving it to the frontend.

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?

Python and JavaScript are both high-leveling scripting languages, meaning they are read line by line as opposed to needing to be compiled before executed. Also, they both have dynamic typing, meaning the language infers the data type of variables by what you assigned to them, as opposed to static typing where you have explicitly state the data type of each variable. I would recommend using Python for this particular project because Python is very readable which makes it easy to get a project up and running. Also, Python has a great built-in package management system and many open-source and proprietary packages for the most common tasks.

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 want to learn how to build a full backend with Python and Django for a project. I want to learn this skill in such a way as to be able to apply it to real-world projects. I hope that putting this project in my portfolio will help me get a job in software engineering.

## Exercise 1.2: Data Types in Python

### Learning Goals

- Explain variables and data types in Python
- Summarize the use of objects in Python

### 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 uses syntax highlighting which makes the scripts easier to read, it has automatic indentation for nested statements, as well as autocomplete commands. These features all make writing Python scripts much easier.

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?
int	Used to store integers, positive and negative numbers	scalar
float	Used to store decimal numbers, positive and negative	scalar
bool	Used to store true or false values	scalar
string	Used to store an array of alphanumeric characters as well as symbols and are surrounded by either single or double quotes	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.

Tuples are immutable whereas lists are mutable meaning you can easily add, remove and modify items in lists but not with tuples. With tuples you have to remake your structure with any updates as opposed to modifying it directly.

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 also use a dictionary to represent the flashcard because you can name the keys to represent the data being stored. This makes your code much more readable and easier to work with than just using the indices of the list for example.

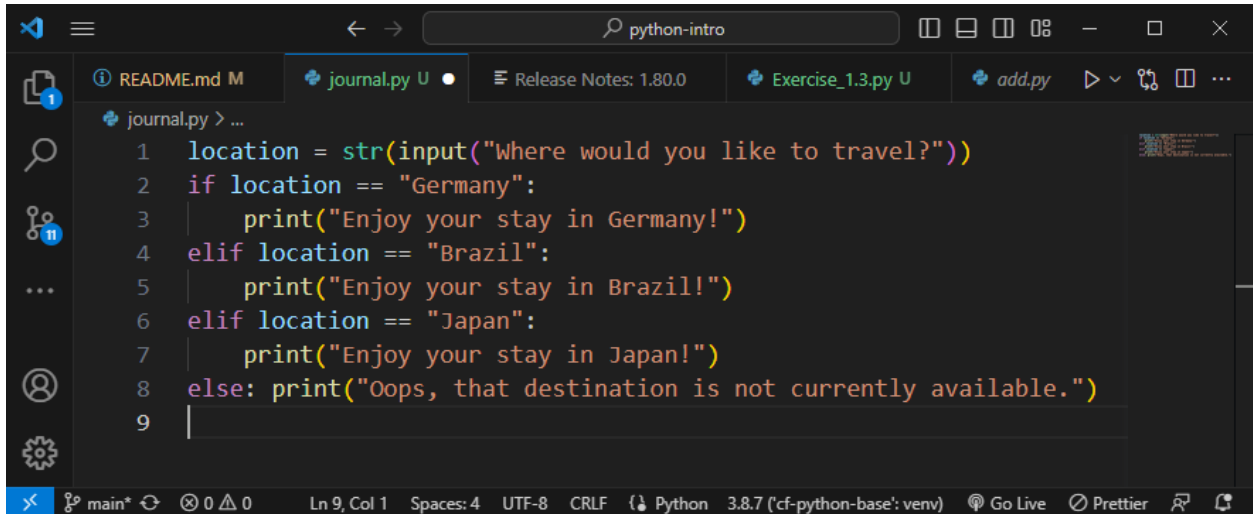
## Exercise 1.3: Functions and Other Operations in Python

### 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 statements should be printed: "Oops, that destination is not currently available."



```
1 location = str(input("Where would you like to travel?"))
2 if location == "Germany":
3     print("Enjoy your stay in Germany!")
4 elif location == "Brazil":
5     print("Enjoy your stay in Brazil!")
6 elif location == "Japan":
7     print("Enjoy your stay in Japan!")
8 else: print("Oops, that destination is not currently available.")
9
```

2. Imagine you're at a job interview for a Python developer role. The interviewer says "Explain logical operators in Python".

Logical operators are what connect logical statements to form more complex logical statements, such as 'and', 'or', and 'not'. The syntax in Python is very straightforward; you simply use the keywords like plain English, except the 'not' operator you put in front of the statement with parentheses.

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

There are many built-in functions in Python that accomplish the most common tasks for you but sometimes you need to do something that is specific to your own code. That is when you have to define a custom function. You should use a custom function when you want to perform the same task multiple times. So instead of writing the same code multiple times you modularize it by writing a function and then can just call that function multiple times instead. This makes your code easier to read, understand, and debug.

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 feel like I haven't learned much in this course so far. Just the basics which I have already learned in Javascript, just with slightly different syntax. I don't think the code I've worked on so far is realistic to the problems I'll see in industry, and I don't think I have anything to put on a portfolio. Feels like most of this course so far is busy work, not actually coding. I have spent way more time formatting the readme file or writing in the journal than actually coding. I feel like the best way to learn is to solve problems with code, not just talk about it—but I'll trust the process.

## 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 in Python because when you run a script any data that you have created and stored in variables will not be saved the next time you run that script. The memory is cleared when the script closes. So if you want to use the data you create the next time you run the script you must save the data to a local file and load that data in the script for the next time you want to run it.

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 a way to store Python objects in local storage in a binary file. You would use them when you want to read or write data in local storage as objects instead of just plain text. It is much easier to just load the data as objects that are ready to be used than to parse text and build your data again by scratch.

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?

In Python, you use the `os.getcwd()` function to find out which directory you're currently in. If you want to change your current working directory you use the `os.chdir()` function. To use either of these functions you must first import the `os` library with `import os`.

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 use a try-except block of code. Place the error prone code in the try block and the error statement in the exempt block. This will prevent the entire script from terminating due to an error. It will simply jump to the exempt block, run that code, then continue with the program.

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.

It is going well. This course is straightforward after just learning Javascript. I'm glad we did file handling because that was one thing not covered in that course. I'm proud that I've been completing the exercise at a good pace. Not struggling with anything but I could always use more practice in everything. I want to practice with a bigger project. These little scripts don't seem very realistic. I'm excited to begin the Django achievement.

## 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?

Object-Oriented Programming is the use of custom data types to structure your code, as opposed to functional programming which is structured around the use of custom functions. The benefits of OOP are a highly structured and organized code base, with little to no repeat code. It is also easier to read and understand because it is often structured around real-world objects and is highly modularized.

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

Everything in Python is an object. Classes are just custom objects. You can buy a pre-built bookcase from the store—that would be a built-in object in Python. But perhaps it doesn't fit the exact dimensions of the room where you want to place the bookshelf. To fix this you build your own bookshelf instead that fits the room perfectly. That would be a custom object or class in Python. You can use classes in Python to create custom objects that fit your exact needs to solve a problem.

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



Method	Description
Inheritance	You can create new classes from previously defined objects. The classes automatically 'inherit' all of the attributes and methods from the parent class. You can add more attributes and methods to the new class to make a more specialized version of the parent class.
Polymorphism	Polymorphism is when a method is defined for two different objects and thus must perform two different but analogous tasks. len() is defined for strings, dictionaries, lists but is defined differently for each.
Operator Overloading	This is just another form of polymorphism that is 'syntactic sugar', which means it is an easier way of writing the code. It is most often used to make operations look similar to mathematical operations. For example, if you want to add two objects you could write the method add(a, b) but to make it look like mathematical addition you could overload the + operator to be defined for this object so you can write a + b.

## 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 an efficient and structured way of storing data. The advantages of using a database are they keep data in a standardized format, they can be made secure through password access, and you can access them using a variety of applications.

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

Data type	Definition
VARCHAR(n)	String of variable length, with n representing the maximum number of characters
INT	Standard integers
FLOAT	Floating-point decimal numbers

3. In what situations would SQLite be a better choice than MySQL?

If you just need a small amount of data for a website or you just want to do some testing without having to set up an entire database engine.

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?

Python is definitely easier to write and remember the syntax because it is closer to plain English. Besides that, they both seem to have all the same features.

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?

I do not know the limitations. I feel like I have just learned the basics. I know nothing about its efficiency compared to other languages or what it is best used for. I'm guessing you can do just about anything in Python because it's so popular—there has to be a library out there for whatever you want to do.

## 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 Object Relational Mapper is a library that relates tables in an SQL database to classes. This allows you to interact with a database using OOP and without generating SQL query strings. Instead you just use functions to query the database.

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?

Building the Recipe app went well. I like how I formatted the print recipes and I like how instead of exiting the program when receiving invalid input I tell the user to try again. If I were to start over I would add more modularization. I feel like there is a lot of repeat code that I could put into a reusable function. I would also try to separate a lot of the print statements and business logic so the code is easier to understand and maintain.

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 have designed and implemented a Python command line application that connects to a MySQL database. It is a food recipe program with the basic CRUD features. It uses SQLAlchemy, an Object Relational Mapper, to interact with the database which makes it easy to query the database and allows for an object-oriented software design. I have implemented data sanitation when receiving input from the user. There is definitely more I could add to this program to improve functionality and usability such as a better user interface.

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?

- a. I stayed on schedule and completed the Achievement at a good pace.
- b. I am proud that I finished it in about a week
- c. The most challenging aspect was connecting the Python to the SQL database and setting up the database correctly.
- d. Yes and Yes
- e. Just keep working at a steady pace.

## Pre-Work: Before You Start Achievement 2

- Was your study routine effective during Achievement 1? If not, what will you do differently during Achievement 2?

My study routine was very effective during Achievement 1 and because of this I don't plan on doing anything different 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?

I was most proud of how I stuck to a schedule for working on the Achievement and how I got it done in about a week. For Achievement 2 I will stick to the same schedule and maybe even take an extra day off from working if I feel like it. I don't want to get burned out.

- 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?

The only difficulty I encountered was setting up the SQL database because the instructions were unclear or not complete. I tried reinstalling the database and following a different approach than what was given in the course in order to solve the problem. This prepared me for future difficulties by giving me the confidence to know that I can figure them out by myself with enough perseverance.

## 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 Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?

The advantages of using Django are that the MVT architecture ensures development is fast and easy. Django has fast processing and can handle network transmission and content delivery at a high speed. Django follows DRY principles and both in how it's implemented and how its users are encouraged to create web applications. Django provides support for Content Delivery Networks and Content Management. The MVT architecture used in Django makes it very simple to add new hardware and components, so scaling a project is no problem. Django is built with security in mind, allowing secure-by-design implementation with built-in automated encryption. Django is open source and has a huge community of contributors so it is usually very easy to get support.

The disadvantages for using Django are that it is highly structured and you have to do things in a certain way, often called 'The Django Way'. So if you want direct control over all the facets of a project then Django may not be suitable. Also, Django offers a lot of out-of-the-box solutions which means it comes with a lot of computing overhead. If you're creating a small, simple project then you may not need all of the features that Django offers meaning you could be making your project unnecessarily large by building it with Django.

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

To me the biggest advantage is not having to write code to fetch data from the database and map it to the URL. You simply need to specify which items to present to the user, and the template will prepare and send it. Sounds great!

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?

I want to learn the most relevant and useful features of Django. I want to learn the best practices of these features and how to implement them in a large-scale, real-world project. I want to be able to get a job working with the Django framework as a Web Developer. After this Achievement I see myself getting paid to build web applications with Django at a tech company.

## 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.

The whole website will be the project while the different features of the site will be apps. Things like login, users, blog, calendar, search would all be different apps within the project.

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

I would create a folder where I want to store the project. Then I would create and run a virtual environment for the project. Then I would install django in the environment and create the project. Then I would run migrations and finally run the server.

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

I would use the Django admin site to create users and manage each user's privileges within the project. I would also use the admin page to create, read, update and delete any records being stored in the database while the site is in production.

## 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.

Models define the structure of stored data, including the field types and possibly also their maximum size, default values, selection list options, help text for documentation, label text for forms, etc. The benefits of using models is that once you’ve chosen what database you want to use, you don’t need to talk to it directly—you just write your model structure and other code, and Django handles all the dirty work of communication with the database for you.

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.

It’s crucial to write test cases from the beginning of a project because you want to catch bugs as they occur instead of having to dig through your code later to find them and debug. It is an awful waste of time to go search for your bugs way down the line as opposed to just addressing them immediately. Trying to debug code you worked on days, weeks, or even months prior is a nightmare, so fixing bugs while the code is still fresh in your mind makes this process much easier. Finally, fixing bugs may reveal structural problems in your code meaning you would have to rethink how your project is structured. Doing this as soon as possible is best so you don’t have to rewrite a bunch of code.

## 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.

A view is the logic that Django runs when a user accesses a URL. It is a Python function or class method that accepts a request, runs Python code, and returns a response. The function or method can be quite basic, simply accepting a request and returning static information—such as an HTML web page, or JSON data—or the function could be as complex as a class that accepts a request, interacts with the database, accepts user input (sent with the request), and determines what will be returned to the user as a response.

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?

If I anticipate that I'll have to reuse lots of code in various parts of the project I will definitely use class-based views, because these are more modular and can be reused more effectively, whereas functions are more specific with more custom code.

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

Templates are a convenient way to generate HTML dynamically. A template contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. Django ships built-in backends for its own template system, creatively called the Django template language (DTL). Django defines a standard API for loading and rendering templates regardless of the backend.