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?
2. What do you know about Python already? What do you want to know?
3. What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.

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

### Exercise 1.1: 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?
   1. Front end development deals with the user interface, in other words, what the users see. Whereas backend web development interacts with the infrastructure of the website such as services, API, and/or data. If I were hired as a backend development, I would expected to interact with building and maintaining API, data validation, web protocols, etc.
2. Imagine you’re working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?

*(Hint: refer to the Exercise section “The Benefits of Developing with Python”)*

1. Now that you’ve had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?
   1. Learn to use Python to analyze data
   2. Learn basic of Python
   3. Conduct data analytics or statistical models in Python

### 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?
   1. It provides color coding, auto completion, and auto indentation
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 | Whole number, integer | scalar |
| Str | alphanumeric | scalar |
| float | Decimal numberes | scalar |
| bool | True or false | scalar |

1. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.
   1. Tuple store linear arrays and all values within a Tuples must be the same type. List on the other can store multiple data type into an array.
2. In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you’re creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language-learning app beyond vocabulary memorization.
   1. I would use a combination of list and dictionary. Dictionary would be used to store individual flashcard, while the list would hold the entirety of all flash cards. This allows looping through each items while providing the ability to display the vocab and definition separately.

### Exercise 1.3: Functions and Other Operations in Python

#### Learning Goals

* Implement conditional statements in Python to determine program flow
* Use loops to reduce time and effort in Python programming
* Write functions to organize Python code

#### Reflection Questions

1. In this Exercise, you learned how to use **if-elif-else** statements to run different tasks based on conditions that you define. Now practice that skill by writing a script for a simple travel app using an **if-elif-else** statement for the following situation:

* The script should ask the user where they want to travel.
* The user’s input should be checked for 3 different travel destinations that you define.
* If the user’s input is one of those 3 destinations, the following statement should be printed: “Enjoy your stay in \_\_\_\_\_\_!”
* If the user’s input is something other than the defined destinations, the following statement should be printed: “Oops, that destination is not currently available.”

Write your script here. *(Hint: remember what you learned about indents!)*

|  |
| --- |
| destination = input(“Where do you want to travel to?”)  if destination == “Hawaii”:  print(“Enjoy your stay in Hawaii!”)  elif destination == “Japan”:  print(“Enjoy your stay in Japan!”)  elif destination == “China”:  print(“Enjoy your stay in China!”)  else:  print(“Oops, that destination is not currently available.”) |

1. Imagine you’re at a job interview for a Python developer role. The interviewer says “Explain logical operators in Python”. Draft how you would respond.
   1. Logical operators evaluate condition and determine if it is true or false. AND operator returns TRUE only if the arguments provided are both TRUE. OR operator returns TRUE as long as one of the provided arguments are TRUE. Finally, NOT operator returns the opposite of the provided arguments
2. What are functions in Python? When and why are they useful?
   1. Functions are define set of instructions that can be called upon to perform a task. They can reduce repetitive tasks within your code and makes it cleaner.
3. In the section for Exercise 1 in this Learning Journal, you were asked in question 3 to set some goals for yourself while you complete this course. In preparation for your next mentor call, make some notes on how you’ve progressed towards your goals so far.
   1. Feeling pretty good as I am learning the basics of Python. Looking to explore further after the course.

### 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?
   1. File storage allows data to be manipulated and maintain, ensuring no lost in data. Without it, data would be lost and user would have to re-enter the data everytime
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?
   1. Pickles is the process of converting complex objects into byte steam. Pickle.dump() serializes an the object and converts it in to simple text. In situation when user needs to store complex objects such as list and dictionaries with the goal of reading it later.
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?
   1. It would be os.getcwd(). To change the current working director, os.chdir() function would be used.
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?
   1. Try-except can be used for the block of codes.
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.
   1. NA

### 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?
   1. Object oriented programming allows users to create custom blueprint with properties and methods associated for it.
2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
   1. Example would be the blueprint of a car, it has the attribute of wheels, color, and vehicle type. It has the method of drive and reverse
3. In your own words, write brief explanations of the following OOP concepts; 100 to 200 words per method is fine.
   1. Use the car example

|  |  |
| --- | --- |
| **Method** | **Description** |
| Inheritance | And SUV is a car, and it inherit the method from the car blueprint including drive and reverse |
| Polymorphism | A truck is a car. When backing up, it back up just like a regular car with the addition of producing a beeping sound for alert |
| Operator Overloading | Operator overloading provides the ability to customize behavior for standard operators. |

### 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?
   1. Database is a organized collections of data
2. List 3 data types that can be used in MySQL and describe them briefly:

|  |  |
| --- | --- |
| **Data type** | **Definition** |
| varchar | Alphanumeric |
| int | Integers |
| float | Decimal numbers |

1. In what situations would SQLite be a better choice than MySQL?
   1. When your work only requires you to use a very simple database storing simple data, or want to test a small set of data.
2. Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?
   1. JavaScript has a wide usages including web development, API, data management. It can be used for both front end and back end. Whereas Python focus more on data management and primarily used for back end development and data analytics.
3. Now that you’re nearly at the end of Achievement 1, consider what you know about Python so far. What would you say are the limitations of Python as a programming language?
   1. Python seems to be more data focus with superb performance, but it seems to lack usage in other area such as front end development.

### 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?
   1. It provides an object-oriented layer between relational database and OOP language without having to write SQL queries. Advantage would be you don’t need SQL knowledge to use ORM.
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?
   1. It was fun to link python to a relational database, providing the ability to user to updating a database without SQL queries for the front end user.
3. Imagine you’re at a job interview. You’re asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.
   1. I created a recipe app for the python shell. It gives user the ability to add, view, update, delete, and search recipes that they’ve added. Technology used include Python and MySQL. Specifically, OOP and ORM.

1. You’ve finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:
   1. What went well during this Achievement?
      1. Learning the basics of Python
   2. What’s something you’re proud of?
      1. Learning how to manipulate data and files using python
   3. What was the most challenging aspect of this Achievement?
      1. ORM is a approach I’ve learned as part of this class. As a SQL person, I feel that it is more efficient to write the SQL statement. But for a front end user, it is more user friendly.
   4. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
      1. Yes, I’ve learned how to manipulate data with Python.
   5. What’s something you want to keep in mind to help you do your best in Achievement 2?
      1. NA

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

### Exercise 2.1: Getting Started with Django

Learning Goals

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

#### Reflection Questions

1. Suppose you’re a web developer in a company and need to decide if you’ll use vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?
   1. Vanilla Python would provide more control, but developer would have to create a lot of custom codes to make it work. Django provides a framework to support quit developments, security, and data support, downside would be losing control.
2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?
   1. With MVT, you don’t have to figure out how to present the data as the framework will prepare it and present it on the template.
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:

* Front end development using Django
* Create beautiful front end with Django, Python, and Data
* Running data analytics and create dashboard

### 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.*)

Facebook for example, have multiple part within the website with each part categorize as project. Upon arrival, the login project will be display to prompt user to login.

1. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.
   1. Create virtual environment -> activate -> install Django -> start new project -> run migration -> create superuser -> startapp -> runserver
2. Do some research about the Django admin site and write down how you’d use it during your web application development.
   1. Django admin panel allow user to create, read, update, and delete models. Allowing users to manage contents directly via the portal with no command lines.

### 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.
   1. Similar to SQL, Django models allow for CRUD operations to database via the admin panel, but without knowledge of SQL
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.
   1. It enhances best practice and avoid any bugs and errors in the early stage.. In the same time, it ensure app is robust and fully functional.

### 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.
   1. Django views are python functions/classes. It receive requests, perform business logic, and send responses of rendered URLs. Overall, process goes user visit app -> app sends requested URL -> Djan fetch URL based on logic and sends corresponding page back to user.
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?
   1. I would use class based views. It allows me to re-use a class multiple times through out the project. This reduces duplicated codes and increase performance.
3. Read Django’s documentation on the [Django template language](https://docs.djangoproject.com/en/3.2/ref/templates/language/#templates) and make some notes on its basics.

### Exercise 2.5: Django MVT Revisited

#### Learning Goals

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

#### Reflection Questions

1. In your own words, explain Django static files and how Django handles them.
   1. Django static files refer to the files that don't change or require processing by Django during runtime, such as CSS, JavaScript, and images. Django handles these files by allowing developers to store them in one or more directories that Django can access. When a Django project is deployed, these static files are collected from their various locations within the project's apps and placed into a single directory for efficient serving to the client. This process is managed by Django's staticfiles app, making it easier for developers to manage and serve these files in a production environment.
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 | The ListView in Django is a generic view that abstracts the pattern of displaying a list of objects. It simplifies the process of writing views that display a collection of objects from the database by handling the fetching of objects and rendering of the template. With ListView, you can easily paginate the list of objects and customize the query to display objects based on certain criteria. |
| DetailView | The DetailView is another generic view provided by Django that is used for displaying the details of a specific object. When you want to show more detailed information about a single item from the database, such as a single blog post or user profile, DetailView comes in handy. It automatically fetches the object based on the provided primary key or slug in the URL and passes it to the template. This simplifies the process of creating detail views for individual items. |

1. 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.
   1. Reflecting on my journey halfway through Achievement 2, I'm feeling positive about the progress I've made. I'm particularly proud of mastering a new technology stack that includes Python and Django. The experience of learning how to integrate Django static files and effectively utilize Django's ListView and DetailView to bring a web application from concept to reality has been incredibly rewarding. While I've enjoyed the learning process, I acknowledge that I need more practice with certain aspects of Django, especially in refining my understanding of its ORM and database models to create more efficient and effective data queries. Looking ahead, I aim to dive deeper into Django's advanced features and continue building my skills to develop more complex and dynamic web applications.

### 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.
   1. Authentication keeps your data secure by verifying user identity. It also protects privacy. For example, the Uber app authenticates the user to keep their credit card info and account info protected.
2. In your own words, explain the steps you should take to create a login for your Django web application.
   1. Create login view -> Create FBV to show login form using Django auth form -> Create HTML template for login in page -> Configure setting file on for auth template location -> register URL for login view to urls.py
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() | This function is a part of Django’s authentication framework, specifically from django.contrib.auth. It takes credentials as parameters (often a username and password) and returns a user object if the credentials are valid for a backend. If the credentials are invalid, it returns None. It's a core part of the login process in Django-based application. |
| redirect() | The redirect() function, part of Django’s shortcut functions, is used to return an HttpResponseRedirect to a particular URL. It can take a model, a view name, or a URL to redirect to. It's commonly used after form submissions or actions to send the user to a new page |
| include() | include() is a function used in Django’s URL handling. When defining URL patterns, include() allows you to reference other URLconfs, essentially allowing you to "include" other URL configurations. This modularizes the URL structure, making it cleaner and more maintainable, especially in larger projects where you might want to split |

### 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](https://docs.djangoproject.com/en/3.2/ref/models/querysets/). 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:
   1. What went well during this Achievement?
   2. What’s something you’re proud of?
   3. What was the most challenging aspect of this Achievement?
   4. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

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