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?

Frontend development deals with all that a user/consumer/client sees, touches and interacts with. Backend development is the server and business logic that enables the front end to return the results it is designed to achieve. If I was hired to do backend development, I would likely be working with databases, authentication and authorization, security, APIs and the like.

1. Imagine you’re working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?

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

The benefits of using Python are in its readability and its ease of maintenance. Its dynamic typing allows for quicker debugging, and there is a massive amount of community support that helps to ensure a continued high level product.

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?

I’d like to become more fluent in the syntax and language, to get a solid feel for how the structure functions. I want to be comfortable indicating why I do or do not prefer this language for a particular project/prospect. Lastly, I would like to be able to learn how to integrate Python with other languages and frameworks to continually develop the best full-stack programming background I can.

### Exercise 1.2: Data Types in Python

#### Learning Goals

* Explain variables and data types in Python
* Summarize the use of objects in Python
* Create a data structure for your Recipe app

#### Reflection Questions

1. Imagine you’re having a conversation with a future colleague about whether to use the iPython Shell instead of Python’s default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

The iPython shell has really beneficial features including automatic indentation, syntax colorization, and code completion which increases readability and speed of use.

1. 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?** |
| Tuple | Linear arrays storing multiple values of any type | Non-scalar |
| List | Mutable sequence of characters that must be wrapped in [ ] | Non-scalar |
| String | Immutable sequence of characters wrapped in “ ” | Non-scalar |
| Dictionary | Set of items, unordered, created and stored in a key-value pair, each key being unique. | Non-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.

The primary difference between tuples and lists is that lists are mutable – can be changed - and tuples are immutable – remain the same.

1. 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 in this case. The key-value pairing lends itself well to the type of data that needs storing, the ability to modify the data in the future is advantageous and the fact that varying types of data can be stored opens up the potential for broader language learning as the users’ skills develop.

### 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!)*

|  |
| --- |
| location = input(‘Choose to go to Alaska, Turks and Caicos, or Joshua Tree: ’)  if location == ‘Alaska’ or ‘Turks and Caicos’ or ‘Joshua Tree’:  print(‘Have a fantastic trip in’ + location)  else:  print(‘Sorry, that’s not a travel option’) |

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.

Logical operators in Python operate fairly simply. “Or”, “Not” and “And” all function as booleans, returning either true or false responses. “And” returns true if all conditions are met, “Or” returns true if even one condition is met, and “Not” returns the opposite of what the output would be if the “Not” operator were not in place.

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

Functions in Python are similar to functions in many other languages. They are defined steps in a piece of code. They are useful particularly in their ability to be called over again without having to rewrite the original block of code over and over. They can serve to keep your code condensed and tidy.

1. 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 I’m continuing to make progress on these goals, particularly in my comfort with using Python and even being able to explain why I may or may not prefer using it for any given project.

### Exercise 1.4: File Handling in Python

#### Learning Goals

* Use files to store and retrieve data in Python

#### Reflection Questions

1. Why is file storage important when you’re using Python? What would happen if you didn’t store local files?

Data has to be stored locally if you’re using Python otherwise, as soon as your shell is closed, data is lost. By keeping it stored locally, work can be continued and recalled as often as needed.

1. 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?

The pickling process allows the usage and storage of pickles –bytes stored as binary data. This allows large and complex information to be stored cleanly and securely while still being able to be read and accessed.

1. 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?

To learn your current directory, you would use an os.getcwd() function. In order to change your working directory, you’d use an os.chdir() function.

1. 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?

To avoid the entire script terminating, I would use try-except blocks to weed out the potentially problematic paths.

1. 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.

So far it’s going well, I think. I’m finding Python fairly easy to use and it seems to be a bit easier for me to understand exactly how it’s working. As with anything, I think repetition and really working with it all helps with the overall comprehension. I also have found, during my time with CareerFoundry, that the more roadblocks I hit and have to overcome, the more I end up understanding in the end. The bug-hunting with Python is pretty straightforward which is nice.

### 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 allows a developer to create objects within classes. This tends to lend itself to clear, concise and organized code that has a high readability. Another significant benefit is in its reusability.

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

Objects are variables and methods and classes create and provide a layout for those objects. If, for example, a house is a class, varying objects fall under that class – a kitchen, living room, den, garage.

1. 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 large part of what makes OOP work – objects inherit properties from classes. Similarly, classes themselves have a hierarchy with subclasses inheriting properties from a parent class. This avoids unnecessarily repetitive code. |
| Polymorphism | Polymorphism allows attributes and methods to have the same name across classes and data types but perform different tasks and operations depending on where and how they are defined. |
| Operator Overloading | Operator Overloading gives the developer an opportunity to adjust and customize the behavior of Python-set operators for objects of a class. Operator Overloading gives a syntax for working with those objects and operators that can give a more readable code. |

### 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 organized and structured set of data that can be stored, accessed and managed efficiently. Advantages of using a database are high levels of structure, accessibility, data integrity and consistency and security for your data.

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

|  |  |
| --- | --- |
| **Data type** | **Definition** |
| INT | Integers |
| VARCHAR(n) | Strings of varying lengths, n being the max number of characters |
| DATETIME | Date/Time values |

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

SQLite might be a better option if the database being constructed and accessed is fairly small and being used for a small application.

1. 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 do think the readability of Python has made it easier to pick up more quickly. There also, by and large, seem to be fewer small nitpicky syntax nuances that inevitably cause bugs to pop up. I think I’ve been able to wind up with a cleaner and more organized code. I do, however, think that JavaScript has more advantages in its frontend capabilities giving it a more broadly applicable scope.

1. 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?

Python’s limitations – when directly looking at it v. JavaScript would be its backend focus. It also seems to execute a bit slower. I may be wrong, but I don’t believe Python is able to be used to develop mobile apps at the moment and that could be a pretty significant drawback.

### 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 (ORM) acts as a go between from your object oriented programming language and your relational database. It allows dealing with the database without having to make cumbersome SQL queries. Advantages include a really tidy code structure, and the ability to use and access databases with fairly limited SQL knowledge.

1. 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 thought it all went fairly smoothly. I really enjoyed that I was able to pick up Python fairly quickly, and that the debugging process felt significantly less cumbersome.

1. 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 used Python to create a command line recipe app and database. It allows users to log, store, edit, delete and search a mySQL database of recipes. Throughout the learning process, I was able to create multiple avenues into and around the app really being able to take a broad approach to learning Python’s capabilities.

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?

Learning the basics of Python and the debugging process went well.

* 1. What’s something you’re proud of?

Diving into this new language (yet another) is feeling less daunting. I know I’m able to learn these things, and to keep expanding my knowledge base.

* 1. What was the most challenging aspect of this Achievement?

I think backend generally is not my complete favorite. Keeping all of the installs and un-seen “where things really live on my machine” pieces straight feels often guided by a “well, it isn’t broken yet” kind of attitude which feels unsteady.

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

Yes, it absolutely did.

* 1. What’s something you want to keep in mind to help you do your best in Achievement 2?

That I can really do this. That the knowledge I’m gaining will be a valuable asset to my job search and hopefully to my future career.

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?

A lot will depend on how you want the final project to exist. Advantages of using Django over vanilla Python are its quick development, scalability, excellent documentation and community support, copious available third party libraries and other resources. It’s limitations are that it does have a steeper learning curve, its fairly large size, and if the project requirements don’t line up with the “Django way”, you may just be out of luck. Plain Python, on the other hand, can be lightweight, have a good bit of flexibility and an easier learning curve. Drawbacks are higher development times an long lasting maintainability.

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

The most significant advantage of MVT over MVC is decoupling – it keeps the data, logic and format to remain separate while with MVC the controller links the model and the view.

1. 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 would like to be able to add another tool to my toolbelt - particularly one that is used so widely throughout the tech community. A lot of where I see myself working, working on, working with following this achievement will depend on what kind of job I manage to get. I would like to play around a bit more with how all of these things can play together – see what I might be able to create for myself.

### 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.*)

The full website would be considered a Django project and the functions and modules within are considered apps – things like photos, chat, login, videos are apps.

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

I would start by creating a virtual environment, installing Django onto that environment and start a new project. This creates the initial file structure. From there I would run the migrations using manage.py. Once this has been initialized, I would stop the server, create a superuser for the admin, and add any apps I need to for the project, and then run the server using manage.py.

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

The Django admin site is an excellent tool to manage a project’s data through the development and administration. The pre-built interface is particularly helpful to be able to manage the contents and users of a project along with the 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

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 objects that are essential in creating and defining the structure of your stored data. The model establishes the format of the data, how it is stored, accessed and manipulated. The benefits of using Django models are that once the model is ordered and created, Django handles the database communication without any assistance. This can save an incredible amount of time, though if your project needs more specific control, Django might not be the framework for you.

1. 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.

Writing test cases early on in a project establishes an environment free from errors. This helps to encourage and make collaboration easier by catching bugs early, preventing regression and enabling scalability. This process also helps to have a solid base upon which a robust app can be built.

### 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.

Django views are functions or classes in Python that receive requests from the user and the URL and handle the business logic as needed and send responses of rendered URLs. They retrieve data from models, pass it to templates, generate the HTML content that goes back to the user and the URL.

1. 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?

In this case, I would use CBVs because, while more complicated to create from the outset, they generate views that are more modular, reusable, and extendable.

1. 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.

Django’s template language uses a simple syntax that lets its users create HTML, XML and other documents generated by a web app.

Templates are stored in separate folders and use a .html extension.

Template syntax is based on variables, tags and filters

Template files use both static and dynamic content and maneuver between the two

### 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.

Static files in Django hold the pieces of information, code, etc. that don’t change on a website – things like background images, stylesheets, JavaScript – but help build the full project.

1. 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 | This is a Django class-based view that displays a list of objects by rendering multiple instances of a table in a database. |
| DetailView | This is another class-based view that shows the full details of a single object from a particular model. |

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.

This particular one took me a bit longer and I had to do a fair bit of wrestling to get everything synched up the way I wanted it to, but overall, I feel like python and django are going well. They feel fairly intuitive and the “django way” seems to keep everything nice and ordered. I, of course, need to keep just working through things. The more I manipulate the code, the more I understand how it’s all working.

### 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.

Authentication is key to maintain data security, create experiences that are custom-tailored to the individual user, keep control, accountability and integrity within your application. It allows the correct users access to the correct information while still protecting sensitive resources.

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

To create a login for a Django web application: set up user authentication, create a user model, define the login and logout views, create the login form, specify the and connect the URLs and views, create templates for login, protect the authorized pages, test and debug.

1. 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() | Handles both authentication and authorization, and authenticate refers to both processes in Django. Aims to be generic and utilizes third parties for things like OAuth, password strength, etc. |
| redirect() | Utility function that takes the URL of the page you want a user directed to, returns the view of that page and its corresponding template. |
| include() | Adds URLs from the apps directory to the main urls.py file in the project directory. |

### 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.

Being able to analyze collected user data is vastly helpful to determine and establish the best user experience for a website/application. Optimizing the UX can drive better engagement, which ends up with better business practices. Being able to make UX decisions that are based in data leads to more sound business.

1. 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.

A QuerySet can be evaluated by slicing, counting, aggregation, iteration, pickling and caching.

1. 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 advantages of QuerySet include it coming built into Django, so it’s already ready to use. It’s designed to handle large data sets smoothly and efficiently and also has a robust API for data manipulation making it pretty easy to create complex queries.

The disadvantages of QuerySet are that it can really only be used with relational databases, limiting the data sources that it can use. It can also be slow if the datasets being used are large or the queries complex.

Advantages of using DataFrame are that it’s highly versatile, can use a broad variety of data sources, has a robust API for data manipulation. It’s optimized for working with data in-memory and can be quite quick with using small to medium sized datasets.

### 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.