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?

The front end is what the client interacts with i.e mobile/web browser or application the backend is the server, business logic and the data. If hired to

to work on the backend of a web application id be working on the web server, api endpoints, authentication and database.

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”)*

Both Python and JavaScript are dynamic high-level scripting languages both are executed line by line however python is build to be more readable and uses understandable command words which assist in keeping the code error free and easier for others to understand. Python has a large number of open-source packages and interactive shell which also assist in keeping code error free as commands can be tested in isolation. Both have a large community support, however Python has one of the largest and most co-operative communities.

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 want to learn more about what Python can be used for

- I want to learn more about automation and scripting

- As I understand Python is the fastest growing language, I want to know how I can

develop my skills to make me stand out from others

### 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 shell automatically indents code for nested statements and auto completion making it more user friendly.
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?** |
| Tuple | Like a list but less flexible and faster | Non-Scalar |
| List | Like an Array in JavaScript, slower and more flexible than a Tuple | Non-Scalar |
| String | A string of zero to multiple characters | Non-Scalar |
| Dictionary | A key-value data structure like a JavaScript Object | 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. A List is faster but a Tuple offers more flexibility since it has more functions to modify it.
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. I would use a dictionary for the vocabularies since this offers key-value pairs. A value I would use a dictionary as well in order to store more than just the translation of the word e.g., some example sentences, definitions and categories.

### 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?:”)  if Destination == ”Chicago”:  print(“Enjoy your stay in Chicago”)  elif Destination == ”New York”:  print(“Enjoy your stay in New York”)  elif Destination == ”Los Angeles”:  print(“Enjoy your stay in Los Angeles”)  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.

I would state that logical operators (and, or, not) are used to combine conditional statements, for example and will return true if both conditions are met, or will return true if one condition is met and not will reverse the result, it returns false if the result is true.

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

Functions are separated block of code that can be reused several times. They have input parameters (arguments) and a return value. They are useful for executing the same operation several times or on different places in the code.

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.

The course material has taught me about automation and scripting, I have researched common uses of python such as its uses in Data analysis and machine learning as well as Web development (booking marking this learning for a later date when I am more experienced with Python <https://www.simplilearn.com/tutorials/data-analytics-tutorial/data-analytics-with-python>)

### 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?

It is important to store data correctly because if the script is terminated either via an error or the script ending the data input could be lost and may have to be re-entered.

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?

Pickles are binary files that can not be read by humans which store python data structures so that python files can retrieve stored data, as well as edit stored data.

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 get the current working directory: **os.getcwd()**

To change the current working directory: **os.chdir()**

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?

Determine what kind of error might occur in order to execute Try-Except blocks to display an error message and kept the script running.

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.

I am proud of where I have come in the last few months, at times the JavaScript was very difficult for me as it was so new, however Python seems to make a lot more sense, I believe this is don’t to my overall knowledge increasing as well as python being a more intuitive language. Sometimes its difficult to write blocks of code from memory I have to refer back to the exercise to break the code down and adapt to the task requirements however I’m confident this will come in time.

### 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?

In OOP everything is an object, using python you can create classes and objects then determine their specific functions. The benefits are it helps create neat and reusable code.

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

A class is a like a blueprint for creating objects. Objects are instances of a class, using its own specific data. For example a Bicycle as a Class, it can define common properties such as has 2 wheels and a frame, Objects could define color, number of gears, size etc. Every Bicycle has 2 wheels and a frame then different objects define its own specific characteristics.

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 means that a class inherits another class. In this case the subclass gets all attributes from the parent class. Also, a subclass can have even more attributes that are defined within the subclass. A subclass can also override an attribute from the parent class. |
| Polymorphism | Polymorphism is when different class names or data types have an attribute with the same name. |
| Operator Overloading | Operator overloading is when you want to use certain operators or functions on objects from classes in order to manipulate or retrieve data. Such operators can be + - < > == and so on or even methods like print. In order to use such operators with self-defined classes, you need to define those methods inside the class with pre-defined method/attribute names. For example, a + operator can be overloaded by defining the \_\_add\_\_ function inside the class or the print function can be specified by defining a method \_\_str\_\_ which returns a readable string that then gets printed by the print function. |

### 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 collections of data to store them permanently. A database can be accessed through a connector from a programming language in order to manipulate the data.

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

|  |  |
| --- | --- |
| **Data type** | **Definition** |
| INT | An integer / whole number like 1 or 5 |
| VARCHAR | A string of characters with a defined maximum length for example 20 characters |
| FLOAT | A decimal number like 1.4 or 3.8 |

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

For testing or small projects that are in development

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?

Both programming languages have their own benefits and disadvantages. The syntax is little different and I feel Python is more user friendly. Python is more for scripts and applications that run on a server and JavaScript is mostly used in the browser on websites.

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 is very nice and offers a lot of opportunities with different use cases. The limitations are that you need to have the correct python version installed on your computer or machine in order to run the script correctly and the virtual environment can be ‘fiddley’.

### 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 makes the communication with a database much easier since it converts the structure of the databases in classes and objects that can easily be handled in python.

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?

It went well, I learned a lot, practiced and now have a working command line app. I did well with the program logic and the program structure. If I were to start over I’d use a different version of Python as I had to keep swapping between versions.

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 know the Python basics and I am able to create full working command line applications with Python. I can read and write files and communicate with a MySQL Database in different ways.

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?

Python understanding in general and program structure and logic.

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

Finishing the Achievement

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

Some database queries where a little challenging and also planning to finish the tasks on time as I set my goals.

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

Yes, I would say it was ok or good but I want and need to learn more about Python.

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

Plan enough time for each exercise and make the application like I want it to work.

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?

Using Django means you have a lot of pre-written code that you can use in order to speed up the development process, scale your app and implement thing more easily. Using vanilla Python means your app is small and lightweight and you have full control over it. But you probably need to write more code in order to achieve the same functionality as with Django depending on what you want to achieve.

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

It is suitable for small and large projects and easy to modify.

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?

I want to learn how to build full-stack applications with user-authentication and everything that you need in order to have a full working app.

* What do you want to get out of this Achievement?

Add a new Framework to my Stack and bee good in that Framework (Django)

* Where or what do you see yourself working on after you complete this Achievement?

I will continue working on the Job preparation course and AlgoExpert after 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.*)

My dream company is Zwift (online cycling app)

The whole website is a project, inside the project 1 there is login, training plans, events, and project 2 is the virtual app itself which shares, login and separate apps such as which bike, avatar, and world.

1. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.
   1. Create virtual env
   2. Install Django
   3. Create a Django project
   4. Migrate the project
   5. Run the server
2. Do some research about the Django admin site and write down how you’d use it during your web application development.

I would use it to show as much data about the application as possible e.g., user data, recipes, statistics, reports and so on

### 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 represent tables in a database. The make it very easy to retrieve database entries and make queries to the database.

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.

It’s important to write test cases in order to ensure a working app all the time. Without test cases one could easily make some changes to the code that may break another part of the code. Test case can detect those changes that make some part of the code not working properly and warn the developer.

### 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 Django view is a function that gets called on a specific URL. A view then runs some logic and returns a template.

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?

I would use class-based-views because they are very good suitable for reusing code.

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.

A template can be any text file like HTML, XML, CSV. It can also hold variables that can get replaced by certain values. This makes templates very powerful.

### 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.

Django can handle static files if one creates a folder for static files and specifies it in the settings.py file. This comes very handy since it allows the developer for example to show static pictures on the website.

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 | Used for displaying a list of objects. |
| DetailView | Used for displaying a single object. |

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.

Its going quite well but the exercises are more time consuming then in the beginning. Its still hard to grasp everything and how things work in Django but I am proud of coming this far.

### 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](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.