CAREER**FOUNDRY**

Python for Web Developers

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](#) of the Orientation course if you're not sure whom to reach out to for help and support.

Exercise 1.1: Getting Started with Python

Learning Goals

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

Reflection Questions

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

   Front end is what the user sees when they view the website, backend is behind the scenes. Backend involves managing data with data bases, and using api's. As a backend developer you're mainly tasked with storing and retrieving data.

2. Imagine you're working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?
   (Hint: refer to the Exercise section "The Benefits of Developing with Python")

   I would explain that python is a high level scripting language just like javascript and is user friendly. Python also offers built-in package management and dynamic typing. Lastly python has a strong community with lots of support.

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?

My first goal is to further understand the basics of python and what benefits it offers compared to other languages. My second goal is to understand what you can do with python and how it is used by companies today. My final goal is to know enough about python to work on my own projects and develop software.

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 allows for easier readability and organization of your code. Ipython allows you to view available methods that can be called on variables in an organized fashion. Ipython changes the color and font of your code and can auto populate information as you type.

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	A whole number can be negative or positive	Scalar
bool	A value that ether hold true or false	Scalar
list	An Array of values that can be changed/added/removed	Non-scalar
dictionary	An Array of values that each have a key and a value pair and can be changed/added/removed	Non-scalar

3. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.

The main difference between a list and a tuple is that lists are mutable, meaning the values throughout the list can be changed. The values can also be rearranged and new values can be inserted anywhere in the list unlike tuples.

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 think a dictionary would be best in this instance, because you could set one side of the flashcard to be the key and the other side to be equal to the value. This way you could change/add/or remove any flashcard in the list and the data would still be organized. This allows for each flashcard to be recognisable and independent from the rest.

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(str("Do you want to travel to Spain, France, or Portugal: "))

      If Destination == "Spain":
            print("Enjoy your stay in Spain!")
      Elif Destination == "France":
            print("Enjoy your stay in France!")
      Elif Destination == "Portugal":
            print("Enjoy your stay in Portugal!")
      Else:
            print("Oops, that destination is not currently available.")

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

A logical operator is any code that returns true or false. So a bool, if, elif, else, ==, >, <, =>, =< are all examples of logical operators.

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

A function is a block of code that can be called at any point in your code. It can have parameters or values that it takes in and uses to create an output. Functions are used to eliminate the need to have to rewrite lines of code. This saves time and keeps code organized.

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 have learned a lot of the basics, which is great. Now I feel like I need to start implementing these basic principles and practice using them so I have a strong foundation of python.

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

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, because a file is a place where you can save the changes that your code makes. Without a file all your variables would be deleted when the code is done executing.

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 convert complex data into bytes which can be stored in a binary file. You would want to use pickles when you are trying to store complex data and keep data encrypted. This is because pickles can only be read by the computer and not humans.

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?

   To find out what directory you're currently in you use the os.getcwd() function. If you want to change current directories you use the os.chdir("<path to desired folder>") function, with the specified new directory.

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?

   You could write a try and except to keep the code flowing even if the error occurs. The try would run the normal code and the except would handle the error that you are worried about and keep the code going with the error.

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.

   I think everything is going well. I'm proud of my knowledge on the foundation of python and I haven't really struggled much. I'm excited to keep moving forward!

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

1. In your own words, what is object-oriented programming? What are the benefits of OOP?
Object-oriented programming is when you create objects to hold key values for your code to operate, and objects are data that is manipulated in your code. The benefits of OOP is it keeps your code more organized and reduces redundancy.

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
An object is one specific example of a class. A class can have as many objects as it wants and each object has the same class traits, but the values of these traits may be different. A real world example would be an ice cream class. The overall theme or topic would be ice cream (the class), but each object or flavor would have different values based on the flavor.

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

Method	Description
Inheritance	Pass in on traits from a parent class to a sub-class. These traits could be attributes or methods.
Polymorphism	When different classes have the same name, but perform different functions, because they are separate classes.
Operator Overloading	Manually assigning basic operation functionality to custom classes since they are not predefined. Examples (+,-,>,<)

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 servers where data can be saved for later use. Databases save information that can be created, read, updated, and deleted. The advantage of databases is they save your information off your device which means they can be accessed from multiple computers.

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

Data type	Definition
VARCHAR(n)	A string in which you can specify the number of characters
DATETIME	A variable that will convert numbers into a date
INT	Any whole number value

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

In a smaller database where not much data is stored and your data can be stored in a .db file instead. These files can be directly manipulated from a python script.

4. Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?

I think the main difference between JavaScript and python is the variable naming conventions and the structure of the code. Using indentation and different syntax. Python is also more OOP focused than JavaScript and in my opinion more user friendly.

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'm not sure what the limitation of python might be, maybe creating advanced user menus and database manipulation and mobile 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?

An object Relational Mapper converts the structure of your database into classes and objects, which makes it easier to send and receive data from the database. This eliminates the need for sql syntax.

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?
   The recipe app went well. I feel like I did well on the methods/ functions that you can call on each recipe. If I were to start over I would improve the user experience and solving more issues with user input.

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 would tell them I have a solid understanding of the foundation of python and have done some work with database management through mysql and sqlalchemy.

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?
      My understanding of each assignment and lesson.
   b. What's something you're proud of?
      I'm proud of my recipe project and my database management skills.
   c. What was the most challenging aspect of this Achievement?
      The most challenging aspect is managing variables and converting data between your script and a database.
   d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?
      Yes, I'm enjoying python a lot and love its syntax.
   e. What's something you want to keep in mind to help you do your best in Achievement 2?
      Keeping track of variables and what type of data type they are and how to manipulate them.

Well done—you've now completed the Learning Journal for Achievement 1. As you'll have seen, a little metacognition can go a long way!

Pre-Work: Before You Start Achievement 2

In the final part of the learning journal for Achievement 1, you were asked if there's anything—on reflection—that you'd keep in mind and do similarly or differently during Achievement 2. Think about these questions again:

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

- Reflect on your learning and project work for Achievement 1. What were you most proud of? How will you repeat or build on this in Achievement 2?
- What difficulties did you encounter in the last Achievement? How did you deal with them? How could this experience prepare you for difficulties in Achievement 2?

Note down your answers and discuss them with your mentor in a call if you like.

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

Exercise 2.1: Getting Started with Django

Learning Goals

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

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?

The pros of django or a framework is they reduce the time it takes to develop code by offering additional functions that the developer can use. If you are working with a database a framework can be a great help from eliminating redundant code. If you're not using a database then a framework might not be necessary.

2. 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 a MVT compared to a MVC is you don't need to map database values to the user view though the url. You can just select the values you want from the database and MVT will handle connecting them to your code.

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?

I would like to learn how to use django to send, get, update and delete data from a database.
- What do you want to get out of this Achievement?
  I want to be able to understand how django functions and what its capabilities are.
- Where or what do you see yourself working on after you complete this Achievement?
  After this achievement I hope to work on more python and django projects.

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.*)
   I would break the website up into different apps and explain how the apps are connected through their urls. The company would have a login app, sign up app, main page app, and different features will also have their own apps. Each app would have its own view and temple.

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

1. Download visual studio code
2. Download python
3. Create an environment
4. Download Django
5. Create a project
6. Migrate and run server locally

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 it to change my database information, manage account permissions, importing/exporting data, and developing/testing.

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.

A Django model is a class that creates variables for your database. Each instance of the class is a row in your database table and each attribute in the class is a column.

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.

Tests help eliminate problems before they occur. When you write tests for your code you get more familiar with your code and find small mistakes that you might have made. Tests also help eliminate problems if you have to make changes to your code or revise your code later.

Exercise 2.4: Django Views and Templates

Learning Goals

- Summarize the process of creating views, templates, and URLs
- Explain how the "V" and "T" parts of MVT architecture work
- Create a frontend page for your web application

Reflection Questions

1. Do some research on Django views. In your own words, use an example to explain how Django views work.

2. Imagine you're working on a Django web development project, and you anticipate that you'll have to reuse lots of code in various parts of the project. In this scenario, will you use Django function-based views or class-based views, and why?

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

Exercise 2.5: Django MVT Revisited

Learning Goals

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

Reflection Questions

1. In your own words, explain Django static files and how Django handles them.

2. Look up the following two Django packages on Django's official documentation and/or other trusted sources. Write a brief description of each.

Package	Description
ListView	
DetailView	

3. You're now more than halfway through Achievement 2! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? You can use these notes to guide your next mentor call.

Exercise 2.6: User Authentication in Django

Learning Goals

- Create authentication for your web application
- Use GET and POST methods
- Password protect your web application's views

Reflection Questions

1. In your own words, write down the importance of incorporating authentication into an application. You can take an example application to explain your answer.

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

3. Look up the following three Django functions on Django's official documentation and/or other trusted sources and write a brief description of each.

Function	Description
authenticate()	
redirect()	
include()	

Exercise 2.7: Data Analysis and Visualization in Django

Learning Goals

- Work on elements of two-way communication like creating forms and buttons

- Implement search and visualization (reports/charts) features
- Use QuerySet API, DataFrames (with pandas), and plotting libraries (with matplotlib)

Reflection Questions

1. Consider your favorite website/application (you can also take CareerFoundry). Think about the various data that your favorite website/application collects. Write down how analyzing the collected data could help the website/application.

2. Read the Django official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.

3. In the Exercise, you converted your QuerySet to DataFrame. Now do some research on the advantages and disadvantages of QuerySet and DataFrame, and explain the ways in which DataFrame is better for data processing.

Exercise 2.8: Deploying a Django Project

Learning Goals

- Enhance user experience and look and feel of your web application using CSS and JS
- Deploy your Django web application on a web server
- Curate project deliverables for your portfolio

Reflection Questions

1. Explain how you can use CSS and JavaScript in your Django web application.

2. In your own words, explain the steps you'd need to take to deploy your Django web application.

3. (Optional) Connect with a few Django web developers through LinkedIn or any other network. Ask them for their tips on creating a portfolio to showcase Python programming and Django skills. Think about which tips could help you improve your portfolio.

4. You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:
   a. What went well during this Achievement?

b. What's something you're proud of?
c. What was the most challenging aspect of this Achievement?
d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

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