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?
    I have taken the Intro to Frontend Development and the Full-Stack Immersion courses from CareerFoundry. Prior to that, I had followed some tutorials on React to familarize myself with coding concepts.
2. What do you know about Python already? What do you want to know?
    I do not know much about Python. I only know it is considered a very convenient language to code in while some do not like how flexible it is. I want to learn more about its application in data science, data visualization, and AI/machine learning. I hope to be able to apply it in my future projects and gain a real practical understanding of how to implement Python in either visualizing my own data or in training models for machine learning such as object detection, etc.
3. What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.
    I am sure there will be some getting used to the different syntax and how code is organized. Consulting online sources and videos should be helpful in facing the 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?

   Frontend is creating a web page that displays information for the user to view. It is important to have a smooth UI that allows users to interact with the web page.

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")

   Javascript and Python are similar in that they have simple commands and have dynamic typing which allows variablesto change types without causing errors. Some differences are they Python comes with a lot of built in packages and pre-installed web operations. I would say that Python is a great language for the project due to its readability, flexibility in code, and its built-in tools that allow for quick development.

3. Now that you've had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?

   I want to learn how to apply Python to what I already know in web developement. I also hope to   learn about data science or machine learning if possible. After completing this Achievement, I see myself working on developing useful web apps that either use data science to provide interesting insight to understanding data or using machine learning to create useful tools such as object detection.

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 highlights syntax and automatically indents texts for nested statements. Each command is also excuted right after typing it so it speeds up the testing of small sections of code. Overall, iPython is more efficient and has better reability which help to speed up coding.

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	Integers are negative or non-negative numbers	Scalar
float	Floats are able to hold decimal numbers	Scalar
tuple	Tuples are linear arrays that hold any types of values.	Non-Scalar
dictionary	Dictionaries are an unordered set of items with key-value pairs.	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.

   Tuples are immutable linear arrays that use a minimal memory. Lists are mutable linear arrays that can be modified or updated and use more memory than 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.

For this language learning app, I would use a list containing dictionaries. Dictionaries are used to save keys for words, definitions and categories. The list contains those "flashcards" or dictionaries and so the list should be mutable to be able to add or remove flashcards. Tuples do not work because they are immutable and do not allow for keys to differentiate values.

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!)*

```
locations = [Japan, France, Spain]

location = input("Where do you want to travel? ")

if location in locations:
    print("Enjoy your stay in " + Location + "!")
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.
   Logical operators determine whether a statement is true or false. By using logical operators, you can create conditions to control when if, elseif, or else statements are triggered.
3. What are functions in Python? When and why are they useful?
   Functions are repeatable blocks of code that can take in parameters/arguments and run code using

those inputs. They are useful when you need to run code many times with different inputs. For example, if you consistently need to find the sum of two numbers, it would be useful to have a sum function instead of re-writing code.

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.
    By learning about how Python functions differently than JavaScript, I am learning about the nuances for using Python such as indentation, lack of curly braces, use of colons, etc. Also, an understanding of data structures like tuples, lists, and dictionaries. This will make it easier to understand how to write code for some of my goals such as data science and machine learning.

Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

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

    File storage is important because after Python scripts finish running, it deletes all existing variables and ends the scripts. So if you did not store local files, there is no way to save data to be used in the future. Basically, without saving data into a local file such as a binary file, there is no persistence of data.

2. In this Exercise you learned about the pickling process with the **pickle.dump()** method. What are pickles? In which situations would you choose to use pickles and why?

    Pickles are a way to transform complex data into a format that can be stored in a binary file. Pickles should be used when you need to store data such as dictionaries into a local file to be accessed in the future.

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?

    os.cwd is how you find which directory you are currently working in. If you wanted to change directories, you would use the os.chdir function which takes a path as an argument.

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?

    Using a try except block, you can control what section of code runs initially, which code runs when an error has occured, and which code run regardless of it was successful or not.

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.

Everything is going quite well. I am proud of my ability to understand the concepts of how Python scripts work which is probably attributed to the detailed Web Development course I took before this. I do not think I am struggling that much with the material, but I do feel that there is a lot of new information and functions that were introduced in a short amount of time. So, it is reasonable to feel like some of various ways of data manipulation could use more practice like list comprehension, enumerate, rstrip, etc.

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

Reflection Questions

1. In your own words, what is object-oriented programming? What are the benefits of OOP?

   Object-oriented programming is when your data attributes and procedural attributes/methods are confined to the object itself which is an instance of a class. The benefits of OOP are that all of the methods and variables are associated with the object so it is organized and the method is always available to the class. In addition, it allows an easy way to create multiple instances of a class or create subclasses which can retain some attributes from the parent class through inheritance but also override attributes that are specific to the subclass.

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

   Classes are like blueprints for an object and contain all the data attributes and procedural attributes for that object. Take an animal as a class example. An Animal class will contain data attributes like the type of animal, size of the animal, or even a method called speak(). An object is when you create an instance of the class and fill in all the info specific to this instance. For example, I can make an Animal object with type of animal = dog, size of animal = medium, and speak() contains print("Woof").

3. 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 when making a subclass of a parent class, it will automatically inherit the attributes of the parent class automatically. So, if the parent class has data attributes and procedural attributes, theywill be passed down to the child class as well. So, an instance of the child object will be able to call

	any methods the parent class had. When necessary, child classes can override the parents methods such as string representation, addition, subtraction, etc by simply writing the same method name in the child class. Inheritance only passes attributes from parent to child, never the other way around.
Polymorphism	When different classes still contain a procedural attribute that is named exactly the same, but functions somewhat differently than each other. For example, having many different animal classes can have a similar procedural attribute like speak(). Although they are named the same, a Dog class might call the method and it would print("Woof!") while a Cat class might call the method and it would print("Meow!"). What this means is that classes have their own class specific outputs despite having similarly named methods. This also applies to built-in functions like lens() which behaves differently when the argument is either a string, list, or dictionary.
Operator Overloading	When creating classes, you also have to create specific methods to choose how the class responds to being called in certain situations. For example, if you were to try to print your class, it will not understand how you want it to function because it is a class, not a string. So by using a reserved function name __str__, you can return how you want the class to behave when used as a string. Similarly, you may want to also specifiy how it would function when being used to add with another object. If you had two instances of the class Height, you would have to manually create a class function to let them know how to behave when added together, subtracted, or using comparison 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?

    Databases are a separate location for storing data that is preserved and can be access at another time. Advantages are that they can store information you want to access again at a future time. In addition, you can use queries to be able to retrieve information based on different fields and characteristics.

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

Data type	Definition
VARCHAR	String with a varying character limit
INT	Integer
FLOAT	Number with decimals

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

    SQLite is a better choice is you just want to test your simple web app because it does not require installation or setup so it is much faster to start using.

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

    Python requires less code in genenral like semi-colons, curly brackets and more, but you have to make sure your indentation is clean because Python relies on indents to understand how code is broken up. Main difference is the formatting and that Python is OOP while JavaScript leans towards functional programming.

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?

    Python as a programming language might be considered too "easy" of a language due to its leniency how variables are declared or how code is formatted using identation, but I do not see any huge limitations on Python. Identation takes time getting used to and may annoy users when errors start to pop up simply due to misaligned block of code.

Exercise 1.7: Finalizing Your Python Program

Learning Goals

- Interact with a database using an object-relational mapper
- Build your final command-line Recipe application

Reflection Questions

1. What is an Object Relational Mapper and what are the advantages of using one?

    An Object Relational Mapper is library used to connect database structure into classes so instead of constantly using SQL commands to make changes in the database, you can simply interact with the classes.

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?

Making the Recipe app was very interesting due to using Python as the coding language instead of JavaScript and noticing the differences between the two. I feel like I did well in creating a UI that felt smooth and natural to use like allowing multiple inputs from the user for entering ingredients for the recipe. If I were to start over, I would try to optimize the codes and see if there are ways to reduce code length.

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.

Through my Python for Web Development course, I have gained lots of experience in using Python to create an efficient and useful application. I have created a Recipe App to allow users to create, save, and edit recipes for future use and stored all relevant information in an MySQL database. I understand how to use Object Relational Mappers such as SQLAlchemy to create relationships between classes and the data structure so interacting with the database can be done using Python instead of SQL syntax. By virtue of data being used as classes, it was important to understand how classes were structured, as well as, how they functioned. Data attributes could be stored within objects/instances of classes and functions/data procedures could be written to interact with those data attributes. It also reinforced the understanding of data structures such as tuples, lists and dictionaries.

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?

        Learning how Python is formatted and written went well and it was not too difficult to adapt to using identation as a way to separate blocks of code.

    b. What's something you're proud of?

        I am proud of learning more of how classes and objects function and how they can be used to easily replicate code and create objects with similarities, as well as, differences in certain cases.

    c. What was the most challenging aspect of this Achievement?

        Understanding all the different ways to handle or manipulate data was somewhat challenging because there were many different ways to approach it. Also, storing the data into the MySQL database as a string and retrieving the data and changing it back to a list was a little tedious.

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

        I think the achievement met my expectations for a good foundation of how Python is supposed to be used. It does give me confidence to start working with Python for future projects.

    e. What's something you want to keep in mind to help you do your best in Achievement 2?
        I should keep in mind to understand how it is similar and different to making apps

using JavaScript. It will help build an understanding of the general concepts involved with web development.

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?
    Yes. I believe my study routine was effective during Achievement 1.
- Reflect on your learning and project work for Achievement 1. What were you most proud of? How will you repeat or build on this in Achievement 2?
    I was most proud of adapting to a new coding language, as well as, understanding how it was similar and different to JavaScript. It allowed me to understand how to use the language in similiar scenarios.
- 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?
    There was some difficulties understanding all the different ways to manipulate data but by looking online, I could find their use cases and how to use them efficiently.

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?

Vanilla Python has advantages of being completely new and thus gives the user more control on how it will function and determine what kind of features will be necessary for the application. Downsides are that features will need to be created from scratch and take longer to develop a working prototype. Django's advantages are that it is a batteries-included framework will features like database access and file operations. It also has great scability to provide services to large numbers of users. Downsides are that Django is very opinionated and must be coded using the "django way" so it is not very flexible on how code is written.

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

MVT is faster than MVC because you do not need to code to fetch data from the database and map it to the URL. You only need to specify the data needed and the template will take care of the rest.

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 want to learn how it is different from React and in what ways is it better and worse.

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

  I want to understand how to create a working website using Django and how I might be able to integrate other libraries into my web app such as data science and maybe machine learning.

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

  I see myself looking into data scienece and machine learning and how to truly integrate them with my web apps for the future.

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

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.*)

   Let's use Google as an example. Google is considered the project that has many apps used for specific functions. Some apps would include login, gmail, images, meet, chat, drive, and more which all have their specific functions.

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

   Open up the terminal. Create a virtual environment using mkvirtualenv command. Activate that virtual environment using workon command. Install Django in virtual environment using pip install django. Navigate to location you wish to create project. Create a new project using django-admin.exe startproject command. Rename root directory to src for clarity using move command. Use python manage.py migrate to create database. Lastly, run server using python manage.py runserver.

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

   The admin site can be used to manage users. It also displays all the models you create to be able to add new entries. You can create a catalog of all the different models and entries, as well as, add list filters and more. Simply, the admin site can be used to manage your database and the data entries contained within.

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 python classes that function like schemas connecting to data in your database. So

instead of having to interact directly with the database using SQL commands, you can just setup a Django model that will represent the data in the database.

2. In your own words, explain why it is crucial to write test cases from the beginning of a project. You can take an example project to explain your answer.

    It is important to write test cases from the beginning of a project so insure that your app is functioning properly. It will save lots of time in the future because it will constantly validate and check your app for mistakes and catch them early on. It also makes your code stronger and more reliable because it is being tested for any weaknesses or bugs as the project is ongoing.

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 kind of like API endpoints. When the user navigates to a specific URL, the views associated with that URL is triggerred. The view is a function or class. The function should then return a render of the assigned template.

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?

    I would use class-based views. Class-based views are perfect for repeatable code and when a view needs to be repeated across multiple apps.

3.   Read Django's documentation on the [Django template language](#) and make some notes on its basics.

    Variables are shown using double brackets {{}}. Filters can be applied using pipes (e.g. {{name| lower}}. Tags are more powerful as they can either create text as output or perform logic like loops and logic statements. Template inheritance is also very powerful. It is used to first create a base template that will be used initially. A child template can be created with different tags. As long as the child template extends the base template, it will copy the base template and only replace sections where the child has the same tags as the base. The childrens tags will override the base template's tags.

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 are files that are not expected to change while using the website and is not generated by the user. Usually static files are stored within apps under the filepath "app_name"/static/"app_name"/images. Within html templates, use {% load static %} to allow the img source to reference the static folder within the app.

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	Allows classes to be created using ListView package as a parent class which has a built-in variable called object_list. The class based view should also contain the model and the template. This object_list contains all the objects/instances of that model which can then be iterated through to create a list with their values.
DetailView	Similar to the ListView, DetailView provides the object variable to display values from the database.

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.

    Everything is going well. I think I am having a decent understanding of how Python works. I am proud of creating a header navigation bar that allows for linking all the different pages efficiently. I am also proud of creating attributes under the models using many-to-many fields and showing ingredient models under the recipe list.

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 very important for an application because it creates stronger security so only certain individuals have access to certain functionalities of the website. In addition, authentication allows for the creation of user specific content that is geared specifically for them.

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

   Firstly, create a template containing the form with method POST. Under views.py, import AuthenticationForm from django.contrib.auth.forms and import authenticate, login, logout from django.contrib.auth. Create a function based view with form = AuthenticationForm and if the form is valid, get the values and use authenticate to make sure they match the user data. If true, login function is called and redirect to another page. There is also a context object that contains a form and an error message. This is how the template knows that kind of form to display or what kind of error message to display. And also provide the render function with the template.

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()	By default, it accepts arguments for username and password and check it against the backend to see if a user matches. If so, it returns a User object.
redirect()	Takes a url name as an argument, and redirects the page to the url specified which is linked to a view that renders an html template.
include()	It is used within urlpatterns inside a urls.py file. It basically adds paths being used in an app's url.py file so it will recognize the app's different endpoints and their corresponding views.

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.

   Analyzing the collected data allows the web developers to understand how users are interacting with the website. By tracking how long users complete tasks or what frequency they work on the assignments lets the tutors understand how well the lesson is written and see whether the material needs to be altered in any way.

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

   Iterations, slicing, pickling, repr(), len(), list() and bool().

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.

   Queryset is fine when working with a small database but when your database is large, you   should convert to a dataframe and use Pandas which is intended for fast and efficient data   analysis.

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.

   Templates can link stylesheets usually from the nearest static folder containing css files. This is one way to style your Django website using templates.

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?

      Learning how python works and the types of data structures it uses. In addition, understanding how Django is organized and strucutured through urls, views, models and templates.

   b. What's something you're proud of?

      I am proud of learning more about the relationships of SQL databases, as well as, learn about how to visualize the data through matplotlib.

   c. What was the most challenging aspect of this Achievement?

      Some of the Django relationships required more complex connections like many-to-many relationships. So when using queryset to retrieve that data is more complex. Also, the dataframes table was confusing to manipulate and add a hyperlink to the recipe.

   d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

I think ths achievement met my expectataions and gave me confidence in beginning to use Django and python libraries in future projects.

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