Reflection Questions Achievement 2

1. **Was your study routine effective during Achievement 2? If not, what will you do differently during Achievement 2?**
   Overall, I think my approach to learning was reasonably effective during Achievement 1. I was able to complete all the tasks within the suggested time windows and fill the requirements of every task. The only thing I want to adjust is my overall pace of learning. For this half of the course, I need to complete the assignments a bit faster, completing 2-3 assignments per week. This would mean dedicating more time to the course than I had been in the month before.
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?**
   I can't say I'm particularly proud of anything from the last Achievement, mostly because I simply did what was required of me. That said, I am pretty happy with how my code is laid out. During this unit I was much more diligent about commenting on blocks of code, and it helped me stay organized.
3. **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?**
   My biggest hurdle was confidence in my own knowledge of coding. I am very new to Python, so I felt the need to double check every function or loop I laid down. It wasn't until I checked other students' submitted works and confirmed my own choices that I was comfortable submitting my work. Honestly, I feel this was an effective strategy and I learned a lot while doing it. It was helpful to see how others approached the tasks laid out for us. Some works made me change my own approach, while others simply showed a different solution that I chose not to utilize.

Reflection Questions Exercise 2.1

1. **Suppose you're a web developer in a company and need to decide if you're using vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?**
   One of the biggest advantages of Django or any similar framework, is the pre-made structure it provides. Using it on a project would save developers time, especially if it needed to be launched quickly and has both a backend and a frontend side to it. Django is well-known for its effective data management, so if there's a database involved, it would be a good idea to work with Django. Conversely, if it's a simple task where the developer doesn't want to be constricted by Django's structure, it would be simpler and faster to use vanilla Python.
2. **In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?**
   The biggest advantage MVT has over MVC is not having to code the controller logic. A framework like React makes using JS pretty intuitive, but as a developer I still have to write out the implementation of data collection. With MVT, the developer needs only to specify which data is necessary and how it's presented, and the framework will take care of the rest.
3. **Now that you're 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:**
   a. **What do you want to learn about Django?**
   b. **What do you want to get out of this Achievement?**
   c. **Where and what do you see yourself working on after you complete this Achievement?**

   Here are three things I would like to work on during this Achievement:
   1. Building up confidence in my code structure decisions.
   2. Finish the Achievement with a stable base of knowledge in Python that I can work on expanding in the future, on my own time.
   3. After completing the Achievement, I intend to use the skills gained to round out my development portfolio and land a job or internship working in web development.