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?
   Django framework offers support in backend without compromising features at frontend. Its MVT architecture ensures fast and easy development. Using a framework help an application to run faster. Django follows DRY principles to keep code non-repetitive, non-redundant and efficient. Django scalability also makes it simple to add new hardware and components. It's also built with security in mind. Most importantly, Django open source and has huge community which means easier to get support.

   However, it has its drawbacks as well. Django must go Django-way which means its firm structure may cause some difficulty as can't diverge from rules. Also depends if the project requires database or file management – which if not required, no need to use Django.

   Also depends if developer wants to have more control over certain parts of system. If wants to have more control, Django may not be suitable.

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?
   With MVT, developers don't have to write code to fetch data from database and map it to URL. Only need to specify which items to present to the user and framework will prepare and send it.

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?
     Increase my knowledge and coding skills using Django.
   - What do you want to get out of this Achievement?
     Create an application out of Django
   - Where or what do you see yourself working on after you complete this Achievement?
     As of now, just want to improve my knowledge and skills on coding.