

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

Django views are the components responsible for processing requests and returning responses. They determine what content to display based on the user's request. For example, in a simple blog application, a view might fetch a list of blog posts from the database and render them to the user's browser.

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?

In a scenario where code reuse is anticipated, class-based views are often preferred in Django projects. Class-based views offer more flexibility and organization, allowing developers to inherit behaviors from parent views and override specific methods as needed. This promotes cleaner and more maintainable code compared to function-based views.

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

Django's template language is a powerful tool for generating dynamic HTML content. It allows developers to embed Python-like code directly within HTML templates to display dynamic data. Basic features include variable interpolation, loops, conditionals, and template inheritance, providing a concise and intuitive way to create dynamic web pages.