

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

-If you are creating a blog and want to display each post in a list format, you would create a view that queries the database for all the posts and send that data to a template. The template then generates an HTML page to display them. When a user visits the page the view gets the request and fetches the data then returns the rendered HTML 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 this case the class-based views would be the better option. CBV offers a way to group related logic into reusable components making it easier to reuse in the future.

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

-Django's Template Language is used to create dynamic HTML pages by embedding Python like syntax within HTML

-Variables are denoted by double curly braces, `{{variable}}`. Variables cannot start with an underscore or a number. The `( . )` is used to access a variables attributes.

-Filters modify variables, `{{name|lower}}`, the name here is the variable and it is passed through the lower filter. Filters can be chained, one filter is applied to the next.

-Tags (for, if, elif, else) look like `{% tag %}`

-Commenting out looks like `{# Greeting #}` hello, would render only hello. Greeting is commented out