Reflection Questions

1. Do some research on Django views. In your own words, use an example to explain how Django views work.
    a. In Django, views are responsible for taking a web request and returning a web response. Views in Django can be written as either function-based or class-based.

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?
    a. For a project where you anticipate needing to reuse a lot of code across different parts of the project, class-based views are generally more suitable. CBVs leverage object-oriented principles like inheritance and encapsulation, making them ideal for creating reusable, extendable, and modular components.

3. Read Django's documentation on the Django template language and make some notes on its basics.
    - Variables: Represented by {{ variable_name }}, variables are placeholders in a template that get replaced with values from a context object passed to the template.
    - Tags: Enclosed within {% tag %}, tags provide logic in the template layer, such as looping through lists, conditionally displaying content, or loading other templates.
    - Filters: Filters transform the display of variables. They are used within variable brackets, such as {{ variable|filter:"parameter" }}, to apply formatting or convert forms.
    - Comments: Anything between {# and #} is treated as a comment and not executed or rendered in the output.