# Exercise 2.4: Django Views and Templates

## Learning Goals

- Summarize the process of creating views, templates, and URLs
- Explain how the “V” and “T” parts of MVT architecture work
- Create a frontend page for your web application

## Reflection Questions

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

Django views are python functions that handle HTTP requests and responses. They process and return data from the backend to the user. Using views, developers can create logic to process and generate data in response to user requests. This makes it easier to build complex web apps.

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?

I would use class-based views because it does a better job when the developer can take advantage of reusing code. I can define reusable view classes that can be subclassed and extended to create new views with minimal code duplication. Class-based views also provides better organization and built-in functionality that can help reduce code duplication.

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

* Django’s template language is a simple and powerful syntax that allows developers to build dynamic HTML, XML, or other types of documents generated by a web app.
* Templates are stored in separate folders with .html extensions
* The template language uses a syntax based on variables, tags, and filters.
* Template files contain both static and dynamic content