

## 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 functions or classes in Python that receive requests from the user and the URL and handle the business logic as needed and send responses of rendered URLs. They retrieve data from models, pass it to templates, generate the HTML content that goes back to the user and the URL.

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, I would use CBVs because, while more complicated to create from the outset, they generate views that are more modular, reusable, and extendable.

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

Django's template language uses a simple syntax that lets its users create HTML, XML and other documents generated by a web app.

Templates are stored in separate folders and use a .html extension.

Template syntax is based on variables, tags and filters

Template files use both static and dynamic content and maneuver between the two