

## EXERCISE 2.4: DJANGO VIEWS AND TEMPLATES

### 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 THE COMPONENTS RESPONSIBLE FOR HANDLING THE LOGIC BEHIND EACH REQUEST THAT A USER MAKES TO A DJANGO WEB APPLICATION. THEY RECEIVE REQUESTS FROM A BROWSER, PROCESS DATA, AND RETURN A RESPONSE, USUALLY BY RENDERING A TEMPLATE
  - DEFINE VIEW IN APP/VIEWS.PY
  - CREATE THE TEMPLATE IN APP/TEMPLATES/APP/HOME.HTML
  - MAP URL TO VIEW IN APP/URLS.PY
  - REGISTER URL IN PROJECT/URLS.PY

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?

- IT WOULD BE MORE EFFICIENT TO USE CLASS-BASED VIEWS (CBVS) RATHER THAN FUNCTION-BASED VIEWS (FBVS), BECAUSE CODE REUSABILITY, EXTENSIBILITY, MAINTAINABILITY, CONSISTENCY. BY LEVERAGING INHERITANCE AND DJANGO'S BUILT-IN GENERIC VIEWS, YOU CAN BUILD FLEXIBLE AND SCALABLE APPLICATIONS WITH LESS EFFORT COMPARED TO USING FUNCTION-BASED VIEWS.

3. READ DJANGO'S DOCUMENTATION ON THE DJANGO TEMPLATE LANGUAGE AND MAKE SOME NOTES ON ITS BASICS.

- THE DJANGO TEMPLATE LANGUAGE (DTL) IS DESIGNED TO HANDLE DYNAMIC HTML GENERATION, SEPARATING THE PRESENTATION FROM LOGIC IN WEB APPLICATIONS AND IS DESIGNED TO MAKE IT EASY TO WRITE CLEAN, REUSABLE, AND MAINTAINABLE TEMPLATES WITHOUT MIXING IN COMPLEX LOGIC.