

1. Do some research on Django views. In your own words, use an example to explain how Django views work.
In Django, views are Python functions or classes that receives web request and return a web response.
Response can be as simple as returning static information or as complicated as interacting with database, accepting user input, and returning user as response.
View holds the logic which Django will run when receiving web request. Views usually are put in views.py file located in apps folder
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?
Will use CBV in this case. As there's anticipation of reusing codes in various parts of code, CBV is more suitable as they're based on Python classes. CBV able to create views that are easier for reusing. It helps to reduce effort in rewriting the code.
3. Read Django's documentation on the Django template language and make some notes on its basics.
Django template system meant to express presentation, not program logic
 - Provides tags: if tag for boolean, for tag for loopingA template is a text file – contains variables and tags.
 - Variables can be replaced with values when template evaluated
 - Tags control logic of templateVariable looks like this {{ variable }}
 - Use dot notation to access attributes of variableTags looks like this {% tag %}
 - More complex than variables