

# Django

① what is Django ? what kind of framework is it ?

Django is a high level python web framework that encourages rapid development and clean pragmatic design. It follows MVT architectural pattern.

② Main benefits of using Django over other framework ?

Django is scalable, secure comes with built in features like admin panel & supports rapid development.

③ compare Django and flask.

<u>Django</u>	<u>Flask</u>
Django is fullstack with built in features	Flask is micro and lightweight
Django uses ORM & MVT	Flask gives more flexible and uses jinja2 templates.

④ what does battery included mean in Django ?

It means django comes with many built in features like authentication, admin panel, ORM, so you don't need third party tools for basic needs.

⑤. how does Django follow MVT/MOT architecture?

Django uses MVT  $\rightarrow$  Model (data)  
View (logic)

Template (UI)  
It separates business logic from presentation layer for better maintainability.

⑥. how do you install Django using pip?

Run `pip install Django`

⑦. which command used to create new Django project?

`django-admin startproject projectname`

⑧. how do you run django development server?

Navigate to the project folder and run `python manage.py runserver`.

⑨. what URL is used to access a running Django project by default?

`http://127.0.0.1:8000/`

⑩. how do you verify if Django is installed & working correctly?

`django-admin --version`

## Day 2: Project Structure:

⑪. what is the purpose of `manage.py` file?

It acts as a command line utility



to manage django apps. you use it for tasks like running server or migrations.

(13) what does settings.py file contain?

It contains configuration settings like database info, installed apps, middleware and static files.

(14) Explain purpose of urls.py?

It defines URL patterns and map them to corresponding views. It acts as router for application.

(15) wsgi.py :- they are entry points for WSGI (for sync apps)

ASGI → (for async apps) servers.

they help deploy Django applications.

(16) role of --init--.py?

It makes a directory as a python package. This help python treat folders like module.

(17) how do you create a new Django app within a project?

python manage.py startapp appname.

(18) where do you register a newly created app so Django recognizes it?

Add app's name to Installed apps in settings.py.

Q18. can a django project contain multiple apps?

yes. a project can have apps like users, blog, and products working together.

Q19. what is a view in Django?

A view is a python function or class that takes a request and returns a response. It contains business logic.

Q20. what are function-based views (FBVs)?

FBVs are simple python function defined in `views.py` that return an `HttpResponse` or render a template.

Q21. how do you map URL to view in Django?

In `urls.py`, use `path('route/', view_func)` to link URL pattern to a view.

Q22. which Django function is used to return plain text response?

use `HttpResponse("Hello")` from django. It helps to return plain text.

Q23. purpose of render() function

render combines a template with context dictionary & returns an `HttpResponse`.



Q4) proper folder structure for templates

place templates inside appname / templates /  
appname / for django to locate them correctly

Q5) how do you include an apps. urls. py  
use `path('app /', include('app. urls'))`  
in project urls. py.

Q6) Django will rise `TemplateDoesNotExist`  
error

Q7) how do you pass data from a view to  
template?

pass a dictionary (context) in `render()`  
function like `render(request, 'template.html',  
{ 'key': value })`.

Q8) explain context of context dictionary.

It is a python dict containing data  
passed from view to be used in template.

Q9) diff between `HttpResponse()` & `render()`

`HttpResponse()` sends raw text as HTML  
while `Render()` combines templates  
with context for dynamic content.

Q10) what is jinja2 & how it is used in Django

jinja2 is a templating engine used by  
Django. It allows embedding python like  
expressions in HTML file.

31) a base template is extended by child templates using `{% extends "base.html" %}` for consistent layout.

32) purpose of `{% block %}` & `{% endblock %}` they define placeholders sections in base templates that child template can override.

33) how do you extend base template?  
use `{% extends "base.html" %}`

34) syntax for outputting variables inside a template?

`{{ variable-name }}`

35) how do you use loop inside a template?

use `{% for item in list %}`

`{% endfor %}`

36) if-else conditions

`{% if condition %}`

`{% else %}`

`{% end if %}`

37) Static files in Django

static files include CSS, JS, images.



33) how do you load 2 use static files  
use `staticfiles` load static files 1. 2 then refer  
with `static` 'path/to/file.css' 1. 2

39) when static files be stored in Django app  
store them in `appname/static/appname/`

40) what is ORM in Django & why it useful?  
ORM lets you interact with database  
using python code instead of SQL.

41) how do you define model in Django?  
create a class in `models.py` inheriting  
from `models.Model` with fields like `name =`  
`models.CharField()`

42) `CharField` → model field type to store  
text  
`TextField` → long text  
`DecimalField` → use / value.

43) steps to create and apply migrations

`python manage.py makemigrations`

`python manage.py migrate`

44) what database does Django use by  
default?

By default, Django uses SQLite &  
a lightweight file-based database.

45) how do you view & test model data using Django shell?

using python manage.py shell to run python commands & interact with models using queries.

46) superuser :-

Run python manage.py createsuperuser & provide username, email, password.

47) what is URL to access django admin panel?

By default its available at <http://127.0.0.1:8000/admin/>.

48) how do you register model so it appears in admin panel?

admin.site.register(ModelName) in admin.py of app.

49) purpose of list\_display, search\_fields, and list\_filter in admin customizer?

They control what fields are shown, searchable, & filterable in admin list view.

50) how do you customize how a model appears in Django admin interface?

create a custom ModelAdmin class and register it using admin.site.register(Model, customAdmin).