Django Interview Questions & Answers (Practical + Conceptual)

This document compiles **interview-ready Django questions and answers** based on real beginner-to-intermediate issues, framed professionally with short examples.

1️⃣What is the difference between a Django project and a Django app?

Answer:
A **Django project** contains global configuration such as settings, URLs, and deployment configuration, while a **Django app** contains business logic like models, views, templates, and migrations.

Example:

```
project/ → settings.py, urls.py
blog app/ → models.py, views.py
```

Django apps are reusable; projects are not.

2️⃣Why doesn't Django create `models.py` in the project folder by default?

Answer:
Django does not create `models.py` in the project folder because **models belong to apps, not the project**. The project layer is only responsible for configuration and wiring apps together.

This enforces **separation of concerns** and keeps apps reusable.

3️⃣Why do we need to create an app even though a project folder already exists?

Answer:
The project folder is not an application. It only contains configuration. Creating an app using `startapp` gives a proper structure for models, views, admin, and migrations.

Example:

```
python manage.py startapp accounts
```

4️⃣What does `python manage.py migrate` do?

Answer:
It applies migration files to the database and creates the required tables so Django features like authentication, admin, and sessions can work.

Without running `migrate`, the database has no tables.

5️⃣Why is running `migrate` important the first time?

Answer:
Because Django's built-in apps (`auth`, `admin`, `sessions`) depend on database tables. Without migrations, these features will crash at runtime.

6️⃣What is the difference between `makemigrations` and `migrate`?

Answer:
- `makemigrations` creates migration files (the plan) - `migrate` applies those migrations to the database (execution)

Analogy:
Blueprint vs building the house.

7️⃣What does `python manage.py createsuperuser` do?

Answer:
It creates an admin user with full permissions to access the Django admin panel.

Example:

```
python manage.py createsuperuser
```

8️⃣Why does Django show a “Pillow is not installed” error when using `ImageField`?

Answer:
Because Django requires the **Pillow** library to process image files, and it is not installed by default.

Fix:

```
pip install pillow
```

9️⃣Why do we install Django inside a virtual environment?

Answer:
To isolate project dependencies so different projects can use different Django versions without conflicts.

Analogy:
`venv` in Python ≈ `node_modules` in Node.js

🔟Is `venv` the same as Docker?

Answer:
No. A virtual environment isolates **Python packages only**, while Docker isolates the entire operating system environment.

1️⃣1️⃣What is `uv` and why is it used?

Answer:
`uv` is a modern, fast Python package and environment manager written in Rust. It replaces tools like pip, virtualenv, and pipx.

It is faster due to caching and parallel installation.

1️⃣2️⃣How is `uv` similar to npm?

Answer:
`uv` is similar to **npm + npx combined**. It installs dependencies, manages environments, and runs tools in isolation.

1️⃣3️⃣Why do we add `INTERNAL_IPS = ["127.0.0.1"]` when using django-tailwind?

Answer:
It tells Django that requests from the local machine are trusted, enabling development-only features like live reload.

1️⃣4️⃣Why does Tailwind CSS not work when using `className` in Django templates?

Answer:
Because Django templates use **HTML**, not JSX. The correct attribute in HTML is `class`, not `className`.

Correct:

```
<h1 class="text-3xl bg-orange-500">Hello</h1>
```

1️⃣5️⃣What is this problem called when using `className` instead of `class`?

Answer:
It is called a **syntax mismatch between JSX (React) and HTML (Django templates)**.

1️⃣6️⃣Does installing `django-tailwind` using `uv` cause future issues?

Answer:
No. `uv` installs the same Python packages as pip. Any Tailwind-related issues usually come from Node.js, not uv.

1️⃣7️⃣What does `pip install "django-tailwind[reload]"` do?

Answer:
It installs `django-tailwind` along with its normal dependencies and extra development dependencies required for live reload.

It does **not** install Tailwind CSS itself.

1️⃣8️⃣Can Django use databases other than SQLite?

Answer:
Yes. Django supports PostgreSQL, MySQL, MariaDB, and Oracle.

Example (PostgreSQL):

```
ENGINE = "django.db.backends.postgresql"
```

🔝Final Interview Takeaway

> **Django enforces clean architecture by separating configuration (project) from functionality (apps), uses migrations for safe database evolution, and relies on isolated environments for stability and scalability.**

✅ This list covers **most beginner-to-intermediate Django interview questions** based on real-world development experience.