# Understanding manage.py in Django

## 1. What is manage.py?

In every Django project, there is a file called manage.py located in the project’s root directory. It is automatically generated when you create a new Django project using the command:  
  
django-admin startproject projectname  
  
The manage.py script acts as a command-line utility for interacting with your Django project. It helps perform administrative tasks like starting the server, creating apps, managing databases, and more.

## 2. Purpose of manage.py

The manage.py file serves as a wrapper around the django-admin command. It sets the DJANGO\_SETTINGS\_MODULE environment variable so Django knows which settings file to use.  
  
In short:  
- It provides an easy way to run Django commands.  
- It ensures all commands are executed in the context of your specific project.

## 3. Common Commands in manage.py

### runserver

Starts the Django development server. Example: python manage.py runserver 8080

### startapp

Creates a new Django app. Example: python manage.py startapp blog

### makemigrations

Generates migration files from model changes.

### migrate

Applies migrations to the database.

### createsuperuser

Creates a superuser for Django admin.

### shell

Opens a Python shell preloaded with Django settings.

### showmigrations

Lists all migrations and their status.

### sqlmigrate

Shows the SQL that Django will run for a migration.

### collectstatic

Collects all static files for production.

### test

Runs all tests in the project.

## 4. Example Workflow

Here’s an example of common manage.py commands used during Django development:  
  
python manage.py runserver  
python manage.py startapp accounts  
python manage.py makemigrations  
python manage.py migrate  
python manage.py createsuperuser

## 5. Summary of Commands

|  |  |
| --- | --- |
| Command | Description |
| runserver | Starts the Django development server. Example: python manage.py runserver 8080 |
| startapp | Creates a new Django app. Example: python manage.py startapp blog |
| makemigrations | Generates migration files from model changes. |
| migrate | Applies migrations to the database. |
| createsuperuser | Creates a superuser for Django admin. |
| shell | Opens a Python shell preloaded with Django settings. |
| showmigrations | Lists all migrations and their status. |
| sqlmigrate | Shows the SQL that Django will run for a migration. |
| collectstatic | Collects all static files for production. |
| test | Runs all tests in the project. |

## 6. Summary

- manage.py is the central tool for managing Django projects.  
- It provides commands for running the server, managing apps, and handling the database.  
- Always use python manage.py instead of django-admin for project-specific operations.