

code
institute



Django Cheat Sheet

Useful Django terminal commands

We recommend **you don't use** these commands before you have been taught about them in the lesson content.

Getting started commands

| Terminal Command | Description | Example |
|---|---|--|
| pip3 install <package_name><optional_version_number> | Installs packages with an optional version | pip3 install Django~=4.2.1 |
| pip3 freeze --local > requirements.txt | Creates/updates the requirements.txt file | pip3 freeze --local > requirements.txt |
| django-admin startproject <proj_name> . | Starts a django project, don't forget the full stop [.] to start the project in the current directory | django-admin startproject my_project . |
| python3 manage.py startapp <app_name> | Creates a new Django app in the Django project | python3 manage.py startapp blog |

Migration and database commands

| Terminal Command | Description | Example |
|---|---|---|
| <code>python3 manage.py makemigrations</code> | Create migrations files for any new or updated models across the entire project | <code>python3 manage.py makemigrations</code> |
| <code>python3 manage.py makemigrations <app_name></code> | Create migrations files for any new or updated models within the specified app | <code>python3 manage.py makemigrations blog</code> |
| <code>python3 manage.py migrate</code> | Migrate your database with any new migrations across the entire project | <code>python3 manage.py migrate</code> |
| <code>python3 manage.py migrate <app_name></code> | Migrate your database with any new migrations within the specified app | <code>python3 manage.py migrate blog</code> |
| <code>python3 manage.py migrate <app_name> zero</code> | Reverts all migrations for a specified app to the initial state, effectively undoing them. | <code>python3 manage.py migrate blog zero</code> |
| <code>python3 manage.py makemigrations --dry-run</code> | Migration preview feature that allows you to see potential model changes without creating actual migration files, enabling you to review changes before committing. | <code>python3 manage.py makemigrations --dry-run</code> |
| <code>python3 manage.py makemigrations --check</code> | New: Checks if the current model states match the database migrations without actually making new migrations. | <code>python3 manage.py makemigrations --check</code> |
| <code>--check</code> is for ensuring all necessary migrations have been made (and is non-interactive), while <code>--dry-run</code> is for previewing what migrations would be made without actually making them. | | |

General commands

| Terminal Command | Description | Example |
|--|---|--|
| <code>python3 manage.py runserver</code> | Run your django app in the browser | <code>python3 manage.py runserver</code> |
| <code>python3 manage.py createsuperuser</code> | Creates an admin user for accessing the Django Admin site. | <code>python3 manage.py createsuperuser</code> |
| <code>python3 manage.py collectstatic</code> | Collects all static files from each of your applications into a single location that can easily be served in production. Only need to use this when DEBUG is set to False as Heroku will automatically collectstatic. | <code>python3 manage.py collectstatic</code> |
| <code>python3 manage.py test</code> | Run all unit tests in files that start with test_ across the entire project | <code>python3 manage.py test</code> |
| <code>python3 manage.py test <app_name></code> | Run all unit tests in files that start with test_ inside the app specified | <code>python3 manage.py test blog</code> |
| <code>python3 manage.py loaddata <fixture_name></code> | Loads data from a fixture into the database. This is particularly useful for setting up a database with predefined data. | <code>python3 manage.py loaddata initial_data</code> |
| <code>python3 -V</code> | Displays your current Python version, for example, if you need it to set up the Heroku Python version | <code>python3 -V</code> |

| | | |
|--------------------------|--|---|
| pip3 show <package_name> | Gives information on the package including its location in your file structure | pip show django-allauth |
| cp -r <from>* <to> | Copies files from one location to another, for example when duplicating package templates so you can personalise them. | cp -r /home/cistudent/.local/lib/python3.9/site-packages/allauth/templates/* ./templates/ |

Other useful commands

| Terminal Command | Description | Example |
|---|--|--|
| python3 manage.py dumpdata <app_name> > <filename>.json | New: Creates a fixture (in JSON format) from the current database. It can be used for backup purposes or to populate another database with the same data. | python3 manage.py dumpdata blog > blog_fixtures.json |
| python3 manage.py flush | New: The flush command is different from migrate zero. When you run python manage.py flush, it doesn't undo migrations; instead, it removes all data from the database and resets primary key sequences for all models. | python3 manage.py flush |