

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

## 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 &lt;app_name&gt;</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 &lt;app_name&gt;</code>	Migrate your database with any new migrations within the specified app	<code>python3 manage.py migrate blog</code>
<code>python3 manage.py migrate &lt;app_name&gt; 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>	<b>New:</b> 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 <b>DEBUG</b> is set to <b>False</b> as Heroku will automatically collectstatic.	<code>python3 manage.py collectstatic</code>
<code>python3 manage.py loaddata &lt;fixture_name&gt;</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>
<code>pip3 show &lt;package_name&gt;</code>	Gives information on the package including its location in your file structure	<code>pip show django-allauth</code>
<code>cp -r &lt;from&gt;* &lt;to&gt;</code>	Copies files from one location to another, for example when duplicating package templates so you can personalise them.	<code>cp -r /home/cistudent/.local/lib/python3.9/site-packages/allauth/templates/* ./templates/</code>

## Testing commands

Terminal Command	Description	Example
python3 manage.py test	Run all unit tests in files that start with <b>test_</b> across the entire project	python3 manage.py test
python3 manage.py test <app_name>	Run all unit tests in files that start with <b>test_</b> inside the app specified	python3 manage.py test blog
python3 manage.py test <app_name>.<file_name>	Run all unit tests in a specific test file inside the app specified	python3 manage.py test about.test_forms
python3 manage.py test <app_name>.<file_name>.<class_name>	Run all unit tests in a specific class within the location specified.	python3 manage.py test about.test_forms.TestCollaborateForm
python3 manage.py test <app_name>.<file_name>.<class_name>. <test_name>	Run a single unit test within the location specified.	python3 manage.py test about.test_forms.TestCollaborateForm.t est_form_is_valid

## Other useful commands

Terminal Command	Description	Example
<code>python3 manage.py dumpdata &lt;app_name&gt; &gt; &lt;filename&gt;.json</code>	<b>New:</b> 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.	<code>python3 manage.py dumpdata blog &gt; blog_fixtures.json</code>
<code>python3 manage.py flush</code>	<b>New:</b> The flush command is different from migrate zero. When you run <code>python manage.py flush</code> , it doesn't undo migrations; instead, it removes all data from the database and resets primary key sequences for all models.	<code>python3 manage.py flush</code>