Welcome to this CoGrammar tutorial:

Django Foundations

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.

Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: Questions

Software Engineering Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures

Skills Bootcamp Progression Overview

✓ Criterion 3: Course Progress

- ***Completion:*** All mandatory tasks, including Build Your Brand and resubmissions by study period end
- *Interview Invitation:* Within 4 weeks post-course
- *Guided Learning Hours:* Minimum of 112 hours by support end date (10.5 hours average, each week)

✓ Criterion 4: Demonstrating Employability

- *Final Job or Apprenticeship Outcome:* Document within 12 weeks post-graduation
- *Relevance:* Progression to employment or related opportunity

CoGrammar

Learning Outcomes

- Define what a web framework is.
- Describe Django
- Explain the benefits of Django
- Describe the MVT structure of Django
- Explain what a model is in Django.
- Create models for your Django project
- Execute database migrations

Learning Outcomes

- Explain what a view is in Django.
- Route views to specific urls.
- Create views for your Django project.
- Explain what a template is in Django.
- Create templates for your Django project.


SKILLS FOR LIFE
SKILLS BOOTCAMPS
Department for Education
CoGrammar
Django
May 2024

What is a Web Framework?

- A web framework is a collection of tools, libraries, and best practices designed to simplify the process of building and maintaining web applications.
- It provides a standardised way to develop web applications by offering predefined components such as templates, forms, URL routing, database access, security features and session management.
- A web framework allows developers to focus on writing application-specific code rather than reinventing the wheel for every new project.
- Promotes code reuse.

What is Django?

- High-level, open-source web framework for Python.
- Used for developing secure and scalable web applications.
- Platforms using Django: Instagram, Spotify, Youtube and many more.

Why use Django?

- Designed to help developers take applications from concept to completion as quickly as possible.
- Architecture is highly scalable, allowing it to handle large amounts of traffic and data efficiently.
- Includes numerous security features out of the box, which help developers protect their applications from common vulnerabilities.
- Django is versatile and can be used to build a wide variety of applications.
- Automatically generated an admin interface, which provides a powerful, web-based interface for managing data.

Why use Django?

- ORM allows developers to interact with the database using Python code instead of SQL.
- Django's templating engine allows developers to create dynamic HTML pages.
- Large list of libraries and tools significantly reduce the amount of time needed to build common functionalities.

Model-View-Template (MVT) Architecture

- Variation of Model-View-controller architecture
- Three main components
  - **Model**: Represents the business logic and data structure of the application.
  - **View**: Handles the interaction between the user and the application, managing the presentation logic.
  - **Template**: Deals with the presentation layer, defining the structure and appearance of the HTML content.

Models

- Models serve as the blueprint for your database schema.
- Each model is a Python class/table that subclasses django.db.models.Model, and its attributes represent the fields/columns of the database table.
- Models are crucial for defining the structure of your data, including field types, default values, and validation rules.
- When you create or modify models and then run python migrations, Django translates these Python classes into SQL commands to create and alter database tables with high-level Python code rather than writing raw SQL .

Views

- Views handle the logic of processing user requests and returning responses.
- Views define the behaviour of our URL patterns.
- Views are Python functions or class methods, that takes a web request and returns a web response.
- They interact with models to retrieve or update data.
- Views return appropriate HTTP responses, such as rendering templates or redirecting.

Templates

- Templates define the structure and layout of the HTML pages.
- Templates can incorporate dynamic data, using placeholders and template tags.
- Templates receive data from views through context dictionaries.
- Templates are stored in the templates directory.
- HTML pages are constructed using template tags for data integration.

Django: Naming Conventions

- Project and Application Names:
  - Use lowercase letters and underscores to separate words.
  - Hyphens are not recommended because they can cause issues in import statements.
  - Choose a name that clearly describes the purpose of the project.
  - Avoid using names that are reserved keywords in Python or common names that might conflict with libraries.
  - Application names are typically singular, representing the main concept or entity the app manages.

Django: Naming Conventions

- Model Names:
  - Models are represented as classes and should use the PascalCase or also known as CamelCase.
  - Model names should generally be singular, as each instance of the model represents a single record in the database.
  - Example: UserProfile

Django: Naming Conventions

- View Names:
  - For Function-Based Views, use lowercase with underscores to separate words.
  - Function names should clearly describe the action or response of the view.
  - For Class-Based Views (Where methods are used), use PascalCase for class names, and the class name should describe what the view does or what it represents.
  - Examples: def register_user(request): for user registration, class ProductListView(ListView): for listing products.

Django: Naming Conventions

- Template Names:
  - Use lowercase letters with underscores to separate words.
  - Template names should be descriptive of the view they are associated with.
  - Example: blog_detail.html, user_profile.html.
  - Organise templates into directories that mirror your application's structure.
  - Create reusable template components (partials) and place them in a partials or includes directory, ie. header.html, footer.html

Let’s get coding!

CoGrammar

Let’s take a short break

CoGrammar

Polls

CoGrammar

Polls

- *Refer to the polls section to vote for you option.*

1. What is the correct command to create a new Django Project?
    a. django-admin startproject projectname
    b. django-admin startapp projectname
    c. django-admin createproject projectname
    d. django-admin createapp projectname

Polls

- *Refer to the polls section to vote for you option.*

2. What is the correct command to create a new Django Application?

    a. django-admin startproject appname

    b. django-admin startapp appname

    c. django-admin createproject appname

    d. django-admin createapp appname

Polls

- *Refer to the polls section to vote for you option.*

3. What is the correct command to run a Django Project?
    a. python manage.py startproject
    b. python manage.py startapp
    c. python manage.py runproject
    d. python manage.py runserver

Polls

- *Refer to the polls section to vote for you option.*

4. What is the correct command to run a database migration?
    a. python manage.py migrate
    b. python manage.py runmigrations
    c. python manage.py startmigration
    d. python manage.py makemigrations

Summary

CoGrammar

Summary – Django

1. A Web Framework is a software design to assist in the development of web applications, services and more
2. Django is a free and open source web framework that is making use of the MVT structure.

Summary – Django

3. Models are Python classes and through migration, django translates these Python classes into SQL commands to create and alter database tables accordingly.
4. Views retrieve necessary data from the database through models, process this data as needed, and then render it using templates.
5. Templates are HTML files that define the structure and layout of the web pages served to users.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar