Welcome to the CoGrammar

Django 1

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
8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

- ***Guided Learning Hours (GLH):*** Minimum of 15 hours
- ***Task Completion:*** First 4 tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

- ***Guided Learning Hours (GLH):*** Minimum of 60 hours
- ***Task Completion:*** *First* 13 tasks

Due Date: 28 April 2024

CoGrammar

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

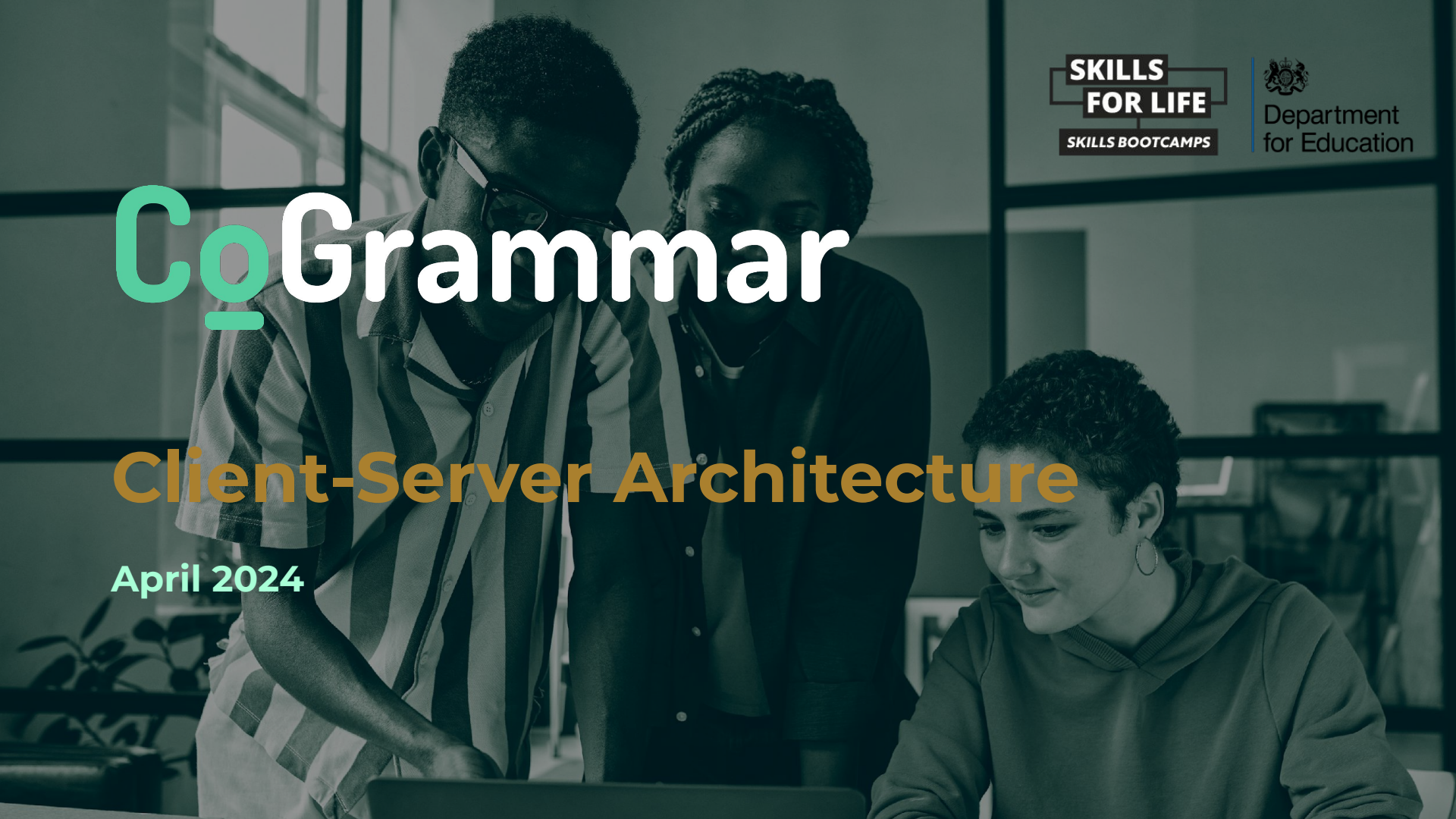
CoGrammar

Learning Outcomes

- Define the client-server architecture
- Explain the request response cycle used in the client server architecture.
- Define HTTP
- Define what a web framework is.
- Describe Django
- Explain the benefits of Django
- Describe the MVT structure of Django

Learning Outcomes

- Explain what a template is in Django.
- Create templates for your Django projects.
- Explain what a view is in Django.
- Route views to specific urls.
- Create views that will render your templates to the user.
- Render templates with context data received from view.



Client-Server Architecture

April 2024

Client-Server Architecture

- Network architecture that breaks down task and workloads between clients and server
- Can reside on same system or linked by a computer network
- Typically consists of multiple workstations, PCs or other devices belonging to users connected to a central server
- Connect through internet connection or other network connection

Client-Server Architecture

- Basic steps
  - Client sends request for data
  - Server accepts request
  - Server processes request
  - Send requested data back to user

Servers and Clients

- Servers
  - Not just a computer clients make requests to
  - Requires appropriate server software running to be a server E.g. Apache, Tomcat, Nginx
- Client
  - Not just any device making requests
  - Requires correct software to make requests
  - Most common client - Web browser
  - Your social media application is also a client

HTTP

- HyperText Transfer Protocol
- Underlying protocol of WWW
- Defines how messages are formed and transmitted between clients and server
- Defines actions clients and server must take in response to various commands

HTTP

- Basic example of HTTP implementation
  - Urls gets entered in a browser
  - Browser send HTTP command to server
  - Command directs server to search for and transmit requested page
  - Response can be HTML in this instance

HTTP

- HTTP is a stateless protocol
- Each request is independent from the previous request

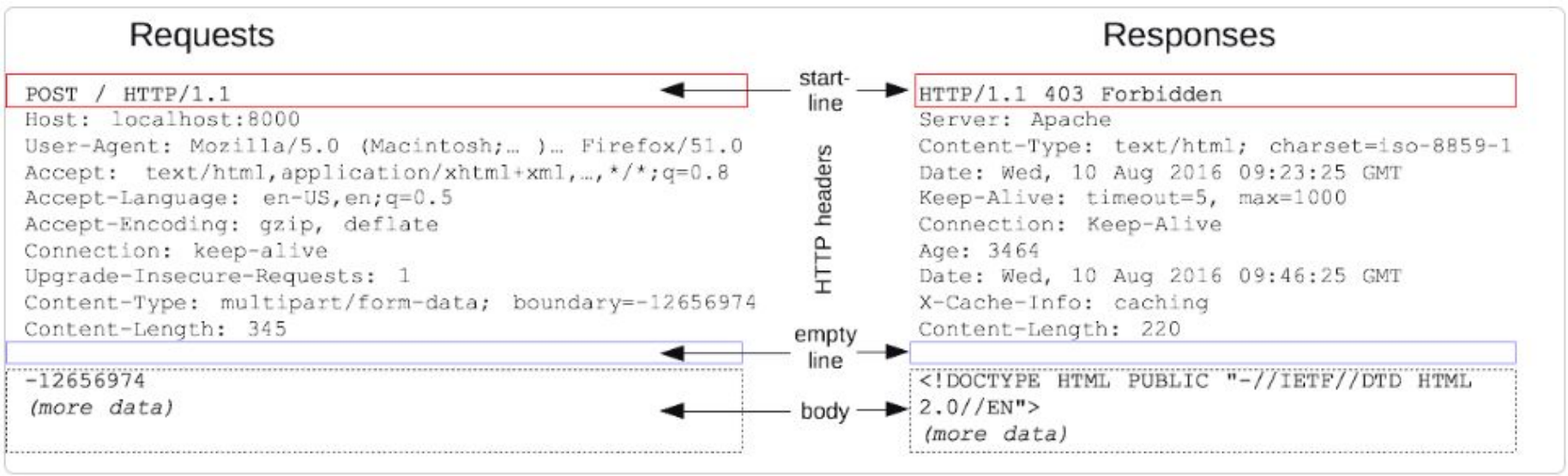
HTTP

- E.g. a request is made for the first ten records in a database and then another request is made for the next ten records
- Stateful protocol
  - Give me the first 10 records
  - Give me the next 10 records
- Stateless protocol
  - Give me records 1-10
  - Give me records 11-20

HTTP Messages

- Used for requests and responses
- Composed of textual information encoded in ASCII and spans multiple lines
- Consists of
  - Start line
  - Headers
    - General
    - Request
    - Representational
  - Body

HTTP Messages



CoGrammar

Status Codes

- Short notes tacked onto a webpage
- Not part of the site's content but messages telling us how things went
- Returned every time your browser interacts with a server
- Helps diagnose and fix website configuration

Status Codes

5 Classes of status codes

- 100s
  - Informational code
  - Indicates request initiated in continuing
- 200s
  - Success code
  - Indicates request was received, understood and processed

Status Codes

- 300s
  - Redirection codes
  - When a new resource in substituted for the requested resource
- 400s
  - Client Error
  - Problem with request
- 500s
  - Server error
  - Request was accepted but a server error has occurred
  -
  -

Django

April 2024

What is a Web Framework?

- Software framework designed to assist in the development of web applications.
- Provides libraries for database access, templating frameworks, and session management.
- Promotes code reuse.

What is Django?

- Open-source web framework
- Used for developing secure and scalable websites and web applications
- Platforms using Django: Instagram, Spotify, Youtube and many more

Why Django?

- Has a large list of libraries and tools
- Allows for the creation of robust data driven applications.
- Code is fast to implement and is very clean and pragmatic

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.

Templates

- Templates define the structure of the HTML pages.
- They incorporate dynamic data using template tags.
- They 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 Template Language

- We build our our templates using the Django Template Language.
- It allows us to create base templates and extend them.
- We can use variables inside our templates.
- It also contains ‘tags’ we can use to create loop structures and boolean checks.

Views

- Views are Python functions we create in the views.py file.
- Views define the behaviour of our URL patterns.
- Views handle user requests and define the logic for processing them.
- They interact with models to retrieve or update data.
- Views return appropriate HTTP responses, such as rendering templates or redirecting.

Summary

- **Client-Server Architecture:** Use a request response cycle, where clients make request to a server that will respond.
- **HTTP:** Underlying protocol of the WWW. Defines how messages are formed and transmitted between clients and server.
- **Web Framework:** Software framework design to assist in the development of web applications, services and more.
- **Django:** Free and open source web framework. Makes use of the MVT structure.
- **Templates:** Define the structure of our HTML pages.
- **Views:** Allow is to perform backend processes and actions, and render our templates to the user.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar