



# Welcome to this **CoGrammar** tutorial: HTML and CSS

The session will start shortly...

Questions? Drop them in the chat.



# 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** throughout this session, should you wish to ask any follow-up questions.
- 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](http://www.hyperiondev.com/support)
- Report a **safeguarding** incident: [www.hyperiondev.com/safeguardreporting](http://www.hyperiondev.com/safeguardreporting)
- We would love your **feedback** on lectures: [Feedback on Lectures](#)

# Enhancing Accessibility: Activate Browser Captions

---

## Why Enable Browser Captions?

- Captions provide **real-time text for spoken content**, ensuring inclusivity.
- Ideal for individuals in noisy or quiet environments or for those with **hearing impairments**.

## How to Activate Captions:

### 1. YouTube or Video Players:

- Look for the CC (Closed Captions) icon and click to enable.

### 2. Browser Settings:

- Google Chrome: Go to *Settings > Accessibility > Live Captions* and toggle ON.
- Edge: Enable captions in *Settings > Accessibility*.

# Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles  
Designated Safeguarding  
Lead



Simone Botes



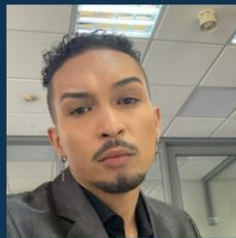
Nurhaan Snyman



Rafiq Manan



Ronald Munodawafa



Tevin Pitts

Scan to report a  
safeguarding concern



or email the Designated  
Safeguarding Lead:  
Ian Wyles

[safeguarding@hyperiondev.com](mailto:safeguarding@hyperiondev.com)

# *Stay Safe Series.*

## Mastering Online Safety One Week or Step at a Time

---

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalisation, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* is designed to guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

## Trustworthy Websites: How to Spot Secure Sites

---

- Look for the padlock.
- Check if there is a valid SSL/TLS certificate.
- **Look for a site seal.**
- **Check if the URL is legitimate.**
- Pop-up and Redirection ads are a red flag.





**SKILLS  
FOR LIFE**

**SKILLS BOOTCAMPS**



Department  
for Education

# CoGrammar

## HTML and CSS



# Polls



# Poll

1. Which of the following is the primary function of HTML in web development?
  - a. Adding interactivity to web pages
  - b. Defining the overall visual style of a webpage
  - c. Setting up server-side logic and databases
  - d. Structuring the content and layout of a webpage
  - e. Communicating with web browsers using high-level commands

# Poll

2. What happens to a webpage if it is not responsive?
  - a. It will not display any content at all
  - b. Images and text might appear blurry
  - c. The layout might not adapt properly to different screen sizes
  - d. It will only work on specific web browsers
  - e. The website will load very slowly

# Learning Outcomes

- Construct a Basic HTML Structure
- Implement Basic Layouts with CSS
- Explain the Box Model Concept
- Describe the difference between static and responsive design
- Link an External CSS Stylesheet

# Introduction



# Similarities and Differences Between Python and HTML/CSS

- Both Python and HTML/CSS are fundamental building blocks for web development.
- **Python:** A high-level, general-purpose programming language used for various tasks like data analysis, automation, and back-end web development logic.
- **HTML/CSS:** Focus is on web development presentation. HTML defines the structure and content of a webpage, while CSS styles its visual appearance.

# Similarities and Differences Between Python and HTML/CSS

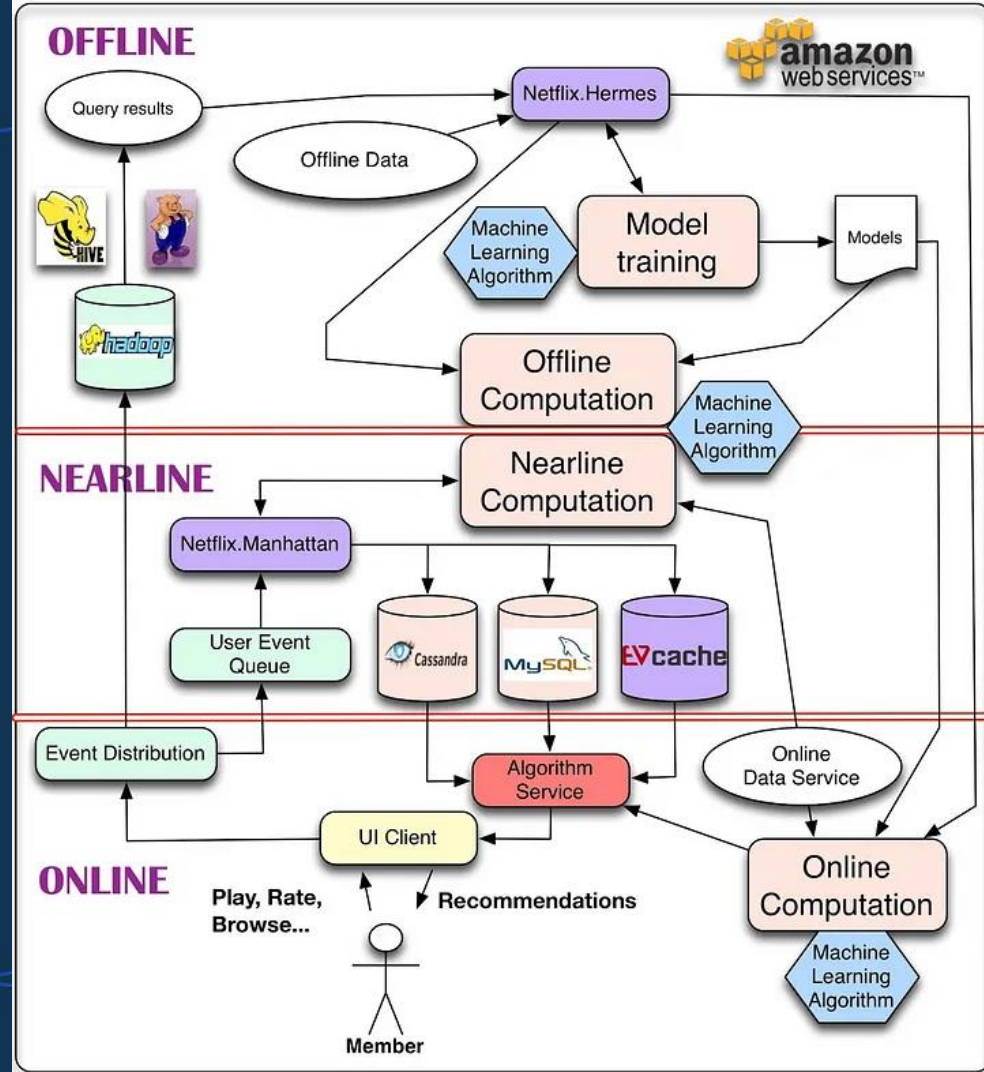
- Both require writing code, but Python has a more complex syntax compared to HTML/CSS.
- **They work together:** Python code can generate dynamic HTML content or interact with data displayed on a webpage styled with CSS.



# Front-End vs. Back-End Development

- **Front-End Development:** Deals with the user interface (UI) and user experience (UX) of a website, including HTML, CSS, and JavaScript for interactivity.  
(This session focuses on front-end development)
- **Back-End Development:** Handles server-side logic, databases, and application functionality, often using languages like Python.

# Software Design Example



## Characters



## Popular



## Everyone's Watching



# HTML Basics



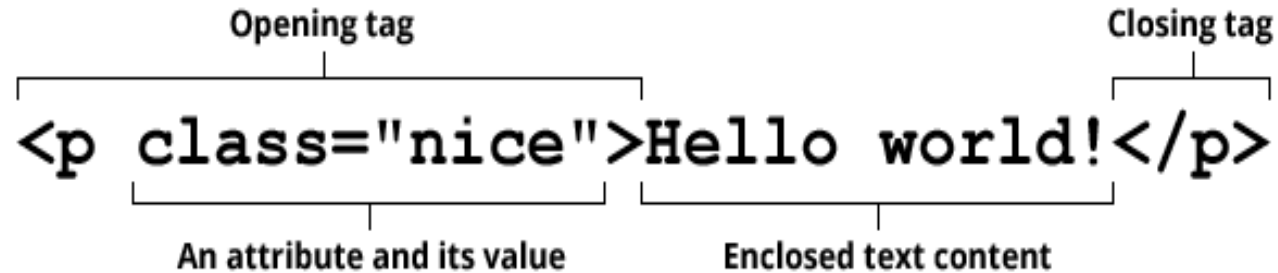


# Dive into HTML : The Language of the Web

- What is the Web?
  - The **Web** or **World Wide Web** is a system of interconnected documents and resources (software) that are accessed over the internet using web browsers.
- What is HTML?
  - **HTML** (HyperText Markup Language) is the code that is used to structure a web page and its content.
- Why is HTML important?
  - HTML is the fundamental language for structuring and defining the content of webpages.

# Decoding the Language: HTML Tags & Attributes

*Anatomy of an HTML element*



# Organising Your Content: Basic HTML Structure

```
<!DOCTYPE html>
<html lang="en">
  <head>

  </head>
  <body>

  </body>
</html>
```



# Building Blocks of Your Webpage

- **Headings** (<h1> to <h6>): Define different heading levels for titles and subtitles.
- **Paragraphs** (<p>): Used for the main body text content.
- **Lists** (<ul> for unordered, <ol> for ordered): Create bulleted or numbered lists.
- **Images** (<img>): Embed images with proper alt text for accessibility.
- **Links** (<a>): Create hyperlinks for navigation or external references.

# The Versatile *div*: A Powerful Container

- The `<div>` tag defines a division or a section in an HTML document.
- The `<div>` tag is used as a container for grouping related HTML elements
- Organisation Power improves webpage structure and simplifies styling later.
- CSS Makes it Shine: Paired with CSS, div lets you control layout and appearance.

# CSS Fundamentals



# Unveiling CSS: The Stylist for Your Webpages

- CSS (Cascading Style Sheets) is a **styling language** specifically designed for webpages.
- It **controls the presentation of HTML elements**, including font styles, colours, backgrounds, and layout.
- This **separation of concerns** keeps HTML focused on structure and CSS focused on styling, promoting cleaner and more maintainable code.

# Applying Styles: Inline, Internal, and External Stylesheets

- There are **three ways** to apply CSS styles to HTML elements:
  - **Inline Styles:** Styles are added directly to the HTML element using the style attribute (less preferred, can make code messy).
    - `<h1 style="color: blue; text-align: center;">Welcome to Inline Styling</h1>`
  - **Internal Styles:** Styles are defined within the `<head>` section of the HTML document using a `<style>` tag (more organised approach).
    - `<style> h1 { color: blue; text-align: center; } </style>`
  - **External Stylesheets:** Styles are placed in a separate CSS file (.css) that is linked to the HTML document (preferred method for larger projects, promotes code reusability).
    - `<link rel="stylesheet" href="styles.css">`

# Targeting Elements with Selectors

- Selectors are used to target specific HTML elements within your webpage for styling.
- Common types of selectors include:
  - **Tag/Element Selectors:** target elements by their HTML tag (e.g., h1, p, img).
  - **Class Selectors:** target elements with a specific class attribute (e.g., .important, .highlight).
  - **ID Selectors:** target a unique element with a specific ID attribute (e.g., #banner, #footer).

# The Box Model:

## Understanding Element Size and Positioning

- The box model is a concept in CSS that defines how elements are sized and positioned.
- It consists of four layers (CPBM):
  - **Content**: The actual content of the element (text, image, etc.).
  - **Padding**: The space between the content and the border (optional).
  - **Border**: The decorative line around the element (optional, can have width and style).
  - **Margin**: The space around the border of the element (can be used for spacing).



**Let's take a short  
break**

**CoGrammar**



Let's get coding!



# Polls



# Poll

1. What is the correct way to link an external CSS stylesheet to an HTML document?
  - a. By embedding CSS code directly within HTML tags
  - b. Including a `<css>` element within the `<head>` section
  - c. Using a `<link>` element with the `rel="stylesheet"` attribute in the `<head>` section
  - d. Saving both HTML and CSS files with the same name
  - e. Referencing the CSS file using a script tag

# Lesson Conclusion and Recap





# Lesson Conclusion and Recap

- **HTML:** The Web's Content Backbone
- **CSS:** Styling the Web Experience
- **Separation of Concerns:** We emphasised the importance of separating HTML's structure from CSS's styling for cleaner and more maintainable code.
- **The Box Model in Play:** We delved into the box model, a key concept in CSS.
- **Responsive Design:** We briefly introduced responsive design.

# Learner Challenge

Choose one of the following:

- **E-commerce Product Catalogue:** Design a catalogue page for an online store, featuring product images, descriptions, prices, and add-to-cart buttons.
- **Blog Archive:** Design an archive page for a blog, listing posts by category, date, or author, with links to individual post pages.
- **Survey Form:** Develop a survey form with various types of input fields (text, radio buttons, checkboxes) to gather feedback or data on a specific topic.
- **Music Player Page:** Create a music player interface with play, pause, skip buttons, and a track list displaying song titles and durations.
- **Your dream project**



# Thank you for attending



Department  
for Education

CoGrammar



# Further Resources

- <https://learnlearn.uk/edexcel-igcse-computer-science/components-of-the-world-wide-web/>
- [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
- <https://developer.mozilla.org/en-US/docs/Glossary/Element>
- [https://html.com/#What\\_are\\_Tags\\_and\\_Attributes](https://html.com/#What_are_Tags_and_Attributes)
- <https://www.codewizardshq.com/html-for-kids/>
- <https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8>