



## Welcome to this session: Responsive Design and building a calculator

**The session will start shortly...**

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



# 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



Rafiq Manan



Charlotte Witcher



Nurhaan Snyman



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)

# Skills Bootcamp Cloud Web Development

---

- 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**

# Skills Bootcamp Cloud Web Development

---

- For all **non-academic questions**, please submit a query:  
**[www.hyperiondev.com/support](https://www.hyperiondev.com/support)**
- **Report a safeguarding incident:** **[www.hyperiondev.com/safeguardreporting](https://www.hyperiondev.com/safeguardreporting)**
- We would love your feedback on lectures: Feedback on Lectures
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

# ***Stay Safe Series:***

Mastering Online Safety One week 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 radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the ***Stay Safe Series*** will 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.

# Think Before You Click:

## Stay Safe during Online Interactions

---

"Think Before You Click" is crucial for maintaining online safety and protecting personal information. It encourages individuals to be mindful of their online actions, helping to prevent cyberbullying, scams, and the spread of misinformation.




# What is the purpose of JavaScript in front-end development?

**(Select all that apply)**

- A. To add interactivity to web pages.
- B. To style HTML elements.
- C. To create server-side applications.
- D. To manipulate the Document Object Model (DOM).





# What is the purpose of an attribute in an HTML element?

- A. To define additional information about the element.
- B. To style the element.
- C. To create animations on the page.
- D. To specify the behavior of the element.





## Learning Outcomes

---

- ❖ **Create responsive layouts** using flexbox and grid to dynamically adjust element positioning across various screen sizes.
- ❖ **Implement breakpoints** in CSS to change styling based on screen dimensions, ensuring the design remains functional on different devices.
- ❖ **Use CSS properties** like display, justify-content, and align-items to control where elements will be placed on the screen.
- ❖ **Test and assess the responsiveness of web designs**, ensuring they maintain usability across multiple devices and screen sizes.

# Lecture Overview

---

- Recap on Introduction to Responsive Design
- Design Layouts
- Responsive Images



# Responsive Design



A method to create web pages that adapt to various screen sizes and orientations while maintaining functionality and readability.

- ❖ Websites may become **cluttered, distorted or unusable** when viewing them on **smaller screens** than they were designed for.
- ❖ This causes many **accessibility challenges**, limiting users who don't have access to larger screens and who may have impairments.
- ❖ To solve this, we use responsive design to adjust pages to screen size, by detecting **size and orientation**.
- ❖ We use HTML and CSS along with **flexible layouts, media queries and responsive units**.

# Responsive Design

## ❖ Key components:

- **Flexible Design Layouts:** Resize and reorganise themselves to fit screen dimensions.
- **Responsive Images:** Scale or switch based on the screens size or device type.
- **Media Queries:** CSS rules that apply styles depending on the viewport's characteristics such as screen width or resolution.

# Responsive Units

- ❖ **Viewport:** The visible area on a device's screen. It changes across devices.
- ❖ **Types of Units:**
  - **Absolute Units:** Fixed units like pixels (px) and centimeters (cm)
  - **Relative Units:**
    - **em:** relative to the parent element's font size.
    - **rem:** relative to the root element's font size.
    - **vh and vw:** percentages of the viewport's height or width.
    - **%:** relative to the parent element.

```
p {  
  font-size: 5vh;  
}
```

# Media Queries

Allow developers to apply different styles based on the screen's characteristics (width, resolution).

- ❖ There are many types of media that we can create designs for: **all, screen, print, speech.**
- ❖ **Breakpoints:** defined points at which layouts adjusts.

```
img {  
  width: 50vw;  
}  
  
@media screen and (max-width: 700px) {  
  img {  
    width: 100px;  
  }  
}
```

# Flexible Design Layouts

## ❖ Grid Layout:

- Uses rows and columns for layout, creating a structured, multi-dimensional grid.
- Commonly consists of 12 columns, allowing for versatile placement of elements.

## ❖ Flexbox Layout:

- Simplifies the alignment of elements within a container, using flexible boxes and works in a single direction (row or column).
  - **flex-direction:** specifies the direction
  - **flex-wrap:** allows items to wrap if they overflow the container.
  - **flex:** controls the amount of available space an item occupies.



Let's take a  
break



# Questions and Answers



# Thank you for attending



**CoGrammar**



Department  
for Education