



Welcome to this **CoGrammar** lecture: Cascading Style Sheets (CSS)

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** 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
- Report a **safeguarding** incident: 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

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 Bootcamp Progression Overview

✓ Criterion 1 - Initial Requirements

Specific achievements **within the first two weeks** of the program.

To meet this criterion, students need to, by no later than **01 December 2024**:

- **Guided Learning Hours (GLH):** Attend a **minimum of 7-8 GLH per week** (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- **Task Completion:** Successfully complete the **first 4 of the assigned tasks**.

✓ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks **within the first half** of the program.

To meet this criterion, students should, by no later than **12 January 2025**:

- **Guided Learning Hours (GLH):** Complete at least **60 GLH**.
- **Task Completion :** Successfully complete the **first 13 of the assigned tasks**.

Skills Bootcamp Progression Overview

✓ Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- Guided Learning Hours (GLH): Complete the **total minimum required GLH**, by the **support end date**.
- Task Completion : **Complete all mandatory tasks**, including any necessary resubmissions, by the end of the bootcamp, **09 March 2025**.

✓ Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- Record an Interview Invite: Students are required to record proof of invitation to an interview by **30 March 2025**.
 - South Holland Students are required to proof and interview by **17 March 2025**.
- Record a Final Job Outcome : Within 12 weeks post-graduation, students are required to record a job outcome.

**SKILLS
FOR LIFE**

SKILLS BOOTCAMPS



Department
for Education

CoGrammar

Cascading Style Sheets (CSS)

Polls



Poll

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

1. What is the primary purpose of CSS?

- a. To define the structure and content of a webpage.
- b. To add interactivity and dynamic behaviour to a webpage.
- c. To enhance the accessibility features of a webpage.
- d. To style the presentation and layout of a webpage.

Poll

- *Refer to the polls section to vote for your option.*
2. Which CSS property is used to change the colour of text?
- a. text-color
 - b. color
 - c. font-color
 - d. text-style

Learning Outcomes

- Define CSS.
- Explain what **selectors** are.
- Identify **different element selectors** such as class, ID and element type.
- Use **common CSS properties** to style elements on your web pages.

Learning Outcomes

- Define the **box model**
- **Implement** the box model for a more structured layout and spacing.
- Explain what a **CSS framework** is.
- **Use** a CSS framework like Bootstrap to create web pages in a streamlined manner.

What is CSS?

- Cascading Style Sheets (CSS) is a language which is used to change the presentation and look of a particular document which has been written in a markup language, such as HTML.
- CSS is usually applied to web pages, but can also be used in other areas, such as XML documents.
- HTML elements are described using attributes and properties. You can style a web page by changing the properties of the elements that make up that webpage.
- For example: font-family (Arial, Times New Roman, etc.), font-style (normal, italics, etc.) and font-size.

Inline Styling



Style Attribute

- Like all other attributes, the **style attribute** goes **inside** the element's **beginning tag**, right after the tag name.
- After **specifying** that you are changing the **style attribute**, you type **=**, and then, within **double quotes**, list the **properties** you want to **change** and **after a colon** specify the **value** for that property.

```
<p style="font-size: 46px; font-style: italic;">
```

↑
Attribute

↖ ↗
Property

↖ ↗
Value

Using Inline Style

- An example of using the **style attribute** to change the font of an element is shown below:

```
<p style="font-family:Georgia;color:■blue;">  
  Look at this stylish paragraph!  
</p>
```

Look at this stylish paragraph!

Inline Style Limitations

- When you **style** an element **individually** by changing that element's properties, it is known as **inline styling**. Inline styling allows you to specify the **style** of an **individual element** in the **line** where that **element** is **declared**.
- What if you wanted to apply **similar styles** to all **elements** of a certain **type**? For example, what if you wanted to change the font of all paragraphs on your web page?
- You can do this by creating a **CSS rule**.

Internal CSS



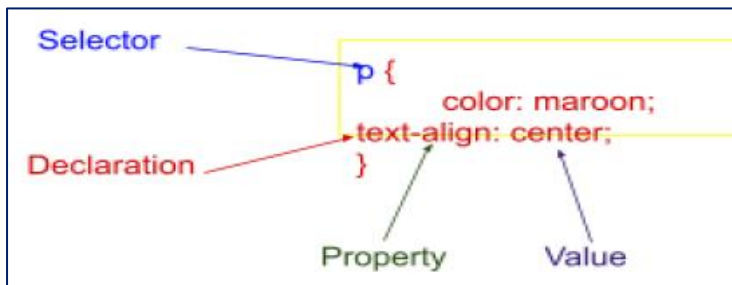
Internal CSS

- You can define a CSS rule in the head part of your HTML template -> This is called internal CSS.
- The CSS rule below will format all paragraphs to the colour red with font-family Arial. If the browser can't find Arial, then it will look for Helvetica. Paragraph backgrounds will be blue.

```
<head>
  <style>
    p{
      color: ■ red;
      font-family: Arial, Helvetica;
      background-color: ■ blue;
    }
  </style>
</head>
```




CSS Syntax

- The **selector** indicates which HTML element you want to style.
- The **declaration** block is surrounded by curly braces and contains one or more declarations with each declaration ending with a semicolon.
- Each declaration includes a **property** and a **value**, separated by a colon.



Element Selectors

- Let's take another look at our CSS rule.

```
p{  
  color:  red;  
  font-family: Arial, Helvetica;  
  background-color:  blue;  
}
```

- Our selector here is an **element selector**.
- All elements of type 'paragraph' will have the properties as defined by the selector above.

What if we don't want all the paragraphs to have the same properties?

Class Selectors

- A **class selector** is used when the selector describes the rule only for elements that have a class attribute with the same name. **Class styling can be applied multiple times.**
- In <head> you will define the class rule in a <style> element preceded by a full stop(.).

```
.changestyle{  
    font-family: 'Times New Roman';  
}
```

- In <body> you will use the **class attribute** to apply the rule.

```
<p class="changestyle">  
    Changed my style! What do you think?  
</p>
```

Changed my style! What do you think?

Id Selectors

- An **id selector** describes the style of an element with a specific id attribute defined. **Id styling can be applied only one time.**
- In <head> you will define the id rule in a <style> element preceded by a hash(#).

```
#head{  
  font-size: 20px;  
  color: ■ red;  
}
```

- In <body> you will use the **id attribute** to apply the rule.

```
<h2 id="head">Welcome to my Page!</h2>
```

Welcome to my Page!

External CSS



External CSS

- If your website consists of **many** HTML files, you likely want to be able to apply the **same style** rules to all the web pages. To accomplish this, use **external CSS** instead of internal CSS.
- To do this, create a **separate file** with the extension **.css**. Within this file write all the **style rules** that you would like to specify. You can then **link** this external CSS file to all the HTML files in which you would like the style rules to apply.
- To link an external CSS file to a specific HTML file, include the below in the <head> section of the HTML file.

```
<link rel="stylesheet" href="style.css">
```

External CSS ...

- Another important reason to **separate CSS from HTML files** is to improve the maintainability of your website.
- If you wanted **to update the look and feel** of a website, this could easily be done **by simply replacing the external CSS file** if only external CSS is used for the website.
- Using external CSS also makes it **easier to debug errors** since all the CSS is in one place.
- You may find, though, that it is necessary to **use a combination of external, internal and inline style**. In this case, it is important to understand the concept of cascade.

Cascading



Cascading

- As we know, CSS stands for **cascading style sheets**. You may have wondered why it is called cascading style sheets. **Cascading** has to do with how the rules are prioritised.
- If your website contains external, internal and inline CSS, the following rules apply:
 - Inline CSS **overrides** internal CSS rules and external CSS files.
 - Internal CSS **overrides** external CSS rules.
 - If there are conflicting rules regarding properties, the general rule is that the **more recently** defined rule **takes precedence**.

Cascading

- When **several** CSS rules **match** the **same** element, they are **all applied** to that element. Only afterwards are any **conflicting properties evaluated** to see which **individual styles** will win over others.
- Another important rule to remember is that the **more specific** a rule is, the **higher** its **precedence**.
- In a stylesheet that uses element selectors, class selectors and id selectors, **element** selectors are the **least specific** whereas **id** selectors are the **most specific**. Therefore, **id** selectors will be **applied over class** and **element** selectors.

Box Model



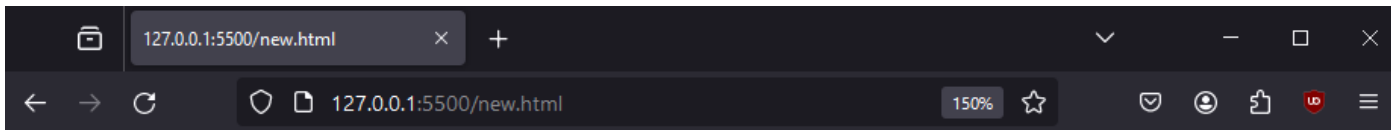
Box Model

- Everything in CSS has a box around it.
- We can use these boxes to build complex layouts on our pages.
- We can set the display type of a box to block and inline.
- This will change the behaviour of our box when certain changes are applied.
- We can then edit the main parts of a box, being content, padding, border and margins.

Box Model: Block

- Block Box
 - A block box type will take up the **full width** of the page.
 - Has a **line break before and after** the box.
 - Below we have the normal display of a paragraph element.

```
<p style="background-color: red;">Hello world!</p>
```



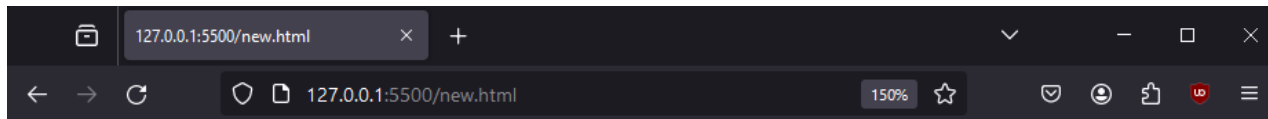
Hello world!

Box Model: Block ...

- When adding another element **inline** and **setting it to use a block display** we can see how the new lines apply and how our second element will also take up as much width as possible.

```
.box{  
  display: block;  
}
```

```
<p style="background-color: red;">Hello world! <a class="box">Click here!</a></p>
```

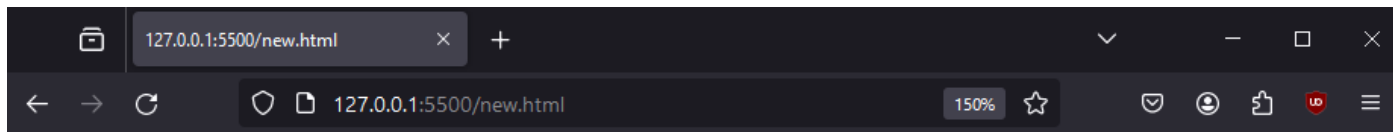


Hello world!
Click here!

Box Model: Inline

- Inline Box
 - An inline box type will take up the width of it's content.
 - Has no line break before or after the box.
 - Below we have the normal display of a paragraph element

```
<p style="background-color: red;">Hello world!</p>
```



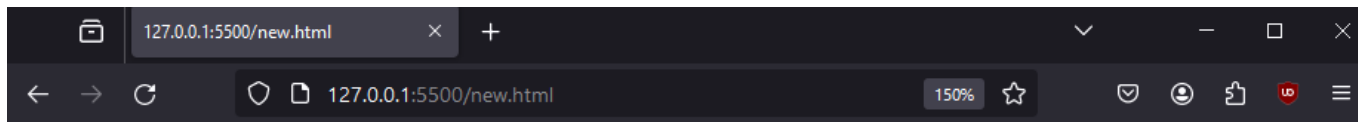
Hello world!

Box Model: Inline ...

- When adding another element **inline** and **setting it to use an inline display** we can see how the new element gets placed next to our previous element.

```
.box{  
  display: inline;  
}
```

```
<p style="background-color: red;">Hello world! <a class="box">Click here!</a></p>
```



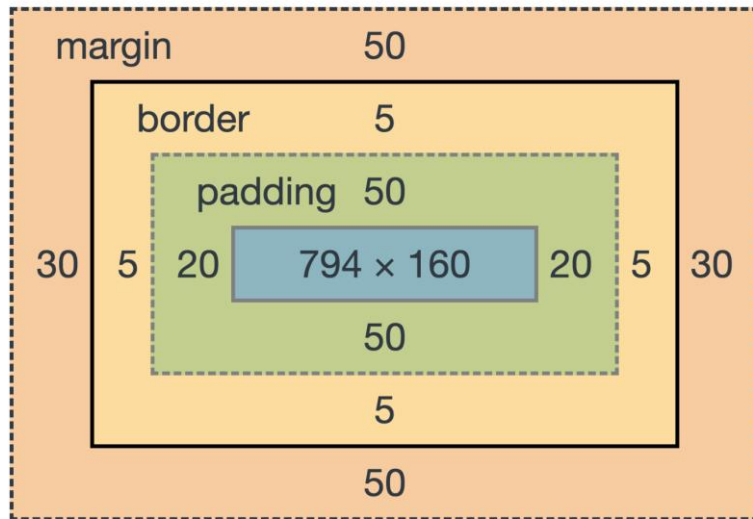
Hello world! Click here!

Box Model: Parts

Now that we have seen some ways to structure our boxes, let's take a look at how we can edit the box itself.

- There are 4 main parts to our box that we can edit.
 - **Content**: The actual content in the block.
 - **Padding**: Space between content and border
 - **Border**: Space between padding and margin
 - **Margin**: Space between border and other elements

Box Model: Layout



Bootstrap



Bootstrap

- Bootstrap is an **open-source CSS framework**.
- It contains **predefined templates** we can use for styling our web pages.
- We **link Bootstrap** with our html pages similarly to how we link our own style sheets.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
```

- Now that we have access to the style rules we can apply them to our pages.

Bootstrap: Buttons

- Below we can create a button and apply the class “btn” and “btn-success” from bootstrap to apply the style.
- There are many types of button.

```
<button type="button" class="btn btn-success">Button</button>
```

Button

```
<button type="button" class="btn btn-warning">Button</button>
```

Button

Bootstrap: Images

- We can also **add some style to images**. Let's say we would like our images to be displayed like thumbnails with rounded corners.

```

```



Bootstrap: Tables

- Here we are adding some style to a table.

```
<table class="table table-striped-columns table-hover">
  <th>Name</th>
  <th>Surname</th>
  <th>Age</th>
  <tr>
    <td>Peter</td>
    <td>Parker</td>
    <td>21</td>
  </tr>
  <tr>
    <td>Tony</td>
    <td>Stark</td>
    <td>38</td>
  </tr>
</table>
```

Bootstrap: Table Style

Name	Surname	Age
Peter	Parker	21
Tony	Stark	38

**Let's take a short
break**



Let's get coding!



Polls



Poll

- *Refer to the polls section to vote for you option.*
1. Which CSS selector targets elements with a specific class?
 - a. #
 - b. .
 - c. :
 - d. @

Poll

- *Refer to the polls section to vote for you option.*
2. Which CSS property is used to add space around an element's content, inside the border?
 - a. margin
 - b. padding
 - c. border-spacing
 - d. spacing

Conclusion and Recap

- **CSS:** Allows us to apply style to our web pages.
- **Inline, Internal, External CSS:** We have different levels where we can write CSS rules and these levels affect how the rules are applied.
- **Selectors:** Selectors help us choose the element for a specific style. We can have style apply to all element of a type, a class or a single element with a specific ID.
- **Cascading:** Inline CSS will override internal CSS will override external CSS. The more specific a selectors override less specific selectors.
- **Box Model:** Think of all your elements as boxes to structure your web pages.
- **Bootstrap:** A web framework with predefined CSS rules that you can apply to your projects to streamline design.

Questions and Answers



Learner Challenge

- **Interests or Hobbies:**

- Create a webpage about your favourite sport, hobbies or interest.
- You can add all the content to your web page using HTML and style your page using CSS.
- Try to use a mixture of inline, internal and external CSS to create an eye-catching web page.

Thank you for attending



Department
for Education

CoGrammar

