Welcome to this CoGrammar session:

Hypertext Markup Language (HTML) & Cascading Style Sheets (CSS)

The session will start shortly...

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

CoGrammar

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.

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

✅ 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

- Understand the basic structure and purpose of HTML.
- Describe the importance of HTML in web development.
- Create simple HTML documents using appropriate tags.
- Differentiate between HTML tags and attributes.
- Identify the role of HTML in separating content from presentation.
- Define CSS.
- Identify different element selectors such as element, class and Id type.

Learning Outcomes

- Use common CSS properties to style elements on your web pages.
- Use a CSS framework like Bootstrap to create web pages in a streamlined manner.
- Define the box model.
- Implement the box model for a more structured layout and spacing.

CoGrammar

CoGrammar

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

HyperText Markup Language (HTML)

June 2024

Introduction

- HTML stands for **H**yper**T**ext **M**arkup **L**anguage.
- It is a language that we use to create files that tell the browser how to lay out or structure text, images, tables, “content” etc. on a web page.
- Its primary role in web development is to define the structure and content of a webpage by using a system of tags and attributes.
- HTML does not include the style of the content (e.g. font, colour, size, etc.), which is done using CSS (Cascading Style Sheets).

CoGrammar

HTML Tags

- Tags are the fundamental building blocks that indicate to the browser what sort of structure the content is contained in.
- A tag is a specific syntax enclosed within angle brackets ("<" and ">") that denotes the beginning and end of an HTML element.
- For example: A paragraph element will have the opening tag <p> and the closing tag </p>

Basic Tags: <html>

- The **<html>** tag serves as the root element of an HTML document, enclosing all other elements.
- It indicates the beginning and end of the HTML document and defines the document type.

CoGrammar

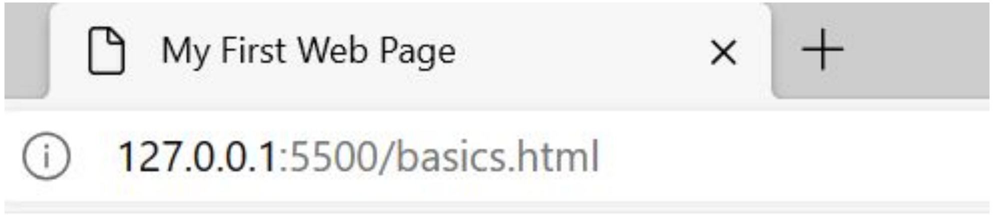
Basic Tags: <head>

- The **<head>** tag contains metadata about the HTML document.
- The metadata includes information such as the document's title, character set, links to external stylesheets or scripts, and other elements that are not displayed directly on the webpage.

CoGrammar

Basic Tags: <title>

- The **<title>** tag specifies the title of the HTML document, which is displayed in the browser's title bar or tab.
- It provides a concise description of the webpage's content and is important for search engine optimisation (SEO) and user experience.



CoGrammar

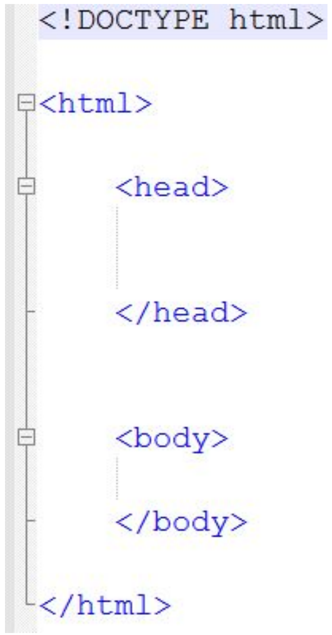
Basic Tags: <body>

- The **<body>** tag encloses the main content of the HTML document that is visible to users when they view the webpage in a browser.
- It includes elements such as text, images, links, headings, paragraphs, lists, and other multimedia content.

CoGrammar

HTML Document Layout

- A DOCTYPE which indicates which version of HTML to load.
- A head which contains metadata about the page.
- A body which contains the actual content.
- Both the head and body are nested inside the html element.



```
<!DOCTYPE html>

<html>

    <head>

    </head>

    <body>

    </body>

</html>
```

CoGrammar

HTML Elements: Headings

- **Headings (<h1> to <h6>)**
- Headings are used to define the headings or titles of sections within a webpage.
- <h1> (most important) to <h6> (least important), moving down a substructure.

CoGrammar

HTML Elements: Paragraphs

- **Paragraphs (<p>)**
- Paragraphs are used to define paragraphs of text.
- It separates blocks of text, making content easier to read and understand.

CoGrammar

HTML Elements: Line Breaks

- **Line Breaks (<br>)**
- This element is used to insert a line break within a paragraph or other block-level element.
- It forces text or content to start on a new line without creating a new paragraph.

HTML Elements: Links

- **Links (<a>)**
- The anchor element, is used to create hyperlinks to locations within the same webpage, other webpages or other files.
- It allows users to navigate between different pages or resources on the internet.

HTML Elements: Images

- **Images (<img>)**
- The image element is used to insert images into a webpage.
- It specifies the location (URL) of the image file and includes optional attributes such as width, height, alt text, and more.

CoGrammar

HTML Elements: Lists

- **Lists (<ul>, <ol>, <li>)**
- Lists are used to organise and present information in a structured format.
- <ul> (unordered list) represents a bulleted list.
- <ol> (ordered list) represents a numbered list.
- <li> (list item) is used to define individual items within a list, whether it's a bullet point or a numbered item.

CoGrammar

HTML Tags vs Attributes

- **Tags** are the fundamental building blocks of HTML and are used to define the structure and content of a webpage.
- **Attributes** provide additional information about the objects created by HTML elements and modify their behaviour or appearance such as size, colour, alignment, links, and more.
- Attributes are specified within the opening tag of an HTML element and are written as name-value pairs.
- While HTML itself does not dictate specific visual styles, attributes can be used to add certain functionalities or visual enhancements to HTML elements.

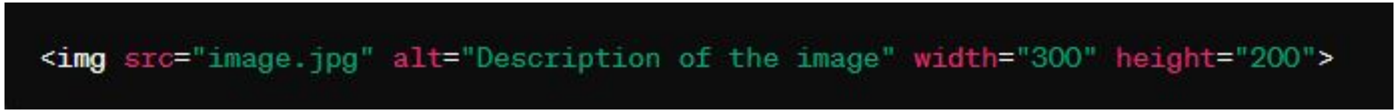
CoGrammar

Attribute Examples: Anchor Tag

```html
<a href="https://www.example.com" target="_blank">Visit Example</a>
```

- **href:** Specifies the URL of the link destination.
- **target:** Specifies where to open the linked document (e.g., in a new window or tab).

Attribute Examples: Image Tag

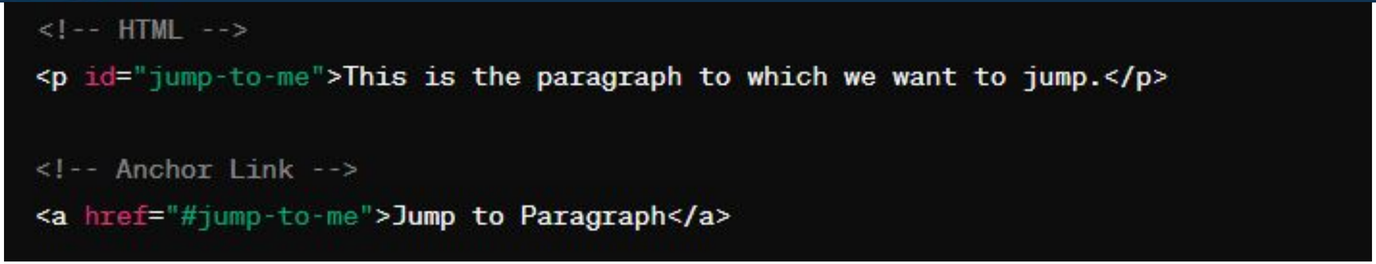


```html
<img src="image.jpg" alt="Description of the image" width="300" height="200">
```

- **src:** Specifies the URL of the image file.
- **alt:** Provides alternative text for the image (useful for accessibility).
- **width:** Specifies the width of the image in pixels or as a percentage.
- **height:** Specifies the height of the image in pixels or as a percentage.

CoGrammar

Attribute Examples: Paragraph Tag



```html
<!-- HTML -->
<p id="jump-to-me">This is the paragraph to which we want to jump.</p>

<!-- Anchor Link -->
<a href="#jump-to-me">Jump to Paragraph</a>
```

- **id:** to uniquely identify an element.
- Other elements also have the id attribute.
- Each id attribute value must be unique within the HTML document.
- We can use the id element to jump to a different place in the document.

CoGrammar

Text Formatting Tags

- Text formatting tags in HTML are essential for structuring and styling textual content on webpages.
- They allow developers to apply various formatting styles to improve readability and convey meaning.
- When conflicting styles are defined in both HTML and CSS, the more specific or recently defined CSS rules will override HTML formatting.

CoGrammar

Text Formatting Tags: Heading

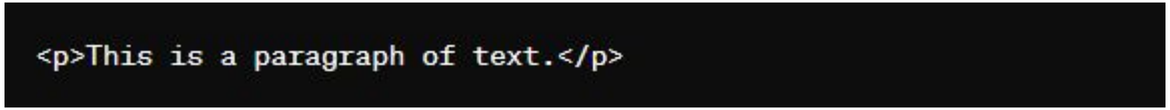
- **Heading Tags (<h1> to <h6>)**
- Heading tags are used to define headings or titles of sections within a webpage.
- Not only does the tag define the heading, but it also formats the header normally from large to smaller font size.

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
```

CoGrammar

Text Formatting Tags: Paragraph

- **Paragraph Tag (<p>)**
- The paragraph tag is used to define paragraphs of text.
- It separates blocks of text, making content easier to read and understand.
- In separating blocks of text, it also applies some line spacing as formatting.



```
<p>This is a paragraph of text.</p>
```

CoGrammar

Text Formatting Tags: Emphasis

- **Emphasis Tag (<em>)**
- The emphasis tag is used to apply emphasis or stress on text.
- It typically renders as italicised text in most browsers, although the exact styling may vary.

```
<p>This text is <em>emphasized</em>.</p>
```

CoGrammar

Text Formatting Tags: Strong

- **Strong Tag (<strong>)**
- The strong tag is used to apply strong importance or importance of greater significance than the surrounding text.
- It typically renders as bold text in most browsers, although the exact styling may vary.

```
<p>This text is <strong>strongly emphasized</strong>.</p>
```

Separation of Concerns (SoC)

- HTML plays a crucial role in separating content from presentation in web development.
- This concept, known as "separation of concerns," is essential for creating maintainable, scalable, and accessible web content.

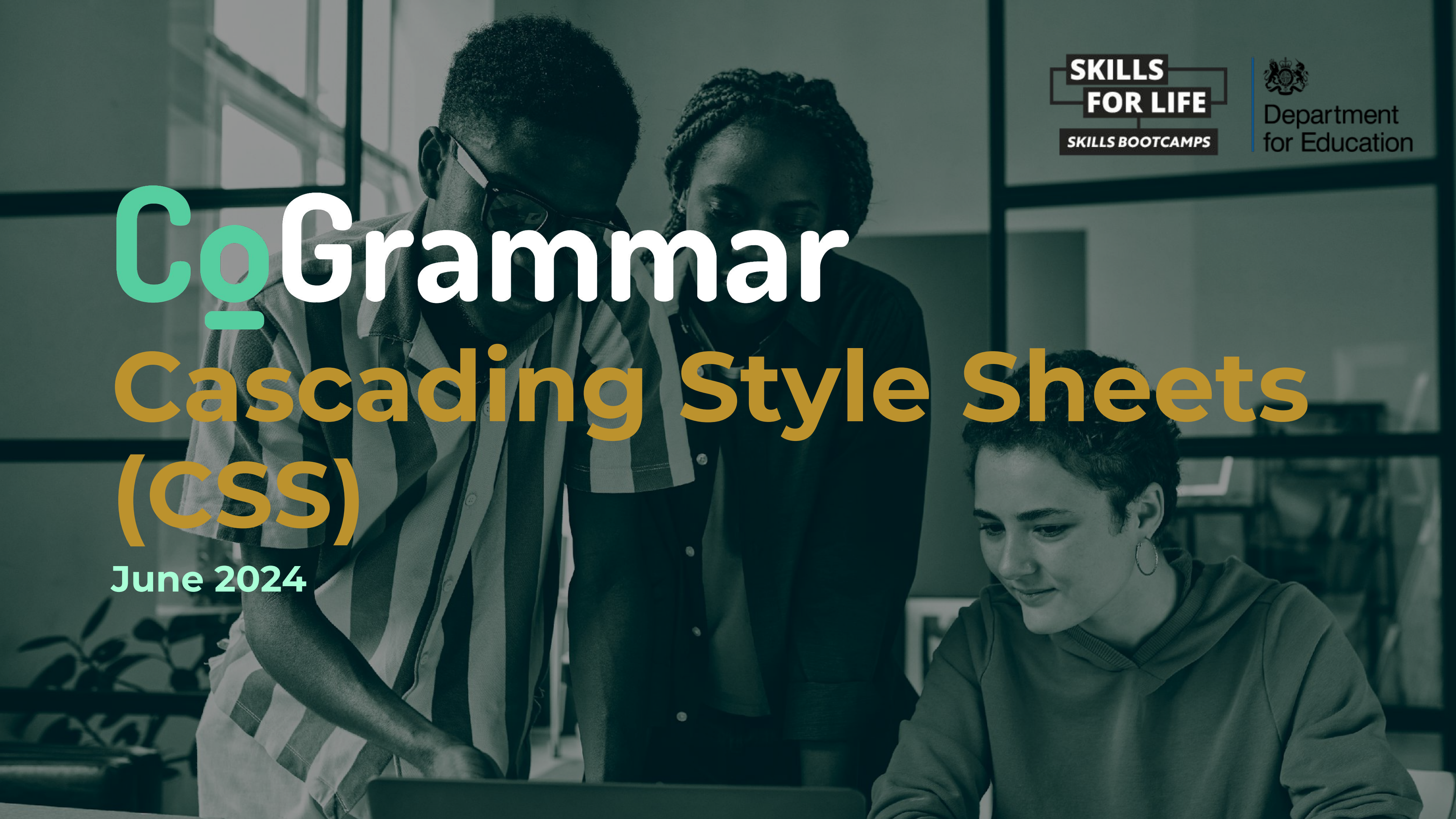
How HTML achieves SoC

- **Semantic Markup:** HTML provides a set of semantic elements that describe the structure and meaning of content rather than its presentation.
- Semantic elements like <header>, <nav>, <main>, <article>, <section>, and <footer> allow developers to structure content in a meaningful way that conveys its purpose and relationship to other elements on the webpage.
- **Content Layer:** HTML serves as the foundation or "content layer" of webpages, housing the actual content of the website, including text, images, videos, links, and other media.

How HTML achieves SoC ...

- **Separation of Style:** HTML delegates the task of styling and layout to CSS (Cascading Style Sheets). While HTML defines the structure and semantics of content, CSS controls how that content is displayed and formatted on the webpage.
- **Responsive Design:** HTML's separation of content from presentation enables the creation of responsive web designs that adapt to different screen sizes and devices.
- **Accessibility:** Semantic markup enhances accessibility by providing clear, structured content that can easily be navigated and be understandable by both humans and assistive technologies.

CoGrammar



SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Cascading Style Sheets (CSS)

June 2024

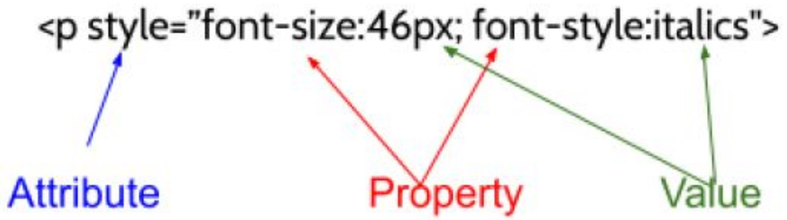
What is CSS?

- 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.

CoGrammar

Using the Style Attribute in CSS

- 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.

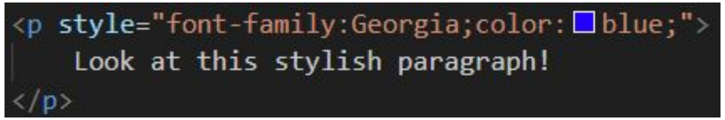


```
<p style="font-size:46px; font-style:italics">
```

CoGrammar

Inline Styling

- HTML elements are described using attributes and properties.
- You can style a web page by changing the properties of elements.
- For example: *font-family* (Arial, Times New Roman, etc), *font-style* (normal, italics, etc) and *font-size*.
- Script:



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

- Result: Look at this stylish paragraph!

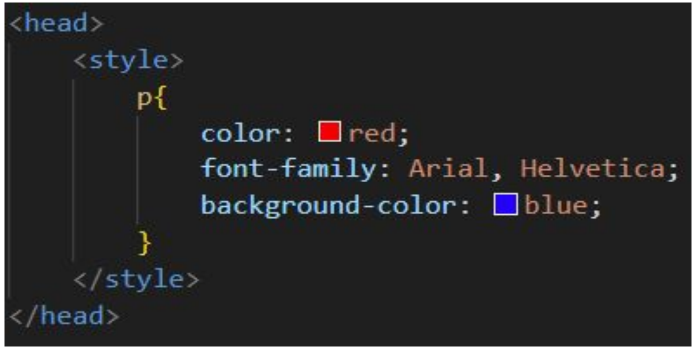
Inline Limitations

- 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.

CoGrammar

Internal CSS

- The example below shows how you can define a CSS rule in the head part of your HTML template -> This is called internal CSS.
- The CSS rule will cause all paragraphs to:



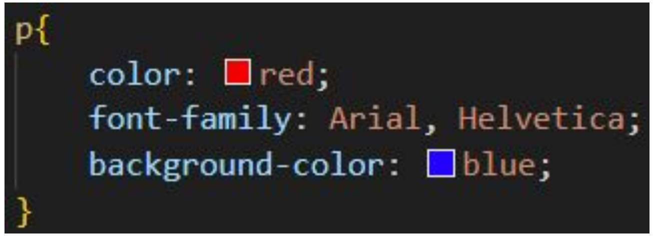
```
<head>
    <style>
        p{
            color: 🟥red;
            font-family: Arial, Helvetica;
            background-color: 🟦blue;
        }
    </style>
</head>
```

  - be in the colour red,
  - be of the font-family Arial and if the browser can't find Arial, then it will look for Helvetica,
  - also have a background colour of blue!

CoGrammar

Element Selectors

- Example:

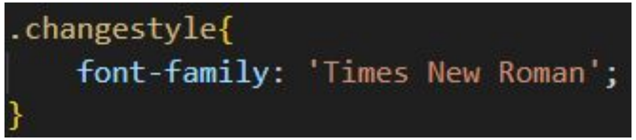


```css
p{
    color: 🟥red;
    font-family: Arial, Helvetica;
    background-color: 🟦blue;
}
```

- In the above example, all elements of type ‘paragraph’ will have the properties as defined by the selector above.

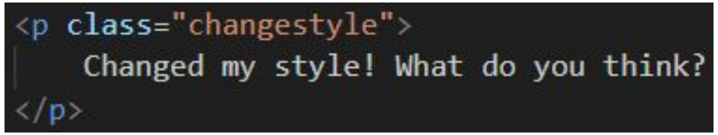
Class Selectors

- A class selector is used when the selector describes the rule for all elements that have a class attribute with the same name.
- In the <head> section, you will define the class rule in a <style> element.



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

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

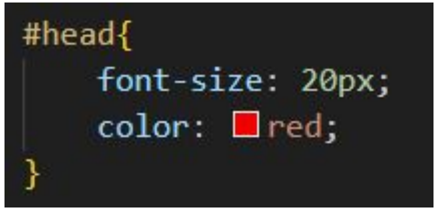


```
<p class="changestyle">
    Changed my style! What do you think?
</p>
```

CoGrammar

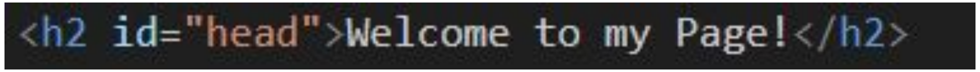
Id Selectors

- An id selector describes the style of an element with a specific id attribute defined.
- In the <head> section, you will define the id rule in a <style> element.



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

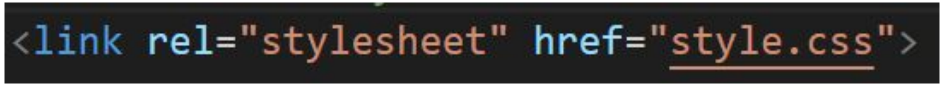
- In <body> you will use the id attribute to apply the rule.



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

External CSS

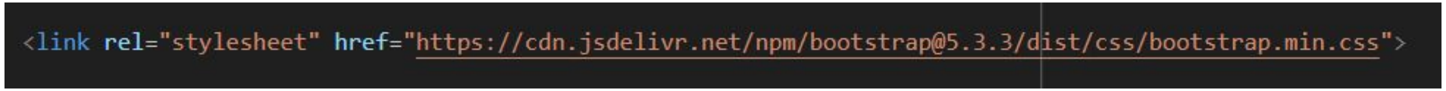
- If your website consists of many HTML files, you likely want to be able to apply the same style rules to all the web pages. To accomplish this, use external CSS instead of internal CSS.
- To do this, create a separate file with the extension .css. In this file write all the style rules that you would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



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

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.



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

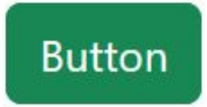
- Now that we have access to the style rules we can apply them to our pages.

CoGrammar

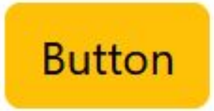
Bootstrap: Buttons

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

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



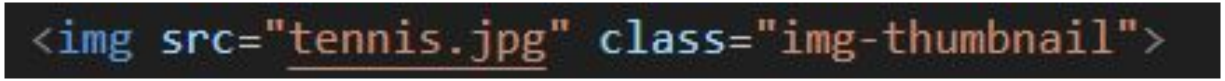
```html
<button type="button" class="btn btn-warning">Button</button>
```



CoGrammar

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.

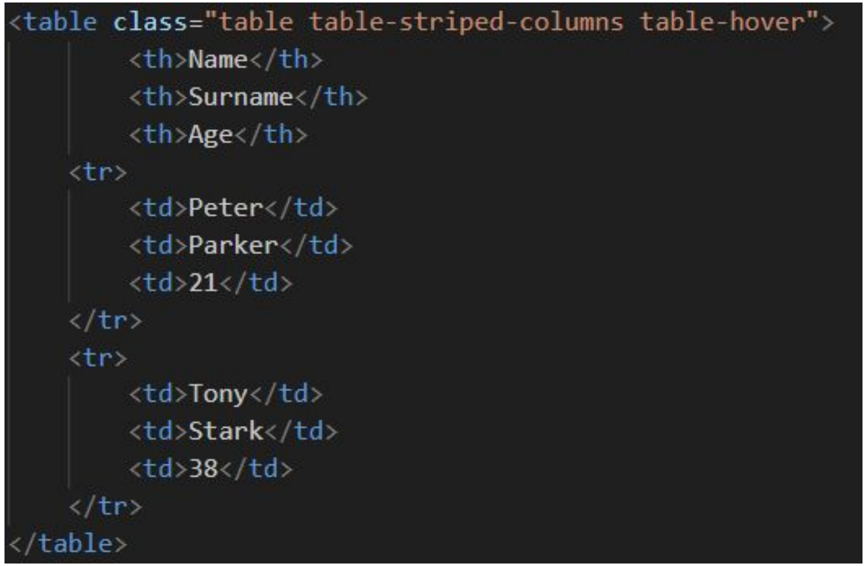


```
<img src="tennis.jpg" class="img-thumbnail">
```

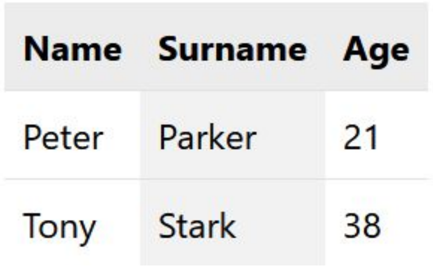
CoGrammar

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>
```



Name	Surname	Age
Peter	Parker	21
Tony	Stark	38

CoGrammar

Style 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 general rule is that the more recently, last defined rule takes precedence.
- The defined or applied order is normally bootstrap, external, internal and then inline styling.

CoGrammar

Element 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.
- The more specific a rule is, the higher its precedence.
- When using element selectors, class selectors and id selectors, element selectors are the least specific (because they could match most elements in a page) whereas id selectors are the most specific.
- Therefore, id selectors will be applied over class selectors and element selectors.

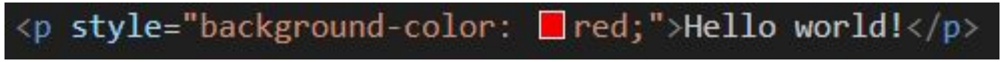
CoGrammar

Box Model

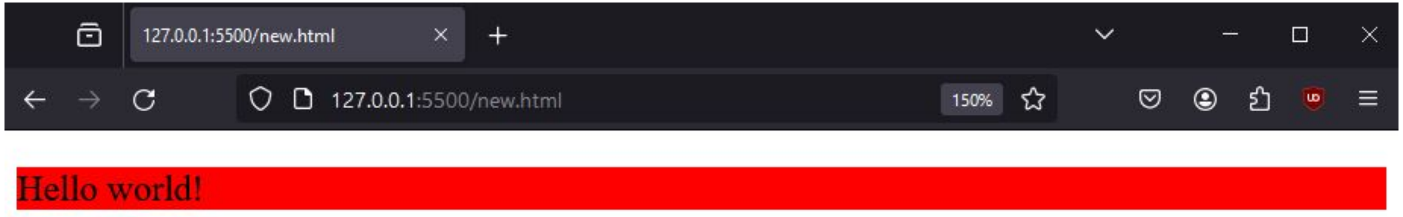
- Everything in CSS has a box around it.
- We can use these boxes to build complex layouts on our pages.
- We can set the display type of a box to **block** or **inline**.
- This will change the behaviour of our box when certain changes are applied.
- We can then edit the main parts of a box, being content,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

Box Model: Block

- A block box type will take up the full width of the page.
- Has a line break (next line) before and after the box.
- Before implementing the Block Box, let's start with below.



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



Hello world!

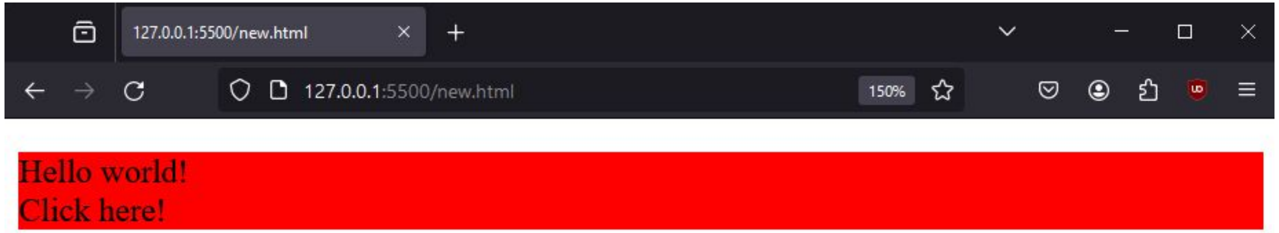
CoGrammar

Box Model: Block Applied

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

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

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

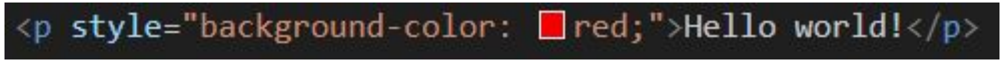


127.0.0.1:5500/new.html

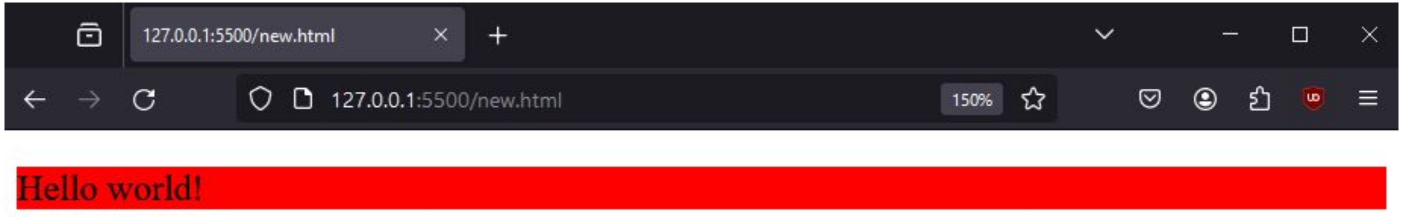
Hello world!
Click here!

Box Model: Inline

- An inline box type will take up the width of it’s content.
- Has no line break (next line) before or after the box.
- Before implementing the Inline Box, let’s start with below.



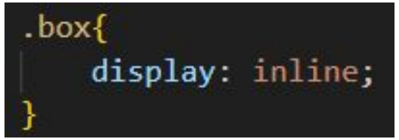
```
<p style="background-color: 🟥red;">Hello world!</p>
```



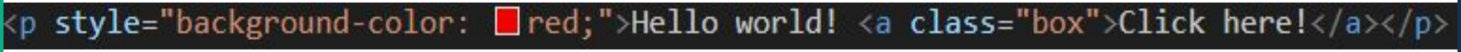
```
Hello world!
```

Box Model: Inline Applied

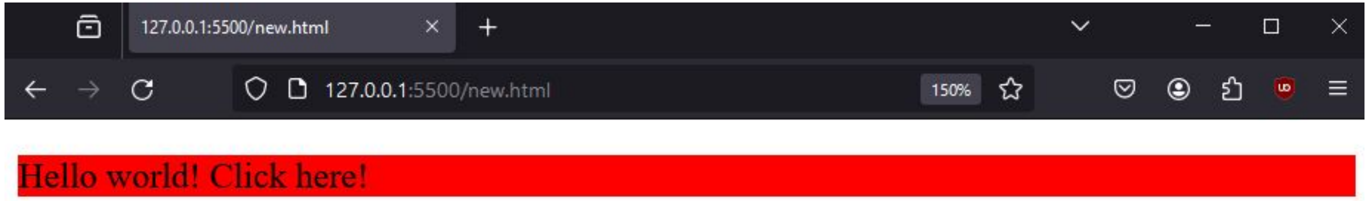
- When adding another element and setting it to use an inline display we can see how the new element gets placed next to our previous element.



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



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



127.0.0.1:5500/new.html

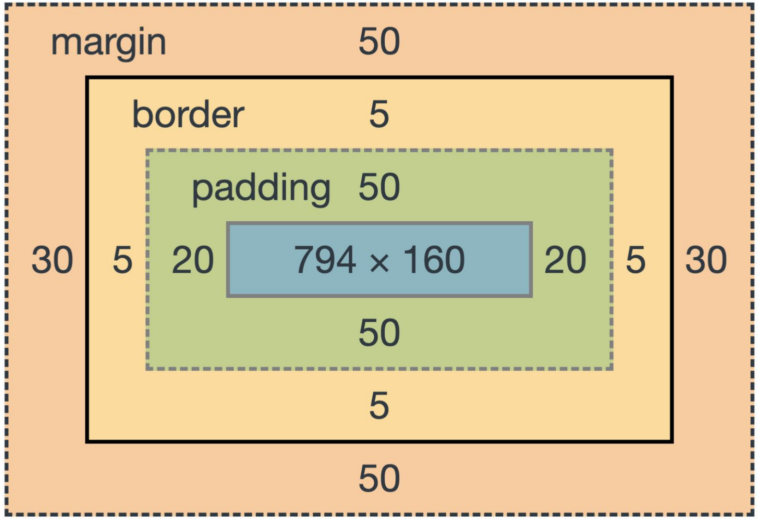
Hello world! Click here!

Box Model: Parts

- Now that we have seen some ways to structure our boxes together 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

CoGrammar

Box Model: Layout



CoGrammar

Let’s take a short break

CoGrammar

Let’s get coding!

CoGrammar

Summary

- Key concepts covered in this lesson are HTML basics, tags, attributes, and document structure.
- We looked at the importance of HTML in web development and its role in separating content from presentation.
- We looked at the different styles of CSS. Inline, internal, external and bootstrap.
- We covered the different selectors we have available to us. Element, Class and Id Selectors.
- Think of how cascading will apply to your projects.
- We spoke about the box model and how you can think of all elements as boxes.

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar