Welcome to the CoGrammar

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.
- 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 Objectives & 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.
- 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.

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

CSS

April 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.

Inline Style

- 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, any text that you add to a web page has several properties that you can change.
- For example: font-family (Arial, Times New Roman etc), font-style (normal, italics etc) and font-size.
- An example of using the style attribute to change the font of an element is shown below:

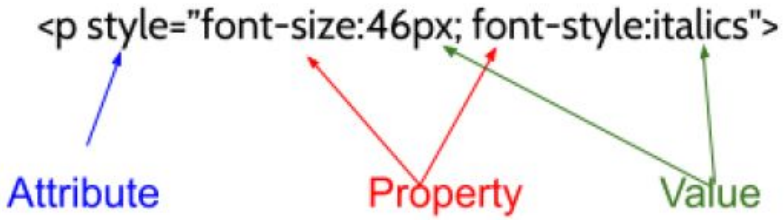
```
<p style="font-family:Georgia;color: blue;">
    Look at this stylish paragraph!
</p>
```

Look at this stylish paragraph!

CoGrammar

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.



CoGrammar

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.

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 below will cause all paragraphs to be in the colour red and be of the font-family Arial. If the browser can't find Arial, then it will look for Helvetica. Paragraphs will also have a background colour of blue!

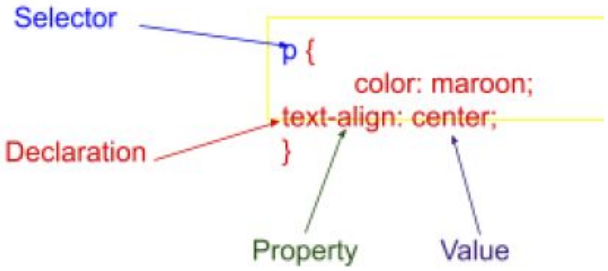
```
<head>
    <style>
        p{
            color: red;
            font-family: Arial, Helvetica;
            background-color: blue;
        }
    </style>
</head>
```

CoGrammar

Syntax

CSS syntax consists of a selector and a declaration.

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



CoGrammar

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 the selector above.

What if we do not want all the paragraphs to have the same properties?

CoGrammar

Class and ID selectors

A class selector is used when the selector describes the rule for all elements that have a class attribute with the same name defined.

In the <head> you will define the class rule. For example:

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

In <body> you will use the class attribute to apply the rule. For example:

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

Changed my style! What do you think?

CoGrammar

Class and ID selectors

An ID selector describes the style of an element with a specific ID attribute defined.

In the <head> you will define the id rule. For example:

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

In <body> you will use the id attribute to apply the rule. For example:

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

Welcome to my Page!

CoGrammar

External CSS

If your website consists of many HTML files, you are likely to 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 applied.

To link an external CSS file to a specific HTML file, do the following:

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

CoGrammar

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

As we know, CSS stands for cascading style sheets. You may have wondered why they are called cascading style sheets. Cascading has to do with how the rules are applied.

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, properties override other properties, but entire rules don't override other rules.

Cascading

When several CSS rules match the same element, they are all applied to that element. Only after that 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.

For example, in a stylesheet that uses element selectors, class selectors and ID selectors, element selectors are the least specific (because they could match the most elements in a page) whereas ID selectors are the most specific. Therefore, ID selectors will be applied over class selectors and element selectors.

Box Model

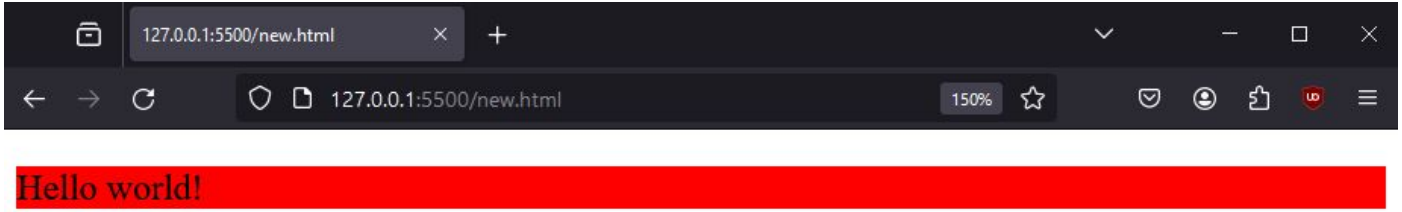
- Everything in CSS has a box around it.
- We can use these boxes to build complex layouts on our pages.
- We can set the display type of a box to block and inline.
- This will change the behaviour of our box when certain changes are applied.
- We can then edit the main parts of our box. Content, padding,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

CoGrammar

Box Model

- Block
  - A block box type will take up the full width of the page.
  - Has a line break before and after the box.

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



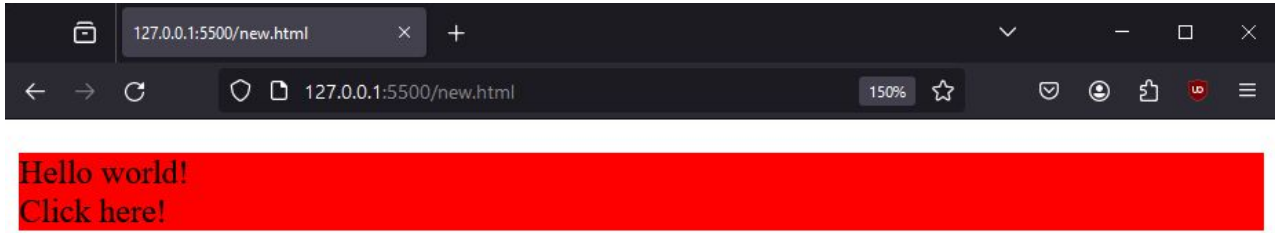
CoGrammar

Box Model

- 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 as much width as possible.

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

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

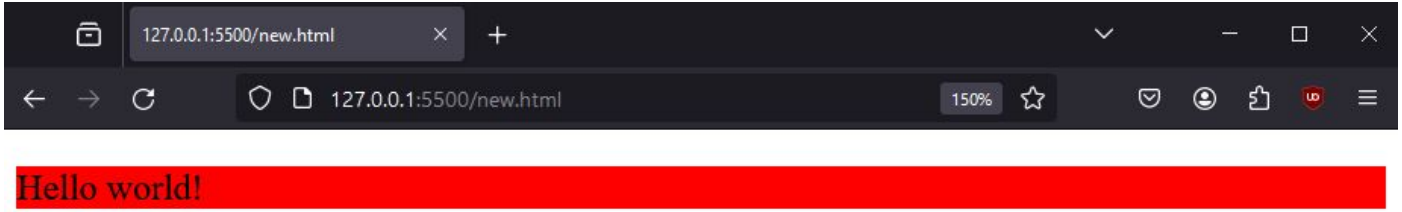


CoGrammar

Box Model

- Inline
  - A block box type will take up the width of the it's content.
  - Has no line break before or after the box.

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



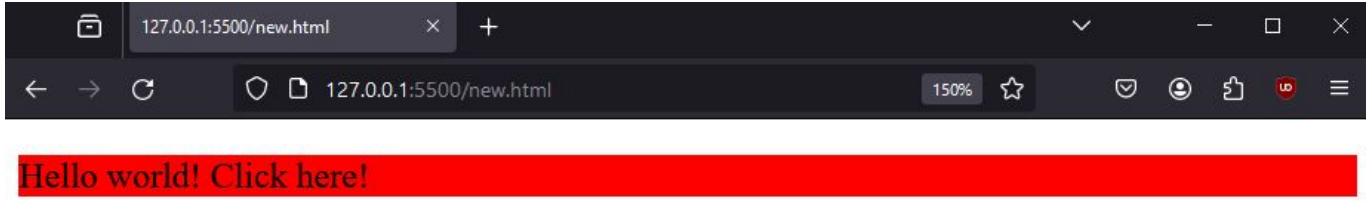
CoGrammar

Box Model

- When adding another element inline and setting it to use a 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>
```



CoGrammar

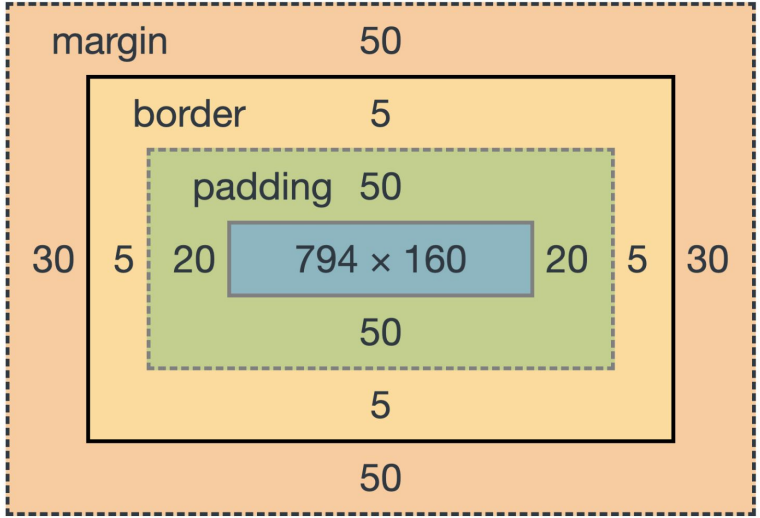
Box Model

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 actually 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



CoGrammar

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

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

CoGrammar

Bootstrap

- 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.

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

Button

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

Button

CoGrammar

Bootstrap

- 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

- 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>
```

CoGrammar

Bootstrap

Name	Surname	Age
Peter	Parker	21
Tony	Stark	38

CoGrammar

Let’s take a short break

CoGrammar

Summary

- **CSS:** Allows us to apply style to our web pages.
- **Inline, Interal, 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 you web pages.
- **Bootstrap:** A web framework with predefined CSS rules that you can apply to your projects to streamline design.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar