



**SE Bootcamp**

Hyperiondev

# CSS Workshop

# Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- ❑ 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.
- ❑ You can also submit questions here:  
<http://hyperiondev.com/sbc4-se-questions>
- ❑ For all non-academic questions, please submit a query:  
[www.hyperiondev.com/support](http://www.hyperiondev.com/support)
- ❑ Report a safeguarding incident:  
<http://hyperiondev.com/safeguardreporting>
- ❑ We would love your feedback on lectures:  
<https://hyperiondev.wufoo.com/forms/zsgv4m40ui4i0g/>

# Objectives

1. Introduction to CSS
2. Inline Style
3. Internal CSS
4. External CSS

# Github Repository – Lecture Examples

[https://github.com/HyperionDevBootcamps/C4\\_SE\\_lecture\\_examples](https://github.com/HyperionDevBootcamps/C4_SE_lecture_examples)

## Resources

<https://www.w3.org/Style/Examples/007/center.en.html>

<https://developer.mozilla.org/en-US/docs/Learn/CSS>

# Introduction to HTML

**HTML** stands for Hypertext Markup Language. It is a language that we use to write files that tell the **browser** how to **lay out** text, images, tables, “content” etc. on a page.

We use HTML **tags** to define how the page must be structured.

# HTML Tags

HTML **tags** are placed on the left and the right of the element you want to **markup**, to wrap around the element.

General syntax:

*<opening tag>Some content that should be Marked up.</closing tag>*

There are a few exceptions, which we will discuss later. The words 'opening tag' and 'closing tag' are just **placeholders** we use to illustrate the pattern. Instead of those words, we are going to use **special keywords**, or elements, that modify the **appearance** of our webpage.

# HTML Tags Cont.

Note that the **syntax** of the opening and closing tags are not the same.

- The opening tag consists of an opening angled bracket (<), **the name of the element**, and a closing angled bracket, (>).
- The closing tag consists of an opening angled bracket, (<), a forward slash, (/), **then the name of the tag**, and finally the closing angled bracket, (>).

Example:

**<p>**This text will be marked up as a paragraph**</p>**

# HTML Tags Cont.

The HTML tags indicate to the browser what sort of **structure** the **content** is contained in.

Note that HTML does not include the **style** of the content (e.g. font, colour, size, etc.), which is done using **CSS** (Cascading Style Sheets), but only the structure and content itself.

CSS will be covered in the next workshop :)



# HTML Elements

An element usually consists of an **opening tag** `<element_name>` and a **closing tag** `</element_name>`, which each contain the element's name surrounded by angle brackets, and the **content** in between these, as follows:

Example:

```
<element_name>...content...</element_name>
```

# Basic Layout (Template) of an HTML Page

A typical HTML document consists of:

- 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 (contained) inside the **html** element.

```
<!DOCTYPE html>

<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

# Attributes

Attributes are “*things*” that **describe** the objects created by HTML elements.

Example:

```
<title id = "myTitle">My first web page</title>
```

In this case, the element is of type **title**. Next, we have an id, which is an **attribute** of the element (title), and has a **value** of “myTitle”.

# Common HTML Elements

- A piece of **metadata** that should be included in all web page is the **<title>** element.
- As you would with a word document, use **headings** to show the structure of your web page. There are 6 types of heading: **<h1>...<h6>**
- Add **paragraphs** of text using the **<p>** element.
- Lists can either be ordered lists **<ol>** or unordered lists **<ul>**. Ordered simply means that the list is numbered, i.e. 1, 2, 3, etc. and unordered is in the form of bullet points.

# Common HTML Elements

- To do the equivalent of pressing enter to get a **line break** between text, use the `<br>` element.
  - This element does not have a matching closing tag.
- By adding the HTML element `<hr>` to your web page you will create a horizontal rule.
  - This element does not have a matching closing tag.

# Introduction to 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.

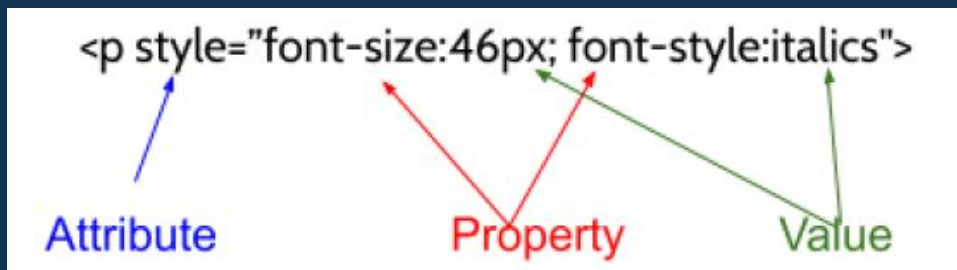
# Inline Style

- 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-size:46px;color:blue;">  
    This is the paragraph where I describe myself.  
</p>
```

# Introduction to 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.





# 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.

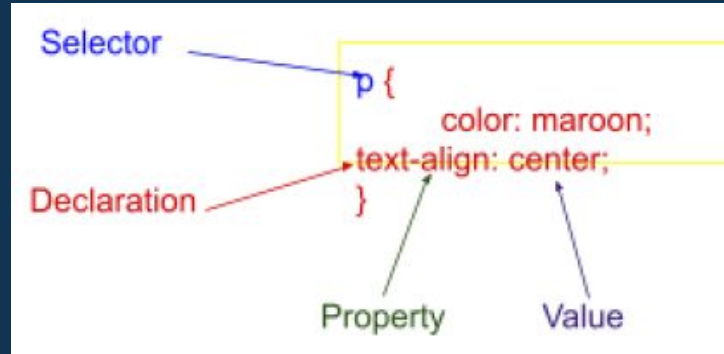
# Internal CSS

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 example below shows a CSS rule that 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!

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

# Introduction to CSS

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.

# Example

The CSS below should look familiar. Here the selector is always an element.

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

Note that all elements of type 'paragraph' will have the properties as defined by the the selector above.

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

# 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: cursive;  
}
```

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

```
<p class="changestyle">Only this paragraph's style will change</p>
```

# 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:

```
#center {  
    text-align: centre;  
}
```

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

```
<p id="center"> This paragraph will be centered</p>
```

Although you can have many elements that have a class attribute with the same value, an ID name must be unique in the document!

# 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">
```

- 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.



# What does CSS actually mean?

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.

# What does CSS actually mean?

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.

Hyperiondev

# Q & A Section

**Please use this time to ask any questions relating to the topic, should you have any.**



Hyperiondev

# Thank you for joining us

Stay hydrated  
Avoid prolonged screen time  
Take regular breaks  
Have fun :)