

# Web Development

Higher Diploma in Science in Computer Science

---

Produced  
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics  
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



# CSS Fundamentals

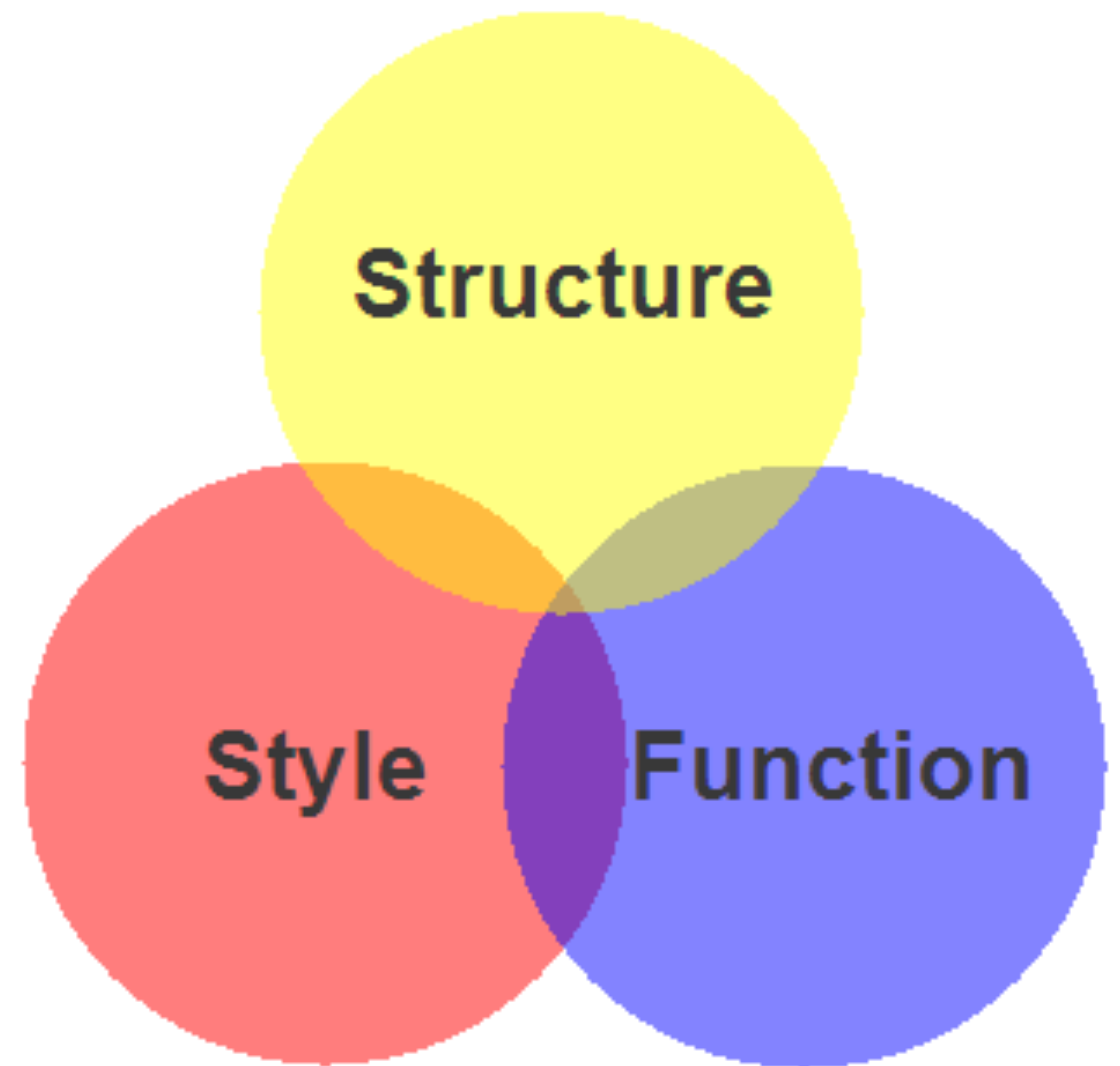
---

Enterprise Web Development

# Markup, Style, Function

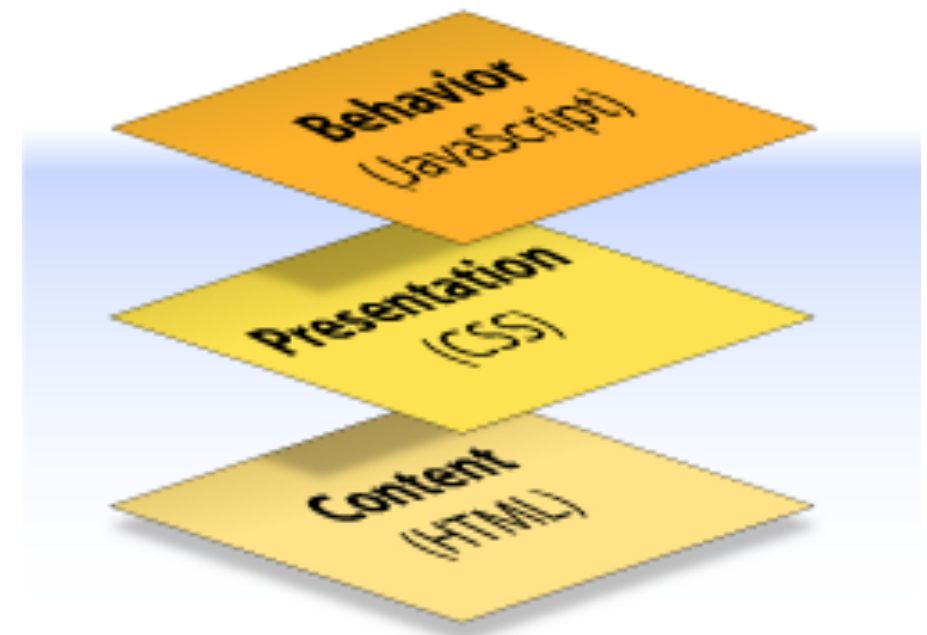
---

- Markup (HTML)
  - Structure
  - Content
- Style (CSS)
  - Style
  - Presentation
  - Appearance
- Function (Javascript)
  - Actions
  - Manipulations



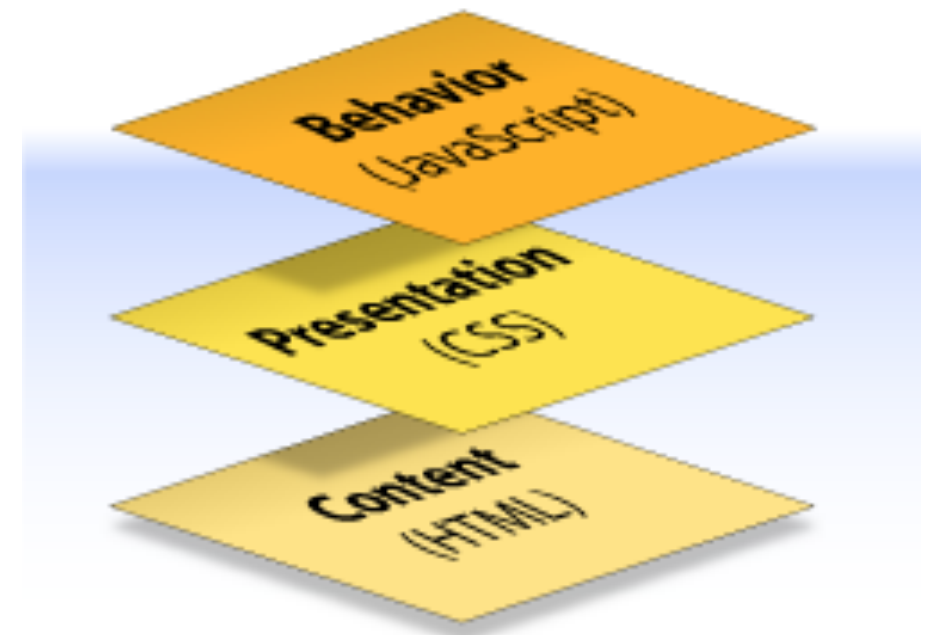
# Content, Presentation, Behavior

- **Content** comprises the information the author wishes to convey to his or her audience, and is embedded within HTML markup that defines its structure and semantics.
- Much of the content on the Web today is text, but content can also be provided through images, animations, sound, video, and whatever else an author wants to publish



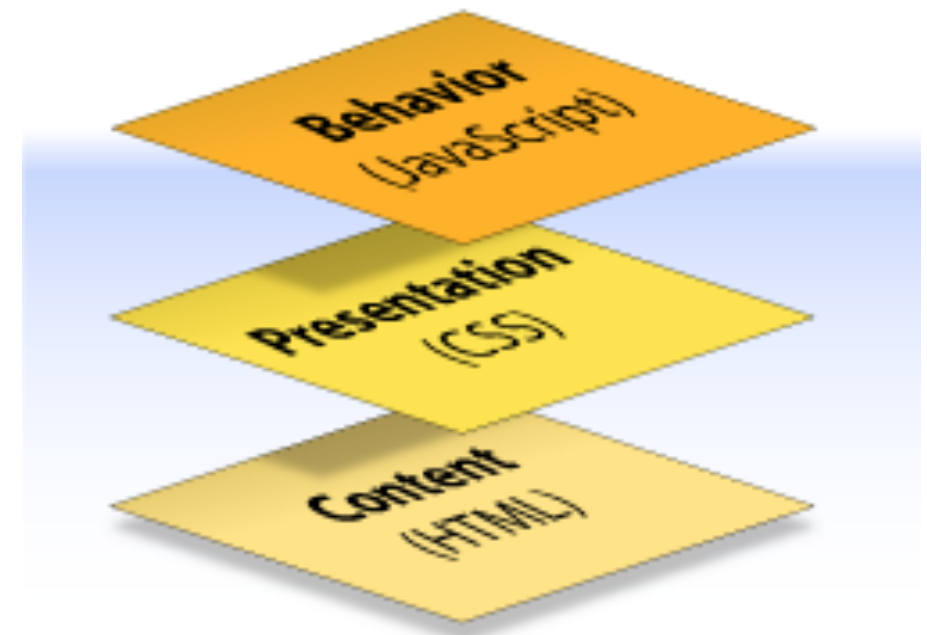
# Content, Presentation, Behavior

- **Presentation** defines how the content will appear to a human being who accesses the document in one way or another
- The conventional way to view a web page is with a regular web browser, of course, but that's only one of many possible access methods. For example, content can also be converted to synthetic speech for users who have impaired vision or reading difficulties



# Content, Presentation, Behavior

- **Behavior** layer involves real-time user interaction with the document.
- This task is normally handled by JavaScript.
- The interaction can be anything from a trivial validation that ensures a required field is filled in before an order form can be submitted, to sophisticated web applications that work much like ordinary desktop programs.



# Separation of Concerns

---

- It's possible to embed all three layers within the same document
  - For example `<em>` and `<b>` can be used to control the presentation of text, and `<hr>` will insert a visible rule element.
  - Sometimes called *Presentation Markup*, these types of elements embed presentation-layer information within the content layer, they negate any advantage we may have gained by keeping the layers separate.
- Keeping them separate gives us one valuable advantage:
  - ***We can modify or replace any of the layers without having to change the others.***

# What is CSS?

---

- Cascading Style Sheets is the recommended way to control the presentation layer in a web document.
- The main advantage of CSS over presentational HTML markup is that the styling can be kept entirely separate from the content.
- e.g. it's possible to store all the presentational styles for a 10,000-page web site in a single CSS file.
- CSS also provides far better control over presentation than do presentational element types in HTML.



# Advantages of CSS

---

- All styling is kept in a limited number of style sheets.
- Saving on bandwidth: The style sheet is cached (kept on client machine) after the first request and can be reused for every page on the site, it doesn't have to be downloaded with each web page. Could reduce bandwidth usage—by more than 50% in many documented cases.
- The separation of content from presentation makes it easier for site owners to reuse the content for other purposes, such as RSS feeds or text-to-speech conversion.
- Separate styling rules can be used for different output media. We no longer need to create a special version of each page for printing—we can simply create a single style sheet that controls how every page on the site will be printed.

# A CSS Rule

The first thing you do is select the element you want to style, in this case the <p> element. Notice in CSS, you don't put <> around the name.

p {  
background-color: red;  
}

Place all the styles for the <p> element in between { } braces.

Then you specify the property you want to style, in this case the <p> element's background color.

There's a colon in between the property and its value.

And you're going to set the background-color to red.

At the end, put a semicolon.

We call the whole thing a RULE.

# More Properties...

---

- You can add as many properties and values as you like in each CSS rule.
- To put a border around your paragraphs:

```
p {  
  background-color: red;  
  border: 1px solid gray;  
}
```

All you have to do is add  
another property and value.

The <p> element  
will have a border...

...that is 1 pixel thick, solid, and gray.

# <style> head element

```
<html>
  <head>
    <title>My DVD Shop</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <style>
      p
      {
        color: maroon;
      }
    </style>
  </head>
  <body>
    <h1>Welcome to the New and Improved DVD Shop</h1>
    <p>
```

**MARCH MADNESS**  
**MIX & MATCH:**  
**2 CDs OR DVDs**  
**FOR €12 OR 3 FOR €17**

We have the biggest selection of DVDs and Games in Ireland. Check out our [new releases](#), This week we recommend *the Crazies*. From our extensive catalog of DVDs and Games you choose and prioritise the DVDs you most want to see or Games you want to play. We have an advanced search facility available for our users to search through our huge catalog of stock. We stock a wide variety for movies from *Manga to Music and Comedy to Childrens*, Games from *Puzzles to Beat'em Up and Party to Action/Adventure* as well as all the latest new releases.

# Multiple Rules

---

```
h1 {  
    font-family: sans-serif;  
    color:      gray;  
}
```

```
h2 {  
    font-family: sans-serif;  
    color:      gray;  
}
```

```
p {  
    color: maroon;  
}
```

Here's the rule to select `<h1>` elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

And here's another rule to do the exact same thing to the `<h2>` element.

# Combining Selectors (1)

- Rules can be combined if they are identical

```
h1 {  
    font-family: sans-serif;  
    color:      gray;  
}  
  
h2 {  
    font-family: sans-serif;  
    color:      gray;  
}  
  
p {  
    color: maroon;  
}
```

Here's the rule to select <h1> elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

And here's another rule to do the exact same thing to the <h2> element.

```
h1, h2 {  
    font-family: sans-serif;  
    color:      gray;  
}
```

To write a rule for more than one element, just put commas between the selectors, like "h1, h2".

```
p {  
    color: maroon;  
}
```

# Border Styles

---

```
border-bottom: 1px solid black;
```

This property controls how the border under an element looks.

We're going to style the bottom border so that it is a 1 pixel thick, solid black line.

- Placing the above rule associated with h1 “selector”, will draw a line - 1 pixel wide - under the heading in our site

**Welcome to the New and Improved DVD Shop**

---



# Combining Selectors (2)

---

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

The first rule stays the same. We're still going to use a combined rule for the font-family and color for both <h1> and <h2>.

```
h1 {  
  border-bottom: 1px solid black;  
}
```

But now we're adding a second rule that adds another property just to <h1>: the border-bottom property.

```
p {  
  color: maroon;  
}
```

- Both h1 and h2 share the font-family and color attributes, however only h1 is underlined



# Separating Style Sheets from HTML

---

- Ideally, a site may have a single style sheet for most purposes,
  - and not a repeated set of rules inside every html file

```
<head>
  <title>My DVD Shop</title>
  <style>
    h1
    {
      font-family: sans-serif;
      color: gray;
    }

    h2
    {
      font-family: sans-serif;
      color: gray;
    }
  </style>
</head>
<body>
```

```
<head>
  <title>New Releases</title>
  <style>
    h1
    {
      font-family: sans-serif;
      color: gray;
    }

    h2
    {
      font-family: sans-serif;
      color: gray;
    }
  </style>
</head>
<body>
```

```
<head>
  <title>FAQ</title>
  <style>
    h1
    {
      font-family: sans-serif;
      color: gray;
    }

    h2
    {
      font-family: sans-serif;
      color: gray;
    }
  </style>
</head>
<body>
```

# <link> element

---

Use the link element to "link in" external information.

The type of this information is "text/css". In other words, a CSS style sheet.

And the style sheet is located at this href (in this case we're using a relative link, but it could be a full-blown URL).

`<link type="text/css" rel="stylesheet" href="lounge.css" />`

The rel attribute specifies the relationship between the XHTML file and the thing you're linking to. We're linking to a style sheet, so we use the value "stylesheet".

<link> is an empty element.

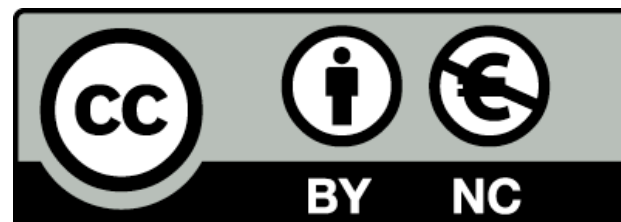
# One Stylesheet - Multiple html pages

---

```
<head>
  <title>My DVD Shop</title>
  <link type="text/css" rel="stylesheet" href="mydvdshop.css" />
</head>
<body>
```

```
<head>
  <title>New Releases</title>
  <link type="text/css" rel="stylesheet" href="mydvdshop.css" />
</head>
<body>
```

```
<head>
  <title>FAQ</title>
  <link type="text/css" rel="stylesheet" href="mydvdshop.css" />
</head>
<body>
```



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

