EBU6305

Interactive Media Design and Production

CSS Basics

Written by Dr. Ethan Lau & Dr. Changjae Oh

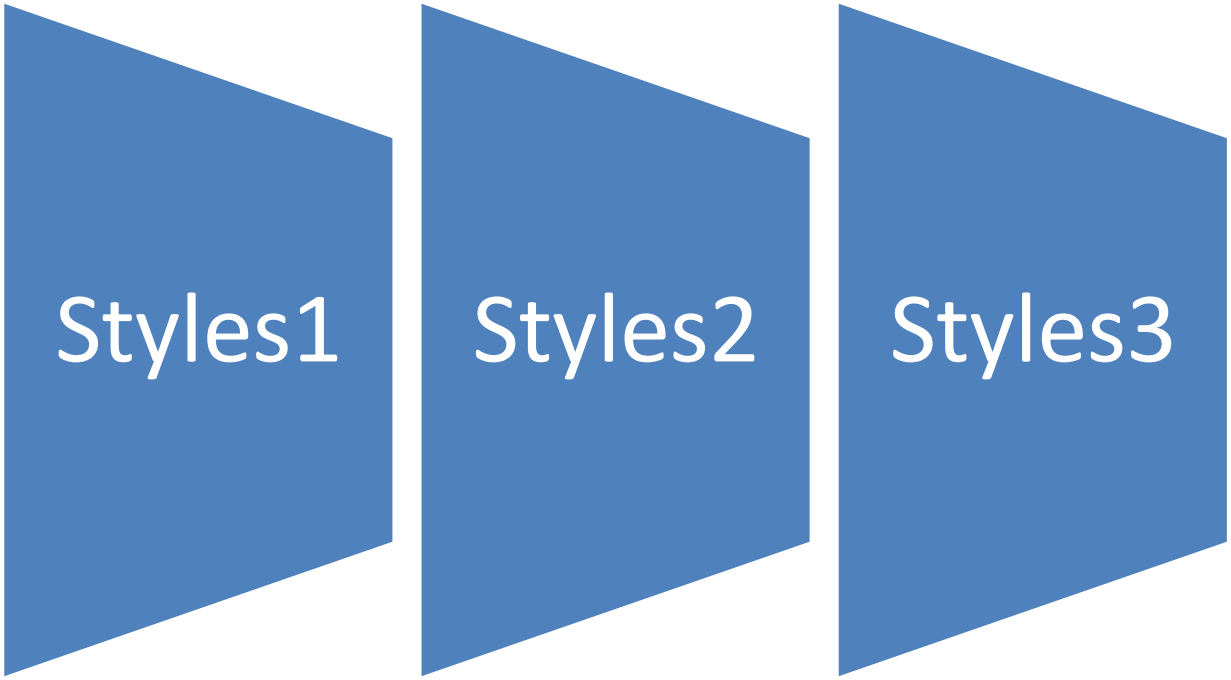
Delivered by Dr. Xianhui Cherry Che

Learning objectives

On completion of this topic, you will be able to:

- Understand the syntax of CSS
- Describe the structure of CSS
- Write CSS!

What is CSS?



What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files**

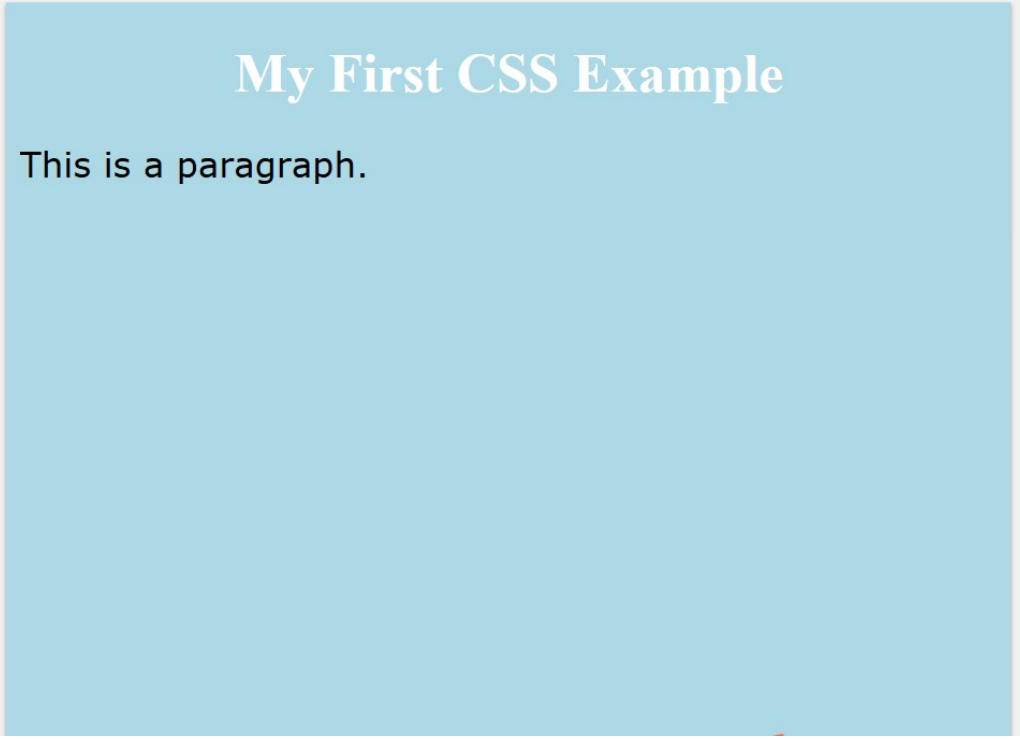
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}

h1 {
    color: white; text-align: center;
}

p {
    font-family: verdana; font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



CSS Topics

- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

CSS Topics

- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

A Style

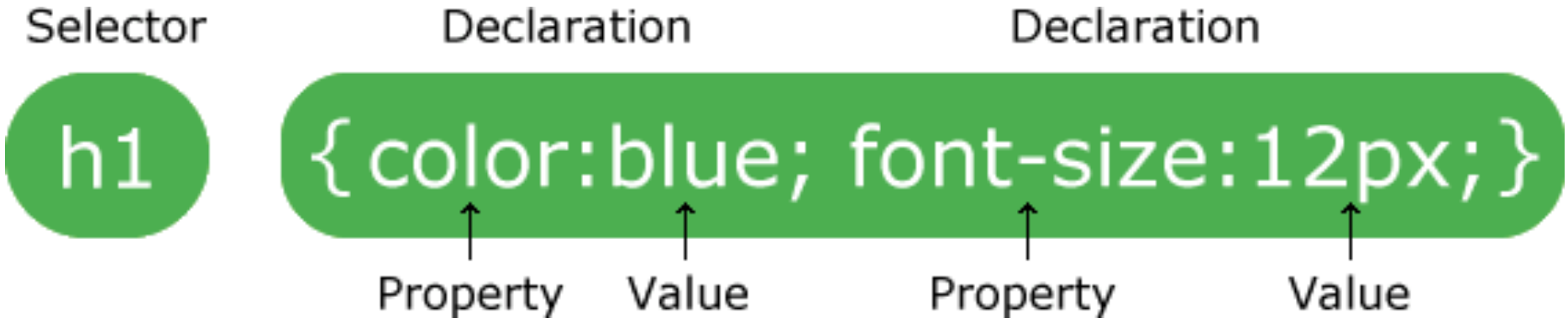
Selector Property Value

```
p { font-family: times; }
```

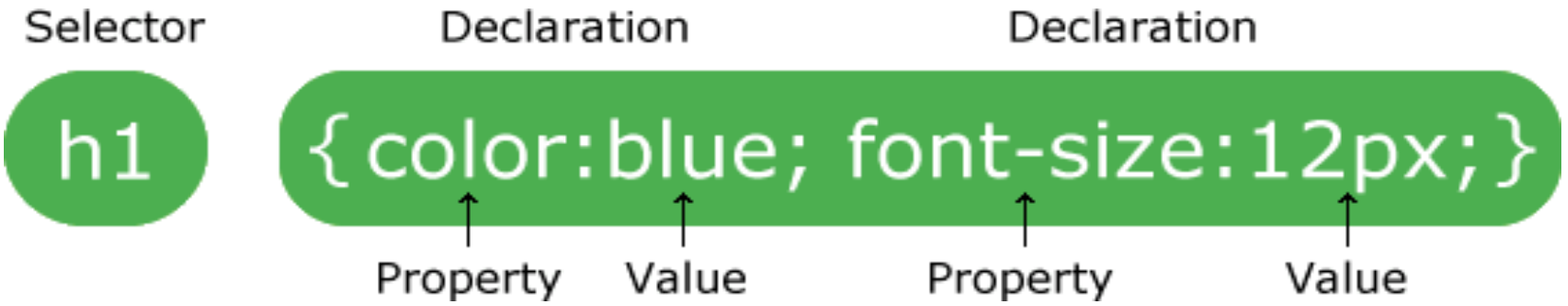
- Note the punctuation: The property is followed by a colon (:) and the value is followed by a semicolon(;)

CSS Rule

- A CSS rule-set consists of a selector and a declaration block:
  - The **selector** points to the HTML element you want to style
  - The **declaration block** contains one or more declarations separated by semicolons.
  - Each **declaration** includes a CSS property name and a value, separated by a colon.
  - A CSS **declaration** always ends with a semicolon, and declaration blocks are surrounded by curly braces.



CSS Rule




```
h1 {
    color: red;
    font-size: xx-large;
    text-align: center;
}
```

Using CSS

Rationale for External CSS redundancy: don't need all of the styles in CSS, may design a lot of codes, but only use part of definition in CSS. (avoid they don't need) re-usability: reuse in another scenario

```
/* melville.css style sheet */
body { font-family: verdana, helvetica, sans-serif;
       background: #f1f1f1;
     }
```

Stylesheet

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>Moby Dick, or, The Whale</title>
    <link rel="stylesheet" href="melville.css"
      type="text/css" />
    <style type="text/css">
    <!--
      h1 { font-style: italic; }
    -->
    </style>
  </head>
  <body>
    <div style="text-align: center;
        width: 70%; border: solid black 1px;">
      <h2>About "Moby Dick, or, The Whale"</h1>
      <p>This story of a obsessive sea captain's quest to
      <span style="color: red;">find and destroy<span>
        an extraordinary whale that had taken his leg in a
        previous encounter is sometimes seen as an allegory
        for all of humanity's obsessive pursuits.
      </p>
    </div>
    <h1>Moby Dick, or, The Whale</h1>
    <p>Call me Ishmael. Some years ago –
      never mind how long precisely – having
      little or no money in my purse, and nothing
      particular to interest me on shore, I thought
      I would sail about a little and see the watery
      part of the world.</p>
  </body>
</html>
```

Style element

Style attribute

Styles can be set in a stylesheet, in a style element in the head or in a style attribute

CSS Syntax and Selectors

- The element Selector
  - Based on the element name
  - p { text-align: center;}
- The id Selector:
  - A hash (#) character, followed by the id of the element
  - #para1 { text-align: center;}
- The class Selector
  - A period (.) character, followed by the name of the class
  - .center { text-align: center;}
- Grouping Selectors
  - For elements with the same style definitions
  - To group selectors, separate each selector with a comma
  - h1, h2, p { text-align: center;}

CSS Topics

- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

What Does "Cascading" Mean?

We use the term "cascading" because there is an established order of priority to resolve formatting conflicts:

priority of execution

1. Inline style (highest priority)
2. Internal style sheet (second priority)
3. External style sheet (third priority)
4. Web browser default (only if not defined elsewhere)

For each element, the browser will see which styles are defined inline and from internal and external style sheets. For any conflicts detected, it will use this priority system to determine which format to display on the page.

Example: the heading text would display in the color specified by the inline style, which outranks all the others.

If multiple, conflicting styles are defined in the **same** style sheet, only the **final one will be applied**. Be careful, as this is another common mistake committed by beginners.

Three Ways to Use CSS

We can add CSS code in any combination of three different ways:

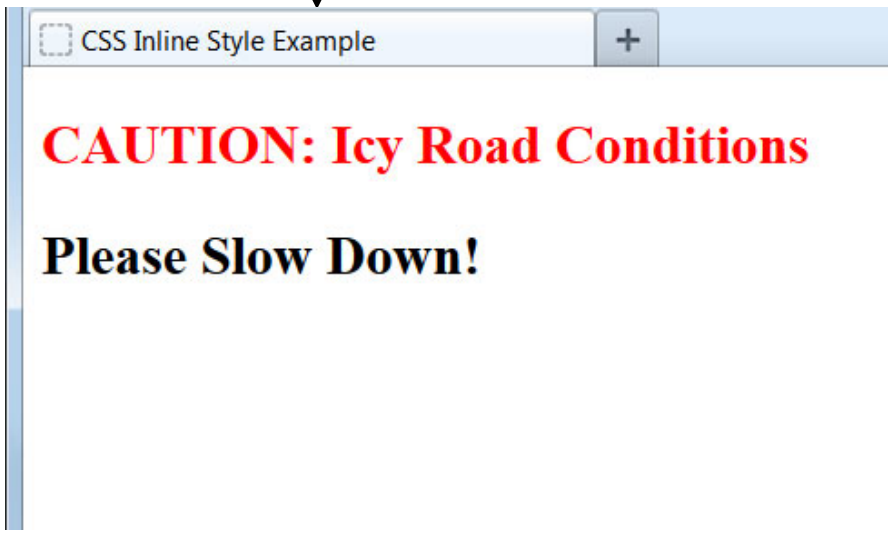
1. **Inline Style** - CSS code is placed directly into an XHTML element within the <body> section of a web page.
2. **Internal Style** - CSS code is placed into a separate, dedicated area within the <head> section of a web page.
3. **External Style** - CSS code is placed into a separate computer file and then linked to a web page.

Let's take a look now at examples of each of these methods.

Inline Style

To define an inline CSS style, we simply add the **style** attribute to an XHTML element with the CSS declaration as the attribute value:

```
<h2 style="color:red;">CAUTION: Icy Road Conditions</h2>
<h2>Please Slow Down!</h2>
```



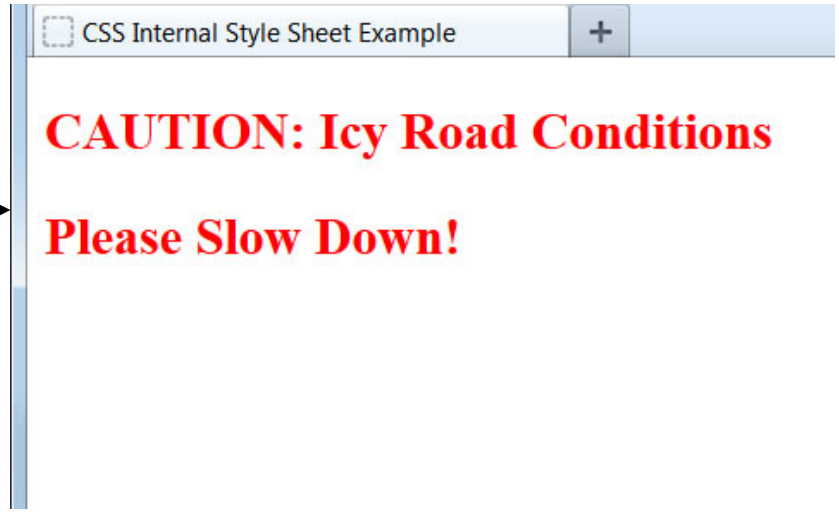
An inline style declaration is highly specific and formats just one element on the page. No other elements, including other <h2> elements on the page, will be affected by this CSS style.

Since inline styles have limited scope and do not separate content from presentation, their use is generally discouraged. We won't be using inline styles much in this class.

Internal Style Sheet

To use an internal CSS style sheet, we add a **<style>** section within the **<head>** of the page. All our CSS declarations go within this section:

```
<head>
  ...
  <style type="text/css">
    h2 {color:red;}
  </style>
</head>
<body>
  <h2>CAUTION: Icy Road Conditions</h2>
  <h2>Please Slow Down!</h2>
</body>
```



Styles declared in the internal style sheet affect all matching elements on the page. In this example, all <h2> page elements are displayed in the color red.

Since formatting declarations are entirely in the <head> section, **away from the actual page content**, **internal CSS style sheets do a much better job than inline styles** at separating content from presentation.

External Style Sheet

To use an external CSS style sheet, we create a new file (with a .css extension) and write our style declarations into this file. We then add a **<link>** element into our HTML file, right after the opening **<head>** tag:

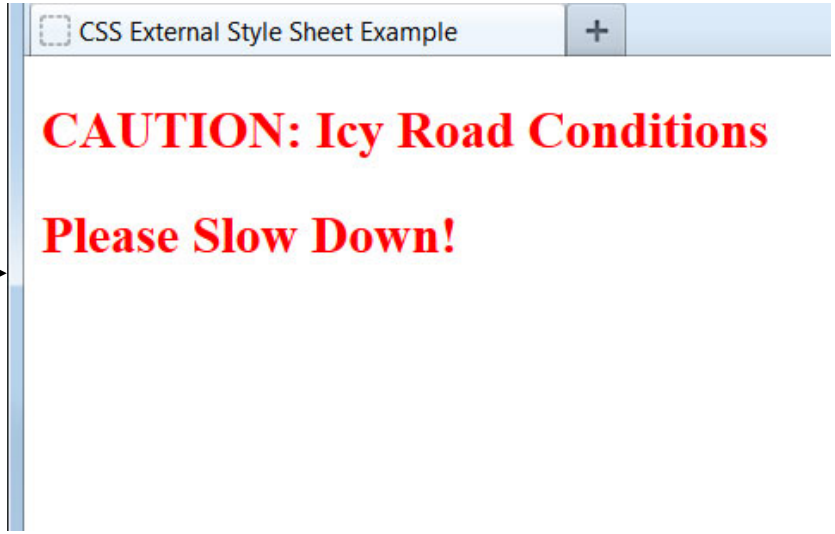
link is outside the style bracket

style.css (separate file):

```
h2 {color:red;}
```

example.html file:

```
<head>
  <link rel="stylesheet" type="text/css"
   href="style.css" />
  ...
</head>
<body>
  <h2>CAUTION: Icy Road Conditions</h2>
  <h2>Please Slow Down!</h2>
</body>
```



The <link> element instructs the browser to load the external file specified by the href attribute and to apply the CSS style declarations contained there.

External Style Sheet – more example

style.css

```
body {
    background-
color: powderblue;
}
h1 {
    color: blue;
}
p {
    color: red;
}
```


```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

CSS Topics

- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

HTML Class

- The HTML **class attribute** is used to define equal styles for elements with the same class name.
- All HTML elements with the same **class** attribute will have the same format and style.
- To **style** elements with a specific class, write a period **(.)** character, followed by the name of the class

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
    background-color: black; color: white;
    margin: 20px; padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
<h2>London</h2> <p>London is the capital of England.</p>
</div>

<div class="cities">
<h2>Paris</h2> <p>Paris is the capital of France.</p>
</div>


</body>
</html>
```

HTML Class

- HTML elements can have more than one class name, each class name must be separated by a space.
- Different tags, like <h2> and <p>, can have the same class name and thereby share the same style.

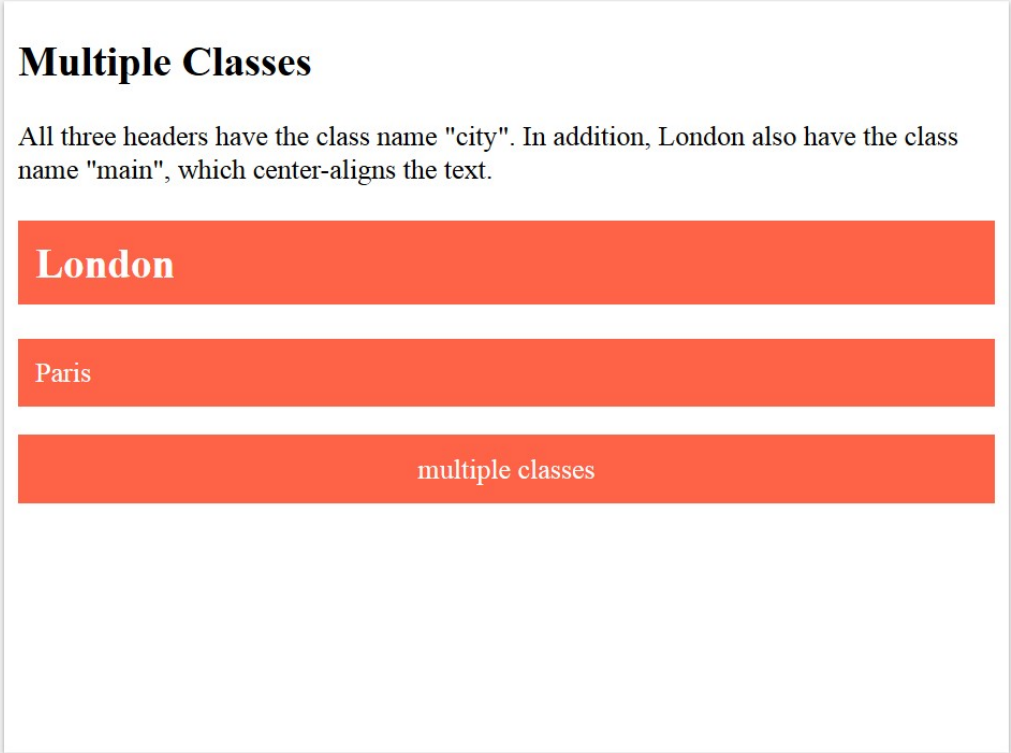
```
<html>
<style>
.city {
    background-color: tomato; color: white; padding: 10px;
}

.main {
    text-align: center;
}
</style>
<body>

<h2>Multiple Classes</h2>
<p>All three headers have the class name "city". In addition,
London also have the class name "main", which center-aligns the
text.</p>

<h2 class="city">London</h2>
<p class="city">Paris</p>
<p class="city main">multiple classes</p>

</body>
</html>
```



HTML ID

- The HTML **id attribute** specifies a unique id for an HTML element (the value must be unique within the HTML document)
- The id value can be used by CSS and JavaScript to perform certain tasks for a unique element with the specified id value.
- To **style** an element with a specific id, write a hash **(#)** character, followed by the id of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
    background-color: lightblue; color: black;
    padding: 40px; text-align: center;
}
</style>
</head>
<body>

<h2>The id Attribute</h2>
<p>Use CSS to style an element with the id "myHeader":</p>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

HTML ID

- Difference Between Class and ID
  - An HTML element can only have one unique id that belongs to that single element
  - A class name can be used by multiple elements

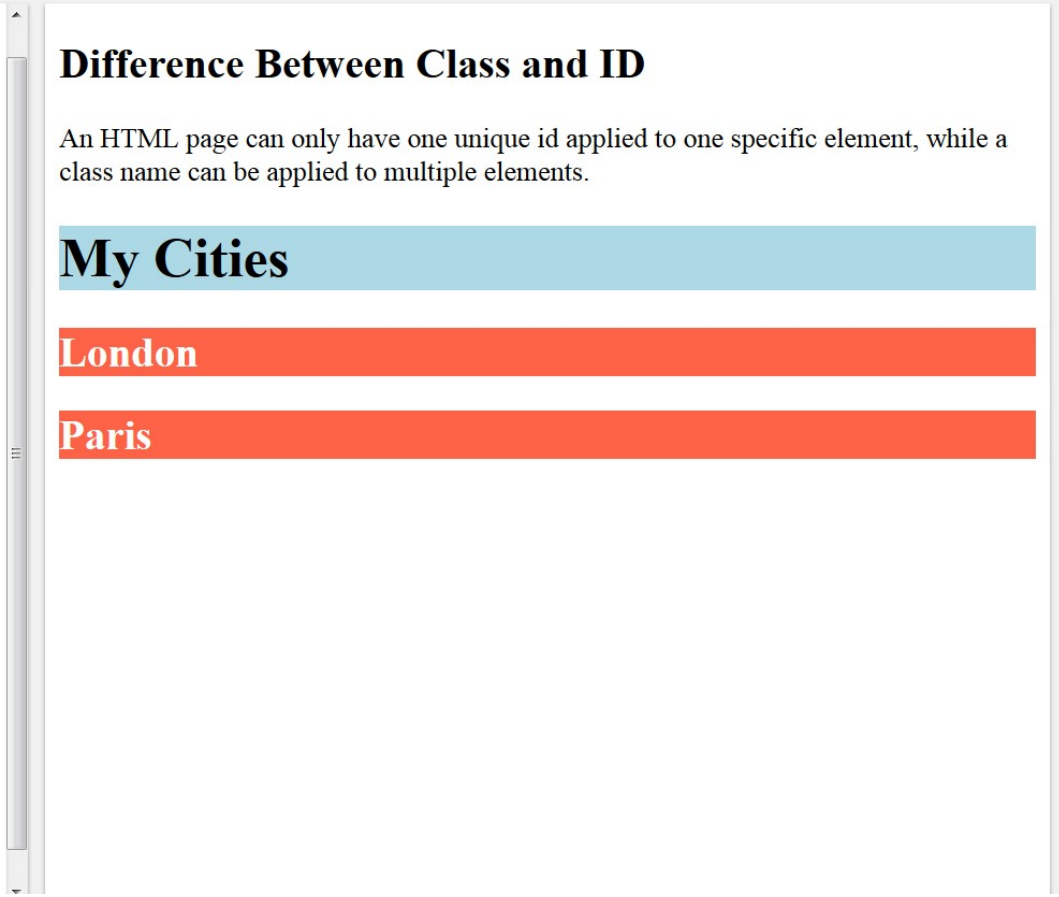
```
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
    background-color: lightblue; color: black;
}
/* Style all elements with the class name "city" */
.city {
    background-color: tomato; color: white;
}
</style>
</head>
<body>

<h2>Difference Between Class and ID</h2>
<p>An HTML page can only have one unique id applied to one
specific element, while a class name can be applied to multiple
elements.</p>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<h2 class="city">Paris</h2>

</body>
</html>
```



CSS Syntax and Selectors

- CSS pseudo-class
  - Style an element when a user mouses over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus
- selector: pseudo-class {property:value;}

```
<html>
<head>
<style>
/* unvisited link */
a:link {color: red;}

/* visited link */
a:visited {color: green;}

/* mouse over link */
a:hover {color: hotpink;}

/* selected link */
a:active {color: blue;}
</style>
</head>
<body>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in
order to be effective.</p>

</body>
```

This is a link

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

CSS Syntax and Selectors

- CSS pseudo-element
  - Style specified parts of an element: the first letter, or line, of an element
- selector: pseudo-element {property:value;}

```
<html>
<head>
<style>
p::first-letter {
    color: #ff0000;
    font-size: xx-large;
}
</style>
</head>
<body>

<p>You can use the ::first-letter pseudo-element to
add a special effect to the first character of a
text!</p>

</body>
</html>
```

You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

CSS Syntax and Selectors

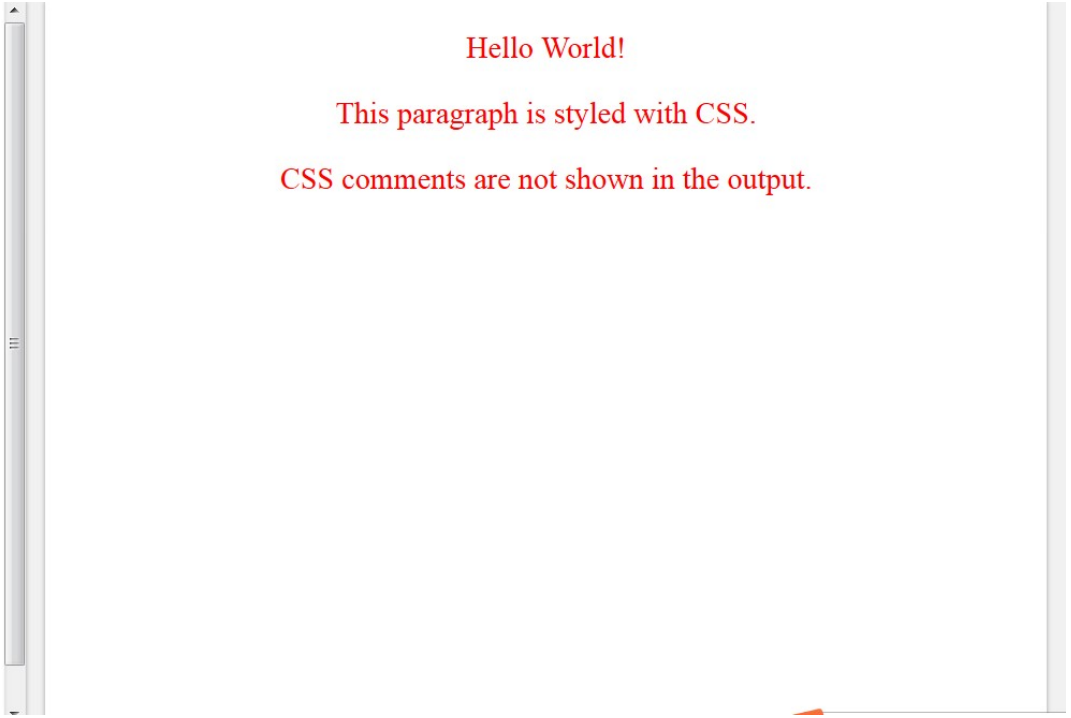
- A CSS comment starts with /* and ends with */. Comments can also span multiple lines.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
</style>
</head>
<body>

<p>Hello World!</p>

</body>
</html>
```



CSS Topics

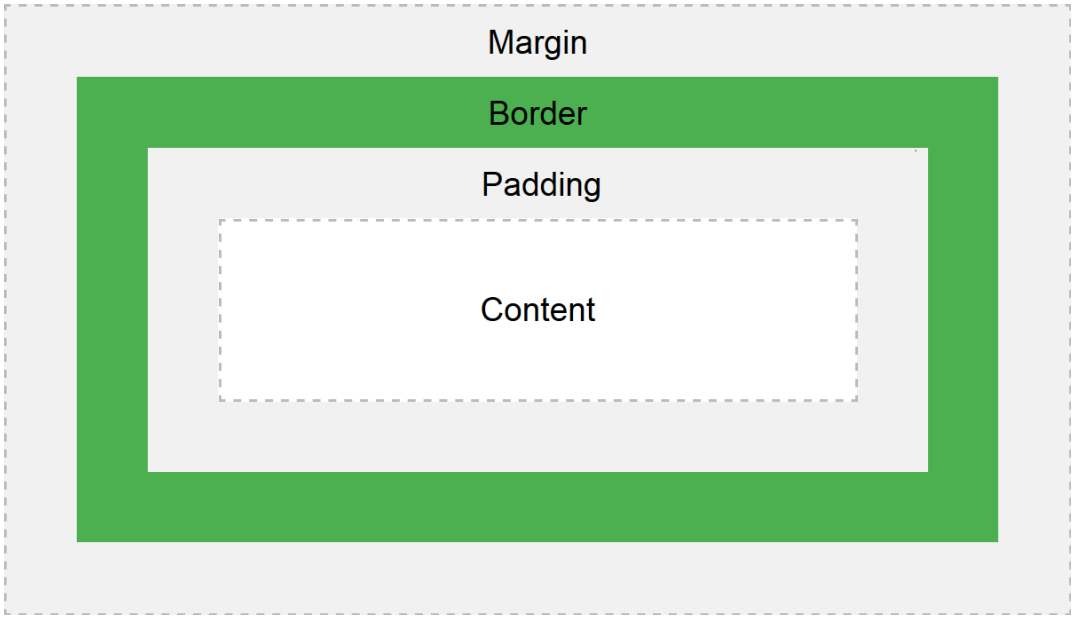
- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

CSS Box Model

- All HTML elements can be considered as boxes

- The CSS box model is essentially a box that wraps around every HTML element. It consists of: **margins**, **borders**, **padding**, and the actual **content**.

CSS Box Model



- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is **transparent**
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is **transparent**

CSS Box Model

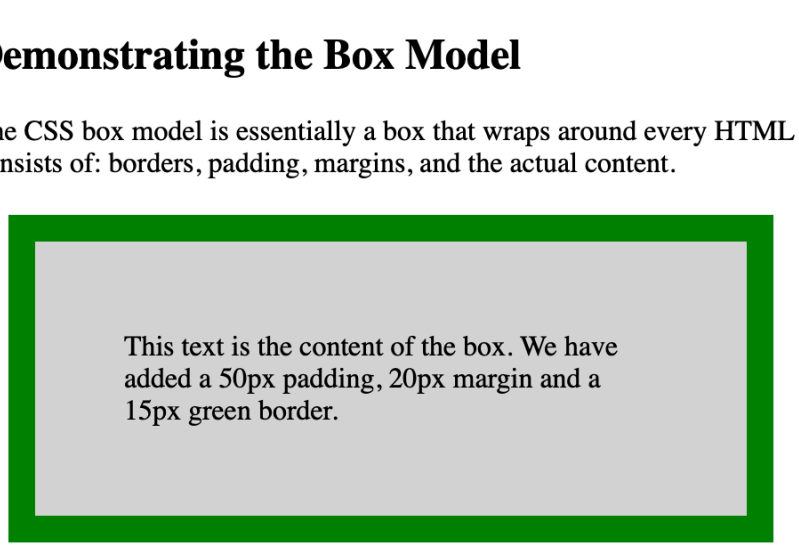
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around
every HTML element. It consists of: borders, padding, margins,
and the actual content.</p>

<div>This text is the content of the box. We have added a 50px
padding, 20px margin and a 15px green border.</div>

</body>
</html>
```

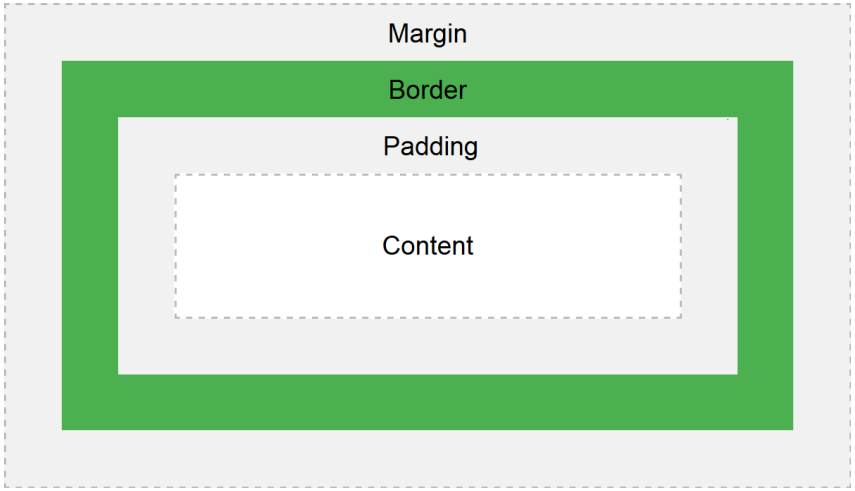


CSS Box Model

- Width and Height of an Element
  - When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**.
  - To calculate the full size of an element, you must also add padding, borders and margins.

```
div {
    width: 320px;
    padding: 10px;
    border: 5px
    margin: 0;
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px



Queen Mary University of London

CSS Topics

- Style & Rule
- Inline, Internal & External Style Sheets
- Class, ID, Pseudo-Class, Pseudo-Element
- CSS Box Model
- CSS Property, units

CSS Property

https://www.w3schools.com/css/

Color
Background
Border
Margin
Padding
Height/width, max-width
Text
Fonts
Links
Lists
Table
Position
Alignment
Float

Colours & Borders	
`color: red;`	Element Colour - eg. red | #FF0000
`background-color: white;`	Background Colour of element
`background-image: url(image.gif);`	Background Colour of element
`border-color: yellow;`	Border Colour of element
`border: 1px solid blue;`	Width, style and colour of border defined together

Text Styles	
`text-align: left;`	Horizontal Alignment - left | center | right
`text-decoration: underline;`	Text Decorations - eg. none | underline | line-through
`font-family: fontname;`	Font Face (Typeface) - eg. Verdana, Arial, Helvetica
`font-size: 16pt;`	Font Size or Height - eg. 12pt | 15px
`font-weight: bold;`	Font Weight (Boldness) - eg. bold | normal | 200

Size and Layout	
`width: 400px;`	Width of HTML element - eg. 100px | 50%
`height: 100%;`	Height of HTML element - eg. 20px | 100%
`margin: 5px;`	Margin - space around an element, or distance between two elements
`margin-top: 1px;`	Top Margin. Also try -bottom: -left: or -right:
`padding: 5px;`	Padding - distance between an elements contents and its border
`padding-top: 1px;`	Top Padding. Also try -bottom: -left: or -right:

CSS Units

- The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.
- Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

CSS Units

- Relative length units specify a length relative to another length property.
- Relative length units scales better between different rendering mediums.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

Special thanks to:

1. W3Schools - https://www.w3schools.com/css/

2. Wang, L. ECS507U – Website Design and Authoring

3. Naeem, U. ECS417U – Fundamentals of Web Technology

4. Tutorialspoint - https://www.tutorialspoint.com/css/index.htm

5. Mai Moustafa, HTML/CSS - https://www.slideshare.net/maimoustafa735/html-css-basics-30843378