Cascading Style Sheets (CSS)

Dr Jon Bird

CSS

CSS stands for **C**ascading **S**tyle **S**heets.

CSS describes how HTML elements are to be displayed on screen.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

Inline - by using the style attribute in HTML elements

Internal - by using a <style> element in the <head> section

External - by linking to a CSS file – this is the method that we recommend

Structure and Styling

Use HTML for structure and CSS for styling (includes layout, appearance, some behaviours).

You can customise anything. For example, if you want emphasised words to be underlined, you can use an tag and style it to be displayed underlined using CSS.

Linking to a stylesheet

<link rel="stylesheet" href="styles.css">

This goes in the head tag of a web page.

External style sheets have several advantages:

One style sheet can control many pages**:** Generally, you have a large number of different pages in a website that all share the same general style. You can define the style sheet in one document and have all the HTML files refer to the same CSS file.

Global changes are easier**:** If you're using external styles, you make a change in one place and it's automatically propagated to all the pages in the system.

Separation of content and design**:** With external CSS, all the design is housed in the CSS, and the data is in HTML.

An absolute URL - points to another web site (like href="http://www.example.com/theme.css")

A relative URL - points to a file within a web site (like href="/themes/theme.css")

CSS

selector { key: value; ... }

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

.lecture {
  font-weight: bold;
}

#currentDate{
  color: blue;
}
```

tag { ... }
.class { ... }
#id { ... }

Descendant selector (space)


```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that are descendants of a specified element.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

The descendant selector matches all elements that are descendants of a specified element. This example selects all <p> elements *inside* elements.

Child selector >


```
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Child Selector</h2>
<p>The child selector (>) selects all elements that are the children of a specified element.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section> <!-- not Child but Descendant -->
  <p>Paragraph 4 in the div.</p>
</div>

<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>

</body>
</html>
```

Child Selector

The child selector (>) selects all elements that are the children of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4 in the div.

Paragraph 5. Not in a div.

Paragraph 6. Not in a div.

The child selector selects all elements that are the children of a specified element.
This example selects all <p> elements that are children of a element.

Adjacent sibling selector +


```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Adjacent Sibling Selector</h2>

<p>The + selector is used to select an element that is directly after another specific element.</p>
<p>The following example selects the first p element that are placed immediately after div elements:</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>

<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>

<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>

</body>
</html>
```

Adjacent Sibling Selector

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are place immediately after div elements:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

Paragraph 8. After a div.

The adjacent sibling selector is used to select an element that is *directly after* another specific element. Sibling elements must have the same parent element, and "adjacent" means "immediately following".

This example selects the first <p> elements that are placed immediately after elements.

General sibling selector ~

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>General Sibling Selector</h2>
<p>The general sibling selector (~) selects all elements that are siblings of a specified element.</p>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
</html>
```

General Sibling Selector

The ~ selector is used to select all elements that are siblings of a specified element.

Paragraph 1.

Paragraph 2.

Paragraph 3.

`Some code.`

Paragraph 4.

The general sibling selector selects all elements that are siblings of a specified element i.e. come after it.
This example selects all <p> elements that are siblings of elements.

Additional selector types

Selector	Meaning	Example	
Descendant selector	Matches all descendants of an element	p a { }	Select <a> elements inside <p> elements
Child selector	Matches a direct child of an element	h1>a { }	Select <a> elements that are directly contained by <h1> elements.
First child selector	Matches the first child of an element	h1:first-child { }	Select the the elements that are the first child of a <h1> element.
Adjacent selector	Matches selector	h1+p { }	Selects the first <p> element after any <h1> element
Negation selector	Selects all elements that are not selected.	body *:not(p)	Select all elements in the body that are not <p> elements.
Attribute selector	Selects all elements that define a specific attribute.	input[invalid]	Select all <input> elements that have the invalid attribute.
Equality attribute selector	Select all elements with a specific attribute value	p[class="invisible"]	Select all <p> elements that have the invisible class.

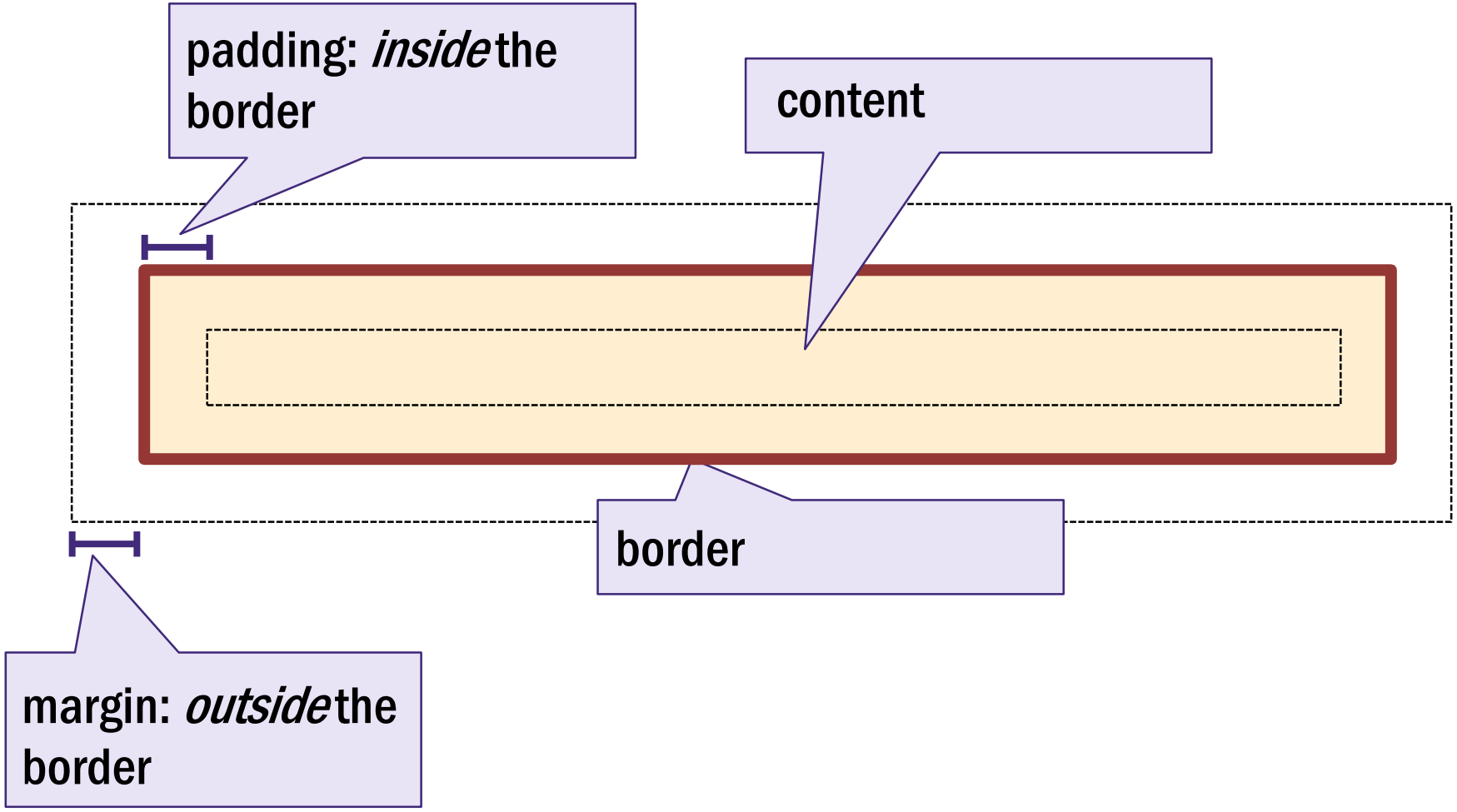
Cascading selectors

What happens if more than one rule applies?

The most *specific* rule takes precedence, otherwise the *last* rule wins

```
em { font-style: normal; }
em.room { font-style: italic; }
```

Box model


padding: *inside* the border
content
border
margin: *outside* the border

Visibility and layout

- Can force elements to be inline or block element.
  - display: inline
  - display: block
- Can cause element to not be laid out or take up any space
  - display: none
  - Useful for content that is dynamically added and removed.
- Can cause elements to be invisible, but still take up space
  - visibility: hidden;

```
<ul>
    <li>Home</li>
    <li>Products</li>
    <li class="coming-soon">Services</li>
    <li>About</li>
    <li>Contact</li>
</ul>
```

```
li {
    display: inline;
    margin-right: 10px; }
li.coming-soon {
    display: none; }
```

Home Products About Contact

```
li {
    display: inline;
    margin-right: 10px; }
li.coming-soon {
    visibility: hidden; }
```

Home Products About Contact

Colours

color: (US spelling) content, i.e. text

background-color, border-color: what it says

full colour (RGB hex): `#ffc0c0`

abbreviation: `#fcc` => `#ffffcc`

named colours: `CornflowerBlue` etc. (x 140)

`rgba(255, 128, 128, 0.9)` or

`hsl(120, 100%, 50%)`

Use accessible colours

This is a great resource:

http://colorbrewer2.org

It enables you to select colour schemes that are colour blind compatible.

Reading

www.w3schools.com

This is a great resource for information on a wide range of web technologies, including HTML5 and CSS