Cascading Style Sheets (CSS)

- W3C standard for defining presentation of web documents (way documents are displayed or delivered to users – typography, colours, layout etc)
- Presentation separated from content
- Presentational elements controlled by author rather than browser
- W3C recommendations
  - CSS1
  - CSS2 (incorporates and extents CSS1)

Benefits of CSS

- More scope in presentation controls
- Less work (particularly if using external style sheets)
- Potentially smaller documents
- Potentially more accessible documents
- Presentation HTML is being used less and less
- CSS well supported by browsers

How CSS works

- Start with HTML or XHTML document – referred to as *structural layer* upon which *presentation layer* is applied
- Write rules for how element should look – rule targets an element and lists properties to be applied
- Attach styles to document

Rule Syntax

```
selector{property:value;}
```

declaration

- `selector` defines element to be styled
- Declaration – style or display instructions
- Example – styling paragraph selector:

```
p{font-size: 11px; font-weight: bold;}
```

Adding Style to a Document

- **Inline Style** – to style an individual element within the HTML tag for that element
- Example:

```
<h1 style="color: red"> This is a red
  heading</h1>
```

- Note: style is tied to individual content elements
- Note use of quotes

Adding Style to a Document

- **Embedded Style** – style embedded in a block at top of HTML document, within `<head>` tags
- Syntax: `<style> …..</style>`
- Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"HTTP://WWW.W3.ORG/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns=http://www.w3.org/1999/xhtm sml:land="en"
   lang="en">
<head>
<style type="text/css">
   h1{color:#666;}
   p{font-size=90%; font-family: Verdana, sans-serif;}
</style>
<title>Style Sheets</title>
</head>
….
```

- Styles all instances of `h1` and `p` in the document

Adding Style to a Document

- **External Style Sheets** – style rules in separate document. Pages link to this document
- External style sheet is text file with suffix `.css`
- Style for whole web site altered by just altering external style sheet

External Style Sheets

- Style sheet document must contain at least one style rule.
- Contains no HTML tags
- Comments:

```
/* This is a comment*/
```

Linking to external style sheets

- Link:

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

`rel` defines linked document's relation to current document

`href` provides address for stylesheet

Note: more than one stylesheet can be linked, all stylesheets will apply

Linking to external style sheets

- **Importing:**

```
<style type="text.css">
<!--
@import url(http://pathname/stylesheet.css);
p{font-face: Verdana;}
-->
</style>
```

Note: multiple style sheets can be applied. With multiple `@import` functions, last file read takes precedence.

`@import` may be used within a style sheet itself.

Placing `@import` within HTML comments means it will be ignored by browsers not supporting CSS.

Conflicting Rule Styles – The Cascade

- Elements may get presentation instructions from several sources – conflict resolved with cascading rules – rules passed down until they are overridden by more important style rules
- The Cascade
  - Style sheet origin
  - Selector specificity
  - Rule order

Style Sheet Origin – ascending order of importance

1. User agent style sheets – built into browser
2. Reader style sheets (user-created rules)
3. Author style sheets
4. Reader `!important` style declarations

Author Style Sheets – ascending order of importance

1. Linked external sheets
2. Imported style sheets (using `@import`)
3. Embedded style sheets (using `<stlye>`)
4. Inline styles (using `<style>` in an element tag)
5. Style declarations marked `!important`

Selector Specificity – selector types in ascending order of importance

1. Individual element (e.g. `p`)
2. Contextual selector (e.g. `h1 strong`)
3. Class selectors (e.g. `p.special`)
4. ID selectors (e.g. `p#intro`)

Rule order

- If several rules are applied to same element in a single style sheet, last rule is most important
- For example:

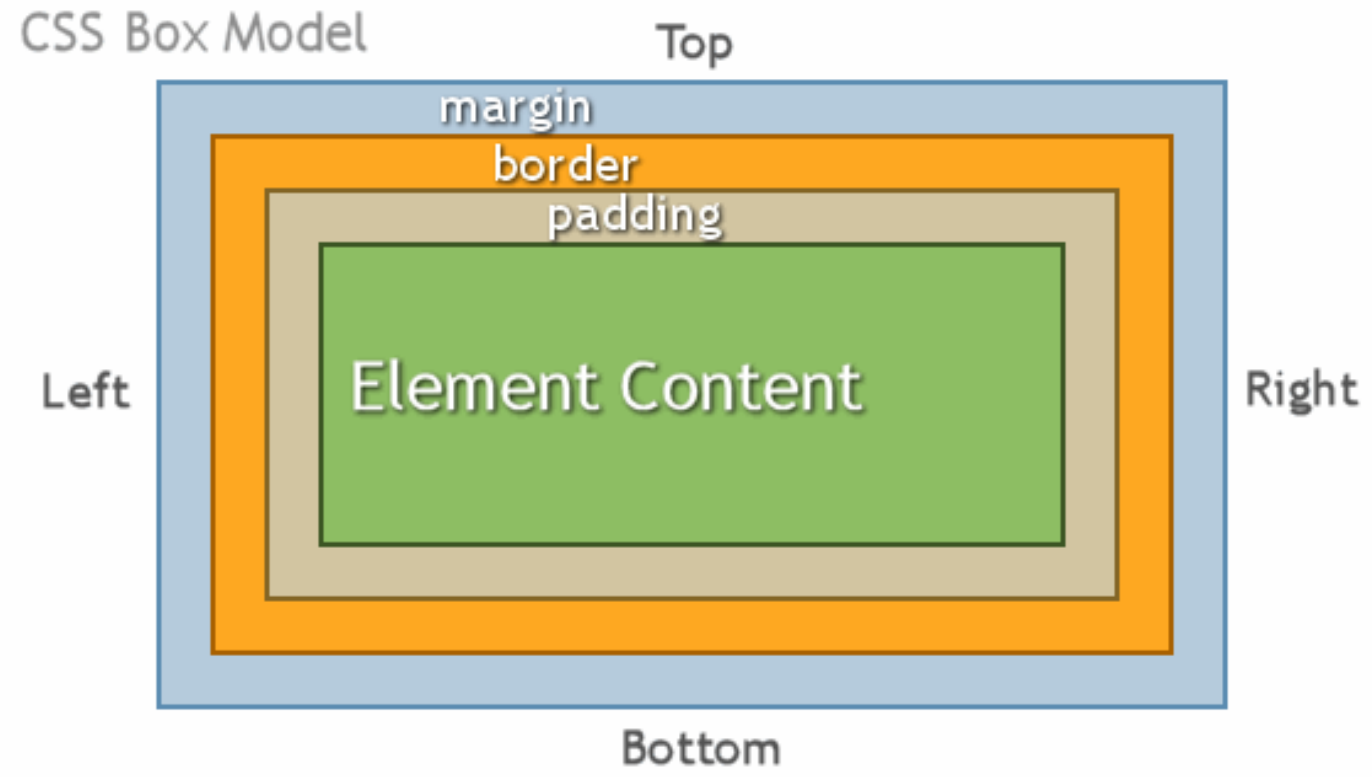
```
h1{color: green;}
h1{color: blue;}
h1{color: red;}
```

Red is colour applied to `h1` headings

The CSS Box Model

- Every CSS element generates rectangular *element box* around itself
- Properties such as borders, margins, background etc. applied to the box
- Boxes used to position elements and lay out page
- Components of element box:
  - *Content*
  - *Padding*
  - *Border*
  - *Margin*

The CSS Box Model



The CSS Box Model

- Padding, borders and margins are optional – if values = 0, they are removed
- Any background color/image extends into padding area
- Borders generated by properties like width, color, style
- Margins always transparent – background color/image shows through
- Width of element is normally content width (note differences between browsers)
- Top/bottom/sides of element box can have differing styles

CSS Selectors

- Type selectors
- Contextual selectors
- Class and ID selectors
- Attribute selectors (not well supported)
- Pseudoclasses
- Pseudoelements

Type (Element) Selector

- Targets an element by name – applies to all instances of that element. Examples:

```
h1{color:blue;}
p{color: red;}
```

- Selectors can be grouped

```
h1, h2, p {color:blue;}
```

Contextual Selectors

- Based on relation of element to another element – e.g. descendent, adjacent sibling
- *Descendent elements* target elements contained in (descended from) another element. Example

```
li em{color: green;}
```

means `em` (emphasised text) is green but only when contained in `li`

- *Adjacent sibling selector* targets elements descended from same parent. Example

```
h1 + p{padding-left: 40}
```

targets first paragraph after a `h1` heading.

Class Selectors

- Target elements you have set up yourself
- Class attribute used to identify a number of elements as part of a group – modify all elements in class in one style rule
- Class element defined in <head> or in external CSS document. To use the class you invoke it in the HTML tag in <body>

Class Selectors - Example

- Identify class “headerfont” in the body:

```
<h1 class=“headerfont”>This is the header</h1>
```

- Code inserted into <head> or external .css file:

```
h1.headerfont{font-size: 15px; font-family:Arial;
   color:red;}
```

Note: period in front of selector signals we are defining a class style

Property can be applied to all elements in a class by omitting the tag name

ID Selector

- Similar to Class Selector, but can only have a single instance of an ID on a page, but may use same ID on several pages on same site
- ID must uniquely name an element
- ID's declared in <head> or external .css file
- Example to invoke an ID:

```
<p id="j0428">New item added today</p>
```

Declaration in `<head>`:

```
p#j0428{color:red}
```

Pseudoselectors

- *Pseudoclasses* – work like classes, usually for anchors, based on a state, for example:

```
a:link {color:red;}
a:visited{color:blue}
```

apply to links, visited links.

Pseudoelements

- Like inserting fictional elements into document structure for styling. For example:

```
:first-line

p:first-line{color: blue; font-size: 20px;}
```

- Other pseudoelements
  - `:first-letter`
  - `:before`
  - `:after`