ICT286
Web and Mobile Computing

Topic 2
Cascading Style Sheets

Objectives

- Understand the importance of separating the structural description from the formatting description in web page design.
- Known how to use the three types of style sheets.
- Understand and be able to use simple selectors, contextual selectors, class selectors, generic selectors, id selectors, and pseudo selectors.
- Understand and be able to use the following types of properties and their values: background, color, fonts, alignment, margin, border, and list.
- Understand how browsers resolve conflicting rules.
- Be able to design your style sheets.

Structure vs Presentation

- The recommended way to design web pages is to separate the structure of the web page from formatting. We describe the structure of a document in HTML, and formatting with CSS.
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents.
- CSS should replace the deprecated HTML tags.

CSS Standards

- Three CSS standards (W3C Recommendations):
  - CSS Level 1 (CSS1): December 1996, revised in April 2008.
  - CSS Level 2 (CSS2): May 1998
  - CSS Level 2, Revision 1 (CSS2.1): published as a W3C Recommendation on 7 June 2011
  - CSS Level 3 (CSS3) builds on Level 2 module by module using CSS2.1 as its core. So far, only four modules are published as W3C Recommendations. Other modules are either in working draft status or in candidate recommendation.

Browser Suport

- Some very older browsers do not support all the features that CSS1 provides.
- CSS1 is fully implemented by IE7 and Firefox2, and Safari.
- CSS2.1 are mostly implemented in major layout engines (Trident, Gecko and Webkit).
- CSS3 has less support at this moment compared to CSS2.

Types of Style Sheets

- There are three types of style sheets
  - Inline - specified for a specific occurrence of a tag and apply only to that element. Not recommended.
  - Internal style sheets - apply to the whole document in which they appear
  - External style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific element in a document, the lowest level style sheet has precedence
  - In a sense, the browser searches for a style property spec, starting with inline, until it finds one (or there isn't one)

Inline Style Sheet

- Inline style sheets appear in the tag itself
- The style sheet appears as the value of the **`style`** attribute in the tag, in the following form:

```
style = "property_1: value_1; property_2: value_2; …
  property_n: value_n;”
```

- Example:

```
<p style="font-style: italic; color: red;">
     This line is displayed in red.
</p>
```

Internal Style Sheet

- Internal style sheets appear in the head element of the document
- Style sheet appears as a list of rules that are the content of a `<style>` element
- Comments in the rule list must have a different form - use C comments (`/*…*/`)
- Example:

```
<head>
    <title> Internal Style Sheet </title>
    <style type="text/css" >
        p { font-style: italic; color: red; }
    </style>
</head>
<body>
        <p>This paragraph is in red. </p>
</body>
```

External Style Sheet

- External style sheets are in separate files, potentially on any server on the Internet
  - Written as text files with the MIME type `text/css`
- A `<link>` tag inside head element is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = "stylesheet"  type = "text/css"
 href = "http://it.www.murdoch.edu.au/~s900432d/css/unit_website.css">
</link>
```

- If the stylesheet is on the same machine, you may use the path of the file to refer to it, as in:

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

External Style Sheet

- External style sheets can be validated

  http://jigsaw.w3.org/css-validator/validator.html#validate-by-upload

- The above link is also available from Unit Resources page of the Unit LMS.

Style Rules

- The content of internal and external style sheets is a list of rules. Each rule is in the form of

```
selector {
   property1: value;
   property2: value;
   . . . . . .
   propertyn: value;
}
```

- Example: display all paragraphs in bold fonts and in blue color.

```
p {
      font-weight: bold;
      color: blue;
}
```

Comments in CSS

- You can have comments in style sheets. They will be ignored by the browser.
- The comments are written in the style of C language, ie, between /* and */ and can be cross lines:

```
/* this is a comment about CSS */
```

Simple Selectors

- The selector is a tag name or a list of tag names, separated by commas
  - `p`
  - `h2, h3`
- Example: the following rule applies to both h2 and h3 elements:

```
h2, h3 {
  color: red;
  text-align: center;
}
```

Contextual Selectors

- Contextual selector :
  - Example:

```
ol p {
  margin-left: 0ex;
  text-align: left;
}
```

  - The above rule only applies to those p elements within an ol element

Class Selectors

- Allow different occurrences of the same tag to use different style specifications
- A style class has a name, which is attached to a tag name

```
p.English {font-weight: bold; color: red; }
p.French {font-style: italic; color: green; }
```

- The class you want on a particular occurrence of a tag is specified with the class attribute of the tag
- For example,

```
<p class = ”English”> Hello </p>
<p class = ”French”> Bonjour </p>
```

Generic Selectors

- A generic class can be defined if you want a style to apply to more than one kind of tags
- A generic class must be named, and the name must begin with a period
- Example,

```
.big { font-size: 200%; }
```

- Use it as if it were a normal style class

```
<h1 class = “big”> Big Heading </h1>
<p class = “big”> Doubling the font size! </p>
```

id Selectors

- An `id` selector allow the application of a style to one specific element
- General form:

  #specific-id {property-value list}

- Example:

```
#section14 {font-size: 20pt;}
<h2 id="section14"> id Selector</h2>
```

Pseudo Classes

- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists
- Names begin with colons
- `hover` classes apply when the mouse cursor is over the element
- `focus` classes apply when an element has focus

Pseudo Class Example

```
<!DOCTYPE html>
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Checkboxes </title>
    <style type = "text/css">
      input:hover {color: red;}
      input:focus {color: green;}
    </style>
  </head>
  <body>
      Your name:
        <input type = “text” />
       <br />
      Your student number:
        <input type = “number” />
  </body>
</html>
```

Properties

- There are 60 different properties in 7 categories:
  - Backgrounds
  - Colors
  - Fonts
  - Alignment of text
  - Lists
  - Margins
  - Borders

Property Values

- Keywords - left, small, …
  - Not case sensitive
- Length - numbers, maybe with decimal points
- Units:
  - px - pixels
  - in - inches
  - cm - centimeters
  - mm - millimeters
  - pt - points
  - pc - picas (12 points)
  - em – current font size of text
  - ex - height of the letter ‘x’
  - No space is allowed between the number and the unit specification e.g., 1.5 in is illegal!

Property Values (cont'd)

- Percentage - just a number followed immediately by a percent sign
- URL values
  - url(protocol://server/pathname)
- Colors
  - Color name
  - rgb(n1, n2, n3)
    - Numbers can be decimal or percentages
  - Hex form: #XXYYZZ
- Property values are inherited by all nested tags, unless overridden

Background Properties

- `background-color:`
  - color value such as `red`, `#FF005C`, `rgb(128, 50, 255)`
  - `Transparent`
- `background-image:`
  - Url value such as in the following example:

```
body { background-image: url("soit.jpeg"); }
```

- `background-repeat:`
  - Possible values: `repeat` (default), `no-repeat`, `repeat-x`, or `repeat-y`
- `background-position:`
  - Possible values: `top`, `center`, `bottom`, `left`, or `right`

Text Colors

- `color`, such as in
  - `p { color: #5F9EA0; }`
- There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors

black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00
maroon	800000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchia	FF00FF	aqua	00FFFF

- There is a much larger set, the Web Safe Colour Chart
  - 216 colors
  - Use hex color values of `00`, `33`, `66`, `99`, `CC`, and `FF`
  - Listed in Appendix B of the textbook
  - See actual color display from: https://htmlcolorcodes.com/color-chart/

Font Properties

- `font-family` property:
  - Value is a list of font names - browser uses the first in the list it has
  - Example:

```
font-family: Arial, Helvetica, Courier
```

- The following generic fonts are defined in CSS. Each browser has a specific font for each them:

```
serif
sans-serif
cursive
fantasy
Monospace
```

- If the font name has more than one word, it should be single quoted.

Font Properties (Cont'd)

- `font-size` property:
  - Possible values: a length number or a name, such as `smaller`, `xx-large`, etc.
- `font-style` property:
  - `italic`, `normal`
- `font-weight` property - degrees of boldness
  - `bolder`, `lighter`, `bold`, `normal`
- `font` property: shorthand for a number of font properties:
  - For specifying a list of font properties

```
font: bolder 14pt Arial Helvetica
```

  - Order must be: style, weight, size, name(s)

Text Decoration Properties

- `text-decoration` property
  - `line-through`
  - `overline`
  - `underline`
  - `blink`
  - `none`
- `letter-spacing` property:
  - `normal`
  - length value

Text Alignment Properties

- The `text-indent` property allows indentation
  - Takes either a length or a % value
- The `text-align` property has the possible values, `left` (the default), `center`, `right`, or `justify`
- Example:

```
p {
    font-family: 'Time New Roman', serif;
    font-size: 14pt;
    text-align: right;
}
```

The Box Model

- Nearly every element has a border with the following properties;
- `border-style:`
  - values: `none`, `dotted`, `dashed`, and `double`
- `border-width`:
  - `thin`, `medium` (default), `thick`, or a length value in pixels
  - Border width can be specified for any of the four borders (e.g., `border-top-width`)
- `border-color:`
  - any color
  - Border color can be specified for any of the four borders (e.g., `border-top-color`)

The Box Model (Cont'd)

- Margin properties – margin is the space between the border of an element and its neighbor element
- The margins around an element can be set with margin property:

```
margin: 10pt;
```

- The top, bottom, left and right margin can be set individually as in

```
margin-left: 10pt;
```

The Box Model (Cont'd)

- Padding properties – padding is the distance between the content of an element and its border
- The padding around an element can be set with padding property:

```
padding: 10pt;
```

- The top, bottom, left and right padding can be set individually as in

```
padding-left: 10pt;
```

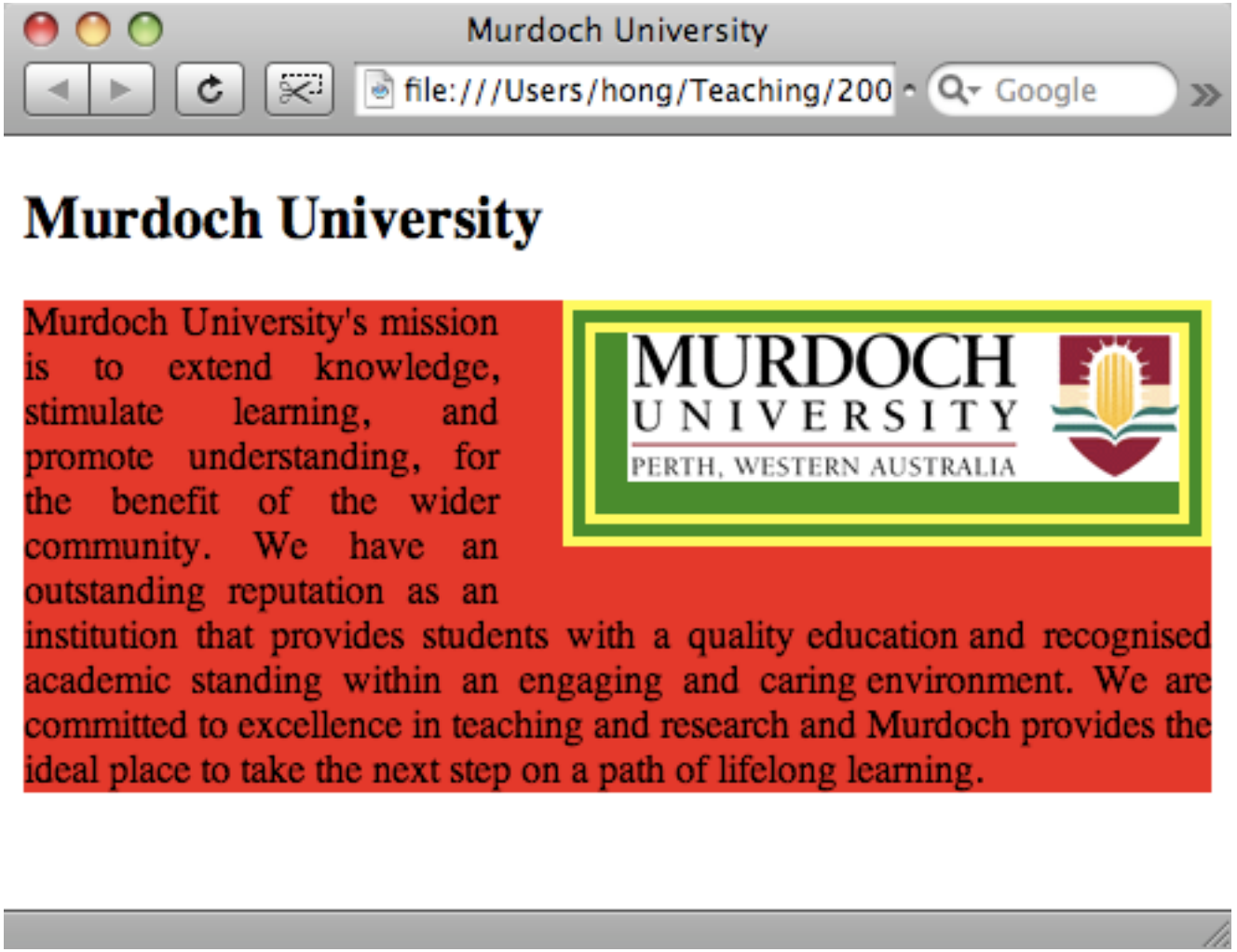
The Box Model - Example

```
<head>
<title> Murdoch University </title>
<style type="text/css">
img.MurdochLogo {
    background-color: green;
    border-style: double; /* none dotted dashed double */
    border-width: 10pt;  /* thin medium thick */
    border-color: yellow;
    padding-left: 10pt;
    padding-bottom: 10pt;
    margin-left: 20pt;
    margin-bottom: 20pt;
    float: right; /* float to the right of the following text and
                     let the following text wrap around the image,
                     note also that img is an inline element,
                     not a block element   */
}
p.Murdoch {
    background-color: red;
    text-align: justify;
}
</style>
</head>
```

The Box Model - Example

```
<body>

<h2> Murdoch University</h2>

<img class="MurdochLogo" src="mu_logo.gif" alt="Murdoch University" />

<p class="Murdoch”>
Murdoch University's mission is to extend knowledge, stimulate
learning, and promote understanding, for the benefit of the
wider community. We have an outstanding reputation as an
institution that provides students with a quality education
and recognised academic standing within an engaging and caring
environment. We are committed to excellence in teaching and
research and Murdoch provides the ideal place to take the next
step on a path of lifelong learning.
</p>

</body>
```

The Box Model - Example



List Properties

- `list-style-type` property can be applied to unordered list (`ul`) and ordered list (`ol`)
- When applied to the unordered lists, bullet can be a `disc` (default), a `square`, or a `circle`
- The property is set on either the `<ul>` or `<li>` tag. When set on `<ul>`, it applies to all list items.
- Example:

```
<h3> Internet Computing Units</h3>
  <ul style = "list-style-type: square">
      <li> ICT286 Web and Mobile Computing</li>
      <li> ICT283 Data Structures and Abstraction</li>
      <li> ICT374 Operating Systems and Systems Programming</li>
  </ul>
```

List Properties (Cont'd)

- When set on <li>, it applies just to that list item.

```
<h3> Internet Computing Units</h3>
<ul style = "list-style-type: square">
    <li style="list-style-type:disc">
        ICT286 Web and Mobile Computing</li>
    <li style="list-style-type:circle">
        ICT283 Data Structures and Abstraction</li>
    <li> ICT374 Operating Systems and Systems Programming</li>
</ul>
```

- Could use an image for the bullets in an unordered list. For example:

```
<li style = "list-style-image: url(bird.jpg)">
```

List Properties (Cont'd)

- *On ordered lists* - `list-style-type` can be used to change the sequence values:

Property value	*Sequence type*	*First four*
`Decimal`	Arabic numerals	1, 2, 3, 4
`upper-alpha`	Uppercase letters	A, B, C, D
`lower-alpha`	Lowercase letters	a, b, c, d
`upper-roman`	Uppercase Roman	I, II, III, IV
`lower-roman`	Lowercase Roman	i, ii, iii, iv

The Element

- In all of previous examples, a style is always applied to the whole element unless it is overwritten by a new style.
- Sometimes, we want to display part of the element content with a different style. This can be achieved with the element.
- The tag is similar to other HTML tags, they can be nested and they have id and class attributes
- Note that the element is an inline element. Hence it does not introduce a line break before and after it and it cannot contain block-level elements such as <p>.

```
<head>
<title> iPhone </title>
<style type="text/css">
   .iPhone {   /* generic selector */
       color: red;
       font-family: Ariel;
       font-style: italic;
       font-size: 150%;
   }
</style>
</head>
```

The Element (Cont'd)

```
<body>
<p>
Introducing <span class="iPhone"> iPhone 3G </span>.  With
  fast 3G wireless technology, GPS mapping, support for
enterprise features like Microsoft Exchange and the new
App Store, <span class="iPhone">iPhone 3G </span> puts even
more features at your fingertips.
</p>
</body>
```

Introducing *iPhone 3G* . With fast 3G wireless technology, GPS mapping, support for enterprise features like Microsoft Exchange and the new App Store, *iPhone 3G* puts even more features at your fingertips.

The Element

- The element defines a new division in the document.
- The new division may contain several block elements. We can use different styles for the division to distinguish it from the surrounding elements.
- Unlike , is a block element. Hence the browser usually places a line break before and after it.

The Element (Cont'd)

```
<body>
<div>
Introducing <span class="iPhone"> iPhone 3G </span>.  With
   <ul>
    <li>fast 3G wireless technology, </li>
    <li> GPS mapping, </li>
    <li> support for enterprise features like Microsoft
         Exchange </li>
   </ul>
and the new App Store,
</div>
<p>
<span class="iPhone">iPhone 3G </span> puts even
more features at your fingertips.
</p>
</body>
```

The Element (Cont'd)

```
<head>
<title> iPhone </title>
<style type="text/css">
    .iPhone {
        color: red;
        font-family: Ariel;
        font-style: italic;
        font-size: 150%;
    }
    div {
        color: green;
        text-align: right;
        font-style: normal;
        font-variant: small-caps;
    }
</style>
</head>
```

Using Multiple Stylesheets

- It is possible to use multiple external stylesheets in the same HTML document. Just include link statements for each one in the <head> element.
- In this case the inheritance works from the top to the bottom. The first stylesheet listed will define styles according to the rules within it. The next stylesheet will inherit all rules from the first, except where a rule is defined in the second stylesheet, in which case the rule is overwritten by the second rule. Similarly for any subsequent stylesheets linked.

Conflict Resolution

- When two or more rules apply to the same tag there are rules for deciding which rule applies
- Internal
  - In-line style sheets have precedence over internal sheets
  - Internal style sheets have precedence over external style sheets
- Within the same level there can be conflicts, eg,
  - A tag may be used twice as a selector
  - A tag may inherit a property and also be used as a selector
- Style sheets can have different sources
  - The author of a document may specify styles
  - The user, through browser settings, may specify styles
- Individual properties can be specified as important such as in

```
body{
    background: url(image.gif) white;
    background-repeat: repeat-x ! important ;
}
```

Precedence Rules

From highest to lowest

1. Important declarations (using !important) with user origin
2. Important declarations with author origin
3. Normal declarations with author origin
4. Normal declarations with user origin
5. Any declarations with browser (or other user agent) origin

Tie-Breakers

- Specificity
  1. id selectors
  2. Class and pseudo-class selectors
  3. Contextual selectors
  4. Simple selectors
- Position
  - Essentially, later has precedence over earlier

Readings

- Sebesta: Chapter 3.
- Useful website for CSS:
  - http://www.w3schools.com/css/
  - In particular, play around the following interactive page to learn more about various types of selectors.
    - http://www.w3schools.com/CSSref/trysel.asp
- Useful book:
  - Mark Myers: A smart way to learn HTML & CSS (amazon kindle book, costs around $10 from amazon.com.au)