Cascading Style Sheets

CITS3403 Agile Web Development
What is CSS?

- CSS stands for *Cascading Style Sheets*
  - stylesheet language for web
  - used to specify the presentation (layout and style) of markup languages
  - can be applied to any XML document (including XHTML)
  - superceded many HTML attributes that mixed presentation with content
Why CSS?

• Separation of content and presentation
• Advantages for the web
  – *Speed* - stylesheet(s) downloaded once, rather than with each page (if content and style information is intermingled)
  – *Maintainability* - can be “centrally” maintained, easier to update
  – *Accessibility* - can make pages appear similar on different browsers and devices
  – *Portability* - eg. printing, porting to new devices
  – *Reduced work* - eg. don’t have to specify alignment every time an element is used
  – *Consistency* - make an organisation’s web pages have consistent “look and feel” - corporate ID, brand (and update as brand updates)
    • eg. UWA...
Science has the power to change our world for the better. UWA's Faculty of Science is harnessing this power to make a positive impact on our planet. Our research and teaching programs focus on solving real-world problems, from climate change to healthcare and more. With our expertise and cutting-edge facilities, we are well-equipped to tackle the challenges of the 21st century.

We are committed to providing high-quality education and research that prepares our students for a successful future. From our undergraduate and postgraduate programs to our research degrees, we offer a range of options to suit your interests and career aspirations.

Whether you are a prospective student, current student, or a researcher, the Faculty of Science has something to offer you. Explore our website to learn more about our courses, research opportunities, and community events.

Join us in shaping the future through science.
Cascading Style Sheets

- **CSS1** introduced styles for the following document features:
  - Fonts
  - Text
  - Color
  - Backgrounds
  - Block-level Elements

- **CSS2** introduced styles for the following document features:
  - Positioning
  - Visual Formatting
  - Media Types
  - Interfaces

- **CSS3** introduces styles for the following document features:
  - User Interfaces
  - Accessibility
  - Columnar layout
  - International Features
  - Mobile Devices
  - Scalable Vector Graphics
CSS3

• Borders
  – border-color
  – border-image
  – border-radius
  – box-shadow

• Color
  – HSL colors
  – HSLA colors
  – opacity
  – RGBA colors

• Text effects
  – text-shadow
  – text-overflow
  – word-wrap

• User-interface
  – box-sizing
  – resize
  – outline
  – nav-top, nav-right, nav-bottom, nav-left

• Selectors
  – attribute selectors

• Basic box model
  – overflow-x, overflow-y

• Generated Content
  – content

• Other modules
  – media queries
  – multi-column layout
  – Web fonts
  – speech
Why “Cascading”?  

• There are three levels of style sheets
  – *Inline* - specified for a specific occurrence of a tag and apply only to that tag
    – This is fine-grain style, which defeats the purpose of style sheets - uniform style
  – *Document-level* 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 tag 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)
Why “Cascading”? 

Levels of Style Sheets

- Inline style sheets appear in the tag itself
- Document-level style sheets appear in the head of the document
- External style sheets are in separate files, potentially on any server on the Internet
  
  - Written as text files with the MIME type `text/css`
In-line Style Specification Format

- Style specification appears as the value of the `style` attribute
  - General form:
    
    ```
    style = "property_1: value_1; property_2: value_2;...
            property_n: value_n"
    ```
  - Example:
    
    ```html
    <p style="background: purple; color: white;">  
        This paragraph will have white text on a purple background.  
    </p>
    ```
Document-level Format

• Style specification appears as a list of rules that are the *content* of a `<style>` tag
• Contained in the document `<head>`
• General form:
  `<style>
      rule list
  </style>

• Form of the rules:

  selector {list of property/values}

  – Each property/value pair has the form:
    property: value

  – Pairs are separated by semicolons, just as in the value of a `<style>` tag
• Example:

```html
<!DOCTYPE html>
<html>
<head>
    <title>My first styled page</title>
    <style>
        body {
            color: purple;
            background-color: #d8da3d
        }
    </style>
</head>
<body>
[etc.]

– from: WC3 CSS Tutorial, http://www.w3.org/Style/Examples/011/firstcss
```
Document-level Format

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

*Made 5 April 2004*

by myself.
A `<link>` tag is used to specify that the browser is to fetch and use an external style sheet file, e.g. Wikipedia style sheet

```
```

- External style sheets can be validated
  
  ```
  http://jigsaw.w3.org/css-validator/
  ```

- Form is a list of style rules

  ```
  selector {list of property/values}
  body {
    color: purple;
    background-color: #d8da3d
  }
  ```

  as in the content of a `<style>` tag for document-level style sheets
Selecting Elements

• There are numerous ways of specifying to which elements style rules apply. Here are examples of some of the more commonly used:

\[
\text{p \{color:red\}} \\
\text{h1,h2,h3 \{\ldots\}} \\
\text{strong \text{em} \{\ldots\}} \\
\text{div[secret=“yes”]} \{\ldots\} \\
\text{span.important \{\ldots\}} \\
\text{p\#1234 \{\ldots\}}
\]

Every p element
Group selector
Contextual selector
Attribute selector
Class selector
ID selector
Selector Forms: Simple

• The selector is a tag name or a list of tag names, separated by commas, eg:
  \texttt{h1, h2 \{font-size: 24pt\}}

• Contextual (or descendant) selectors, eg:
  \texttt{body b em \{font-size: 14pt\}}

<table>
<thead>
<tr>
<th>Selector</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Any element in the hierarchy</td>
</tr>
<tr>
<td>e</td>
<td>The specified element in the hierarchy, where e is the specified element</td>
</tr>
<tr>
<td>e1, e2, e3, ...</td>
<td>The group of elements e1, e2, e3, ...</td>
</tr>
<tr>
<td>e f</td>
<td>The element f when it is a descendant of the element e</td>
</tr>
<tr>
<td>e &gt; f</td>
<td>The element f when it is a direct child of the element e</td>
</tr>
<tr>
<td>e + f</td>
<td>The element f when it is immediately preceded by the sibling element e</td>
</tr>
</tbody>
</table>
<html>
  <head>
    <style type="* {color:red;}
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2.a</h2>
    <p>First paragraph. </p>
    <p>Second paragraph has a <b>bold</b> and a <span>span with another <b>bold</b></span>. </p>
    <h2>Heading 2.b</h2>
    <p>Third paragraph has a <b>bold</b> also. </p>
  </body>
</html>
Any selector example

**Heading 1**

**Heading 2.a**

First paragraph

Second paragraph has a **bold** and a span with another **bold**.

**Heading 2.b**

Third paragraph has a **bold** also.
Example: Selector p

```html
<html>
<head>
    <style>
        p {color:red;}
    </style>
</head>
<body>
    <h1>Heading 1</h1>
    <h2>Heading 2.a</h2>
    <p>First paragraph.</p>
    <p>Second paragraph has a <b>bold</b> and a <span>span with another<b>bold</b></span>.</p>
    <h2>Heading 2.b</h2>
    <p>Third paragraph has a <b>bold</b> also.</p>
</body>
</html>
```
Example: Selector p

Selector p example

Heading 1

Heading 2.a

First paragraph

Second paragraph has a **bold** and a span with another **bold**.

Heading 2.b

Third paragraph has a **bold** also.
<html>
<head>
  <style type="text/css">
    h1, h2, span {color:red;}
  </style>
</head>
<body>
  <h1>Heading 1</h1>
  <h2>Heading 2.a</h2>
  <p>First paragraph. </p>
  <p>Second paragraph has a <b>bold</b> and a <span>span with another <b>bold</b></span>. </p>
  <h2>Heading 2.b</h2>
  <p>Third paragraph has a <b>bold</b> also. </p>
</body>
</html>
Example: Selector h1,h2,span

Selector h1,h2,span example

Heading 1

Heading 2.a

First paragraph.

Second paragraph has a **bold** and a `span` with another **bold**.

Heading 2.b

Third paragraph has a **bold** also.
Example: Selector p b

```html
<html>
<head>
  <style>
    p b {color:red;}
  </style>
</head>
<body>
  <h1>Heading 1</h1>
  <h2>Heading 2.a</h2>
  <p>First paragraph. </p>
  <p>Second paragraph has a <b>bold</b> and a <span>span with another <b>bold</b></span>. </p>
  <h2>Heading 2.b</h2>
  <p>Third paragraph has a <b>bold</b> also. </p>
</body>
</html>
```
Example: Selector p b

Heading 1

Heading 2.a

First paragraph.

Second paragraph has a bold and a span with another bold.

Heading 2.b

Third paragraph has a bold also.
Example: Selector p>b

```html
<html>
<head>
    <style type="text/css">
        p > b {color:red;}
    </style>
</head>
<body>
    <h1>Heading 1</h1>
    <h2>Heading 2.a</h2>
    <p>First paragraph. </p>
    <p>Second paragraph has a <b>bold</b> and a <span>span with another <b>bold</b></span>. </p>
    <h2>Heading 2.b</h2>
    <p>Third paragraph has a <b>bold</b> also. </p>
</body>
</html>
```
Example: Selector p > b

Heading 1

Heading 2.a

First paragraph

Second paragraph has a bold and a span with another bold.

Heading 2.b

Third paragraph has a bold also.
Example: Selector h2+p

```html
<html>
  <head>
    <style type>
      h2+p {color:red;}
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2.a</h2>
    <p>First paragraph. </p>
    <p>Second paragraph has a <b>bold</b> and a <span>span with another <b>bold</b></span>. </p>
    <h2>Heading 2.b</h2>
    <p>Third paragraph has a <b>bold</b> also. </p>
  </body>
</html>
```
Example: Selector h2+p

```html
<body>
  <h1>Heading 1</h1>
  <h2>Heading 2.a</h2>
  <p>First paragraph</p>
  <p>Second paragraph has a <strong>bold</strong> and a span with another <strong>bold</strong>.</p>
  <h2>Heading 2.b</h2>
  <p>Third paragraph has a <strong>bold</strong> also.</p>
</body>
```
Class Selectors

- Used to 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

```html
p.narrow {property/value list}
p.wide  {property/value list}
```

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

```html
<p class = "narrow"> ...
  ...
</p>

<p class = "wide"> ...
  ...
</p>
```
Generic Selectors

• A *generic class* can be defined if you want a style to apply to more than one kind of tag
• A generic class name must begin with a period
• Example,

  ```
  .really-big {font-size: 60pt; ...}
  ```

• Use it as if it were a normal style class

  ```html
  <h1 class = "really-big"> ... </h1>
  ...
  <p class = "really-big"> ... </p>
  ```
id Selectors

• An id selector allows the application of a style to one specific element
• General form:
  #specific-id {property/value list}
• Example:
  #breadcrumbs {
    top: 60px; width: 100%; height: 23px;
    text-indent: 15px; padding-top: 1px;
    color: white;
  }

  <p id="breadcrumbs">
    <a href="http://web.csse.uwa.edu.au/">School Home</a> | 
    <a href="http://web.csse.uwa.edu.au/current/">Current Students</a> | 
    <a href="http://undergraduate.csse.uwa.edu.au/units/CITS4230/">Internet Technologies</a>
  </p>
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

```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>
<form action = "">
  <p> Your name:   <input type = "text" />
  </p>
</form>
</body>
</html>
```
Conflict Resolution

• When two or more rules apply to the same tag there are rules for deciding which rule applies

• Document level
  – In-line style sheets have precedence over document style sheets
  – Document style sheets have precedence over external style sheets

• Within the same level there can be conflicts
  – 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
Precedence Rules

- From highest to lowest
  1. Important declarations with user origin ({key: value !important;})
  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. General selectors
- Position
  - Essentially, later has precedence over earlier
CSS Properties

- CSS1 included 60 different properties in 7 categories:
  - Fonts
  - Lists
  - Alignment of text
  - Margins
  - Colors
  - Backgrounds
  - Borders

CSS Property Groups

- Color
- Background and Borders
- Basic Box
- Flexible Box
- Text
- Text Decoration
- Fonts
- Writing Modes
- Table
- Lists and Counters
- Animation
- Transform
- Transition
- Basic User Interface
- Multi-column
- Paged Media
- Generated Content
- Filter Effects
- Image/Replaced Content
- Masking
- Speech
- Marquee
Font Properties

- font-size
  - Possible values: a length number or a name, such as smaller, xx-large, etc.
- font-style
  - italic, oblique (useless), normal
- font-weight - degrees of boldness
  - bolder, lighter, bold, normal
  - Could specify as a multiple of 100 (100 – 900)
- font
  - For specifying a list of font properties
  - font: bolder 14pt Arial Helvetica
  - Order must be: style, weight, size, name(s)
- Examples: fonts.html, fonts2.html
- The text-decoration property
  - line-through, overline, underline, none
  - letter-spacing – value is any length property value
List properties

- `list-style-type`
- **Unordered lists**
  - Bullet can be a disc (default), a square, or a circle
  - Set it on either the `<ul>` or `<li>` tag
    - On `<ul>`, it applies to list items...

<h3> Some Common Single-Engine Aircraft </h3>
<ul style = "list-style-type: square">
  <li> Cessna Skyhawk </li>
  <li> Beechcraft Bonanza </li>
  <li> Piper Cherokee </li>
</ul>

On `<li>`, `list-style-type` applies to just that item

```html
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
  <li style = "list-style-type: disc"> Cessna Skyhawk </li>
  <li style = "list-style-type: square"> Beechcraft Bonanza </li>
  <li style = "list-style-type: circle"> Piper Cherokee </li>
</ul>
```
Colors

- Color is a problem for the Web for two reasons:
  1. Monitors vary widely
  2. Browsers vary widely

- The `color` property specifies the foreground colour of elements
  ```html
  <style type = "text/css">
    th.red {color: red}
    th.orange {color: orange}
  </style>
  ...
  <table border = "5">
    <tr>
      <th class = "red"> Apple </th>
      <th class = "orange"> Orange </th>
      <th class = "orange"> Screwdriver </th>
    </tr>
  </table>
  ```

- The `background-color` property specifies the background color of elements
Colors

• There are three color collections
  1. There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors

<table>
<thead>
<tr>
<th>Name</th>
<th>Hexadecimal Code</th>
<th>Name</th>
<th>Hexadecimal Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>000000</td>
<td>green</td>
<td>008000</td>
</tr>
<tr>
<td>silver</td>
<td>COC0C0</td>
<td>lime</td>
<td>00FF00</td>
</tr>
<tr>
<td>gray</td>
<td>808080</td>
<td>olive</td>
<td>808000</td>
</tr>
<tr>
<td>white</td>
<td>FFFFFF</td>
<td>yellow</td>
<td>FFFFF0</td>
</tr>
<tr>
<td>maroon</td>
<td>800000</td>
<td>navy</td>
<td>000080</td>
</tr>
<tr>
<td>red</td>
<td>FF0000</td>
<td>blue</td>
<td>0000FF</td>
</tr>
<tr>
<td>purple</td>
<td>800080</td>
<td>teal</td>
<td>008080</td>
</tr>
<tr>
<td>fuchsia</td>
<td>FF00FF</td>
<td>aqua</td>
<td>00FFFF</td>
</tr>
</tbody>
</table>

2. There is a much larger set, the Web Palette

3. Any one of 16 million different colors
   • #000000, #000001, #000002, . . . , #FFFFFF, #FFFFF

Some material Copyright © 2013 Pearson Education, Inc. Publishing as Pearson Addison-Wesley.
Alignment of Text

- 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`
- Sometimes we want text to flow around another element - the `float` property
  - The `float` property has the possible values, `left`, `right`, and `none` (the default)
  - If we have an element we want on the right, with text flowing on its left, we use the default `text-align` value (`left`) for the text and the `right` value for `float` on the element we want on the right

```html
< img src = "c210.jpg" style = "float: right" />
```
- Some text with the default alignment - `left`

Some material Copyright © 2013 Pearson Education, Inc. Publishing as Pearson Addison-Wesley.
Working with the Box Model

- The **box model** is an element composed of four sections:
  - Margin
  - Border
  - Padding
  - Content

- Borders – every element has a `border-style` property
  - Controls whether the element has a border and if so, the style of the border
  - `border-style` values: none, dotted, dashed, and double
  - `border-width`: thin, medium (default), thick, or a length value in pixels
  - `border-color`: any color
  - Example: [borders.html](borders.html)
The Box Model

• Margin – the space between the border of an element and its neighbor element
• The margins around an element can be set with margin-left, etc. - just assign them a length value

```html
<img src = "c210.jpg " style = "float: right; 
margin-left: 0.35in; 
margin-bottom: 0.35in" />

• Padding – the distance between the content of an element and its border
  – Controlled by padding, padding-left, etc.
→ Example: marpads.html
• The background-image property
→ Can also specify background-image property
  – Repetition can be controlled
    • background-repeat property
    • Possible values: repeat (default), no-repeat, repeat-x, or repeat-y
```
The `<span>` and `<div>` tags

- One problem with the font properties is that they apply to whole elements, which are often too large
  - Solution: a tag to define an element within a larger element - `<span>`

- Use `<span>` to apply a document style sheet to its content

  ```html
  <style type = "text/css">
    bigred {font-size: 24pt; font-family: Ariel; color: red}
  </style>

  <p>
    Now is the
    <span class = "bigred"> best time </span> ever!
  </p>
  ```
Positioning

- **Normal Flow** - block formatting of block boxes, inline formatting of inline boxes, relative positioning of block or inline boxes
- **Floats** - laid out according to normal flow, then shifted
- **Absolute positioning** - box is removed entirely from normal flow
- Values: static, relative, absolute, fixed
- Offsets: top, right, left, bottom
- Each CSS box is laid out on the screen (or page) in one of the three ways: in its normal position, relative position or at an absolute position.

- **Relative Positioning**
  - If no top and left properties are specified, the element is placed exactly where it would have been placed if no position property were given
  - But it can be moved later using JavaScript

```html
<p> This is some text. 
    <span style="position:relative; top: -1em">Up we go!</span> Here is some more text. 
</p>
```
Absolute Positioning

- **Absolute Positioning**
  - The element is positioned relative to its first positioned (not static) ancestor

```html
<p style = "position: absolute; left: 50px; top: 100px;">

- If an element is nested inside another element and is absolutely positioned, the top and left properties are relative to the enclosing element

```html
<html><head><style type="text/css">
  .one {position:absolute; top: 200px; left:300px}
  .two {position:absolute; top: 100px; left:200px}
  .three {position:static}
</style></head>
<body>
<p class="one">This is the text of the first paragraph. This is the text of the first paragraph. </p>
<p class="two">This is the text of the second paragraph. This is the text of the second paragraph. </p>
<p class="three">This is the text of the third paragraph. This is the text of the third paragraph. </p>
</body></html>
Stacking Elements

- Specify stacking order with:
  - `z-index: value`

- In CSS2, each box has a position in three dimensions. In addition to their horizontal and vertical positions, boxes lie along a “z-axis” and are formatted one on top of the other.

- Positioning on the z-axis is controlled by the `z-index` property (0, 1, 2, 3 …). The higher `z-index`ed objects are stacked above objects with lower `z-index`.

![Stacking Elements Diagram](image)
A positive catalyst for the evolution to exciting technologies
“… force the vendors and the Working Group to work together to devise the
tests necessary to determine interoperability. Those tests can then guide
those who follow, helping them to achieve interoperable status much faster.
They could literally ship the prefixed implementation in one public beta and
drop the prefix in the next.”

```
.foo {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  border-radius: 10px;
}
```
2D Transform – Scale Transform

- Scale Transform
  - scale(2.0) – twice as larger of the original
  - scale(0.5) – half of the original size
- Notice how the transform doesn’t disturb the rest of the elements in the document, and zooms the photo out from the center, without affecting the layout around it.
- You can also set a `transform-origin` that will dictate where the scaling will expand from: top, bottom, center, or a percentage

Can also do shadows, rotations, skew, translations, gradients.
CSS3 Online Resources

- CSS3 Information and Preview
  - http://www.css3.info
- CSS3 online rule generator
  - http://css3please.com/

W3C Standards CSS3 Relevant Modules

- CSS Transitions (http://www.w3.org/TR/CSS3-transitions/)
- CSS 2D Transforms (http://www.w3.org/TR/CSS3-2d-transforms/)
- CSS 3D Transforms (http://www.w3.org/TR/CSS3-3d-transforms/)
- Media Queries (http://www.w3.org/TR/CSS3-mediaqueries/)
- Multi-Column Layout (http://www.w3.org/TR/CSS3-multicol/)
- Web Fonts (http://www.w3.org/TR/CSS3-webfonts/)