
COMP 4021
Internet Computing

More on CSS
(Cascading Style Sheets)

David Rossiter

Use of CSS

- ❑ CSS can be applied to all xml-compliant (tag-based) languages
- ❑ CSS can be used to display:
 - XML
 - Any XML based language, i.e.
 - ❑ HTML
 - ❑ SVG
 - ❑ SMIL (XHTML & Time)
 - ❑ MathML
 - ❑ ChemML
 - ❑ And so on...

Recall: Inline CSS

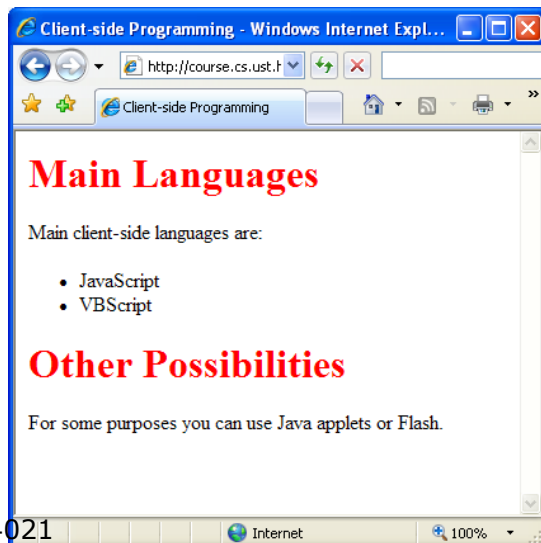
- Inline style:

```
<h1 style="font-size:48pt; font-family:Arial; color:red;" >  
  This is My Report</h1>
```

- These style parameters will apply only to this single instance of h1, not to other instances of h1
- What if you want the same visual information to be used for *all* paragraphs in the web page?
- Inline CSS is ‘bad’; the main idea of css is to provide a *central* set of style rules
- A web site designer wants to find all visual data in the style section so the ‘look and feel’ can easily be changed

Recall: Internal CSS

- ❑ Styles are directly embedded in the **HMTL document**; generally put inside the `<head>` element
- ❑ Still not very good: You need to repeat the styles in each HTML documents that use them!



```
<html>
<title>Client-side Programming</title>
<style>
h1 { color: red }
</style>
<body>
```

All h1 in the web page will be red

```
<h1>Main Languages</h1>
<p>Main client-side languages are:</p>
<ul>
<li>JavaScript</li>
<li>VBScript</li>
</ul>
<h1>Other Possibilities</h1>
<p>For some purposes you can use Java applets
or Flash.</p>
</body>
</html>
```

CSS

Recall: External Styles

- Styles are put in a separate “CSS” file

File: my_style.css

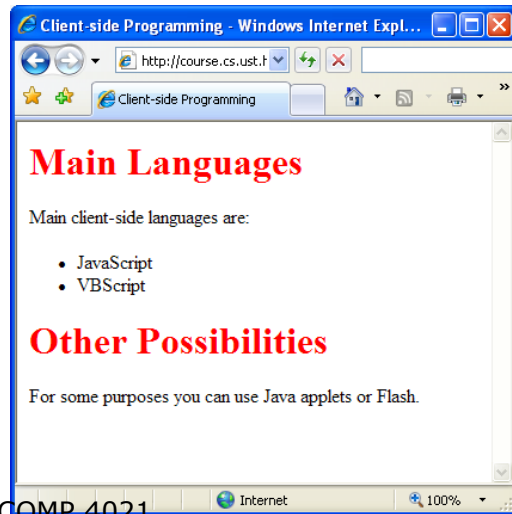
```
h1 { color: red }
```

File: css_simple.html

```
<html>
<title>Client-side Programming</title>
<link rel="stylesheet" href="my_style.css"
      type="text/css"/>
<body>
<h1>Main Languages</h1>
<p>Main client-side languages are:</p>
<ul>
<li>JavaScript</li>
<li>VBScript</li> </ul>
<h1>Other Possibilities</h1>
<p>For some purposes you can use Java applets
or Flash.</p>
</body> </html>
```

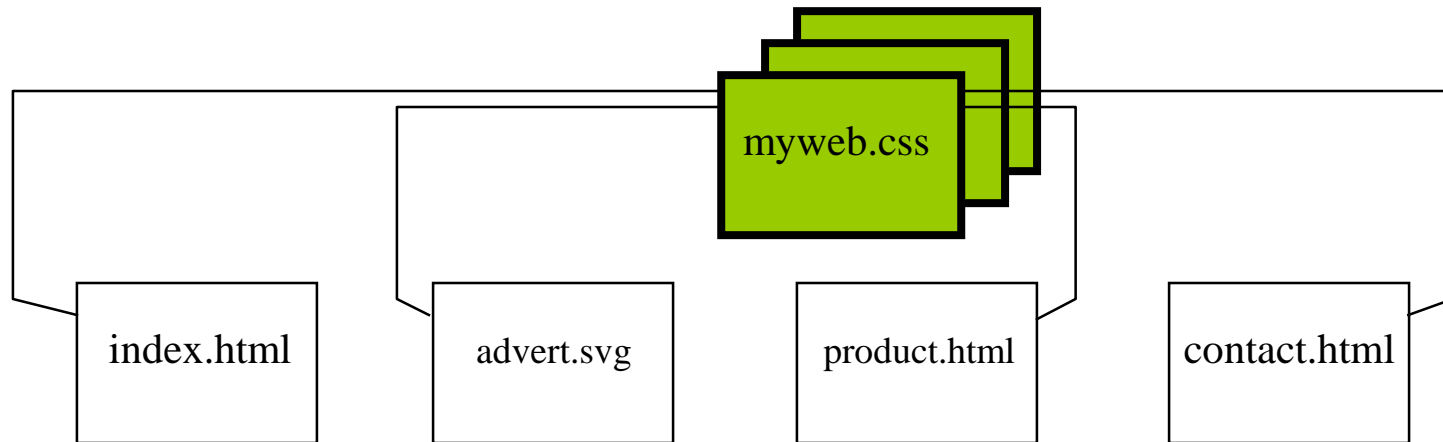
CSS

- The visual result is the same as before:



Typical Web Site Usage

- ❑ CSS means that a complete set of web pages can be developed which all point to the same CSS files
- ❑ The files can even have different languages (i.e. SVG, XHTML) all pointing to the same style information



Why is CSS called “Cascading”?

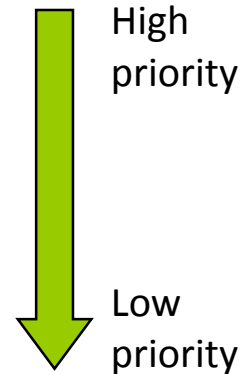
- ❑ Styles in multiple rules defined on the same element are merged
- ❑ When two rules conflict, prioritize them (priority from high to low):

1. Author/Designer-created styles sheets:

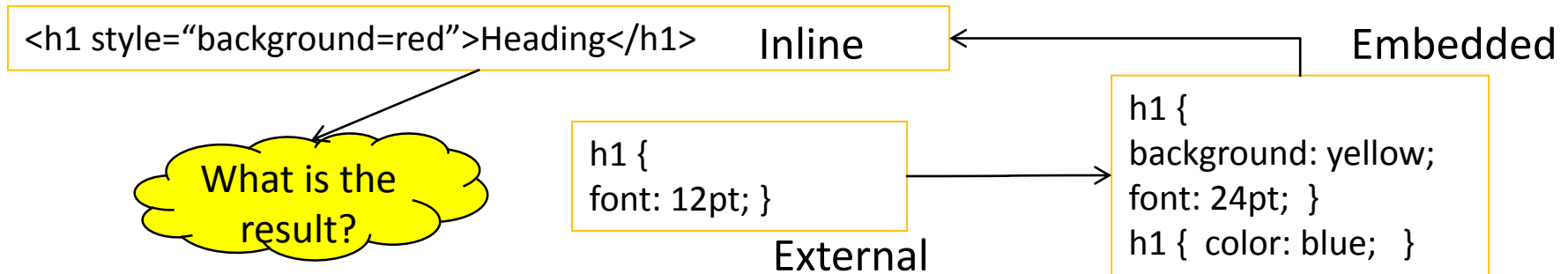
- ❑ Inline styles: style attribute included within a tag
- ❑ Embedded style: CSS rules inside the HTML itself
- ❑ External style sheets: CSS files referenced from the HTML itself

2. User style: Local CSS file specified by the user on the browser

3. User agent style: browser’s default style sheet



- ❑ Given two identical embedded rules, the LAST takes precedence



Website Advantages

- ❑ Separation of contents and styles
- ❑ Styles can be separately managed by visual designers
 - Facilitate global controls and updates to styles
 - Cascade allows local overwrite of styles
- ❑ Every page has a consistent 'look and feel'
- ❑ Style sheet can be altered, result is immediately seen across whole web site - for example, web site can have a different look and feel for Chinese New Year, then later change back
- ❑ Easier for debugging/ handling (just one set of style files controls everything)

Recall: CSS Terminology

- A style rule:

Selector *Property* *Value*

h1 { color: red }

Declaration

- You can define a rule for:

	Selector Syntax	Examples
Element Type	Element_name	h1, div, p
Class	.class_name	.highlight
Element ID	#ID	#myDiv

ID Selector

- Define a rule for a particular element using element **ID**, e.g.,

```
#big_title { font-size: 48pt;  
             font-family: Arial; color: red }
```

```
<h1 id="big_title">My Report</h1>
```

```
<h2 id="small_title">Conclusion</h2>
```

Class Selector

- ❑ Create a rule for a **class** of heterogeneous elements (having different element names):

```
.zappy { font-weight: bold; font-family: Impact; color: blue }
```

The rule will be applied to both of the followings:

```
<p class="zappy">Hi! This is my zappy style!</p>
```

```
<div class="zappy">My name is Zebedee!</p>
```

- ❑ Class can be restricted to a particular set of elements:

```
p.zappy { ... declaration ... }
```

```
div.zappy { ... declaration ... }
```

- p.zappy is applied to `<p class="zappy"> ...`
- div.zappy is applied to `<div class="zappy">...`

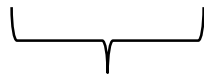
Anchors (Pseudo Class)

- ❑ Style anchor text to distinguish it from normal text
- ❑ However, properties like whether a link has been visited or not is available only to the browser, not the author/designer
- ❑ **Pseudo classes** are classes not defined by human but provided by browser

A:link { text-decoration: underline }

A:visited { text-decoration: none }

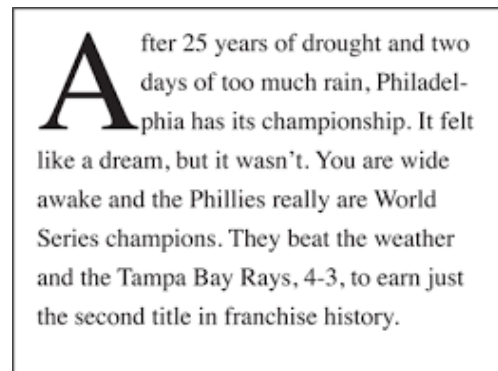
A:hover { background: black }



Pseudo-class names

Pseudo Elements

- ❑ **Pseudo Elements** refer to parts of an element's content, e.g.,
 - FIRST-LETTER and FIRST-LINE
- ❑ To achieve the “drop letter” effect, you can:
`After ...`
and define the style for `<drop-letter>`
- ❑ Using CSS built-in pseudo element:



Class name (defined by a rule not shown here)

P.INITIAL:FIRST-LETTER { font-size: 200%; float:left }

Pseudo-class names

Display at the left of the parent element (i.e., P)

CSS For HTML

□ Style parameters that can be controlled with CSS:

- Text font
- Text size
- Text colour
- Background colour
- Background image
- Margins
- Padding (space between element and margins)
- Borders (including colour, style, width)
- Word spacing
- Letter spacing
- Text decoration (such as underline and blink)
- Vertical alignment
- Control over capitals (upper case, lower case)
- Text indentation
- List styles (many parameters)

Setting Multiple Attributes in a Rule

```
h1 {color: maroon;  
    font: italic 1em Times, serif;  
    text-decoration: underline;  
    background: yellow url(titlebg.png) repeat-x;  
    border: 1px solid red; padding: 5px; }
```

- All h1 in the web page will use dark red, the most commonly used font for paper, is italicized, is underlined, has a background image that is repeated horizontally (not vertically) but will use yellow for the background image if the image cannot be loaded, uses a 1 pixel red border that is separated from the text by 5 pixels

CSS - Large Example 1/4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Basic CSS Example</title>
```

```
<style type="text/css">
```

```
body                {background-color: black;}
```

```
div.page            {background-color: #FFD040;  
                      color: black;  
                      margin: 50px 10px 50px 10px;  
                      padding: 10px 10px;  
                      width: 90%;  
                      height: 90%;}
```


CSS - Large Example 2/4

h1	<code>{font-size: 24pt; font-family: Comic Sans Ms, Cursive; text-align: center;}</code>
.blackonwhite	<code>{color: black; background-color: white;}</code>
.whiteonblack	<code>{color: white; background-color: black;}</code>
p	<code>{font-family: Arial, Sans-serif; font-size: 16pt; line-height: 200%; text-align: justify; text-indent: 20px;}</code>
.style	<code>{color: blue; font-family: Arial; font-style: oblique;}</code>

CSS – Large Example 3/4

.size {font-size: x-large;}

</style>

</head>

<body>

<div class="page">

<h1>

CSS

Demo

</h1>

CSS – Large Example 4/4

```
<hr/>
```

```
<p>Cascading style sheets (CSS) can be used to determine  
everything from <span class="style">font styles</span> and  
<span class="size">sizes</span>to <span id="letterspace">letter  
spacing</span> and line heights.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Applying a Rule to Multiple Tags

```
h1 { background: yellow; color: blue }
```

```
h2 { background: yellow; color: blue }
```

```
h3 { background: yellow; color: blue }
```

- ❑ The above can be more efficiently written as
h1, h2, h3 { background: yellow; color: blue }

Applying a Rule to Multiple Tags

```
h1 { background: yellow; color: blue; font: 24pt; }
```

```
h2 { background: yellow; color: blue; font: 20pt; }
```

```
h3 { background: yellow; color: blue; font: 16pt; }
```

- ❑ The above can be more efficiently written as

```
h1, h2, h3 { background: yellow; color: blue}
h1 { font: 24pt; }
h2 { font: 20pt; }
h3 { font: 16pt; }
```
- ❑ One rule sets the common properties for all three tags
- ❑ An individual rule tailors the font size of each tag
- ❑ Two rules are defined for the same tag

CSS in HTML5

- ❑ CSS is already a powerful language, HTML5 makes it more powerful to meet the imagination of all users
- ❑ Standardization of separation of CSS into modules
- ❑ More selectors: E::nth-child(n), E::not(s)
- ❑ Color: saturation, lightness, alpha-channel
- ❑ Background and Borders: stretch a background image, box shadow, rounded box corners
- ❑ Multi-column layout
- ❑ @media rules: display size, color depth, aspect ratio

Take Home Message

- ❑ Separation of content from styling is a major idea in content management
- ❑ Major problem is to identify which subset of elements a rule applies to
 - Lead to very complicated “pattern matching” or “addressing” schemes
- ❑ CSS has become more and more powerful, from CSS1 to CSS2 and to CSS3 ...
 - CSS goes beyond styling to include animation, 2D/3D transformation