

# **COMP 4021**

## **Internet Computing**

### **Introduction to Cascade Style Sheet (CSS)**

Dr. Kenneth LEUNG

Slides created by Prof. Dik Lun LEE

# Why Cascade Style Sheets (CSS)?

- CSS separates visual parameters (colour, spacing, etc.) from actual content (words)
- <http://www.w3.org/Style/>
- You have already seen how style can be used for individual elements:

```
<p style="color:red; background-color:yellow; font-size:46pt; font-style:italic">A pretty paragraph.</p>
```

- But what if you want the same visual information to be used for *all* paragraphs in the web page?

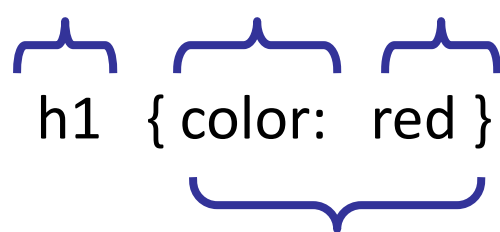
# CSS Terminology

- A style rule:

*Selector*   *Property*   *Value*

h1 { color: red }

*Declaration*

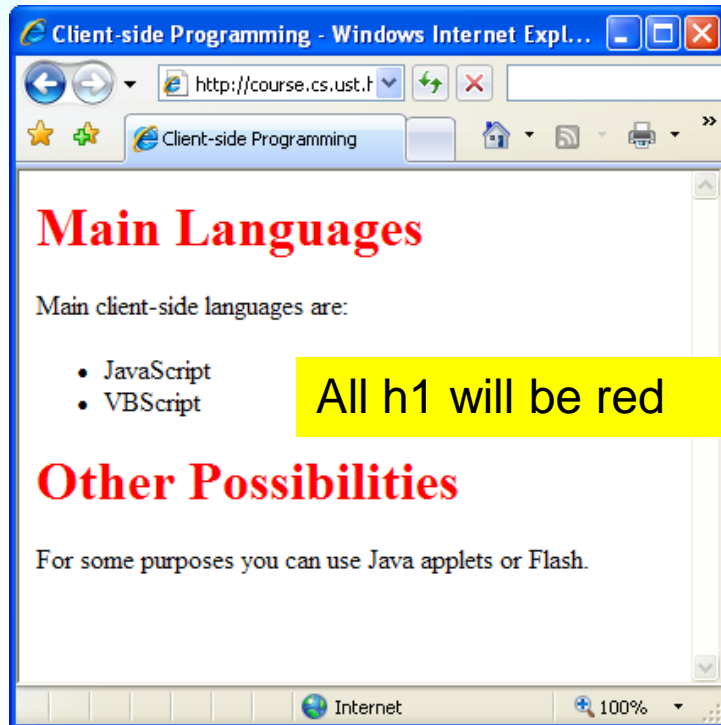


- You can define a rule for:

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

# CSS with Element Selector

- Declare “style” once and apply it to all paragraphs
- You can put a style section at the top of the page



```
<style>
```

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

```
</style>
```

```
... ..
```

```
<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>Java applets or Flash</p>
```

```
</body>
```

Rule:

**h1 { color: red }**

selector ↑    ↑ style

# Style Using a Relative Link

- An alternative is to separate style and data into two files:

*File: my\_style.css*

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

*File: css\_simple.html*

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

- The visual result is the same as before

```
<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> Java applets or Flash.</p>
</body>
```

# ID Selector

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

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

```
#small_title { font-size: 24pt;  
              font-family: Arial; color: blue }
```

```
<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 }
```

```
<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">...`

# Nice Way to Style a Div

- Typically you would first define the style information for a div (such as the position and colours):

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

```
.layer_style1 {
```

```
    position:absolute; top:20px; left:5px;
```

```
    color:#CC00EE; width:200px;
```

```
}
```

```
</style>
```

A **style class** is created. Note the dot before the class name



# Declaring the Div

- The div is defined using the style rule:

Style class created in the last slide is used

```
<div id="layer_name1" class="layer_style1" >  
  <h1>Layer 1</h1>  
  <p>Content for layer 1 goes here.</p>  
</div>
```

# Anchor (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 provided by the browser

```
a:link { text-decoration: underline }
```

```
a:visited { text-decoration: none }
```

```
a:hover { background: black }
```

```
a:active { color: blue }
```

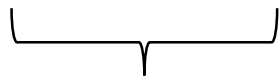
└───┘

Pseudo-class names

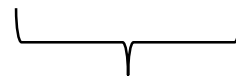
# Pseudo Elements

- **Pseudo Elements** refer to parts of an element's content, e.g.,  
:first-letter and :first-line
- To achieve the “drop letter” effect:  
`.drop-letter { font-size: 48pt }`  
`<p><span class=“drop-letter”>A</span>fter...</p>`
- Using CSS built-in pseudo element:

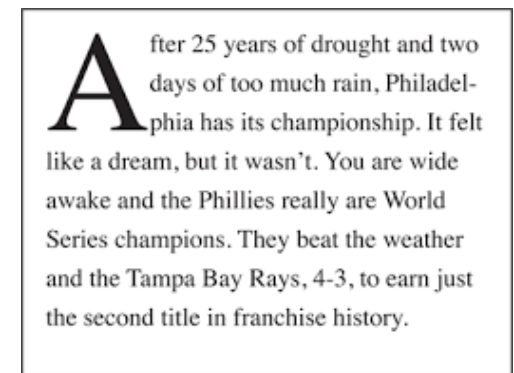
`p:first-letter { font-size: 200%; float:left }`



Pseudo-class names



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



# Take Home Message

- CSS separates content and style, making webpages easier to **read** and **maintain**
  - Powerful “selector” make selecting DOM elements easy
- CSS is much more powerful than covered here
  - In jQuery, we will learn more CSS selectors