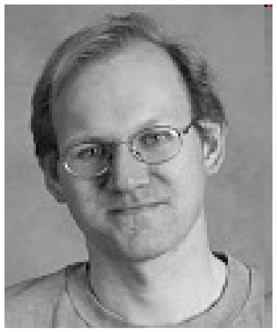
CSCC09F
Programming on the Web

CSS: Cascading Style Sheets

associating style with structure,
example style properties,
the cascade, compatibility

Separation of Style and Content



Bert Bos and Hakon Lie

- CSS = Cascading Style Sheets
  - cumulative effect with overrides, hence "cascading"
  - "A simple mechanism for adding style to Web documents"
    - fonts, colors, spacing, layout, …
- CSS applies generically to all forms of XML (including XHTML), as well as HTML4 and HTML5

Separation of Style and Content

- Particularly optimized for use with HTML
- In original HTML, style was specified using HTML elements and attributes.
  - e.g. **`<i>, <b>, <blink>, <p align=…>, <td bgcolor=…>`**
  - A consequence of this approach is that style is localized to individual elements/attributes – hard to maintain, update
- In HTML 4, these style elements/attributes mostly deprecated in favour of attaching to external style languages not defined as part of HTML.
  - CSS is an example (XSL is another)
  - With CSS, a single style definition may apply to many elements or contexts (tree structures)

CSS Advantages

- Simplified site maintenance: define-once, use-many
  - localize change (lowers cost of maintenance)
  - single definition of style-related data
  - multiple renderings reference this definition
- Precise control over presentation
- User can override default styles
- Faster downloads
  - CSS files can be cached for reuse
  - Eliminate redundant style tags in document body
- Media-specific rendering
  - Accessibility (e.g., Braille, aural, handheld, projector)
  - Print-specific formatting – e.g. pagination

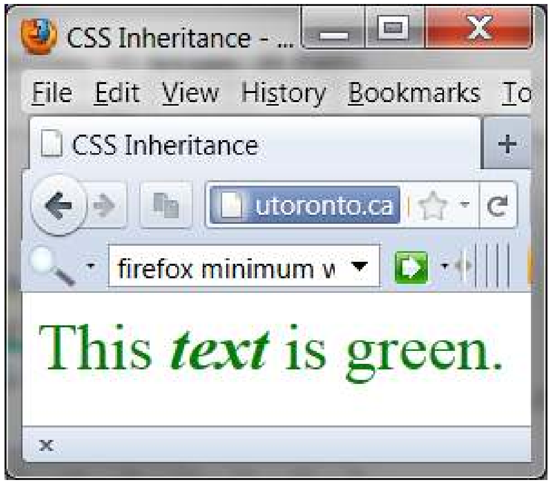
The CSS Language

- Using CSS, one associates style property-values with HTML or XML elements (including XHTML elements)
  - by element type
    - <p>, <h1>, <body>, …
    - <span> used to associate style some portion of inline text – can appear wherever text allowed
    - <div> used to associate style with a block-level element – can appear wherever a block element is allowed
  - by element "class"
    - `<p class="my-style-class-name">`
  - by unique individual-element "id"
    - `<p id="a-specific-paragraph">`
  - by element in context, e.g. **`<em>`** is within a **`<p>`** context
    - `<p>This <em>word</em> is emphasized</p>`

Inheritance

- Most styles are inherited into nested elements.
- One way to set a "default" document style is by setting style property values for the **<body>** element.
  - inherit.html

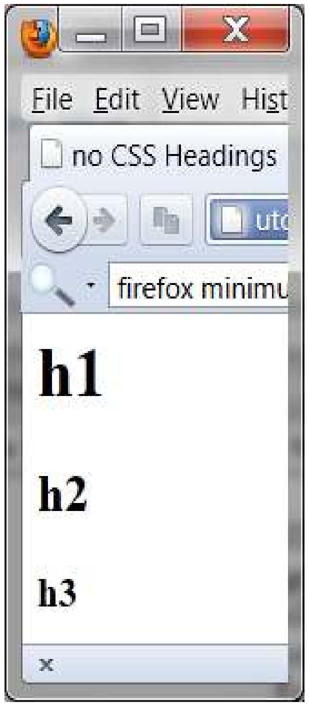
```
<style type="text/css">
    body { color: green;
           font-size: 200%
    }
    em { font-weight: bold }
</style>
  ...
<p>This <em>text</em>
        is green.</p>
```



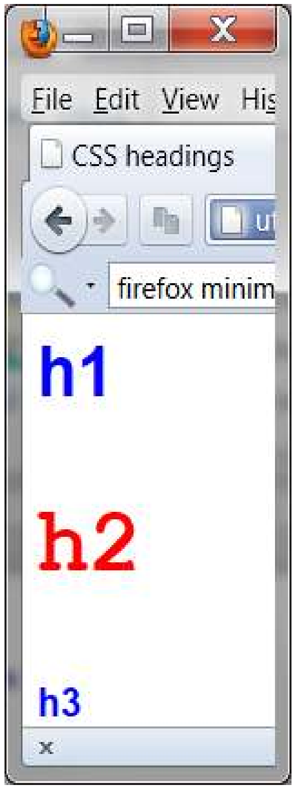
This *text* is green.

Grouping as Selector

- Multiple <u>comma-separated</u> elements can be grouped, with common style applied to all.



<u>h123.html</u>



<u>h123css.html</u>

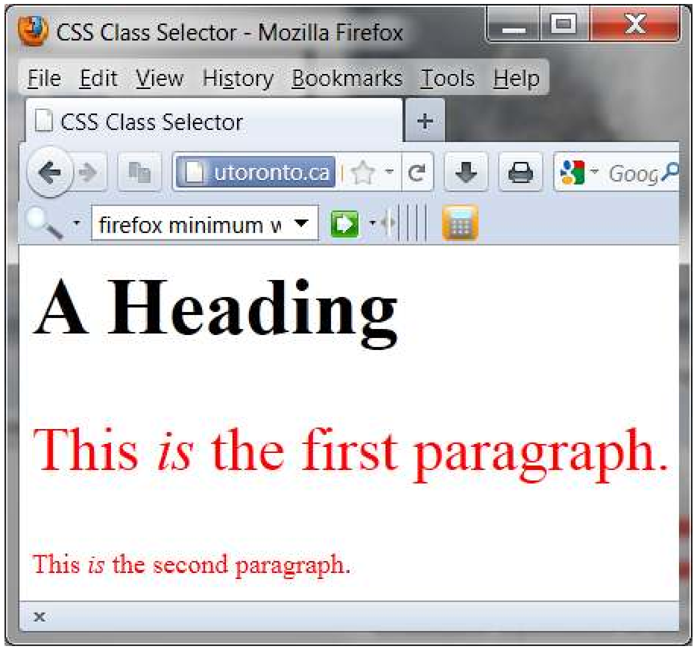
```
<style type="text/css">
   h1, h3 { color: blue;
        font-family: helvetica
   }
   h2 { font-size: 36pt;
        color: red;
        font-family: courier new
   }
</style>
…
<body>
    <h1>h1</h1>
    <h2>h2</h2>
    <h3>h3</h3>
</body>
```

class as Selector

- HTML elements can be tagged with possibly many classes.
- Style properties can be set across all elements of a given class.

classcss.html

```
<style type="text/css">
    .red { color: red }
    .large { font-size: 30pt }
    h1.large { font-size: 40pt }
</style>
...
<h1 class="large">A Heading</h1>
<p class="large red">This
     <em>is</em> the
     first paragraph.</p>
<p class="red">This <em>is</em>
     the second paragraph.</p>
```


CSS Class Selector - Mozilla Firefox
File Edit View History Bookmarks Tools Help
CSS Class Selector
utoronto.ca
firefox minimum w

A Heading

This *is* the first paragraph.

This *is* the second paragraph.

15CSS CSCC09 Programming on the Web 9

id as Selector

- HTML4 and XML allow any element to have a special "id" attribute that is unique within the document
  - Can be used as the target for a hyperlink
  - Can be used to associate style properties with a particular element
    - idcss.html
  - Whereas a class selector may apply to several elements, an id selector applies to at most one element within a given XML (or XHTML) document

```
<style type="text/css">
   #p1 { font-size: 30pt;
          color: blue }
   #p2 p { font-size: 40pt }
</style>
…
<div id="p1">This is a
       <em>p1</em> div
       section.</div>
<p id="p2">This is a p2
   <em>paragraph</em>.</p>
```

15CSS CSCC09 Programming on the Web 10

Contextual selectors

- CSS can match a search pattern on a stack of open elements (designated by whitespace separated list of selectors).
  - contextcss.html
- Ancestors, not just parents
- Can mix and match the various types of selectors into sentences:
  - div.chapter p.first { font-size: 120% }
  - #x23a p .foo { color: red; }

Pseudo Classes and Elements

- A hack to account for attributes and elements whose presence varies through time – as the user interacts with the document.
- Why called “pseudo”?
  - because these classes/elements do not appear in the document text (!) - instead the browser dynamically inserts pseudo-class attributes and elements based on page usage
- Link pseudo classes:

```
a:link { color: blue } /* unvisited link */
a:visited { color: red } /* previously-visited link */
```

  - No effect on elements other than <a>, so 'a' may be omitted.
- Dynamic pseudo classes:

```
:active { color: green } /* active element */
:hover { font-size: 110% } /* cursor hover over elt */
:focus { color: grey } /* element selected for input */
```

Pseudo-Element as selector

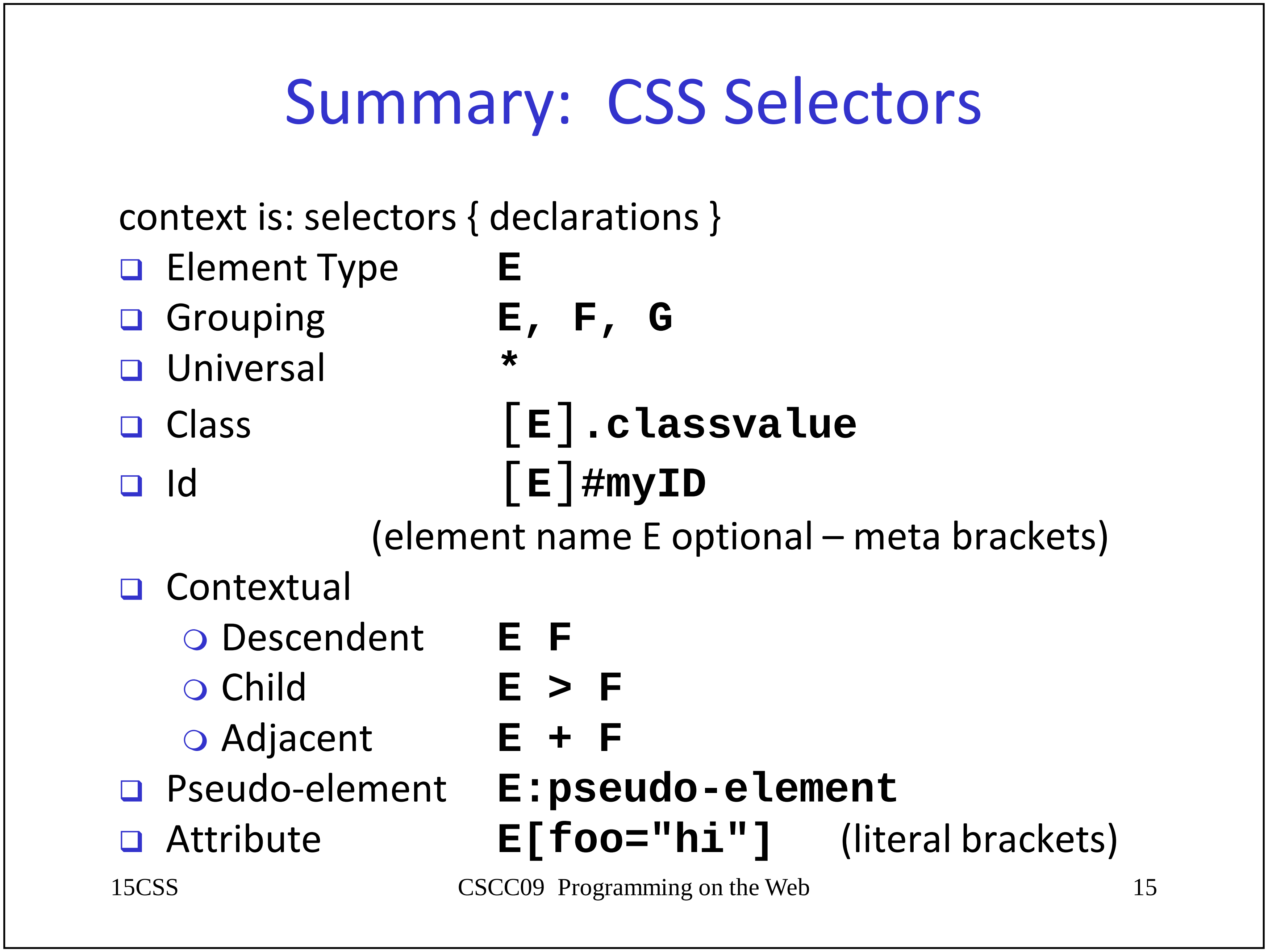
- Used for typographically important regions of text that are not delimited (not bracketed by open/close tags)
  - First-line formatting

    `:first-line { font-variant: small-caps }`
  - First-letter (drop-caps) formatting

    `:first-letter { font-size: 200%; float: left }`
    - browser defines what's "in" the first letter (e.g., opening quotes).
  - example: pseudocss.html
- can combine with other selectors, e.g.
  - `p.key:first-line { color: red }`

Pseudo Element generated text

- Can add text content (but not markup) before or after XML data
- Pseudo-elements
  - `:before`
  - `:after`
- Example
  - XML or XHTML:
    - `<author>Sebesta</author>`
  - CSS:
    - `author:before {content: "Author’s Name -";}`
  - Browser:
    - **Author’s Name - Sebesta**



Summary: CSS Selectors

context is: selectors { declarations }

❑ Element Type	`E`
❑ Grouping	`E, F, G`
❑ Universal	`*`
❑ Class	`[E].classvalue`
❑ Id	`[E]#myID`
	(element name E optional – meta brackets)
❑ Contextual	
❍ Descendent	`E F`
❍ Child	`E > F`
❍ Adjacent	`E + F`
❑ Pseudo-element	`E:pseudo-element`
❑ Attribute	`E[foo="hi"]` (literal brackets)

15CSS CSCC09 Programming on the Web 15