COMP2155

BUILDING WEB APPLICATIONS

DR. CURTIS GITTENS

LECTURE 8

Cascading Style Sheets

CRASH COURSE ON CSS

What's the Diff?

- Same class? Same effect?

```
.redwarning {
    color: red;
}

<html>
...
<h2 class="redwarning">Mild Danger!</h1>
...
</html>
```

```
h1.redwarning {
    color: red;
}

<html>
...
<h1 class="redwarning">Danger! Danger!</h1>
...
</html>
```

Some Quick Info

- Multiple Selectors
  - How are multiple selectors for an element handled?
    - For example
  - The most specific rule(s) is/are applied

```
span { font-size: large; }
.toptext {font-size: x-large;}
.boldtext {font-weight: bolder; }
.yellowtext {font-weight: normal; color: yellow; }
.redtext {font-weight: normal; color: red; }
```

  - Given the above rules, which rule(s) is/are applied to the span element?

Some Quick Info

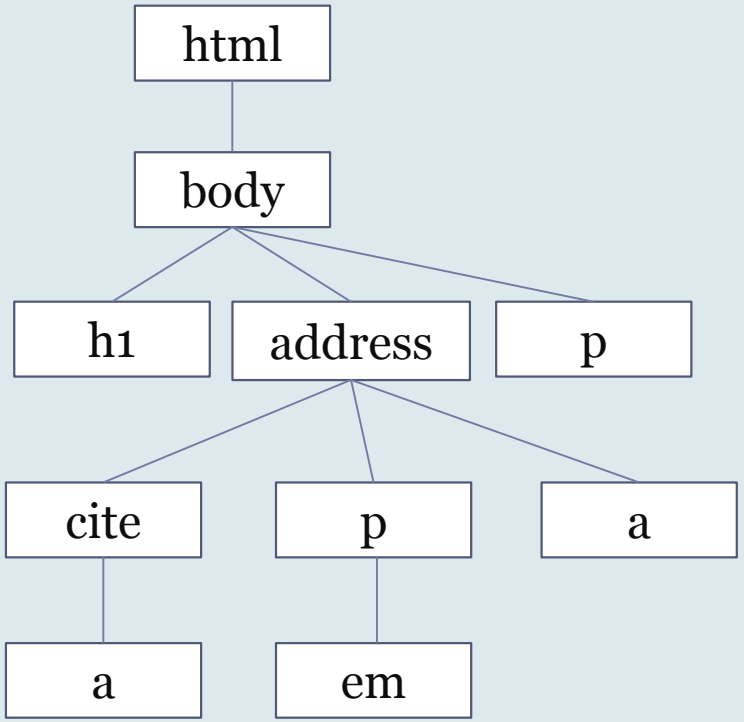
- The id attribute – similar but different
  - Similar to adding an element to a class, **but** the attribute is called “id”, not “class”
- More importantly
  - An element can’t have multiple ids,
  - There can’t be multiple elements on a page with the same id
- Example

```
#footer { color: red; }  /* the element with id footer */
p#footer { color: red; } /* p element with id footer */
.footer { color: blue; } /* class footer */
p.footer { color: orange; } /* paragraph footer class */
```

Some Quick Info

- Descendants
  - Provides a way in CSS to select only elements that are descendants of (nested in) a particular element
  - More specific than inheritance
  - Example:

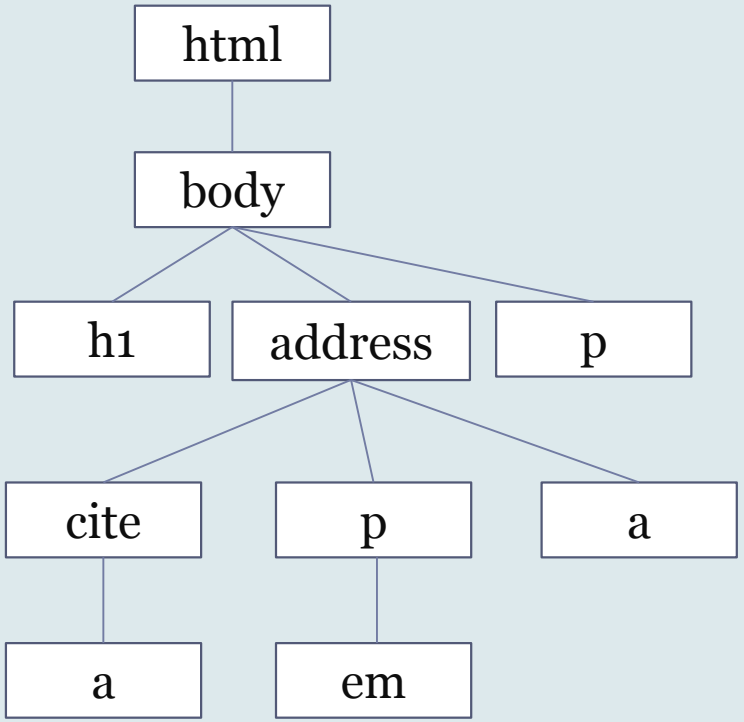
What happens if just these two rules are defined?
p { color: purple; }
address {color: orange; }
… and then this is added?
address p {color: blue; }
… and then this?
address cite a { color: maroon; }



Some Quick Info

- Direct Descendants
  - Descendant selectors apply the rule to **all** descendants of the element
  - It does not stop at the child element or other elements that are nested within a different element
  - To select a child element only use the '>'
  - Example

```
address > a { color: chocolate; }
```



Pseudo-Elements & Pseudo-Classes

- Pseudo-elements
  - Create abstractions about the document tree beyond those specified by the document language.
    - E.g. DOM offers no mechanisms to access the first letter or first line of an element's content.
    - Pseudo-elements provide a mechanism to refer to this otherwise inaccessible information.
    - Pseudo-elements may also provide a way to assign style to content that does not exist in the source document
      - E.g. the :before and :after pseudo-elements give access to generated content

Pseudo-Classes

- Pseudo-classes
  - Classify elements on characteristics other than their name, attributes or content;
    - Characteristics that cannot be deduced from the document tree.
  - May be dynamic,
    - An element may acquire or lose a pseudo-class while a user interacts with the document.
  - Exceptions:
    - ':first-child', which *can* be deduced from the document tree
    - ':lang()', which can be deduced from the document tree in some cases.

Some Quick Info

- Pseudo-classes
  - These are classes defined by the browser. There are seven pseudo-classes*:

Pseudo-class name	Description
:first-child	Styles an element that is the first child of another element
:active	Styles an element that is activated
:focus	Styles an element that has keyboard input focus
:hover	Styles an element when you mouse over it
:lang	Styles an element with a specific lang attribute
:link	Adds a style to an unvisited link
:visited	Adds a style to a visited link

* Descriptions from w3schools.com

Some Quick Info

- :first-child pseudo-class
  - matches an element that is the first child element of some other element.
  - Example
    - Matches any <p>element that is the first child of a element.
    - `div > p:first-child { text-indent: 0 }`
    - Sets the font weight to 'bold' for any element that is some descendant of a P element that is a first child:
    - `p:first-child em { font-weight : bold }`
    - The following two selectors are equivalent:
    - `* > a:first-child /* A is first child of any element */`
    - `a:first-child /* Same */`

Some Quick Info

- Link pseudo-classes: :link and :visited
  - Browsers commonly display unvisited links differently from previously visited ones.
    - Pseudo-classes ':link' and ':visited' used to distinguish them
  - The :link pseudo-class applies for links that have not yet been visited.
  - The :visited pseudo-class applies once the link has been visited by the user.
  - Browsers may return a visited link to the (unvisited) ':link' state at some point.
    - The two states are mutually exclusive

Some Quick Info

- The dynamic pseudo-classes: :hover, :active, and :focus
  - Browsers sometimes change the rendering in response to user actions.
  - CSS provides three pseudo-classes for common cases:
    - The :hover pseudo-class applies while the user designates an element (with some pointing device), but does not activate it.
    - The :active pseudo-class applies while an element is being activated by the user.
    - The :focus pseudo-class applies while an element has the focus (accepts keyboard events or other forms of text input).
  - An element may match several pseudo-classes at the same time.

Some Quick Info

- Rules for using anchor tag pseudo-classes
  - a:hover MUST come after a:link and a:visited
  - a:active MUST come after a:hover
- Examples*

```
a:link {color:#FF0000}      /* unvisited link */
a:visited {color:#00FF00}  /* visited link */
a:hover {color:#FF00FF}  /* mouse over link */
a:active {color:#0000FF}  /* selected link */

a.red:visited {color:#FF0000}
<a class="red" href="css_syntax.asp">CSS Syntax</a>
```

* Examples taken from w3schools.com

Some Quick Info

- Pseudo-elements

Pseudo-element name	Description
:first-line	Applies special styles to the contents of the first formatted line of a paragraph.
:first-letter	Must select the first letter of the first line of a block, if it is not preceded by any other content (such as images or inline tables) on its line
:before and :after	Used to insert generated content before or after an element's content

* Descriptions from www.w3.org

Determining Specificity

- There is a five-step process the browser uses to determine which rule to apply to an element
  1. Collect **all** style sheets including user-defined style sheets
  2. Find all declarations for the property that has a matching selector e.g. looking for the color property for h1 tags
  3. Sort all matches found in order of importance: author, reader and browser
  4. Sort declarations by specificity, the more specific the more important
  5. Sort conflicting rules by order of appearance. Rules declared last have a higher importance
- The rule at the top of the list is the one applied

Welcome to the "What's my specificity game"

To calculate the specificity you start with a set of three numbers, like this:

0 0 0

In the old days we used four numbers, but that was before XHTML... aren't you glad you're learning this now?

And then we just tally up various things from the selector, like this:

Does the selector have any ids? One point each.

Does the selector have any classes or pseudo-classes? One point each.

Does the selector have any element names? One point for each.

0 0 0

For instance, the selector "h1" has one element in it, so you get:

Read this as the number one. → 0 0 1

Both "h1" and "h1.blue" have one element, so they both get a "1" in the right most number column.

"h1.blue" also has one class, so it gets a "1" in the middle number column.

Neither have ids in their selectors, so they both get a "0" in the left number column

As another example, the selector "h1.blue" has one element and one class, so you'd get:

Read this as the number eleven. → 0 1 1

After you've tallied up all the ids, classes, and elements, the bigger the specificity number, the more specific the rule. So, since "h1.blue" has a specificity of 11, it is more specific than "h1", which has a specificity number of 1.

Determining Specificity

- Use the previous steps to calculate the specificity values for the selectors below

h1.greentea	_____	ol li p	_____	em	_____
p img	_____	.green	_____	span.cd	_____
a:link	_____	#elixirs h1	_____	#sidebar	_____