Putting on the style

intermediate web design and development

Week 2 – Selectors, Classes and IDs

Glenn Clarkson

Learning outcomes for tonight

Understand:

- Units of measurement
- Element type selectors
- Class and ID selectors
- Pseudo-class/element selectors
- Descendent selectors
- Parent-child selectors
- Adjacent selectors
- Attribute selectors
- Selector grouping
- Comments

Assignment

- Simple web site demonstrating what you have learned on the course
- 3 pages of consistent structure and overall style
- All pages linked to each other with a common navigation scheme
- Employs at least one external stylesheet
- Pages and style sheet(s) must validate for CSS3 and the HTML5 DOCTYPE
- Include a description of the site's aims, its audience and an overview of the main styling you have covered
- To be submitted by 11 April 2014

Working the exercises

- Take your time to study the texts in the exercises
- Don’t just rush to copy the code displayed further down the exercise
- As the course unfolds there will be less and less code examples,

  and increasing amounts of instructions to follow!

Units of measurement

Unit	Description
%	percentage
in	*inch*
cm	*centimeter*
mm	*millimeter*
em	1em is equal to the CSS font size of the current parent element. 2em means 2 times the size of the current parent element. Very useful unit in CSS, since browsers can resize the font to meet the reader's needs.
ex	one ex is the x-height of a font (x-height is usually about half the font-size)
pt	*point (1 pt is the same as 1/72 inch)*
pc	*pica (1 pc is the same as 12 points)*
px	*pixels (a dot on the computer screen)*

relative values ***absolute values***

Selectors

- Recall from last week that the selector in a style rule is the part that defines the HTML element(s) to which the rule is applied:

```
h1 {
     font-size: 1.5em;
}
```

- Can be multiple selectors:

```
h1, h2 {
     color: rgb(255,0,0);
}
```

Selectors

- Element Type selector: the most common, it specifies one or more HTML element types with no qualifiers:

```
h1 {
     font-size: 1.5em;
}
```

- Class selector: applies a style rule to arbitrary elements in the document and is implemented in the target elements by the use of the **`class`** attribute:

```
.special {
     font-size: 1.5em;
}
```

```
<p class="special">A paragraph with a different style.</p>
```

Selectors

- Element Type selector: the most common, it specifies one HTML element type with no qualifiers:

```
h1 {
     font-size: 1.5em;
}
```

- Class selector: applies a style rule to arbitrary elements in the document and is implemented in the target elements by the use of the **`class`** attribute:

```
.special {
     font-size: 1.5em;
}

<p class="special">A paragraph with a different style.</p>
```

* The name of the selector is not prescribed

Selectors

- ID selector: lets you target a single element in the document
- Much like a class selector except that the ID must be unique within the document.
- It is implemented by use of the ID attribute:

```
#unique {
     font-size: 1.5em;
}
```

```
<p id="unique">A paragraph with a special style.</p>
```

Selectors

- ID selector: lets you target a single element in the document
- Much like a class selector except that the ID must be unique within the document.
- It is implemented by use of the ID attribute:

```
#unique {
     font-size: 1.5em;
}
```

```
<p id="unique">A paragraph with a special style.</p>
```

* The name of the selector is not prescribed

Selectors

- Pseudo-element selector: has no equivalent HTML element, hence the term ‘pseudo’, the selector matches a specified part of the element:

```
p::first-line {
     font-style:italic;
}
```

`::first-letter`	targets the first character of the first line of text within an element
`::first-line`	targets the first formatted line of text
`::before`	specifies content to be inserted before another element
`::after`	specifies content to be inserted after another element

Selectors

- Pseudo-element selector: has no equivalent HTML element, hence the term ‘pseudo’, the selector matches a specified part of the element:

```
p::first-line {
     font-style:italic;
}
```

`::first-letter`	targets the first character of the first line of text within an element
`::first-line`	targets the first formatted line of text
`::before`	specifies content to be inserted before another element
`::after`	specifies content to be inserted after another element

CCS2 precedes pseudo-elements with : Whereas CSS3 uses ::

Selectors

- Pseudo-class selector: works exactly the same as pseudo-element selectors except they match the whole element not a selected part:

```
a:hover {
        text-decoration:underline;
}
```

:link	matches link elements that are determined to be unvisited
:visited	matches link elements that are determined to have been visited
:active	matches any element that's in the process of being activated
:hover	matches any element that's being designated by a pointing device
:focus	matches any element that has keyboard input focus
:first-child	matches an element only if it is the first child element of its parent element
:lang(*language code*)	matches elements with the lang attribute starting with a specified value, the language code

Selectors

- The order in which you place dynamic pseudo-classes in your stylesheet is very important to ensure they execute in the correct sequence:

Anchor Pseudo-class	Corresponding Hyperlink State
`a:link`	not yet visited
`a:visited`	visited
`a:hover`	cursor positioned over the link but the mouse not being clicked
`a:active`	being clicked on at the moment

love hate

Selectors

- Descendent selectors: applies the rule to an element only when it is a descendent of another specified element:

```
h1 p {
     color:#666;
}
```

Oxford

This article is about the city of Oxford in England.

History

Oxford was first settled in Saxon times, and was initially known as "Oxenaforda", meaning "Ford of the Oxen"; fords were more common than bridges at that time.

- The descendent relationship need not be an immediate parent-child relationship

Selectors

- Parent-child selector: applies a style rule to all elements that are the immediate children of a specified element (the parent):

```
ul > li {
    color:#900;
}
```

```
<ul>
   <li>First list item</li>
   <li>Second item
      <ol>
         <li>Sub list 1</li>
         <li>Sub list 12</li>
      </ol>
   </li>
</ul>
```

First list item
Second item
1. Sub list 1
2. Sub list 2

Selectors

- Adjacent selector: selects all elements that are the adjacent (immediately following) siblings of a specified element:

```
h3 + p {
    …declarations
}
```

```
<h3>Heading</h3>
<p>The selector matches this paragraph.</p>
<p>The selector does not match this paragraph.</p>
```

Selectors

- Attribute selectors: match elements on the basis of either the presence of an attribute, or the exact or partial match of an attribute value

[*attribute*]	selector matches any element that has an ***attribute*** attribute
[*attribute*="*value*"]	matches only if the attribute has a value of ***value***
[*attribute*~="*value*"]	matches only if the attribute is defined with a space separated list of values, one of which exactly matches ***value***
[*attribute*|="*value*"]	matches only if the attribute is defined with a hyphen-separated list of 'words', and the first of these words begins with ***value***
[*attribute*^="*value*"]	matches if the attribute value starts with the characters ***value***
[*attribute*$="*value*"]	matches when the attribute contains a value ending with the characters ***value***
[*attribute="*value*"]**	matches when the attribute contains a value with the characters ***value***

Selectors

- Attribute selectors examples

`[href]`	matches all elements with the **`href`** attribute
`[type="submit"]`	matches all elements where the **`type = submit`**
`[class~="caution"]`	**matches `<p class="caution">`** and **`<strong class="important caution">`** and **`<div class="caution highlight">`**, **but not `<p class="my-caution">`** or **`<ul class="cautions">`**
`[hreflang|="en"]`	matches **`hreflang`** attribute values of **`"en"`**, **`"en-US"`**, **`"en-GB"`** etc.
`[href^="http:"]`	matches elements that have an **`href`** attribute value which starts with the characters **`"http:"`**
`[src$=".pdf"]`	matches elements with a **`src`** attribute value that ends with the characters **`".pdf"`**
`[id*="bar"]`	matches elements whose **`id`** attribute value contains the characters **`"bar"`**.

Selectors

- Attribute selectors examples

`[href]`	matches all elements with the **`href`** attribute
`[type="submit"]`	matches all elements where the **`type = submit`**
`[class~="caution"]`	matches **`<p class="caution">`** and **`<strong class="important caution">`** and **`<div class="caution highlight">`**, **but not `<p class="my-caution">`** or **`<ul class="cautions">`**
`[hreflang|="en"]`	matches **`hreflang`** attribute values of **`"en"`**, **`"en-US"`**, **`"en-GB"`** etc.
`[href^="http:"]`	matches elements that have an **`href`** attribute value which starts with the characters **`"http:"`**
`[src$=".pdf"]`	matches elements with a **`src`** attribute value that ends with the characters **`".pdf"`**
`[id*="bar"]`	matches elements whose **`id`** attribute value contains the characters **`"bar"`**.

`img[src$=".png"]` matches all images with **`.png`** file format

Selectors

- Selector grouping: applies a style rule to several different elements in a document:

```
h1, h2, #special {
    font-family:Tahoma, Geneva, sans-serif;
}
```

sets:

all **h1**, **h2** elements, and the element with the **ID** of '**special**'

to use the same font family

New pseudo-classes

- CSS3 provides four powerful pseudo-classes that allow you to select multiple elements according to their positions in a document tree
- The pseudo-classes are:
  - :nth-child(N)
  - :nth-last-child(N)
  - :nth-of-type(N)
  - :nth-last-of-type(N)
- The argument, N, can be a keyword, a number, or a number expression of the form an+b

New pseudo-classes

- These pseudo-classes accept the keywords **odd**, for selecting odd-numbered elements, and **even**, for selecting even-numbered elements.
- If the argument N is a number, it represents the ordinal position of the selected element. For example, if the argument is 5, the fifth element will be selected.
- The argument N can also be given as an+b, where a and b are integers (for example, 3n+1).

New pseudo-classes

n	2n+1	4n+1	4n+4	4n	5n-2	-n+3
0	1	1	4	-	-	3
1	3	5	8	4	3	2
2	5	9	12	8	8	1
3	7	13	16	12	13	-
4	9	17	20	16	18	-
5	11	21	24	20	23	-

- 4n+1 will match the first, fifth, ninth, thirteenth, seventeenth, twenty-first, and so on, elements if they exist, while the expression -n+3 will match the third, second, and first elements only.
- The difference between the nth- and nth-last- is the nth-last- count from the bottom up

New pseudo-classes

- For a more detailed explanation visit:

  http://reference.sitepoint.com/css/css3psuedoclasses

Comments

- In HTML you use **`<!– this an HTML comment -->`** to write a comment inside your code to act as reminders for you when you look back at the code
- In CSS you use **`/* this is a CSS comment */`** to write a comment in CSS
- Comments in CSS are completely ignored by the browser in exactly the same way as they are in HTML
- Comments are visible when ‘viewing source’ in a browser