

- **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

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

```
h2 + p {  
    ...declarations  
}
```

```
<h2>Heading</h2>  
<p>The selector matches this paragraph.</p>  
<p>The selector does not match this paragraph.</p>
```

- **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

- **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;  
}
```

|                       |   |
|-----------------------|---|
| <b>::first-letter</b> | targets the first character of the first line of text within an element |
| <b>::first-line</b>   | targets the first formatted line of text                                |
| <b>::before</b>       | specifies content to be inserted before another element                 |
| <b>::after</b>        | specifies content to be inserted after another element                  |

CCS2 precedes pseudo-elements with : CSS3 uses ::

- **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;  
}
```

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

- **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

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

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

- Attribute selector examples

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

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

- 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  $an+b$

- 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,  $3n+1$ ).

| 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  $n$ th- and  $n$ th-last- is the  $n$ th-last- count from the bottom up

- For a more detailed explanation visit:

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