

Web Development

Produced
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA FHORT LÁIRGE



Exploring CSS Selectors

Web Development

Agenda

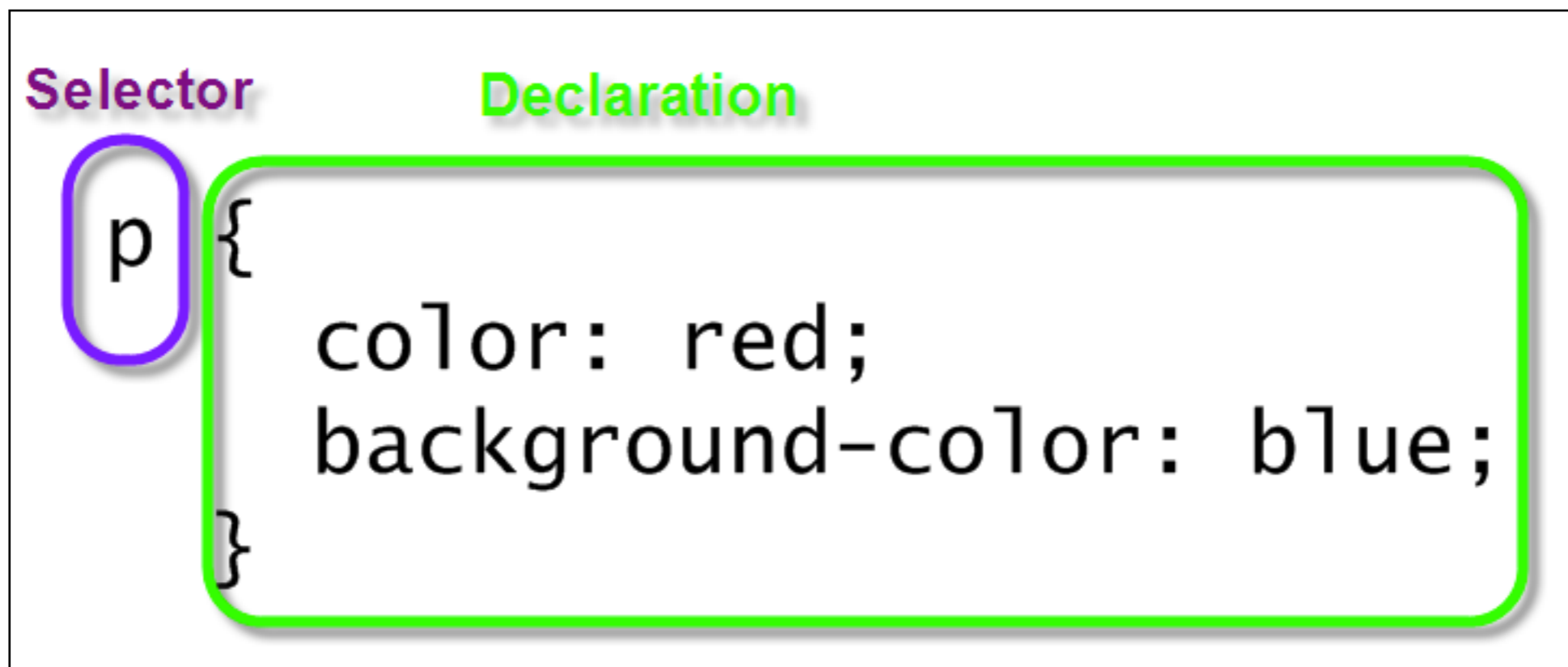
- CSS
 - Rules, Selectors, Declarations, Properties, Values
 - CSS Binding
 - Combining Rules & Selectors
 - Html recap: attributes & Span elements
 - Classes and class based styling
 - Selector rules and the “Cascade” in CSS

CSS Rule

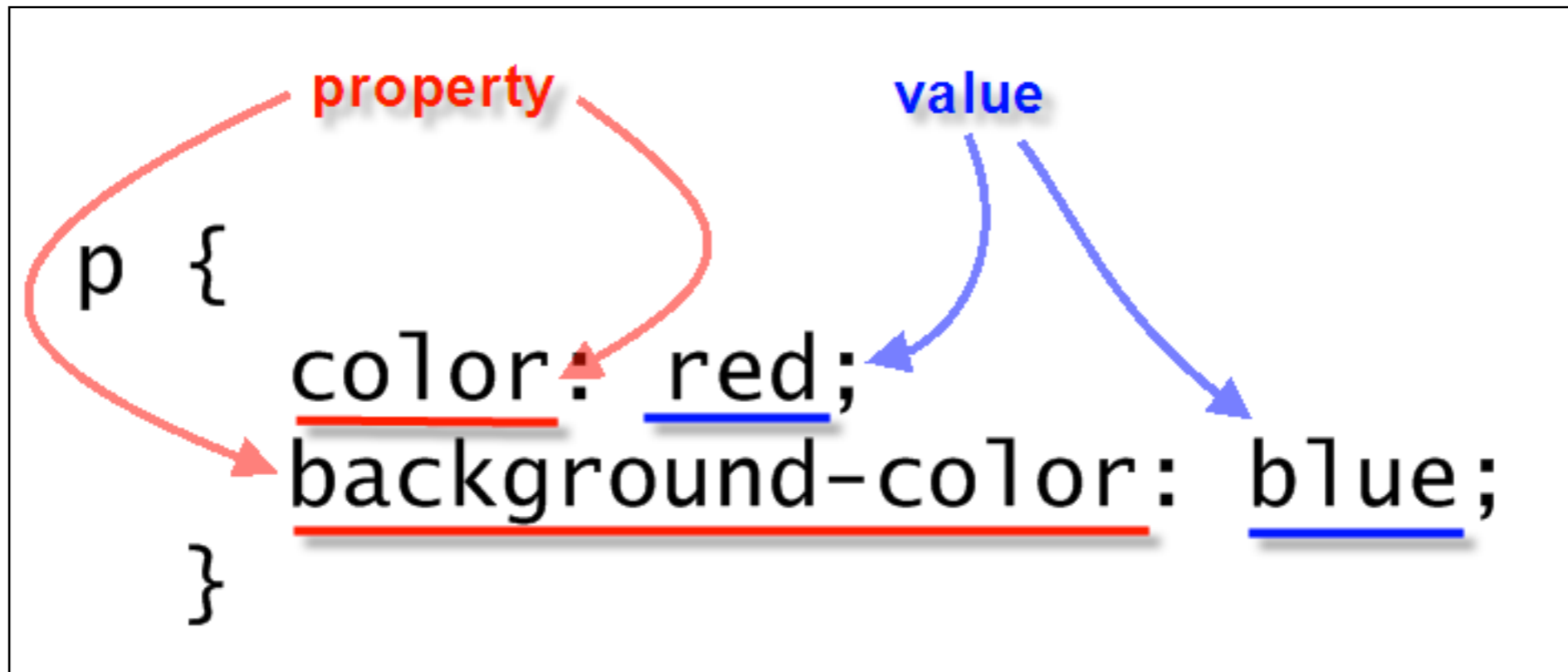
```
p {  
    color: red;  
    background-color: blue;  
}
```

CSS
Rule

Selector and Declarations



Properties & Values



Binding Styles to Markup

3 Ways:

1. Style attribute in element

2. `<style />` element in HTML head

3. External CSS document, through the link element in HTML head

1: Style attribute in element

```
<body>  
  <p style="color: black; background-color: teal; padding: 1em;  
    font-family: helvetica, sans-serif; text-align: justify; margin: 2em;">  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec  
    facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci. Fusce velit.  
    Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis  
    ante sit amet urna. Maecenas condimentum hendrerit turpis.  
  </p>  
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci. Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

2: <style /> element in HTML head

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
  <style type="text/css">
    p {
      color: black;
      background-color: teal;
      padding: 1em;
      font-family: helvetica, sans-serif;
      text-align: justify;
      margin: 2em;
    }
  </style>
</head>
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec
    facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci. Fusce velit.
    Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis
    ante sit amet urna. Maecenas condimentum hendrerit turpis.
  </p>
</body>
</html>
```

3: External CSS Document

```
p
{
  color: black;
  background-color: teal;
  padding: 1em;
  font-family: helvetica, sans-serif;
  text-align: justify;
  margin: 2em;
}
```

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Insert title here</title>
  <link href="example.css" type="text/css" rel="stylesheet"/>
</head>
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec
    facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci. Fusce velit.
    Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis
    ante sit amet urna. Maecenas condimentum hendrerit turpis.
  </p>
</body>
</html>
```

Combining Rules

- Rules can be combined. The following two sets of style rules would produce identical results
- Rules can be listed separately:
- Or, rules can be grouped. Property:Value pairs need to be separated by a semicolon.

```
p {color: black;}
p {background-color: teal;}
p {padding: 1em;}
p {margin: 1em;}
p {font-family: helvetica, sans-serif;}
p {text-align: justify;}
```

```
p
{
  color: black;
  background-color: teal;
  padding: 1em;
  margin: 1em;
  font-family: helvetica, sans-serif;
  text-align: justify;
}
```

Combining Selectors

- Selectors can be combined into comma-separated groups.
- We combine the selectors so that a single declaration applies to multiple selectors.

```
h1 { color: maroon; }  
h2 { color: maroon; }  
h3 { color: maroon; }  
h4 { color: maroon; }  
h5 { color: maroon; }  
h6 { color: maroon; }
```

```
h1, h2, h3, h4, h5, h6 { color: maroon; }
```

Problem

```
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci
    nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p>
    Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
    commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
    hendrerit turpis.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
  <p>
    Lorem ipsum dolor sit amet, <span>consectetur adipiscing elit</span>. Cras
    sollicitudin, orci
    nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
</body>
```

- How to style these paragraphs differently?
- Just using p as the selector will set the style for them all.

```
p
{
  color: black;
  background-color: teal;
}
```

Recap: Attributes

Attributes define information about elements

```
<ElementName> - start tag  
    Attribute(s)  
    Content  
</ElementName> - end tag
```

```
<a href="mydvdsnews.html">  
  news  
</a>
```

```

```

Span element

“For an element that offers no semantic information about the content inside and also provides no styling change, or any other visual change to speak of, the lowly span element is one of the most useful elements in your HTML toolbox.”

- When you wrap text with an opening `` and closing ``, you’re simply providing a hook—one that allows you to add styles later
- Span is an *inline* element

```
<p>  
  Lorem ipsum dolor sit amet,<span>consectetuer adipiscing elit</span>. Cras  
  sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut  
  sagittis enim velit at orci.  
</p>
```

Common Attributes

- Common attributes that may be applied to all the HTML elements
- May be used for the purposes of:
 - styling elements with CSS
 - for scripting purposes
 - to identify language of web page
 - specify “tooltip” text
 - to identify elements uniquely

class Attribute

- While the name of an element specifies its *type*, the class attribute lets you assign to it one or more *subtypes*.
- These subtypes may then be used in CSS code for styling purposes.
- In particular, several elements may share a class and only those elements can be identified for restyling

Using **class** to identify elements

```
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci
    nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p class="withstyle">
    Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
    commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
    hendrerit turpis.
  </p>
  <p class="warn">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
  <p>
    Lorem ipsum dolor <span class="warn">consectetur adipiscing elit</span>.
    Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis
    enim velit at orci.
  </p>
  <p class="warn">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
</body>
```

Using Classes in CSS

- Class names are referenced in CSS as
 - element.classname

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

p.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

p.warn
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

Classes Independent of Elements

- May not have an element name preceding the period:

`.classname`

- Selector now matches *any* element of the given class

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

.warn
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

IDs

- Ids play a similar role, it is a common attribute.
- Provides a unique identifier for an element within the document.
- The most important aspect of the id attribute is that it must be absolutely unique.
- Unlike the class attribute, which may apply the same value to many elements in a page, an id that's applied to an element must not match an id used anywhere else on the same page.

```
<p id="mainpoint">
  Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
  commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
  hendrerit turpis.
</p>
```

```
#mainpoint
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

- ‘#’ instead of ‘.’ in CSS

Rules, Classes, Elements - which ones get selected?

?

Rules, Classes, Elements - which ones get selected?

(1) *Explicit Match*: Do any selectors select your element?

- Examine CSS rules for *explicit match* for element.

(2) *Inheritance Match*: What if no rules match the element:

- Rely on *inheritance*.
- Look at the element's parents, and parents' parents, and so on, until you find the property defined.

(3) *Default Match*: still no explicit or inherited match

- use the *default* value defined by the browser

(4) *Most Specific Match*: What if more than one match (of different rules)?

- select rule that is the *most specific*

Examples

```
p { color: black; }
```

Here's a rule that selects any old paragraph element.

```
.greentea { color: green; }
```

This rule selects members of the greentea class. That's a little more specific.

```
p.greentea { color: green; }
```

And this rule selects only paragraphs that are in the greentea class, so that's even more specific.

```
p.raspberry { color: blue; }
```

```
p.blueberry { color: purple; }
```

These rules also select only paragraphs in a particular class. So they are about the same in specificity as the p.greentea rule.

Specific & Ordering Match

- If we had an element that belonged only to the greentea class
 - there would be an obvious winner:
 - the p.greentea selector is the most specific, so the text would be green.
- But you have an element that belongs to all three classes: greentea, raspberry, and blueberry.
 - So, p.greentea, p.raspberry, and p.blueberry all select the element, and are of equal specificity. What do you do now?
 - You choose the one that is listed last in the CSS file.

5: Ordering Match: - If you can't resolve a conflict because two selectors are equally specific

- use the ordering of the rules in your style sheet file. That is, you use the rule listed last in the CSS file (nearest the bottom). And in this case, that would be the p.blueberry rule.

Example

```
p { color: black; }
.greentea { color: green; }
p.greentea { color: green; }
p.raspberry { color: blue; }
p.blueberry { color: purple; }
```

1- Explicit Match

2- Inheritance Match

```
<p>
  My Normal Tea <br>
  Customers say they <q>really like</q> this one!
</p>
```

```
<blockquote>
  All of the best teas
</blockquote>
```

```
<p class="greentea">
  My Green Tea
</p>
```

```
<p class="greentea blueberry">
  My Mixed Tea - what colour is it?
</p>
```

3- Default Match

4- Most Specific Match

5- Ordering Match

```
p          { color: black; }
.greentea  { color: green; }
p.greentea { color: green; }
p.raspberry { color: blue; }
p.blueberry { color: purple; }
```

```
<p>
  My Normal Tea <br>
  Customers say they <q>really like</q> this one!
</p>
<blockquote>
  All of the best teas
</blockquote>
<p class="greentea">
  My Green Tea
</p>
<p class="greentea blueberry">
  My Mixed Tea - what colour is it?
</p>
```

My Normal Tea
Customers say they "really like" this one!

All of the best teas

My Green Tea

My Mixed Tea - what colour is it?

CSS Properties: Categories

- Box Properties

- Layout Properties

- List Properties

- Table Properties

- Color and Backgrounds

- Typographical Properties

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

p.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

p.warn
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

CSS Selectors Summary (simple)

Selector	Applies to
p	All paragraphs in the document
.about	All elements in the document with a class value of about
#corporatehistory	The element in the document with an id value of corporate history (if present)
h1,h2,h3	All first-, second-, and third-level headings in the document
.privacy, .copyright	All elements with a class of privacy or copyright
#header,#footer	The element assigned an id of header, and the element assigned an id of footer
p.footnote	All paragraphs assigned a class of footnote

CSS Selectors Summary (advanced)

Selector	Applies to
#bodycopy.usergenerated	An element that has been assigned both an id of bodycopy and a class of usergenerated
.navigation a	All links with an ancestor parent assigned a class of navigation
#primarynavigation li.current	All list items with a class of current and an ancestor parent with an id of primarynavigation
.about #bodycopy	Any element on the site with an id of bodycopy and an ancestor parent assigned a class of about
body#personalproducts, body#proproducts, body#enterpriseproducts	The body elements within the site assigned the ids personalproducts, proproducts, and enterpriseproducts
body#personalproducts #bodycopy, body#proproducts #bodycopy, body#enterpriseproducts #bodycopy	The elements assigned an id of bodycopy, within the documents suggested by the previous example
ol li ol li ol li	A list item in the third level of a nested ordered list



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

