CSS selectors

CSS Selectors

CSS style rules can be configured for:

✓ 1. HTML element selector
2. class selector
3. id selector
4. descendant selector
5. pseudo selector
6. grouped
7. compound

HTML element selectors

HTML

```
<p>blah blah blah</p>
```

CSS

```
p {
        font-weight: bold;
        color: red;
}
```

Result: ALL <p> elements are styled

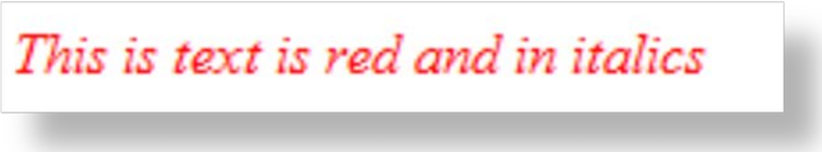
Using CSS with “class”

- class Selector
  - Apply a CSS rule to a certain "class" of elements on a web page
  - Does not associate the style to a specific HTML element
- Configure with .classname
  - code CSS to create a class called “wow” with red italic text.
- Apply the class:

<p class=“wow”>This is text is red and in italics</p>

CSS

```
.wow {
            color: #FF0000;
            font-style: italic;
}
```



This is text is red and in italics

Class selectors

HTML

```
<p class=“loud”>
        blah blah blah
</p>
```

CSS

```
.loud {
        font-weight: bold;
        color: red;
}
```

Result: elements that have the “loud” class applied are styled; can be used multiple times in the HTML document and combined with other classes

Using CSS with “id”

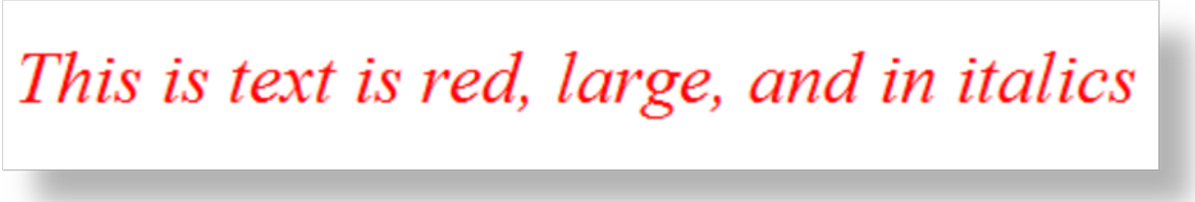
- id Selector
  - Apply a CSS rule to ONE element on a web page.
- Configure with #idname
  - Code CSS to create an id called “top” with red, large, italic text.

CSS

```
#top { color: #FF0000;
          font-size:2em;
          font-style: italic;
}
```

- Apply the id:

```
<p id=“top”>This is text is red, large, and in italics</p>
```



This is text is red, large, and in italics

ID selectors

HTML

```
<p id=“loud”>
        blah blah blah
</p>
```

CSS

```
#loud {
        font-weight: bold;
        color: red;
}
```

Result: <p> elements that have the “loud” ID applied are styled; can be used once only in any HTML document

CSS Descendant Selector

- Specify an element within the context of its container (parent) element.
- AKA contextual selector

CSS

```
.content p {
            color: #00ff00;
}
```

- The example configures a green text color only for p tags located *within* an element assigned to the class named *content*
- Advantage of contextual selectors:
  Reduces the number of classes and ids you need to apply in the HTML

Descendant selectors

HTML

```
<header>
        <p>blah blah blah</p>
</header>

<p>blah blah more blah</p>
```

CSS

```
header p {
        font-weight: bold;
        color: red;
}
```

Result: <p> elements are styled only if they appear nested inside a <header> element

Descendant Selectors

To be more-specific, separate selectors with a space...

CSS

```
.main-menu ul {
      /* target only unordered lists that appear nested inside an
         element with the class “main-menu” */
      margin: 0;
      padding-left: 0;
}
```

HTML

```
<nav class="main-menu">
      <ul>
      ...
```

Descendant Selectors are very important when building navigation menus!

CSS Pseudo-classes

- Pseudo-classes and the anchor element
  - **link** – default state for a hyperlink
  - **visited** – a hyperlink that has been visited
  - **focus** – triggered when the hyperlink has focus
  - **hover** – triggered when the mouse moves over the hyperlink
  - **active** – triggered when the hyperlink is being clicked

```
a:link    {color:#000066;}
a:visited {color:#003366;}
a:focus   {color:#FF0000;}
a:hover   {color:#0099CC;}
a:active  {color:#FF0000;}
```

CSS Structural Pseudo-classes

Pseudo-class	Purpose
:first-of-type	Applies to the first element of the specified type
:first-child	Applies to the first child of an element (CSS2 selector)
:last-of-type	Applies to the last element of the specified type
:last-child	Applies to the last child of an element
:nth-of-type(n)	Applies to the “nth” element of the specified type Values: a number, odd, or even

Pseudo Class Selectors

The basics

```
a:link { color: red; }      /* unvisited link; default */
a:visited { color: purple;}/* visited link */
a:hover { color: green; }  /* mouse over link */
a:active { color: blue; }  /* selected link */
```

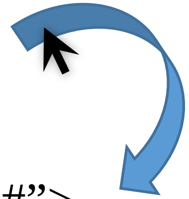
More

```
p:first-child {/* Every <p> element that is the first child of its parent
*/}
p:last-child  {/* Every <p> element that is the last child of its parent
*/}
p:before {
     /* Insert content before every <p> element */
     content: "- start here: ";

}
p:after {
     /* Insert content after every <p> element */
     content: "! ...the end.";

}
```

Pseudo selectors

HTML

```
<a href=“#”>
        blah blah blah
</a>
```



```
<a href=“#”>
        blah blah blah
</a>
```

CSS

```
a { color: blue; }
a:hover { color: red; }
```

Result: <a> elements are styled, and when hovered with a mouse the other style is applied

Pseudo selectors (continued)

HTML

```
<p>blah blah blah</p>
<p>blah blah blah</p>
<p>blah blah blah</p>
<p>blah blah blah</p>
```

CSS

```

p:first-child {
        font-weight: bold;
        color: red;
}
```

Result: the first occurrence of <p> elements are styled

Grouped Selectors

To style multiple selectors at once…

CSS

```
.lead, h1, h2 {
     font-weight: normal;
}
```

Compound Selectors

Normal classes in CSS

CSS

```
.lead {
  //target all elements with the
    class “lead”
  font-size: 18px;
}
```

HTML

Works on any element that uses “lead”…

```
<div class="lead">...
```

--or--

```
<p class="lead">...
```

A compound selector in CSS

To be more-specific, combine selectors with element targets with no space in-between…

CSS

```
p.lead {
  //target only paragraphs
    with the class “lead”
color: #333333;
}
```

HTML

Only…

```
<p class="lead">...
```