

AN INTRODUCTION TO CSS 2.1 & CSS 3

Who am I?

It's nice to meet you all

- I'm Marc
- My wife and I live in Beverly
- I started designing and coding back in 2000
- I created Boston Web Studio in 2005
- I co-created the Build Guild last summer

A Few Questions

How many of you...

- Have written CSS before?
- Write CSS once or twice a **month**?
- Write CSS once or twice a **week**?
- Write CSS once or twice a **day**?
- Are writing CSS right now?
- Hate raising your hand?

The History of CSS

The new stuff is old & it's out of order

- **CSS 1** became a W3C recommendation in December of 1996 <http://www.w3.org/TR/CSS1/>
- **CSS 2** became a W3C recommendation in May of 1998 <http://www.w3.org/TR/2008/REC-CSS2-20080411/>
- **CSS 3** became a W3C working draft in April of 2000 <http://www.w3.org/TR/2000/WD-css3-selectors-20000410>
- **CSS 2.1** became a W3C working draft in August of 2002 <http://www.w3.org/TR/2002/WD-CSS21-20020802/>

Disclaimer(s)

Setting expectations & protecting my butt

- This is an introduction to CSS 2.1 and CSS 3
- CSS 2.1 and CSS 3 are not fully supported
- Nearly everything you'll see will not work in Internet Explorer 6
- We don't like Internet Explorer 6
- Or 7
- 8 is OK

Lets See Some Examples

We'll see examples of CSS that will...

1. Improve element and content targeting
2. Replace outdated techniques for achieving certain design trends (lets stop making rounded corners and drop-shadows with images, eh?)
3. Mimic some actions and behaviors currently found in JavaScript

Lets See Some Examples

We'll see examples of CSS that will...

1. Improve element and content targeting
2. Replace outdated techniques for achieving certain design trends (lets stop making rounded corners and drop-shadows with images, eh?)
3. Mimic some actions and behaviors currently found in JavaScript

Child Selector >

Select any element that's a child of another element

```
#childSelector > p {  
  color: red;  
}
```

```
<div id="childSelector">  
  <p>This content is targeted.</p>  
  <ul>  
    <li>This content is NOT targeted.</li>  
    <li>This content is NOT targeted.</li>  
  </ul>  
  <blockquote>  
    <p>This content is NOT targeted.</p>  
  </blockquote>  
  <p>This content is targeted.</p>  
  <p>This content is targeted.</p>  
</div>
```

This content is targeted.

- This content is NOT targeted
- This content is NOT targeted

This content is NOT targeted.

This content is targeted.

This content is targeted.



Adjacent Sibling Selector +

Select an element that's a direct sibling of another element

```
blockquote + p {  
  color: red;  
}
```

```
<p>This content is NOT targeted.</p>  
<blockquote>  
  <p>This content is inside of a blockquote.</p>  
</blockquote>  
<p>This content is targeted.</p>  
<p>This content is NOT targeted.</p>
```

This content is NOT targeted.

This content is inside of a blockquote.

This content is targeted.

This content is NOT targeted.

 7  8

 2  3  3.5

 3  3.1  4

 9

General Sibling Selector ~

Select an element that's a sibling of another element

```
blockquote ~ p {  
  color: red;  
}
```

```
<p>This content is NOT targeted.</p>  
<blockquote>  
  <p>This content is inside of a blockquote.</p>  
</blockquote>  
<p>This content is targeted.</p>  
<p>This content is targeted.</p>  
<div>  
  <p>This content is NOT targeted.</p>  
</div>  
<p>This content is targeted.</p>
```

This content is NOT targeted.

This content is inside of a blockquote.

This content is targeted.

This content is targeted.

This content is NOT targeted.

This content is targeted.

 7  8

 2  3  3.5

 3  3.1  4

 9

Attribute Selector [name=value]

Select an element that has a particular attribute & value

```
input[type=text] {  
  border-color: red;  
}
```



```
<input type="password" name="one" />  
<input type="text" name="two" />
```



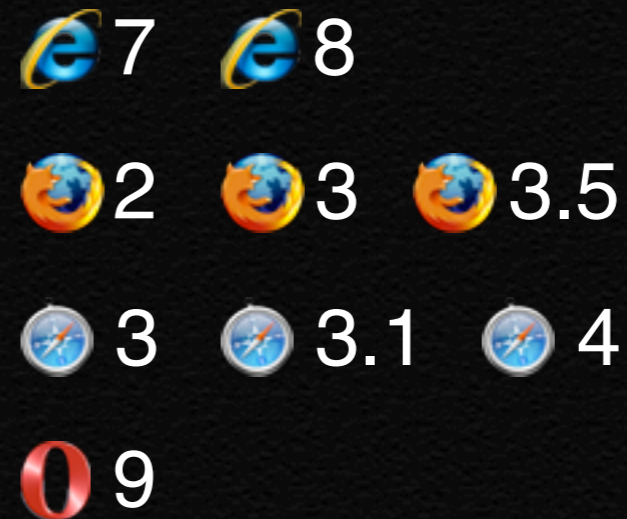
First Child Pseudo-class :first-child

Select only the first child element among many of its siblings

```
ul li:first-child {  
  color: red;  
}
```

- This content is targeted
- This content is NOT targeted
- This content is NOT targeted
- This content is NOT targeted
- This content is NOT targeted

```
<ul>  
  <li>This content is targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
</ul>
```



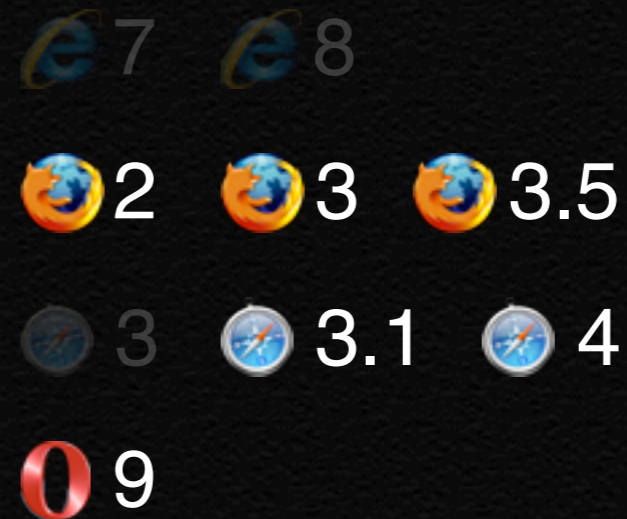
Last Child Pseudo-class :last-child

Select only the last child element among many of its siblings

```
ul li:last-child {  
  color: red;  
}
```

- This content is NOT targeted
- This content is NOT targeted
- This content is NOT targeted
- This content is NOT targeted
- This content is targeted

```
<ul>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is NOT targeted</li>  
  <li>This content is targeted</li>  
</ul>
```



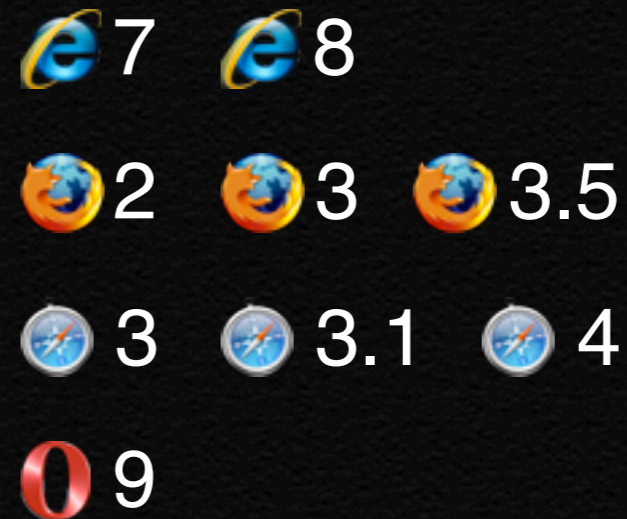
First Letter Pseudo-element :first-letter

Select the first letter in any element

```
p:first-letter {  
  color: red;  
  font-size: 2em;  
}
```

```
<p>The first letter in this paragraph is  
targeted.</p>
```

The first letter in this paragraph is targeted.



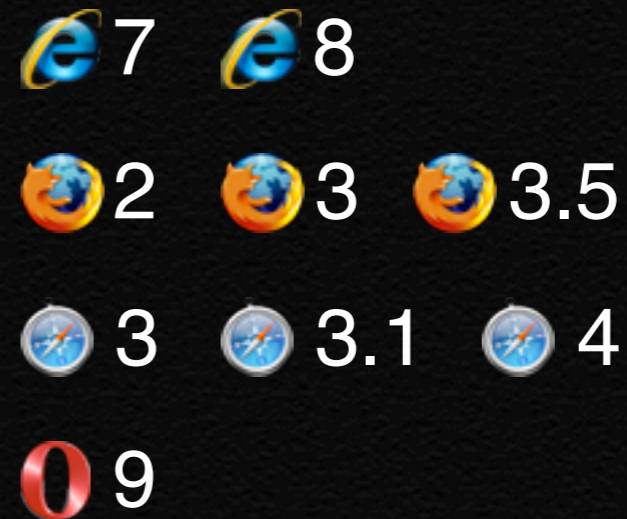
First Line Pseudo-element :first-line

Dynamically select the first line of text in any element

```
p:first-line {  
  color: red;  
}
```

`<p>`The first line of this paragraph is targeted. Resize your browser window and watch how, regardless of where words move, only those words found in the first line remain targeted. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.`</p>`

The first line of this paragraph is targeted. Resize your browser window and watch how, regardless of where words move, only those words found in the first line remain targeted. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.



Enabled, Disabled, & Checked Pseudo-classes

:enabled, :disabled, :checked

Change the appearance of form inputs based on their status

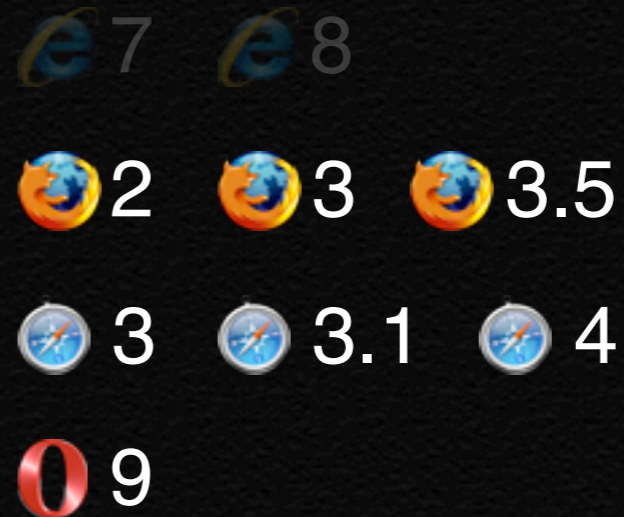
```
:enabled {  
  border-color: green;  
}  
:disabled {  
  border-color: red;  
}  
:checked {  
  margin-left: 50px;  
}
```

```
<input type="text" value="This input is  
enabled" />  
<input type="text" disabled="disabled"  
value="This input is disabled" />  
<label>  
  <input type="checkbox" checked="checked" />  
  This checkbox is initially checked  
</label>
```

This input is enabled

This input is disabled

This checkbox is initially checked



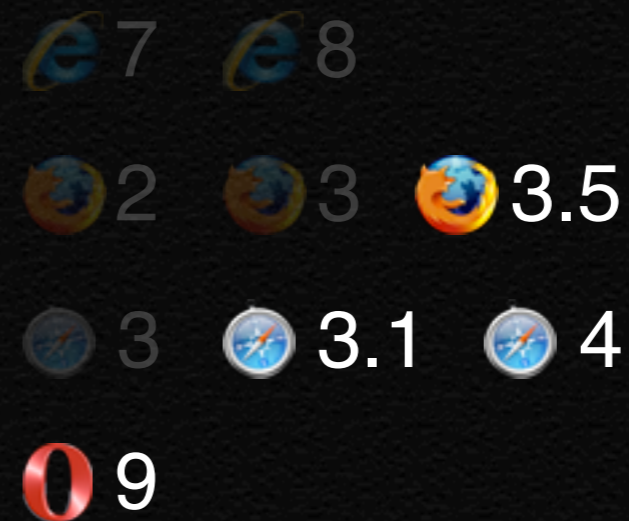
nth of Type Pseudo-class :nth-of-type()

Select multiple instances of sibling elements based on a formula

```
tbody tr:nth-of-type(2n+1) td {  
  background-color: gray;  
}
```

```
<table>  
  <thead>  
    <tr>  
      <th scope="col">Monday</th>  
      <th scope="col">Tuesday</th>  
      <th scope="col">Wednesday</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>  
      <td>14</td>  
      <td>234</td>  
      <td>54</td>  
    </tr>  
    <tr>  
      <td>15</td>  
      <td>236</td>  
      <td>51</td>  
    </tr>  
    ...and so on...
```

Monday	Tuesday	Wednesday
14	234	54
15	236	51
18	230	53
14	233	50
14	236	55



Lets See Some Examples

We'll see examples of CSS that will...

- ✓ Improve element and content targeting
- 2. Replace outdated techniques for achieving certain design trends (lets stop making rounded corners and drop-shadows with images, eh?)
- 3. Mimic some actions and behaviors currently found in JavaScript

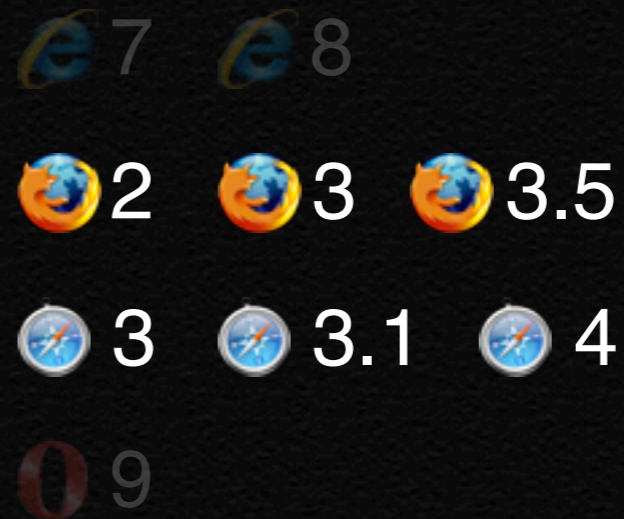
Border Radius Property `border-radius`

Create rounded-corners on any block-level element

```
p {  
  border: 5px solid red;  
  padding: 1em;  
  -moz-border-radius: 10px;  
  -webkit-border-radius: 10px;  
}
```

`<p>`This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.`</p>`

This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.



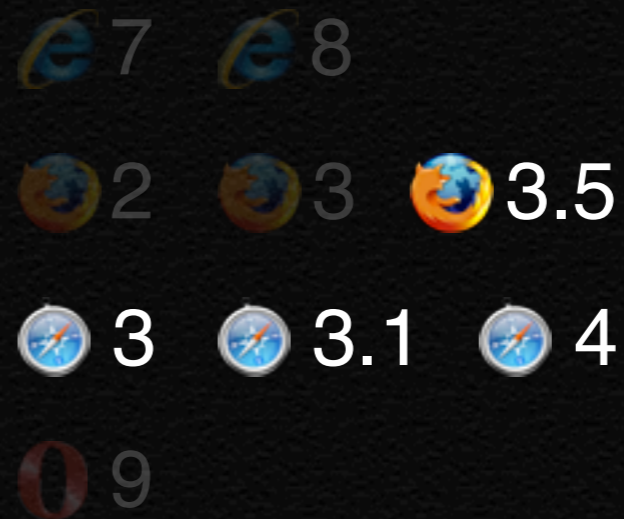
Box Shadow Property `box-shadow`

Create a drop-shadow behind any block-level element

```
p {  
  background-color: black;  
  padding: 1em;  
  -moz-box-shadow: 3px 5px 15px white;  
  -webkit-box-shadow: 3px 5px 15px white;  
}
```

`<p>`This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. `</p>`

This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.



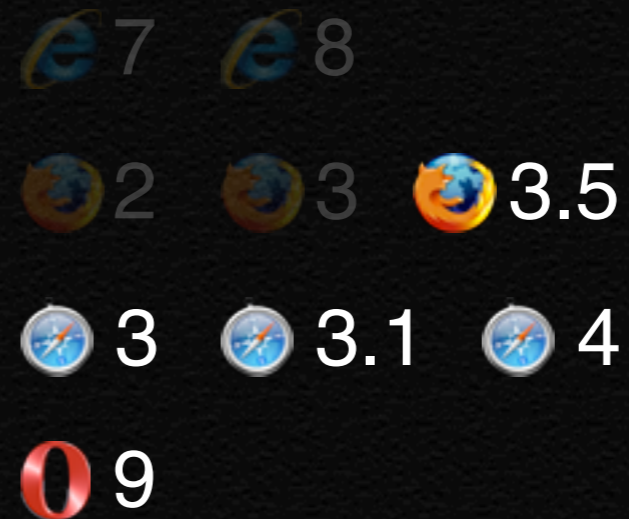
Text Shadow Property `text-shadow`

Create a drop-shadow behind text

```
p {  
  background-color: white;  
  color: white;  
  font-size: 2.5em;  
  font-weight: bold;  
  text-shadow: 2px 2px 6px black;  
}
```

This content has a
text-shadow.

`<p>This content has a text-shadow.</p>`

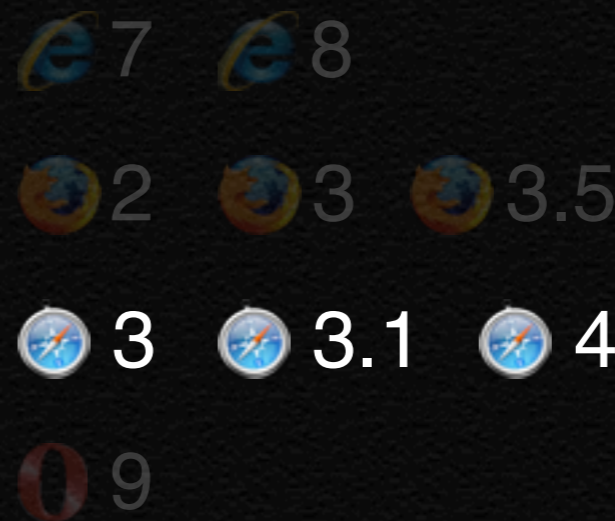
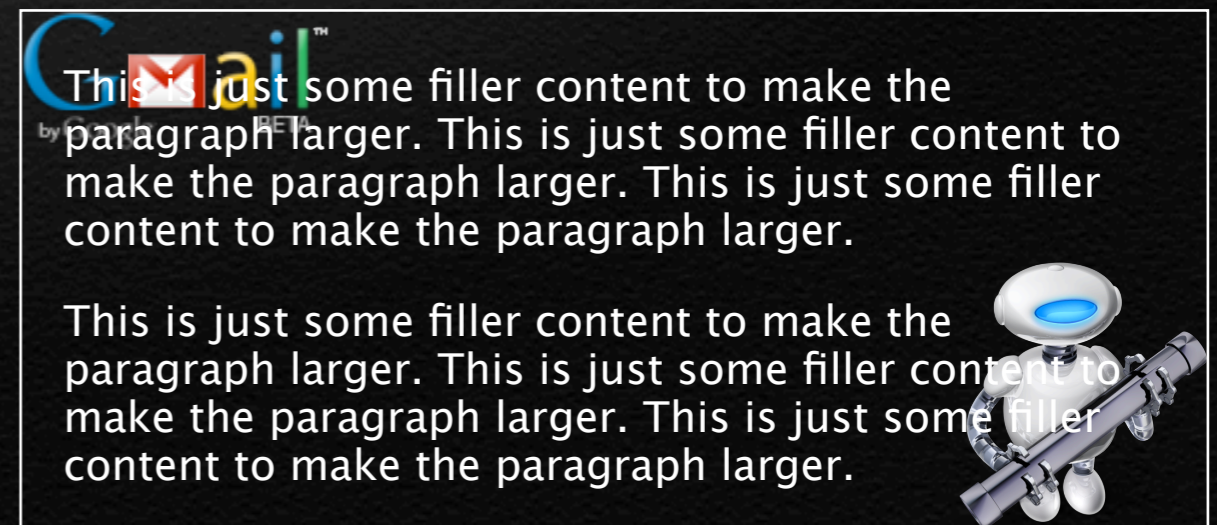


Multiple Background Images

Use more than one background image in an element

```
#multipleBackgroundImages {  
  background-image: url(/images/gmail.png),  
                  url(/images/automator.gif);  
  background-position: top left,  
                    bottom right;  
  background-repeat: no-repeat;  
  border: 1px solid white;  
  padding: 0 1em;  
}
```

```
<div id="multipleBackgroundImages">  
  <p>This is just some filler content to  
  make the paragraph larger. This is just  
  some filler content to make the paragraph  
  larger. This is just some filler content  
  to make the paragraph larger.</p>  
  <p>This is just some filler content to  
  make the paragraph larger. This is just  
  some filler content to make the paragraph  
  larger. This is just some filler content  
  to make the paragraph larger.</p>  
</div>
```



Lets See Some Examples

We'll see examples of CSS that will...

- ✓ Improve element and content targeting
- ✓ Replace outdated techniques for achieving certain design trends (lets stop making rounded corners and drop-shadows with images, eh?)

3. Mimic some actions and behaviors currently found in JavaScript

Column Properties

column-width & column-gap

Automatically create columns

```
#columnProperty {
  -moz-column-gap: 2em;
  -moz-column-width: 10em;
  -webkit-column-gap: 2em;
  -webkit-column-width: 10em;
}

<div id="columnProperty">
  <p>These three paragraphs are inside of a
  single div that has column properties applied
  to it. Look closely, there is no floating or
  positioning going on here.</p>
  <p>This is just some filler content to make
  the paragraph larger. This is just some
  filler content to make the paragraph larger.
  This is just some filler content to make the
  paragraph larger. This is just some filler
  content to make the paragraph larger.</p>
  <p>This is just some filler content to make
  the paragraph larger. This is just some
  filler content to make the paragraph larger.
  This is just some filler content to make the
  paragraph larger.</p>
</div>
```

These three paragraphs are inside of a single div that has column properties applied to it. Look closely, there is no floating or positioning going on here.

This is just some filler content to

make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.

paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.



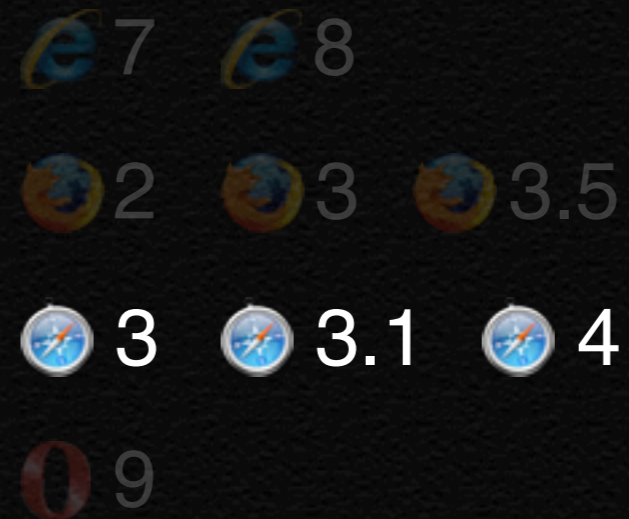
Resize Property `resize`

Allow an element to be resized by the visitor

```
p {  
  overflow: auto;  
  resize: both;  
  width: 325px;  
}
```

Browser demo time!

`<p>`This paragraph can be resized both horizontally and vertically. It looks like a text input field, but it's not.`</p>`



Text Overflow Property `text-overflow`

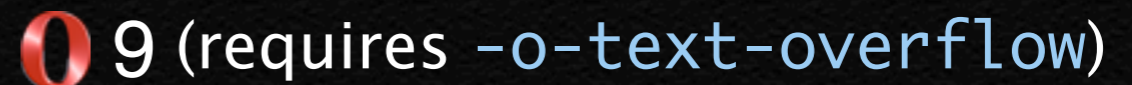
Characters that approach the edge of an element become an ellipsis

```
p {  
  border: 1px solid white;  
  overflow: hidden;  
  padding: 1em;  
  text-overflow: ellipsis;  
  white-space: nowrap;  
}
```

`<p>`This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger. This is just some filler content to make the paragraph larger.`</p>`

This is just some filler content to make the paragra...

Browser demo time!



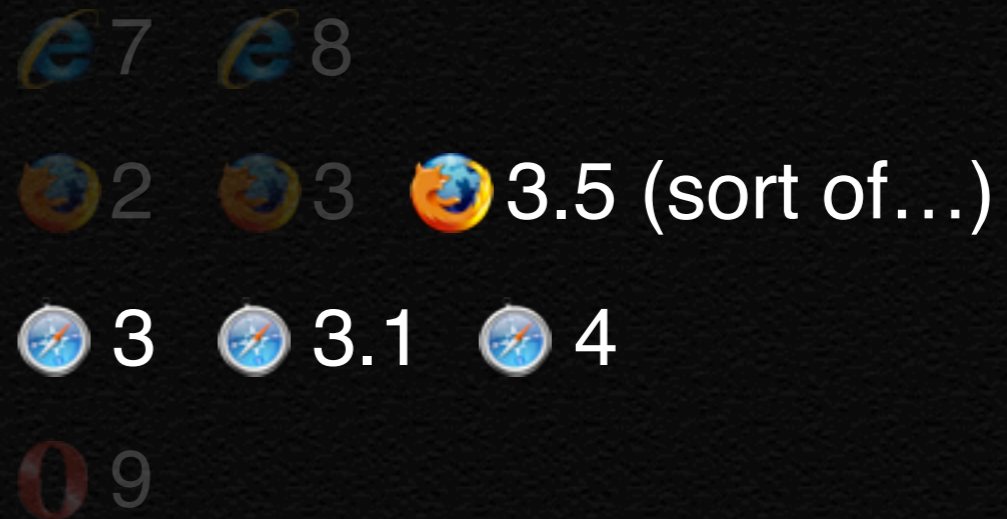
Transition Property -webkit-transition

Basic animation, with CSS

```
p {  
  border: 1px solid black;  
  float: left;  
  margin-right: 2em;  
  padding: 1em;  
  text-align: center;  
  width: 100px;  
  -webkit-transition: all 1s ease-in-out;  
}  
.scale:hover {  
  -moz-transform: scale(2);  
  -webkit-transform: scale(2);  
}  
.slideLeft:hover {  
  -moz-transform: translate(-3em,1em);  
  -webkit-transform: translate(-3em,1em);  
}  
.slideRight:hover {  
  -moz-transform: translate(3em,0);  
  -webkit-transform: translate(3em,0);  
}  
.rotate:hover {  
  -moz-transform: rotate(30deg);  
  -webkit-transform: rotate(30deg);  
}
```

```
<p class="scale">Example 1</p>  
<p class="slideLeft">Example 2</p>  
<p class="slideRight">Example 3</p>  
<p class="rotate">Example 4</p>
```

Browser demo time!



Lets See Some Examples

We'll see examples of CSS that will...

- ✓ Improve element and content targeting
- ✓ Replace outdated techniques for achieving certain design trends (lets stop making rounded corners and drop-shadows with images, eh?)
- ✓ Replace some actions and behaviors currently found in JavaScript

`</fin>`

Conclusion

Allow myself to conclude myself

- Browser support varies, drastically in some cases. Choose wisely based on your audience and the browsers they use.
- That said, a handful of these selectors and properties can be used as “added awesome,” visible only to those who use better browsers without ruining the experience for those using older browsers.

Questions?

(re)sources:

- CSS Current Work
<http://www.w3.org/Style/CSS/current-work>
- Quirksmode (Peter-Paul Koch)
<http://www.quirksmode.org/css/contents.html>
- CSS3.info
<http://www.css3.info>
- My Brain
localhost

links:

- The demonstration page
<http://marcamos.com/presentation/intro-css2.1-css3/>
- My personal website
<http://www.marcamos.com>
- Boston Web Studio
<http://www.bostonwebstudio.com>
- The Build Guild
<http://www.buildguild.org>



</awesomePresentation>