

What you need to know about IE8 and Standards

Aaron Gustafson



easy designs

First, a little history...

1996

IE3 becomes the first browser to support CSS

1997

IE4 adds support
for DHTML
(gentlemen, start your pop-ups)

1998

IE5 for Windows
improves CSS
support

1998 | but gets the box
model wrong

2000

IE5 for Mac
introduces the
DOCTYPE switch

20000

and the most
accurate CSS
engine to date

2000

IE5.5 for Windows
adds even yet still
more CSS support

2000 | but still has the
box model wrong

2001

IE6 adds more CSS support & the DOCTYPE switch

2003 | IE6 begins to gather dust

2004

IE6 is rolled into a corner, where it begins to smell

2005

It appears IE6
will never die;
enter * html

2006

IE7 materializes
with much better
standards support

2006 | and breaks a
bunch of sites

2008

IE8B1 debuts a completely new rendering engine

microsoft.com/windows/products/winfamily/ie/ie8/

2008 | with a focus on being standards- compliant

microsoft.com/windows/products/winfamily/ie/ie8/

So here we are

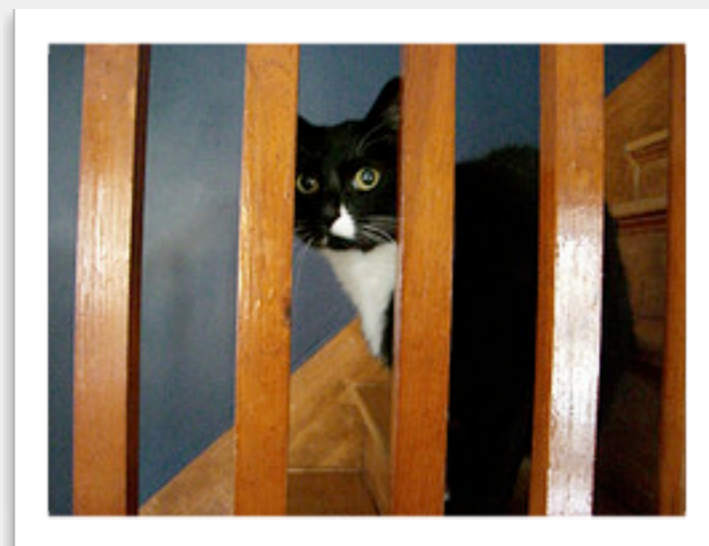
What's been fixed?

hasLayout is history!!!

- ◉ IE8 has a completely new rendering engine built from the ground up with CSS2.1 as its guiding principle.
- ◉ the old layout strategy was responsible for numerous bugs:
 - ▶ content disappearing (and reappearing)
 - ▶ content only half-rendering
 - ▶ floats or positioning causing odd stuff to happen
 - ▶ and much more...
- ◉ for more on **hasLayout**:
<http://msdn.microsoft.com/en-us/library/bb250481.aspx>

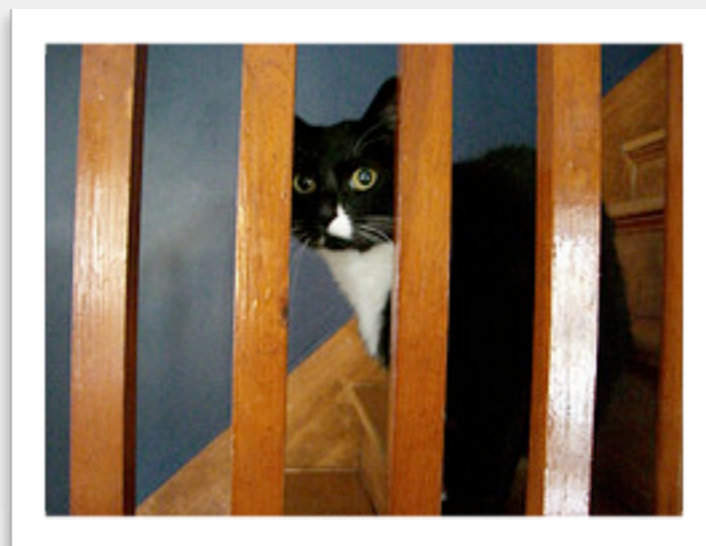
object fallbacks work now

```
<object data="/img/test.tif" type="image/tiff">  
  <object data="/img/test.png" type="image/png">  
    <object data="/img/test.jpg" type="image/jpeg">  
      <p>A photo of my cat.</p>  
    </object>  
  </object>  
</object>
```



object fallbacks work now

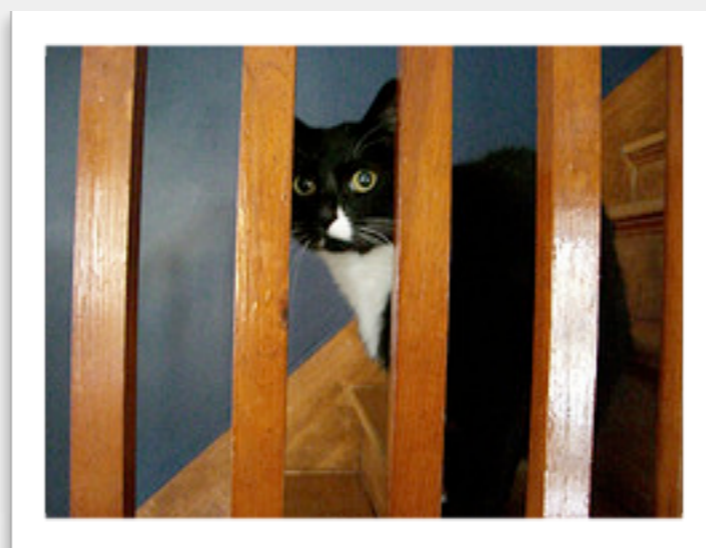
```
<object data="/img/test.tif" type="image/tiff">  
  <object data="/img/test.png" type="image/png">  
    <object data="/img/test.jpg" type="image/jpeg">  
      <p>A photo of my cat.</p>  
    </object>  
  </object>  
</object>
```



test.tif

object fallbacks work now

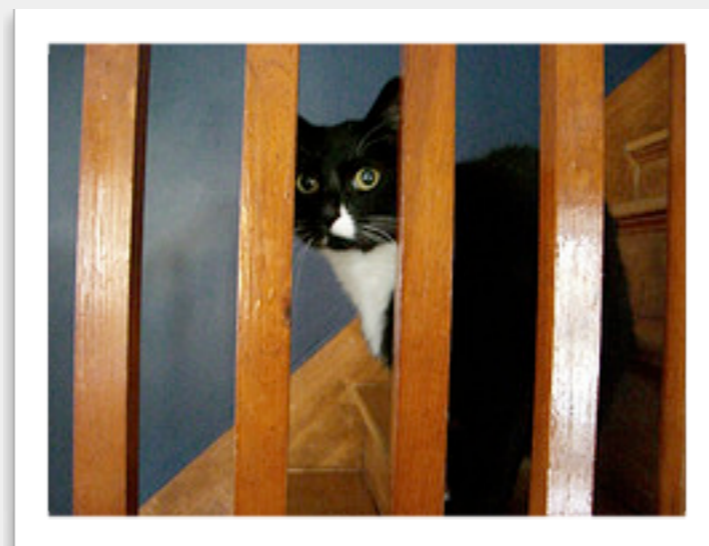
```
<object data="/img/test.tif" type="image/tiff">  
  <object data="/img/test.png" type="image/png">  
    <object data="/img/test.jpg" type="image/jpeg">  
      <p>A photo of my cat.</p>  
    </object>  
  </object>  
</object>
```



test.png

object fallbacks work now

```
<object data="/img/test.tif" type="image/tiff">  
  <object data="/img/test.png" type="image/png">  
    <object data="/img/test.jpg" type="image/jpeg">  
      <p>A photo of my cat.</p>  
    </object>  
  </object>  
</object>
```



test.jpg

object fallbacks work now

```
<object data="/img/test.tif" type="image/tiff">  
  <object data="/img/test.png" type="image/png">  
    <object data="/img/test.jpg" type="image/jpeg">  
      <p>A photo of my cat.</p>  
    </object>  
  </object>  
</object>
```

A photo of my cat.

:hover, :focus & :active*

```
button {  
  background: #ccc;  
  border: 1px solid #333;  
  color: #333;  
}  
button:hover {  
  background: #999;  
}
```

my button

* IE8b1 does not support this, but it is planned for the final release

:hover, :focus & :active*

```
button {  
  background: #ccc;  
  border: 1px solid #333;  
  color: #333;  
}  
button:hover {  
  background: #999;  
}
```



* IE8b1 does not support this, but it is planned for the final release

:hover, :focus & :active*

```
input:focus {  
  background: #fcfab0;  
}  
button:focus {  
  background: #999;  
}
```

Email

Password

* IE8b1 does not support this, but it is planned for the final release

:hover, :focus & :active*

```
input:focus {  
  background: #fcfab0;  
}  
button:focus {  
  background: #999;  
}
```

Email	<input type="text" value="bob@foo.com"/>
Password	<input type="password" value="*****"/>
	<input type="button" value="log in"/>

* IE8b1 does not support this, but it is planned for the final release

:hover, :focus & :active*

```
input:focus {  
  background: #fcfab0;  
}  
input:active {  
  background: #f00;  
}
```

Email

Password

* IE8b1 does not support this, but it is planned for the final release

:hover, :focus & :active*

```
input:focus {  
  background: #fcfab0;  
}  
input:active {  
  background: #f00;  
}
```

Email

Password

* IE8b1 does not support this, but it is planned for the final release

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: decimal-leading-zero;  
}
```

```
01. item 1  
02. item 2  
03. item 3
```

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: lower-greek;  
}
```

```
α. item 1  
β. item 2  
γ. item 3
```

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: lower-roman;  
}
```

- a. item 1
- b. item 2
- c. item 3

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: upper-roman;  
}
```

- A. item 1
- B. item 2
- C. item 3

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: armenian;  
}
```

Ա. item 1
Բ. item 2
Գ. item 3

list-style-type completed

Support for more values:

```
ol {  
  list-style-type: georgian;  
}
```

```
ς. item 1  
δ. item 2  
Ϸ. item 3
```

background-position fixed

Fixed the bug where background positioning set in ems failed to use the font size of the element it was applied to as the baseline against which to calculate the distance

```
p {  
  background: url(img.jpg) 10em 0 repeat-y;  
  font-size: 2em;  
}
```

Spec: |

IE7-: |

white-space completed

Support for additional values:

```
<p>This is the test paragraph.
```

```
    It has all kinds of odd tabs    and spacing in the  
    HTML source code.</p>
```

```
p {  
  background: #999;  
  white-space: pre;  
  width: 100px;  
}
```

```
This is the test paragraph.
```

```
    It has all kinds of odd tabs    and s  
    HTML source code.
```

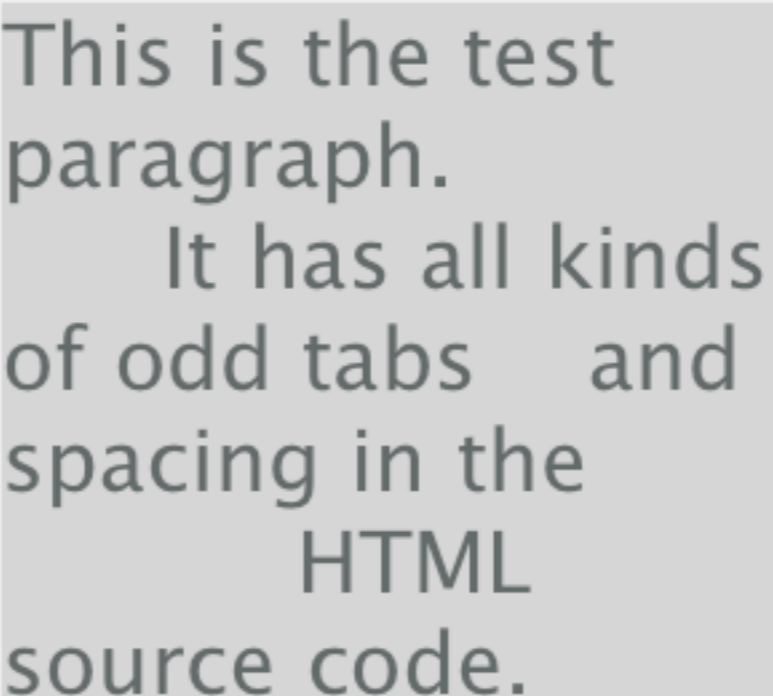

white-space completed

Support for additional values:

```
<p>This is the test paragraph.
```

```
    It has all kinds of odd tabs    and spacing in the  
    HTML source code.</p>
```

```
p {  
  background: #999;  
  white-space: pre-wrap;  
  width: 100px;  
}
```



```
This is the test  
paragraph.  
    It has all kinds  
of odd tabs    and  
spacing in the  
    HTML  
source code.
```

white-space completed

Support for additional values:

```
<p>This is the test paragraph.
```

```
    It has all kinds of odd tabs    and spacing in the  
    HTML source code.</p>
```

```
p {  
  background: #999;  
  white-space: pre-line;  
  width: 100px;  
}
```

```
This is the test  
paragraph.  
It has all kinds of  
odd tabs and  
spacing in the  
HTML source code.
```

word-spacing with ems*

Corrected the error where if an element did not “have layout”, font-size used to compute the value of **word-spacing** is not the one specified on the box itself (as it should be), but the one on the parent

```
p {  
  word-spacing: 1em;  
  font-size: 2em;  
}
```

Spec: This is the test paragraph.

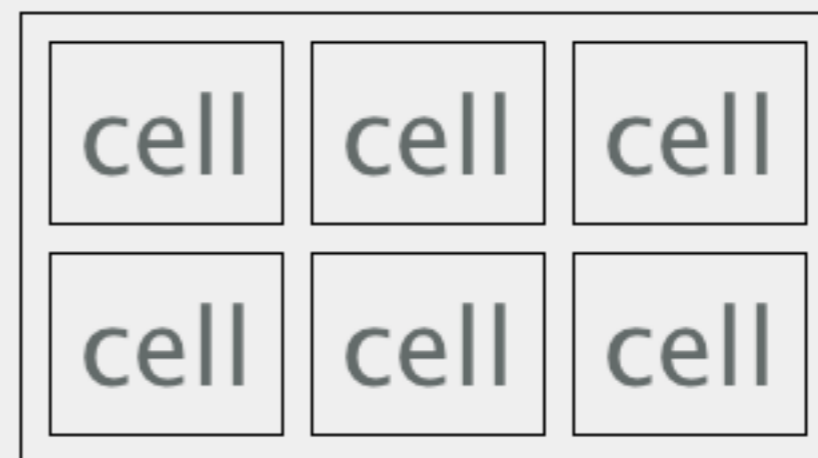
IE7-: This is the test paragraph.

* IE8b1 does not support this, but it is planned for the final release

Table styles

- ◉ Various **border-collapse** fixes
- ◉ Added support for **border-spacing**:

```
table {  
  border: 1px solid #000;  
  border-spacing: 10px;  
}  
td {  
  border: 1px solid #000;  
  padding: 10px;  
}
```



cell	cell	cell
cell	cell	cell

Table styles

- Added support for `border-style: hidden;`

```
table {  
  border: 1px solid #000;  
  border-collapse: collapse;  
}  
td {  
  border: 1px solid #000;  
  padding: 10px;  
}  
tr + tr td + td + td {  
  border-style: hidden;  
}
```

cell	cell	cell
cell	cell	cell

Table styles

- Added support for **caption-side**:

```
table {  
  border: 1px solid #000;  
  border-collapse: collapse;  
  caption-side: bottom;  
}  
td {  
  border: 1px solid #000;  
  padding: 10px;  
}  
caption {  
  font-size: .6em;  
  line-height: 2em;  
}
```

cell	cell	cell
cell	cell	cell

This is a caption

Table styles

- ◉ Added support for **empty-cells***

```
table {  
  border: 1px solid #000;  
  border-spacing: 10px;  
  empty-cells: show;  
}  
td {  
  border: 1px solid #000;  
  padding: 10px;  
}
```

cell		cell
cell	cell	

* IE8b1 does not support this, but it is planned for the final release

Positioning offsets*

Corrected the recalculation errors when using ems to offset an absolutely-positioned element:

```
p a {  
  position: absolute;  
  left: -999em;  
}  
p a:hover {  
  left: 0;  
}
```



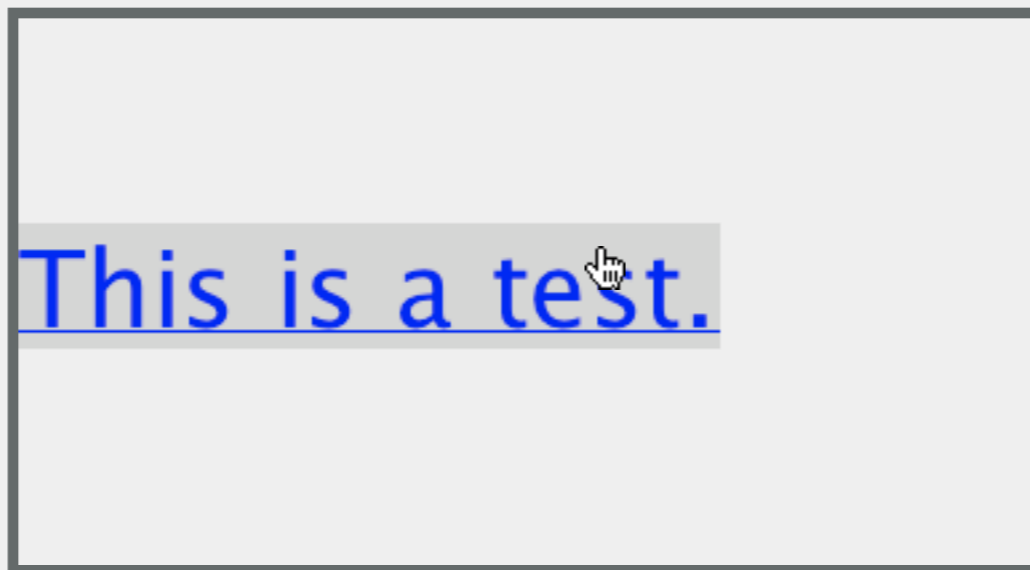
his is a test.

* IE8b1 does not support this, but it is planned for the final release

Positioning offsets*

Corrected the recalculation errors when using ems to offset an absolutely-positioned element:

```
p a {  
  position: absolute;  
  left: -999em;  
}  
p a:hover {  
  left: 0;  
}
```



* IE8b1 does not support this, but it is planned for the final release

Complete display support*

- ◉ `table`, `inline-table`, `table-row`, `table-row-group`, `table-column`, `table-column-group`, `table-cell`, `table-caption`
- ◉ `run-in`:

```
p {  
  display: run-in;  
}  
p + p {  
  display: block;  
}
```

This is a run-in
paragraph.This is a normal
paragraph.

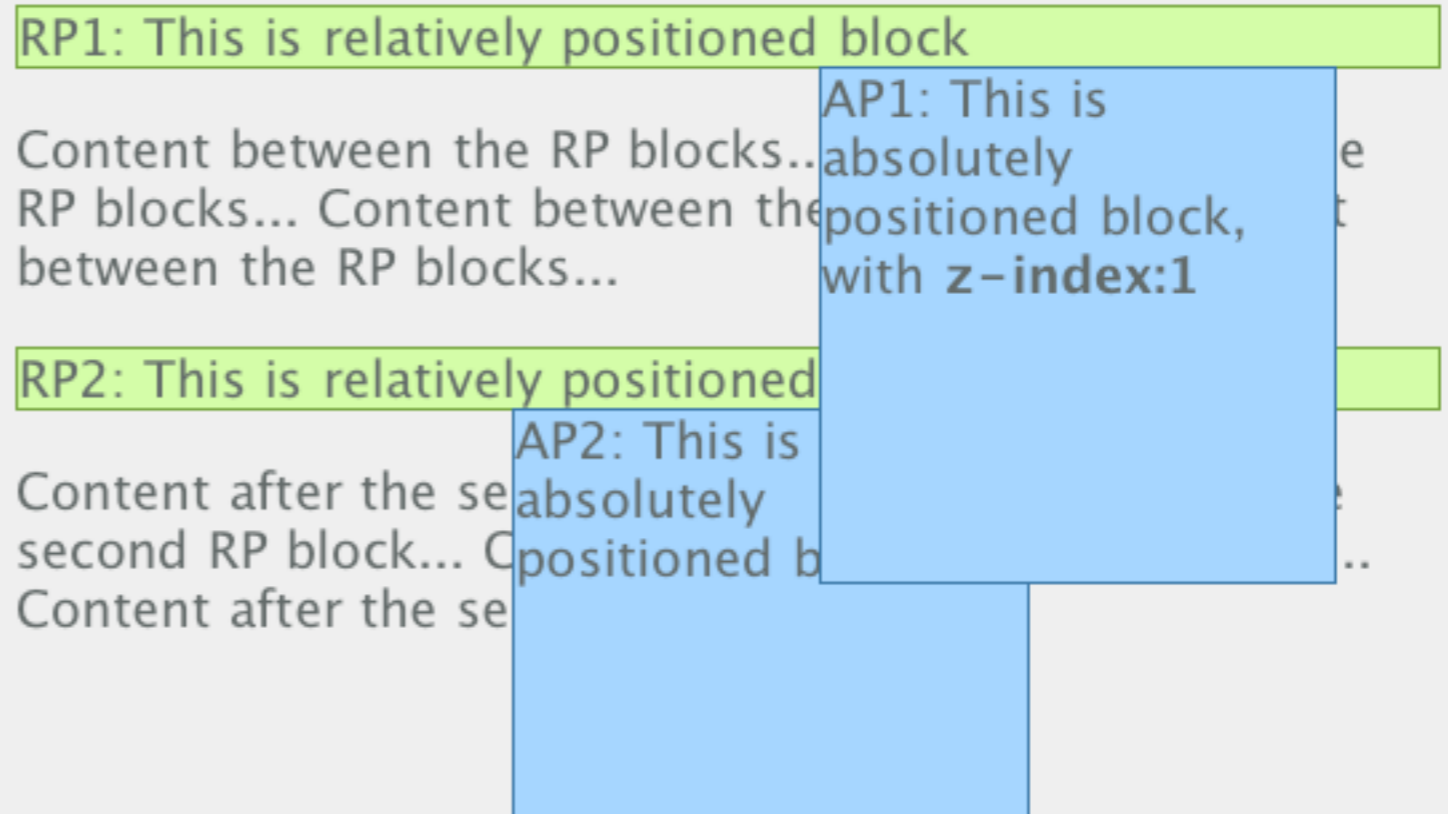
This is a normal
paragraph.

* IE8b1 does not support this, but it is planned for the final release

Mended **z-index**

IE's support for **z-index** incorrectly established a positioning context even in the presence of the "auto" value.

```
div {
  background: #cf9;
  border: 1px solid #693;
  position: relative;
}
div p {
  background: #9cf;
  border: 1px solid #369;
  position: absolute;
  right: 2em;
  width: 10em;
  height: 10em;
  z-index: 1;
}
p + div p {
  right: 8em;
  height: 8em;
  z-index: auto;
}
```



Example adapted from <http://aplus.co.yu/lab/z-pos/>

Fixed inheritance

IE wouldn't inherit a value for a non-inheritable property, even in the presence of the "inherit" keyword

```
div {  
  background: #cf9;  
  border: 1px solid #693;  
  position: relative;  
}  
div p {  
  border: inherit;  
}
```

Spec:

This is some content.

IE7-:

This is some content.

Get what you ask for

```
<form name="foo">First</form>  
<div id="foo">Second</div>
```

Running

```
var el = document.getElementById( 'foo' );  
    el.style.display = 'none';
```

Used to result in the `form` being hidden instead of the `div`.

style is available via DOM...

```
<div id="foo" style="border: 1px solid">Text here.</div>
```

Running

```
var el = document.getElementById( 'foo' );  
    alert( el.attributes['style'].value );  
    alert( el.getAttributeNode('style').value );
```

Both used to return `null`, now they return

```
BORDER-RIGHT: 1px solid; BORDER-TOP: 1px solid; BORDER-LEFT:  
1px solid; BORDER-BOTTOM: 1px solid
```

Note 1: CSS shorthand is expanded and properties are ALL CAPS.

Note 2: the `attributes` array doesn't have a length in IE8b1.

...almost*

```
<div id="foo" style="border: 1px solid">Text here.</div>
```

Running

```
var el = document.getElementById( 'foo' );  
    alert( el.hasAttribute('style') );
```

Returns `false` when it obviously shouldn't.

* This has been submitted as a bug.

Setting `style` works too

Running

```
var el = document.getElementById( 'foo' );  
    el.setAttribute( 'style', 'color: #f00;' );
```

Just remember that it overwrites the **whole** attribute.

Note: IE **does not** support setting event handlers this way:

```
el.setAttribute( 'onclick', 'alert(\'hello\')' );
```

But that's not a huge loss as there are better ways to write event handlers.

Find by **name** now

```
<input type="text" name="foo" value="text" />
```

Running

```
var el = document.getElementsByName( 'foo' )[0];  
    el.style.border = '1px solid #f00';
```

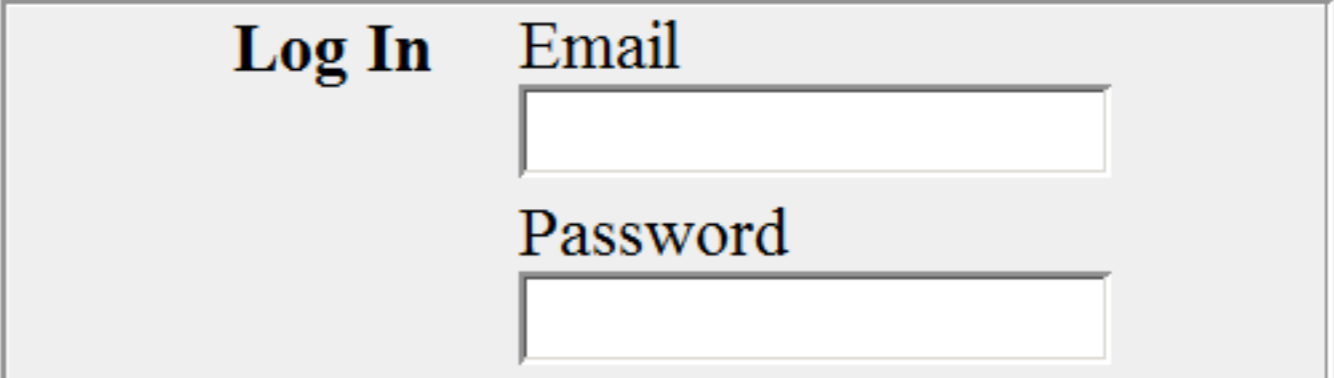
Note: IE's implementation currently ignores custom elements.

Alright, so what's new?

Legend is styleable!!!

The element can now be positioned

```
fieldset {  
  position: relative;  
  padding: 0 0 0 120px;  
}  
legend {  
  font-weight: bold;  
  width: 100px;  
  text-align: right;  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```



Log In	Email
	<input type="text"/>
	Password
	<input type="text"/>

Legend is styleable!!!

And it can wrap!!!

```
Legend {  
  width: 100px;  
}
```

This is some
really really
really long text

:Lang is supported

Take the following example:

```
<dl class="conversation">
  <dt>Whiny dude</dt>
  <dd>Here's your hamburger. Anyway, as I was saying about how
bad my life is&#8230;</dd>
  <dt>Other dude</dt>
  <dd>Yum, burger. <abbr lang="lol" title="Ok, thank you,
goodbye">kthxbye</abbr>. *leaves*</dd>
  <dt>Whiny dude</dt>
  <dd>I have no friends.</dd>
</dl>
```

Example adapted from <http://www.urbandictionary.com/define.php?term=kthxbye>

:Lang is supported

Apply a little CSS:

```
:lang(lol) {  
  font-style: italic;  
}
```

And end up with

Whiny dude: Here's your hamburger.
Anyway, as I was saying about how bad my
life is...

Other dude: Yum, burger. *kthxbye*. *leaves*

Whiny dude: I have no friends.

Example adapted from <http://www.urbandictionary.com/define.php?term=kthxbye>

Generated content, baby!

That's right. In fact, we just used some:

```
<dl class="conversation">
  <dt>Whiny dude</dt>
  <dd>Here's your hamburger. Anyway, as I was saying about how
bad my life is&#8230;</dd>
  ...
```

Whiny dude: Here's your hamburger.
Anyway, as I was saying about how bad my
life is...

```
.conversation dt:after {
  content: ': ';
}
```

Generated content, baby!

You can also do images*:

```
#content p:first-child:after {  
  content: url(fancy-break.png);  
  display: block;  
  margin-top: .5em;  
  text-align: center;  
  width: auto;  
}
```

This is the first paragraph and you can see
the nifty line below it.



This is the second paragraph, it has no line.

* IE8b1 does not support this, but it is planned for the final release

Support for `attr()`

Can be fun to use with your generated content:

```
abbr[title]:after {  
  content: " (" attr(title) ")";  
}
```

Yum, burger. *kthxbye* (Ok, thank you, goodbye).

Could also be useful in debugging:

```
[id]:hover:after {  
  color: #f00;  
  content: " [id=" attr(id) "]" ;  
}
```

CSS can count now*

Using counters, you can do all sorts of interesting stuff.

```
body {
  counter-reset: section;
}
h1:before {
  content: counter(section) " ";
  counter-increment: section;
}
h2:before {
  content: counter(section) "." counter(subsection) " ";
  counter-increment: subsection;
}
h1 + h2 {
  counter-reset: subsection;
}
h3:before {
  content: counter(section) "." counter(subsection)
  "." counter(subsubsection) " ";
  counter-increment: subsubsection;
}
h2 + h3 {
  counter-reset: subsubsection;
}
```

1 Heading

1.1 Subhead

1.1.1 Sub-subhead

1.1.2 Sub-subhead

1.1.3 Sub-subhead

1.2 Subhead

1.2.1 Sub-subhead

1.2.2 Sub-subhead

1.3 Subhead

2 Heading

* IE8b1 does not support this, but it is planned for the final release

You can control **quotes***

The **quotes** property provides access to which quotes are generated into the document.

```
<p>Douglas Adams famously said <q>Flying is learning how to  
throw yourself at the ground and miss.</q></p>
```

Applying some styles:

```
:lang(en) {  
  quotes: "\201C" "\201D" "\2018" "\2019";  
}
```

Douglas Adams famously said “Flying is learning how to throw yourself at the ground and miss.”

* IE8b1 does not support this, but it is planned for the final release

Control page breaks*

- ◉ With **widows** and **orphans**:

```
@media print {  
  p {  
    widows: 3; /* at least 3 lines @ top of page */  
    orphans: 3; /* at least 3 lines @ bottom of page */  
  }  
}
```

- ◉ With **page-break-inside**:

```
@media print {  
  pre {  
    page-break-inside: avoid;  
  }  
}
```

* IE8b1 does not support this, but it is planned for the final release

Control outlining

Outlines are like borders except they take up no space.

```
input:focus {  
  outline: 1px dotted #f00;  
}
```



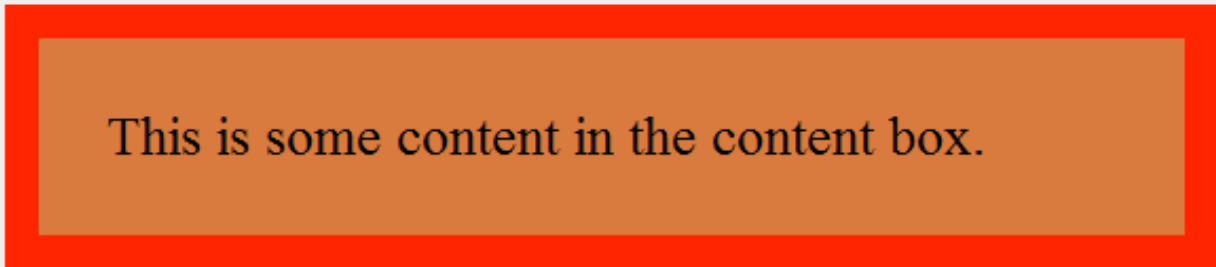
Could also be written as

```
input:focus {  
  outline-color: #f00;  
  outline-style: dotted;  
  outline-width: 1px;  
}
```

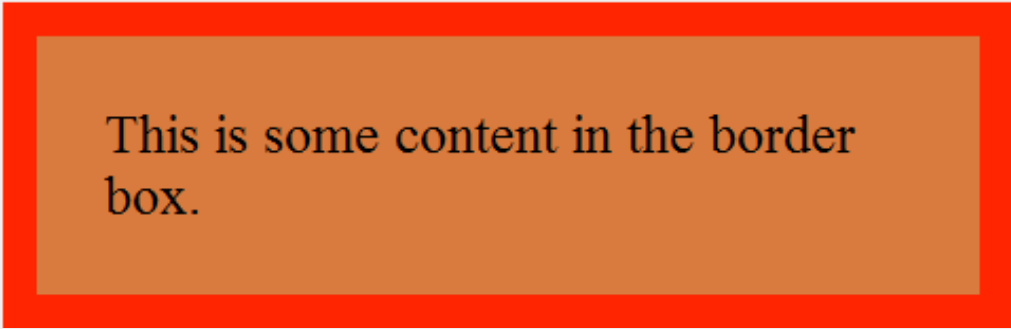
Choose your box model

A little CSS3 action lets you choose between the content (W3C) or border (IE) box model.

```
div {  
  background: #c63;  
  border: 10px solid #f00;  
  margin: 10px;  
  padding: 20px;  
  width: 300px;  
}  
div.border-box {  
  -ms-box-sizing: border-box;  
}
```



This is some content in the content box.



This is some content in the border box.

Track page updates

IE8 introduces a **proprietary** new event called **hashchange** that can be used to key into changes to the hash portion (fragment) of a URL

Could be **very** useful for tracking changes in Ajax widgets.

```
window.onhashchange = function(){  
    alert( window.location.hash );  
}
```

(use your own event listener adding script here)

Support for **data**: URIs

A means of including small data items inline as though they were an external source.

So, instead of writing

```
#nav a {  
    background: url(arrow.png) center right no-repeat;  
}
```

you could write

```
#nav a {  
    background: url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABQAAAAUCAYAAACN  
iR0NAAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAALFJREFUeNpi/P//PwM1ARMD  
lQELjBEWFkay5lWrVtHehegG5gBxNRBzUstAHiBuAeKrQBxIzUhRBOJ1QHwAiA2oGcv2QHwGiKcBsRC1  
kg0zEGcC8T0gLoLyqZIO+YG4F4gvAbEHNRM2G7Vc+BGIi4FYG4h3EMwpBMBcIC4D4ndEZZ0c4CAQFwDx  
BUqTzSMgjjBiB1IMwbCL0BcA8R9QPydotIGCqYMuvKQcdCX2AABBgADYx7aDLgnbAAAAABJRU5ErkJggg  
g==) center right no-repeat;  
}
```


Support for **data**: URIs

Pros:

- ▶ no additional HTTP traffic
- ▶ frees up a browser's limited number of download connections for other content
- ▶ page requires fewer cache entries
- ▶ can be used to embed content when it is not allowed or impractical to reference external files
- ▶ reduces the overhead of HTTPS requests
- ▶ possible to manage a multimedia page as a single file

Support for **data**: URIs

Cons:

- ▶ maintenance nightmare
- ▶ page does not benefit from the browser's cache when assets are re-used
- ▶ some browsers limit URI lengths
- ▶ data is included as a simple stream and many browsers may not support using containers to provide greater complexity such as metadata, data compression, or content negotiation
- ▶ encoded data could be 33–200% larger than the binary file

Will IE8 break sites?

Only if you let it ;-)

Issue #1: DOCTYPE ≠ contract

- ◉ The DOCTYPE switch was a great idea for the time
 - ▶ only standardistas were validating, so it made sense
 - ▶ it was an easy way to keep old content from breaking while simultaneously allowing standards-support to increase
- ◉ Content tools kinda ruined its usefulness
 - ▶ in their efforts to support standards, Dreamweaver, *et al* began adding DOCTYPEs by default
 - ▶ users didn't opt into the DOCTYPE switch's contract

Issue #2: IE stagnated

- ◉ When the IE team stopped improving on IE's standards support in 2001 (after the launch of IE6), many developers
 - ▶ didn't know better and began using it as the standard against which they built websites,
 - ▶ assumed its rendering to be accurate,
 - ▶ weren't aware of how to best manage its "idiosyncrasies" *vis a vis* the browsers that conformed to the W3C standards.
- ◉ And, unfortunately, those developers who did understand IE's inadequacies used unsustainable hacks to get IE to play nicely

When IE7 came along...

- ◉ ...it was a huge improvement in terms of standards support and bug fixes
- ◉ ...many sites broke because either
 - ▶ they weren't built with standards in mind and the sites' developers weren't aware they had been opted into wanting the most accurate standards-based layouts by their software, or
 - ▶ developers had used unsustainable hacks (rather than Conditional Comments) to bend IE to their will.

Who wants a repeat?

(not me)

Enter version targeting

IE team wanted to offer a means of explicitly stating what version of a given browser your site is known to work on and approached WaSP's Microsoft Task Force for help with that.

Together, we came up a solution that was implementable in either of two ways:

```
<meta http-equiv="X-UA-Compatible" content="IE=7" />
```

or an HTTP header

```
X-UA-Compatible: IE=7
```

This instruction tells IE8 to present the page as though the browser was IE7.

It is **optional**, but can provide a means of migrating your site to supporting future IE versions on your own timetable.

Questions?

What you need to know about IE8 and Standards

slideshare.net/AaronGustafson

Aaron Gustafson



easy designs