

CSS Don't

Last Updated Saturday, 15 March 2008

Here are some technologies you should try to avoid when using CSS. Internet Explorer Behaviors What is it? Internet Explorer 5 introduced behaviors. Behaviors are a way to add behaviors to HTML elements with the use of CSS styles. Why avoid it? The behavior attribute is only supported by Internet Explorer. What to use instead? Use JavaScript and HTML DOM instead. Example 1 - Mouseover Highlight The following HTML file has a <style> element that defines a behavior for the <h1> element:

```
<html>
<head>
<style type="text/css">
h1 { behavior: url(behavior.htc) }
</style>
</head>
<body>
<h1>Mouse over me!!!</h1>
</body>
</html>
```

The XML document "behavior.htc" is shown below: <attach for="element" event="onmouseover" handler="hig_lite" />
<attach for="element" event="onmouseout" handler="low_lite" />

```
<script type="text/javascript">
function hig_lite()
{
element.style.color='red';
}
function low_lite()
{
element.style.color='blue';
}
</script>
```

The behavior file contains a JavaScript and event handlers for the elements. If you use Internet Explorer, try it yourself (mouse over the text in the example). Example 2 - Typewriter Simulation The following HTML file has a <style> element that defines a behavior for elements with an id of "typing":

```
<html>
<head>
<style type="text/css">
#typing
{
behavior:url(behavior_typing.htc);
font-family:'courier new';
}
</style>
</head>
<body>
<span id="typing" speed="100">IE5 introduced DHTML behaviors.
Behaviors are a way to add DHTML functionality to HTML elements
with the ease of CSS.<br /><br />How do behaviors work?<br />
By using XML we can link behaviors to any element in a web page
and manipulate that element.</p>
</span>
</body>
</html>
```

The XML document "typing.htc" is shown below: <attach for="window" event="onload" handler="beginTyping" />

```
<method name="type" />
<script type="text/javascript">
var i,text1,text2,textLength,t;
function beginTyping()
{
i=0;
text1=element.innerText;
textLength=text1.length;
element.innerText="";
text2="";
t=window.setInterval(element.id+".type()",speed);
}
}
```

```
function type()
{
text2=text2+text1.substring(i,i+1);
element.innerText=text2;
i=i+1;
if (i==textLength)
{
clearInterval(t);
}
}
</script>
```