CSS Tooltips - Part Two

By: John Gallant , Holly Bergevin ,

This tutorial is meant for those familiar with basic CSS syntax and usage. If this is not you, the following discussion will make more sense after you have studied these articles:

- Under the Hood — The Basics of HTML: Part Three (http://www.communitymx.com/abstract.cfm?cid=AB474C9E02102677)
- The Link and Dynamic Pseudo-classes (http://www.communitymx.com/abstract.cfm?cid=B0934)
- Flowing and Positioning: Two Page Models (http://www.communitymx.com/abstract.cfm?cid=EA1A82A6F65A33F5)
- Absolute Positioning and the Top Property (http://www.communitymx.com/abstract.cfm?cid=AAA7C45E7CD65D33)

In Part One (http://www.communitymx.com/content/article.cfm?cid=4E2C0) of this CSS Tooltips series, we covered the basic coding necessary to create a hovered pop-up that can serve as a supplemental "tooltip" explanation for a link. The next step is to arrange for this effect to be used with any page element, and not just links.

A Matter Of Support

The heart of the CSS tooltip method depends on the **:hover (http://www.w3.org/TR/REC-CSS2/selector.html#x33)** pseudo-class. The W3C makes no restrictions as to which elements may be in the **:hover** state. In theory this means it can apply to any page element. Most modern browsers now support this CSS 2 feature, but sadly, **Internet Explorer for Windows** lags behind in this area. IE/Win supports the CSS **:hover** pseudo-class *only on link elements*. It's really too bad, because having clean CSS tooltips for an image or a complex math formula would be highly desirable to most authors.

However, it happens that IE5/Win and above *can* be made to support hovering on any element, via a (freely available) **jscript** called from within the CSS file. Microsoft calls this proprietary "pathway" for the Jscript a ***behavior***. Calling a script from within a CSS file is a non-valid use of CSS, and normally we would not recommend any such thing. But in this case it's a matter of making IE "support" a standard CSS rule that it otherwise would not.

Further, it's possible to place this "behavior call" in a separate style sheet and link that CSS into a page from within a **conditional comment (http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/overview/ccomment_ovw.asp)** (also Microsoft proprietary code) that appears to other browsers as just a normal comment. Only IE will look inside and see the behavior code lurking within!

Thus, for us code purists the result is nice clean HTML and CSS, with all the invalid code squirreled away out of sight. Explorer can see that invalid code only because it is capable of looking inside HTML comments for extra code it can parse. The W3C specifications don't care what lies within any HTML comments, so the page will validate just fine.

Explorer for the Mac is the one "modern" browser that does not properly execute these tooltip-type hovers, although it is possible to get a pop-up in that browser with a certain code arrangement. Unfortunately, the IE5/Mac method is not very compatible with the primary method, and that limited pop-up is far less useful. For these reasons, and because IE/Mac is no longer being developed, we are not going to attempt a description of this IE5/Mac pop-up method.

Hopefully, having dispensed with major objections to the use of invalid code, we can take a look at this "behavior" stuff with open minds. Shall we begin?

Making Explorer Behave

To quote from **Peter Nederlof (http://www.xs4all.nl/~peterned/hovercraft.html)** , the originator of this method:

> To recreate the hover in Internet Explorer, only a few things need to be done:
>
> - Search the CSS code for rules that contain :hover.
> - Transform them into rules that *do* work in IE.
> - Apply them to the necessary elements.

Peter does this by adding a carefully crafted **Jscript** file (csshover.htc) to the page via a Microsoft proprietary "behavior" rule in the CSS file. We don't want our CSS to fail validation, so a **conditional comment** is created to contain a separate style sheet to hold our behavior rule.

For the purposes of this method, **Jscript** is no different than **JavaScript** in the way that it functions.

So let's look at the code that calls in the magic **Jscript**:

```
<head>

...title element, meta tags, etc...

 <style type="text/css">
 /* some ordinary CSS rules... */
 </style>

 <!-- The line below starts the conditional comment -->
 <!--[if IE]>
  <style type="text/css">
   body {behavior: url(csshover.htc);}
  </style>
 <![endif]--> <!-- This ends the conditional comment -->

</head>
```

Code Block

Notice in the **Code Block** that the **conditional comment** is placed in the head element of an HTML

document. This allows regular HTML comments to be used, which are generally ignored by browsers as they parse and render a page. Because our **conditional comment** begins just like a regular HTML comment, most browsers will ignore its contents, but IE browsers notice that there is "something extra" in there that makes them take notice. The **[if IE]** tells IE/Win browsers that they are to pay attention to the following information in the comment.

Once inside the **conditional comment** in our example, IE notices an embedded style sheet. Our style sheet contains a single selector, **body**, with a single "property," in this case, the **Jscript** behavior call. The behavior call in body {behavior: url(csshover.htc);}, applies the Jscript **csshover.htc** to the body element, and thence to all the children of body. Being Jscript neophytes, we have no idea how this works, but boy does it ever!

Since the **conditional comment** (CC) is an actual HTML comment, care must be taken that no additional HTML comments are placed within this CC. Comment nesting is not allowed. If you wish to add a comment to remind yourself, or someone else what the CC is for, you can put it an HTML comment before the CC as we have done in the Code Block, or you can use CSS comments within the embedded style sheet.

To get the .htc file itself, visit **Peter's site** (http://www.xs4all.nl/~peterned/csshover.html) where you can **view the file** (http://www.xs4all.nl/~peterned/htc/csshover.htc) , then save or copy the complete file into your site where it can be accessed by the above code. That's all there is to it!

Alternate Ways To Call The File

If you prefer, the csshover.htc file may also be **linked** or **@imported** into the page. Just place the link or @import call within the conditional comment instead of the embedded style sheet. Just as before, only IE/Win will be able to call that style sheet. Then, within that remote sheet will be the behavior call to the csshover.htc file.

You are free to use this .htc file as long as you do not redistribute it for financial gain, and you must retain Peter's credit comment without alterations. Other than that, you are officially permitted to use it as you like. However, be aware that the Jscript, while fairly robust at this point, may be developed and improved further by Peter, so check back on occasion for updates.

Mind the MIME Type!

In the past we would be done, but when Microsoft's **XP Service Pack 2** is installed on a user's computer, the Windows OS will *reject the .htc file,* unless your site server serves that file with the MIME-type: **text/x-component**. You cannot set this yourself unless you have administrator privileges on that server, so you will probably have to get the server administrator to make sure that the served MIME-type is correct for **.htc** files. A properly configured server should already be this way, but we all know how things work in the real world, sometimes.

What Next?

Now that **:hover** has been enabled in (nearly) all modern browsers and for all page elements, a large range of possible pop-up effects open up before us. In the next installment of this series we will delve

deeply into those possibilities, and examine some of the pros and cons you will encounter there. It's going to be a lot of fun, but to keep you happy we have prepared a test page, complete with a few simple examples you may play with until then. **Note:** You'll need to download the **csshover.htc** (http://www.xs4all.nl/~peterned/htc/csshover.htc) before viewing the file in IE/Win. Enjoy!

Thanks are extended to Peter Nederlof for allowing us to link directly to his site from this article to facilitate the downloading of the csshover.htc file.

Keywords
CSS, tooltip, tooltips, popup, popups, :hover, pseudo-class, csshover.htc, hovering elements, hover behavior, conditional comment, Jscript, [if IE], MIME-type, x-component