

GREASEMONKEY HACKS™

*Tips & Tools for Remixing
the Web with Firefox*



O'REILLY®

Mark Pilgrim
*Foreword by Aaron Boodman,
Creator and Lead Developer, Greasemonkey*

HACK
#28

Make Pop-up Titles Prettier

Spice up those boring link tool tips.

Many web pages include title attributes on links. When you hover over the link, the browser displays a tool tip that gives more information about the link. The font and color of the tool tip are determined by the theme settings of the underlying operating system. This means you have some control over what tool tips look like, but they'll still look pretty boring. This hack makes link tool tips sexier and more functional at the same time by replacing the tool tip with a translucent floating window that contains both the title and the link URL.

The Code

This user script runs on all pages. It works by finding all the links on the page (using the `document.links` collection) and adding `mouseover`, `mouseout`, `focus`, and `blur` events to each one. On `mouseover` or `focus`, it creates a wrapper `<div>` containing the link title and URL and positions it on the page just below the cursor. On `mouseout` or `blur`, it removes the `<div>` element. It sounds simple, but determining the exact position and dimensions of the `<div>` element is quite complicated, as you can see in the `showNiceTitles` function.

Also, I would like to point out that the nice title `<div>` is styled with rounded corners, using the `-moz-border-radius` CSS rule. It is also slightly translucent, thanks to the `opacity` rule.



The `-moz-border-radius` property is a Mozilla-specific extension to CSS. The upcoming CSS 3 specification will likely include a `border-radius` property. The Mozilla developers just couldn't wait to implement it, but because the syntax might change in the final CSS 3 specification, they implemented it as `-moz-border-radius` to avoid future compatibility problems.

Save the following user script as *nicetitles.user.js*:

```
// ==UserScript==
// @name         Nice Titles
// @namespace    http://www.kryogenix.org/code/
// @description  render link titles with translucent floating window
// @include     *
// ==/UserScript==

// based on code by Stuart Langridge
// and included here with his gracious permission
```

```
// http://www.kryogenix.org/code/browser/nicetitle/

var CURRENT_NICE_TITLE;

function makeNiceTitles() {
  var arlinks = document.links;
  for (var i = arlinks.length - 1; i >= 0; i--) {
    var elmLink = arlinks[i];
    if (elmLink.title) {
      elmLink.setAttribute("nicetitle",elmLink.title);
      elmLink.removeAttribute("title");
      elmLink.addEventListener("mouseover",showNiceTitle,true);
      elmLink.addEventListener("mouseout",hideNiceTitle,true);
      elmLink.addEventListener("focus",showNiceTitle,true);
      elmLink.addEventListener("blur",hideNiceTitle,true);
    }
  }
}

function findPosition( oLink ) {
  if (oLink.offsetParent) {
    for (var posX = 0, posY = 0; oLink.offsetParent;
        oLink = oLink.offsetParent) {
      posX += oLink.offsetLeft;
      posY += oLink.offsetTop;
    }
    return [ posX, posY ];
  } else {
    return [ oLink.x, oLink.y ];
  }
}

function showNiceTitle(event) {
  if (CURRENT_NICE_TITLE) {
    hideNiceTitle(CURRENT_NICE_TITLE);
  }
  var elmTarget;
  if (event && event.target) {
    elmTarget = event.target;
  }
  if (!elmTarget) { return; }
  if (elmTarget.nodeType == Node.TEXT_NODE) {
    elmTarget = getParentElement(elmTarget);
  }
  if (!elmTarget) { return; }
  attrNiceTitle = elmTarget.getAttribute("nicetitle");
  if (!attrNiceTitle) { return; }

  var elmWrapper = document.createElement("div");
  elmWrapper.className = "nicetitle";
  tnt = document.createTextNode(attrNiceTitle);
  pat = document.createElement("p");
  pat.className = "titletext";
```

```
pat.appendChild(tnt);
elmWrapper.appendChild(pat);
if (elmTarget.href) {
    tnd = document.createTextNode(elmTarget.href);
    pad = document.createElement("p");
    pad.className = "destination";
    pad.appendChild(tnd);
    elmWrapper.appendChild(pad);
}
var h_pixels, t_pixels, w, h, mpos, mx, my;
STD_WIDTH = 300;
if (elmTarget.href) {
    h = elmTarget.href.length;
} else { h = attrNiceTitle.length; }
if (attrNiceTitle.length) {
    t = attrNiceTitle.length;
}
h_pixels = h*6; t_pixels = t*10;
if (h_pixels > STD_WIDTH) {
    w = h_pixels;
} else if ((STD_WIDTH>t_pixels) && (t_pixels>h_pixels)) {
    w = t_pixels;
} else if ((STD_WIDTH>t_pixels) && (h_pixels>t_pixels)) {
    w = h_pixels;
} else {
    w = STD_WIDTH;
}
elmWrapper.style.width = w + 'px';
mpos = findPosition(elmTarget);
mx = mpos[0];
my = mpos[1];
elmWrapper.style.left = (mx+15) + 'px';
elmWrapper.style.top = (my+35) + 'px';
if (window.innerWidth && ((mx+w) > window.innerWidth)) {
    elmWrapper.style.left = (window.innerWidth - w - 25) + "px";
}
if (document.body.scrollWidth && ((mx+w)>document.body.scrollWidth)) {
    elmWrapper.style.left = (document.body.scrollWidth - w - 25)+"px";
}
document.body.appendChild(elmWrapper);
CURRENT_NICE_TITLE = elmWrapper;
}

function hideNiceTitle(e) {
    if (CURRENT_NICE_TITLE) {
        document.body.removeChild(CURRENT_NICE_TITLE);
        CURRENT_NICE_TITLE = null;
    }
}

function getParentElement(node) {
```

```

    while (node && (node.nodeType !== Node.ELEMENT_NODE)) {
        node = node.parentNode;
    }
    return node;
}

function getMousePosition(event) {
    x = event.clientX + window.scrollX;
    y = event.clientY + window.scrollY;
    return [x,y];
}

function addGlobalStyle(css) {
    var elmHead, elmStyle;
    elmHead = document.getElementsByTagName('head')[0];
    if (!elmHead) { return; }
    elmStyle = document.createElement('style');
    elmStyle.type = 'text/css';
    elmStyle.innerHTML = css;
    elmHead.appendChild(elmStyle);
}

addGlobalStyle(
'div.nicetitle {' +
'  position: absolute;' +
'  padding: 4px;' +
'  top: 0px;' +
'  left: 0px;' +
'  background-color: black;' +
'  color: white;' +
'  font-size: 13px;' +
'  font-family: Verdana, Helvetica, Arial, sans-serif;' +
'  width: 25em;' +
'  font-weight: bold;' +
'  -moz-border-radius: 12px !important;' +
'  opacity: 0.75;' +
'}' +
'div.nicetitle p {' +
'  margin: 0; padding: 0 3px;' +
'}' +
'div.nicetitle p.destination {' +
'  font-size: 9px;' +
'  text-align: left;' +
'  padding-top: 3px;' +
'}');

window.addEventListener("load", makeNiceTitles, true);

```

Running the Hack

After installing the user script (Tools → Install This User Script), go to <http://www.w3.org> and hover your cursor over one of the links in the main naviga-

tion bar. Instead of the normal tool tip, you will see a rounded translucent tool tip with both the title and the URL of the link, as shown in [Figure 3-15](#).



Figure 3-15. Nice titles on w3.org

Hacking the Hack

Currently, this script checks only for links (using the `document.links` collection). But links aren't the only thing on web pages with titles. Virtually any element can have a title attribute. With a simple XPath query, we can find every element on the page with a title attribute and make a nice title out of it.

Replace the `makeNiceTitles` function with this version:

```
function makeNiceTitles() {
    var snapTitles = document.evaluate("//*[@title]",
        document, null, XPathResult.UNORDERED_NODE_SNAPSHOT_TYPE, null);
    for (var i=0; i<snapTitles.snapshotLength; i++) {
        var elm = snapTitles.snapshotItem(i);
        elm.setAttribute("nicetitle",elm.title);
        elm.removeAttribute("title");
        elm.addEventListener("mouseover",showNiceTitle,true);
        elm.addEventListener("mouseout",hideNiceTitle,true);
        elm.addEventListener("focus",showNiceTitle,true);
        elm.addEventListener("blur",hideNiceTitle,true);
    }
}
```

Now, go to the Greasemonkey home page at <http://greasemonkey.mozdev.org/> and hover your cursor over the word *Search* in the pane on the left. This is an `<h4>` element with a title attribute, and when you hover your cursor over it, you'll see a nice title pop up, as shown in [Figure 3-16](#).

This hack can be extended in other ways, too. Although few pages use it, HTML has tags for marking text as inserted or deleted: `<ins>` and ``, respectively. These elements can have a `datetime` attribute to declare when the text was inserted or deleted. We can extend the `makeNiceTitles` function to display nice titles for inserted and deleted text.

Replace the `makeNiceTitles` function with this version:

```
function makeNiceTitles() {
```



Figure 3-16. Nice titles on nonlink elements

```

var snapTitles = document.evaluate("//*[title or @datetime]",
    document, null, XPathResult.UNORDERED_NODE_SNAPSHOT_TYPE, null);
for (var i=0; i<snapTitles.snapshotLength; i++) {
    var elm = snapTitles.snapshotItem(i);
    if (elm.dateTime) {
        var sDate = elmIns.dateTime;
        var dtIns = new Date(sDate.substring(0,4),
            parseInt(sDate.substring(4,6)-1),
            sDate.substring(6,8),
            sDate.substring(9,11),
            sDate.substring(11,13),
            sDate.substring(13,15));
        if (elm.nodeName == 'INS') {
            elm.setAttribute("nicetitle",
                "Inserted on " + dtIns.toString());
        } else {
            elm.setAttribute("nicetitle",
                "Deleted on " + dtIns.toString());
        }
    } else {
        elm.setAttribute("nicetitle",elm.title);
        elm.removeAttribute("title");
    }
    elm.addEventListener("mouseover",showNiceTitle,true);
    elm.addEventListener("mouseout",hideNiceTitle,true);
    elm.addEventListener("focus",showNiceTitle,true);
    elm.addEventListener("blur",hideNiceTitle,true);
}
}

```

On any site that properly uses the ins and del elements, you can hover over the inserted or deleted text to see the date and time it was modified. Three cheers for semantic markup!