

## **CSS Menus**

By [Roger Keays](#), 16 April 2007

Building menus in HTML can dramatically improve the navigation and useability of your website. Getting it just right can be a very time consuming task though. There are a lot of javascript/dhtml implementations of menus out there but they almost all suffer from excessive bloat, inflexibility and complicated declarations to create the menu.

This blog describes how you can turn *any* existing HTML unordered list into a dropdown menu that is easily customisable through CSS. No need to call any fancy, indecipherable javascript, hardcode colors or sizes or fiddle with 3rd party library code: just import a single stylesheet and you're done.

We've set the following requirements for our menus:

1. Require no additional markup (e.g. class names, id's, anchors) other than regular `<ul>` and `<li>` tags.
2. Use no javascript.
3. Use no browser-dependent code.
4. Be independent of font size or screen size.

CSS menus aren't a new idea. Originally, I think credit for the idea goes to Eric Meyer [1]. Our method also uses ideas from howtcreate.co.uk [2] to support Microsoft Internet Explorer. Improvements made over these approaches include fixing flashing menu problems; cleaning up the css; supporting horizontal menus and fixing some problems displaying borders.

### **The Clean Implementation**

Let's take the following markup, and convert it into a horizontal dropdown menu.

```
<ul>
  <li>
    <a href="#">Music</a>
    <ul>
      <li>
        <a href="#">Classical</a>
        <ul>
          <li><a href="#">Mozart</a></li>
          <li><a href="#">Rossini</a></li>
          <li><a href="#">Pachelbel</a></li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

```

        </li>
        <li>
            <a href="#">Popular</a>
            <ul>
                <li><a href="#">70s</a></li>
                <li><a href="#">80s</a></li>
                <li><a href="#">90s</a></li>
            </ul>
        </li>
    </ul>
</li> <!-- music -->
<li>
    <a href="#">Dance</a>
    <ul>
        <li><a href="#">Salsa</a></li>
        <li><a href="#">Cha Cha</a></li>
        <li><a href="#">Waltz</a></li>
        <li>
            <a href="#">Swing</a>
            <ul>
                <li><a href="#">East Coast Swing</a></li>
                <li><a href="#">Lindy Hop</a></li>
            </ul>
        </li>
    </ul>
</li> <!-- dance -->
</ul> <!-- menu -->

```

Conceptually, we do the following steps with the CSS styling:

1. Remove list style types
2. Float the top-level <li> elements to arrange them horizontally
3. Use CSS2 positioning to position second and third-level <ul> elements
4. Hide the second and third-level <ul> elements using display: none;
5. Use a li:hover ul selector to display second and third-level <ul> elements when the mouse moves over their containing <li>

In detail, the styles we use are:

```
/* top level menu container */
```

```

ul {
    list-style: none;      /* no list bullets */
    margin: 0px;          /* don't try to indent lists */
    padding: 0px;         /* don't try to indent lists */
    background: wheat;
}

/* top level menu items */
li {
    position: relative;   /* makes this a containing block */
    float: left;          /* align menu horizontally */
    width: 5em;           /* make each item the same width */
}

/* second level menu container */
ul ul {
    border: 1px solid black;
    display: none;        /* don't show this menu by default */
    position: absolute;   /* use absolute positioning for submenu */
    top: 100%;            /* display directly under menu bar */
}

/* second level menu items */
li li {
    float: none;          /* makes this list a vertical one */
    width: 8em;           /* our second level menus are wider */
}

/* position third level menu container */
ul ul ul { top: 0px; left: 100%; }

/* make the anchor fill the li */
li a { display: block; padding: 3px; }

/* highlight effect on hover */
li a:hover { background: yellow; }

/*
 * The magic which shows the menus. The > selector selects only an
 * immediate child. So this selector says 'The ul directly below the
 * li being hovered over'.

```

```

*/
li:hover > ul { display: block; }

/* clear the floated elements */
ul:after {
    content: ".";
    display: block;
    height: 0;
    clear: both;
    visibility: hidden;
}

```

[Take a look at the example.](#)

## Internet Explorer

There are, as you might expect, several problems getting this to work on Microsoft Internet Explorer. The biggest of these is that IE 6 and 5.x don't support the :hover psuedoclass for <li> elements (quite ironic, seeing as it was Microsoft who invented the :hover psuedoclass, first using it for the <a> element). IE7 also has problems implementing this selector, particularly when the user changes the font size.

Unfortunately, we must look to javascript for a solution here. We can take advantage of IE's proprietary behavior CSS property to make this as transparent as possible:

```
li { behavior: url('IEmenus.htc'); }
```

With this file containing the javascript required to mimic the required :hover functionality:

```

<attach event="onmouseover" handler="rollOver" />
<attach event="onmouseout" handler="rollOff" />
<script type="text/javascript">

function rollOver() {

    /* fix style */
    element.className += ' hover';

    /* prevent redraw of entire menu */
    window.event.cancelBubble = true;

    /* change display of child */
    for (var x = 0; element.childNodes[x]; x++) {

```

```

        if (element.childNodes[x].tagName == 'UL') {
            element.childNodes[x].style.display = 'block';

            /* force IE to draw the child properly */
            element.childNodes[x].style.visibility = 'visible';
        }
    }
}

/*
 * Called when the mouse moves off the li element.
 */
function rolloff() {

    /* fix style */
    element.className = element.className.replace(' hover', '');

    /*
     * Prevent redraw of entire menu by cancelling event bubble when moving
     * onto children. Otherwise you get a lot of flickering in IE with large
     * menus.
     */
    var onto = window.event.toElement;
    if (onto != null) {
        do {
            if (onto == element) {
                window.event.cancelBubble = true;
                return;
            }
        } while ((onto = onto.parentElement) != null);
    }

    /* change display of child */
    for (var x = 0; element.childNodes[x]; x++) {
        if (element.childNodes[x].tagName == 'UL') {
            element.childNodes[x].style.display = 'none';
        }
    }
}
</script>

```

IE also has some typical rendering defects:

- Gaps are shown between the second and third-level menu items. This is fixed with `li li { vertical-align: bottom; }`.
- Anchor elements sometimes ignore the `:hover` event. This is fixed by putting the elements in `hasLayout` mode with `li a:hover { zoom: 1; }`.
- IE doesn't support the `:after` psuedoclass for clearing the floated `<li>s`. IE will clear elements when the containing block is in `hasLayout` mode though, so the fix is simply: `ul { zoom: 1; }`. You could also use `width: 100%;` here if that is appropriate for your design.
- For IE versions before 7, the position of the submenu is out by 1px when the containing `<li>` has an odd height in pixels. This is fixed with the following IE6-specific css expression:  
`ul ul { _margin-top: expression(this.parentNode.clientHeight % 2 == 0 ? 0 : 1); }`

That's it.

- [Download IEmenu.htc](#)

## References

- [1] <http://meyerweb.com/eric/css/edge/menus/demo.html>  
 [2] <http://www.howtocreate.co.uk/tutorials/testMenu.html>

## About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.