# Billy’s Extras ☺

## IE CONDITIONAL COMMENTS

## Edge Mode & GCF

Early in the document, target IE9 and below to force them into edge mode and to run Google Chrome Framework if they have it installed.

<!DOCTYPE html>  
<html lang="en-gb" class="no-js">  
<head>  
<meta charset=UTF-8">  
<!--[if (lte IE 9) & (!IEMobile)]><meta http-equiv="X-UA-Compatible" content="IE=edge;chrome=1"><![endif]-->

### Main Style-sheet

Load the main style-sheet that will be seen by all browsers. Included a querystring for version control. This style-sheet contains valid CSS3 rules and will validate against the W3C.

<link rel="stylesheet" href="/c/style.css?v=1">

### Vendor Style-sheet

Why should IE browsers have to suffer the load of vendor specific style rules?

<!--[if (gte IE 10) | (IEMobile)]><!-->  
 <link rel="stylesheet" href="/c/vendor.css?v=1">  
<!--<![endif]-->

### IE Style-sheet

Why should non-IE browsers have to load IE specific style rules? While we are at it, let’s also polyfill IE8 and below to understand media queries so that we can build fluid, elastic, mobile friendly web sites.

<!--[if (lte IE 9) & (!IEMobile)]>  
 <link rel="stylesheet" href="/c/ie.css?v=1"/>  
 <script src="/j/libs/respond.min.js"></script>  
<![endif]-->

### Inside the IE Style-sheet

Why have numerous conditional comments to load version specific IE bug fixes. Place them all in a single style-sheet and target IE versions as follows:

/\* =============================================================================

IE Version Targeted Hacks

==========================================================================

.ie {

background: pink; // targets IE9 and below. Use only if we need to fix IE9

background: purple\0/; // targets IE8 only. Use to over-ride IE9 setting or to target only IE8 and not below

\*background: green; // targets IE7 and below

\_background: red; // targets IE6 and below (IE5/5.5 cannot style html tag)

\_bac\kground: brown; // targets IE6 only

}

\*/

## CONDITIONAL IE

Alternatively, you can target IE versions by adding a version class name to the <html> tag.

<!--[if lt IE 7 ]> <html class="no-js ie6" lang="en"> <![endif]-->

<!--[if IE 7 ]> <html class="no-js ie7" lang="en"> <![endif]-->

<!--[if IE 8 ]> <html class="no-js ie8" lang="en"> <![endif]-->

<!--[if (gte IE 9)|!(IE)]><!--> <html class="no-js" lang="en"> <!--<![endif]-->

* Must come before style-sheets.
* Can all be on one line.
* The last line is a reversed conditional comment seen by all browsers and IE9+
* If JavaScript is available, replace the class, "no-js" with "js". Notice in the next topic that Modernizr does this for you.

## CONDITIONAL CSS

<html class=" js no-flexbox no-canvas no-canvastext no-webgl no-touch no-geolocation postmessage no-websqldatabase no-indexeddb no-hashchange no-history draganddrop no-websockets no-rgba no-hsla no-multiplebgs no-backgroundsize no-borderimage no-borderradius no-boxshadow no-textshadow no-opacity no-cssanimations no-csscolumns no-cssgradients no-cssreflections no-csstransforms no-csstransforms3d no-csstransitions fontface no-video no-audio localstorage sessionstorage no-webworkers no-applicationcache no-svg no-inlinesvg no-smil no-svgclippaths" lang="en-gb">

* Modernizr adds class names to the <html> element that indicates whether a CSS property/ HTML5 feature is supported.
* Notice the js class, indicating JavaScript is available.
* Remember that if JavaScript is not available then these rules would not be applied.
  + Do not use if you want to ensure all browsers receive the rule.
  + Only use this technique to progressively enhance a feature that also requires JavaScript to be available or when a browser does not support a feature.

For example, given a block element with an appropriate class name:

<section class="multicolumn">

...

</section>

We can now use media queries to start using multiple columns above a certain width:

@media screen and (min-width: 810px) {

.multicolumn {

column-rule: 1px solid #c4c8cc;

column-gap: 40px;

column-count: 2;

}

}

With additional columns as the width increases:

@media screen and (min-width: 972px) and (max-width: 1296px) {

.multicolumn {

column-count: 3;

}

}

You can continue adding additional columns as the width continues to increase but unfortunately, IE does not support multiple columns, so we need a fallback. What about a centred single column of ideal reading width:

/\* Only applied if both classnames exist \*/

.no-csscolumns .multicolumn {

width: 33em; /\* Ideal reading width \*/

margin: 0 auto; /\* Auto-centred \*/

}

## CDN Fallback

You want to take advantage of a CDN because you know that when your customers first visit your site, they will probably already have the CDN version of the library already cached in their browser. However, you are worried that the CDN server/s may be down and your application would break or that you may need to develop your application whilst offline. On this course we utilize jQuery quite often, so we could use the CDN but if not available, the jQuery object would not exist so we can load from our own server.

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"> </script>

<script>window.jQuery || document.write("\x3Cscript src='/j/jquery-1.7.2.min.js'>\x3C/script>");</script>

Did you also notice the CDN URL? It allows for http or https requests.

## ZEN CODING

* Use CSS3 selectors to create markup.
* Download Zen coding from <http://code.google.com/p/zen-coding/downloads/list>
* It’s great fun playing around with the selectors.
* To get you started, visit this tutorial <http://code.google.com/p/zen-coding/wiki/ZenHTMLSelectorsEn>
* Given this Zen:

div#page>(header>hgroup>h1+h2)+(article#book>section.chapter\*4)+nav>ul#toc>li\*8>a>img

* Will produce this markup:

<div id="page">  
 <header>  
 <hgroup>  
 <h1></h1>  
 <h2></h2>  
 </hgroup>  
 </header>  
 <article id="book">  
 <section class="chapter"></section>  
 <section class="chapter"></section>  
 <section class="chapter"></section>  
 <section class="chapter"></section>  
 </article>  
 <nav>  
 <ul id="toc">  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 <li><a href=""><img src="" alt="" /></a></li>  
 </ul>  
 </nav>  
</div>

Note also that the div and the navigation ul can be targeted to give a basic layout and theme to IE8 and below when JavaScript is unavailable to create the new HTML5 elements for these browsers.

Try:

html:5

It outputs:

<!DOCTYPE HTML>  
<html lang="en-US">  
<head>  
 <meta charset="UTF-8">  
 <title></title>  
</head>  
<body>

</body>  
</html>

How often have you had a list of items and you would like to wrap each item in markup. Very fiddly but zen coding to rescue. Given:

Oranges  
Bananas  
Apples  
Pears  
Cherries

Highlight these items and press the keyboard shortcut (Ctrl+Shift+a) default. Type into the abbreviation textbox:

nav>ul>li\*>a>strong

this will produce:

<nav>  
 <ul>  
 <li><a href=""><strong>Oranges</strong></a></li>  
 <li><a href=""><strong>Bananas</strong></a></li>  
 <li><a href=""><strong>Apples</strong></a></li>  
 <li><a href=""><strong>Pears</strong></a></li>  
 <li><a href=""><strong>Cherries</strong></a></li>  
 </ul>  
</nav>

Zen Coding is installed in Notepad++ on the course load in the html5 virtual machine. Try these examples and experiment yourself.

Good Luck

Enjoy ☺

Billy