



# CSS 101:

An Impossibly Fast Introduction to the World of  
Cascading Style Sheets.

# Who is this Guy Anyway?

- 13+ years HTML/CSS/JavaScript. My day job since 1999.
- Interface Architect at Isobar (AKA Molecular)
- PAST: Cramer, AdvisorTech, Compete, Demandware, The Weekly Dig, Gillette, Museum of Science, Boston, PC Connection, State Street, Webex

# What Are We Going to Talk About

- Introduction to CSS
- CSS Fundamentals
- Specificity
- CSS Versions
- CSS in Action
- Frameworks, Abstractions, etc.
- Testing
- Resources

# CSS?

## Cascading Style Sheets

CSS is a style sheet language used to determine the formatting of an HTML document.

Before we had CSS (and before it was widely adopted) all of this formatting information was embedded directly in the document- either in the form of attributes like width or bgcolor (background color) or in the form of purely presentational tags like font.

Combined with the abuse of the table tag to create complicated layouts, the landscape for layout and design on the web was an unmanageable mess.

CSS fixed all that (kind of.)

Using separate style sheets for an entire site, leveraging semantic markup and identifiers like ids (for unique page elements) and classes (for multiple, like elements) a developer can apply styles across a whole site while updating a single (cacheable) file.

# What It Looked Life Before

```
<p align="center">
  <font face="Papyrus"></font>
</p>
<p align="center">
  <font face="Papyrus"> Welcome to The Fancy lad Site! </font>
</p>
<p align="center">
  <font face="Papyrus">This web-page is the semi-official home of Fancylads on the World Wide Web.</font>
</p>
```

# Not So Bad? Try This.

```
<table width="158" border="0" align="center" cellpadding="0" cellspacing="0">
<tr bgcolor="#006699">
  <td valign="top" bgcolor="#000066"><div align="center"> <strong> <font color="#FFFFFF" size="-1" face="Verdana, Arial, Helvetica, sans-serif"> Sponsors:
    </font> </strong> </div></td>
</tr>
<tr bgcolor="#CCCCCC">
  <td class="body-small"><div align="center"><font color="#666666" size="-2"><a
    href="http://www.packaginggraphics.net/packaging-design.html" target="_blank">Packaging Graphics Co.</a></font></div></td>
</tr>
<tr bgcolor="#CCCCCC">
  <td height="22" class="body-small"><div align="center"> <font color="#666666"
    size="-2"><a href="http://www.brochure-design.com" target="_blank">Brochure Design & Printing</a></font></div></td>
</tr>
<tr>
  <td height="10" bgcolor="cccccc"></td>
</tr>
</table>
```



Drunken



Enter  
CSS



# Enter CSS (The timeline)

CSS1

December 1996

CSS 2

Became a W3C Recommendation in May 1998

CSS 3

CSS level 3 has been under development since December 15, 2005



# Enter CSS

It took a while for CSS to catch on with developers and browser vendors.

So... 1996 really turned into 2000 or later for relatively widespread adoption.

Before that it was `<font>`city all the way.



# CSS Fundamentals

- The Separation of Style, Content and Behavior
- One BIG Core Concept
- Getting the style sheet on the page
- The anatomy of a style sheet

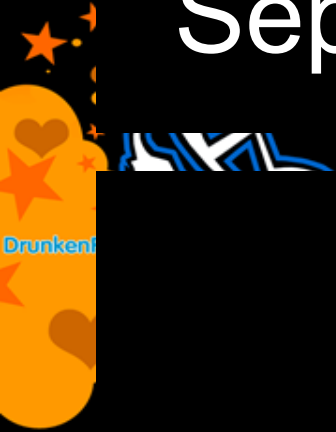
# The Separation of Style, Content and Behavior

- Core Concept of web development
- HTML + CSS + JavaScript
- Content + Style + Behavior

# Separation of Content and Style?



Let's see that in action



# Our New HTML

```
<h1>Fancy Lads</h1>
<p>Welcome to The Fancy lad Site!</p>
<p>This web-page is the semi-official home of Fancy lads on the World Wide Web.</p>

<!--How much simpler is that?-->

<p align="center">
  <font face="Papyrus"></font>
</p>
<p align="center">
  <font face="Papyrus"> Welcome to The Fancy lad Site!</font>
</p>
<p align="center">
  <font face="Papyrus">This web-page is the semi-official home of Fancylads on the World Wide
  Web.</font>
</p>
```

# Let's See the Associated Style Sheet

```
h1{
  background:url(fancy-header.png) no-repeat;
  width:207px;
  height:279px;
  text-indent:-9999px;
}
p {
  text-align:center;
  font-family:papyrus;
}
```

# So, How Does It Work?

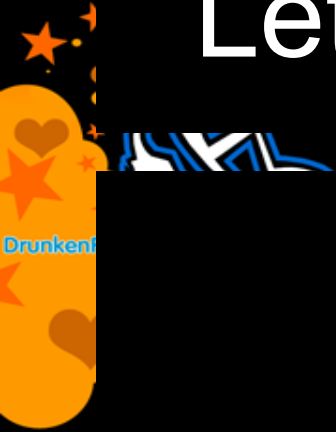
You create a style sheet, the browser downloads it, parses it and then the browser:

Matches elements  
on the page

And then it ->

*Styles Them*

Let's look at some more code



Drunken



# Getting the Style Sheet on the Page

```
<!-- This is in the HEAD of your document -- >
<!-- XHTML -- >
<link rel="stylesheet" type="text/css" href="/_assets/styles/style.css" />
<!-- HTML5-- >
<link type="text/css" href="/_assets/styles/style.css" >
```

# Basic Anatomy of a Style Sheet

```
/* A single tag */
/* Many elements will inherit from this tag, since it's high up in the document
   */
body {
    background: #CCC url(/_assets/styles/images/page-bg.png) repeat-x;
    font: normal .825em/1.65 Verdana, Arial, Helvetica, sans-serif;
    color: #333;
}
/*an ID */
#container {
    background:#fff;
    height:auto;
    margin:auto;
    overflow:auto;
    position:relative;
    width:980px;
}
```

# Basic Anatomy of a Style Sheet

```
/* A single tag */
h1 {
  color: #999;
  font-size: 200%;
  text-transform: uppercase;
  font-weight:normal;
}
/* A series of ID/tag combinations, with the same rules applied */
#main h2, #main h3, #main h4, #main h5 {
  font-weight:normal;
  line-height:1.4;
  margin:7px auto;
}
```

# Basic Anatomy of a Style Sheet

```
/* A class */
.more-link {
    font-weight:bold;
    text-transform:uppercase;
    font-size:110%;
    text-decoration:none !important;
}
/* An ID/class combo */
#main .share {
    margin-top:7px;
}
/* An ID/class/tag combo */
#main .share strong {
    background: url(/_assets/styles/images/share.png) 0px 3px no-repeat;
    color:#393;
    padding-left: 19px;
    text-transform:uppercase;
}
```

# Basic Anatomy of a Style Sheet

```
/* A class */
.more-link {
    font-weight:bold;
    text-transform:uppercase;
    font-size:110%;
    text-decoration:none !important;
}
/* An ID/class combo */
#main .share {
    margin-top:7px;
}
/* An ID/class/tag combo */
#main .share strong {
    background: url(/_assets/styles/images/share.png) 0px 3px no-repeat;
    color:#393;
    padding-left: 19px;
    text-transform:uppercase;
}
```

# Vocabulary

```
.selector {  
  /*declaration*/  
  property:value  
}
```

# Shorthand properties

```
.verbose {  
  font-family: "Times New Roman", Times, serif;  
  font-size: 12px;  
  font-style: italic;  
  line-height: normal;  
  font-weight: bold;  
  color: #003333;  
  background-color: #FFFCC;  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 15px;  
  padding-left: 20px;  
  background-image: url(images/bg.png);  
  background-repeat: repeat;  
  background-position: 10px 5px;  
}
```

# Shorthand properties

```
.shorthand {  
  background: #ffffffcc url(images/bg.png) 10px 5px repeat;  
  color: #003333;  
  font: italic bold 12px/normal "Times New Roman", Times,  
  serif;  
  padding: 5px 10px 15px 20px;  
}
```



# Shorthand properties

Remember:

**T**<sub>(op)</sub> **R**<sub>(ight)</sub> **B**<sub>(ottom)</sub> **L**<sub>(eft)</sub>

# Formatting

```
#main article strong {
    font-weight:bold;
}
#text #main article blockquote {
    background:#efefef url(_assets/styles/images/bq-bg.png) no-repeat;
    color:#600;
    font-style: italic;
    margin: 15px auto 30px auto;
    padding: 30px 30px 15px 75px;
}
#text #main article blockquote cite {
    color:#333;
    font-size:90%;
    font-style:normal;
}
#text #main article ul {
    font-size:14px;
    margin: auto auto 30px 15px;
}
```

# Formatting

```
.post-list li a{
  color:#A8A8A8;
}

.post-list li a:hover{
  color:#000;
  text-decoration:none;
}

.post-list li .author a, .post-list li .author a:hover{
  color:#F30;
  text-transform:uppercase;
}
```

# Formatting

```
#wrapper {width:800px; margin:0 auto;}
#header {height:100px; position:relative;}
#feature .post {width:490px; float:left;}
#footer {clear:both; font-size:93%; float:none;}
#footer .wrapper {float:none;}
```

# Formatting

- Whatever style you use, it's good practice to minify your CSS before pushing to production so that all the extra characters you pump into your sheets for ease-of-use as a developer don't slow down the experience of your users.

I use:

<http://developer.yahoo.com/yui/compressor/>

# Specificity/The Cascade

- One of the most important things in CSS is understanding the way rules are inherited and applied in the browser. This is one of those things that many developers “get” intuitively but don’t necessarily understand at a granular level.
- There’s actually an algorithm, so if you’re stumped, you can actually count it out. It works like this:

# Specificity/The Cascade

- First, find all rules that apply to the target element/property. This will be some combination of **browser default > style sheet default > targeted rules**.

# Specificity/The Cascade

- Once all the rules are gathered calculations are made to decide which ones are to be followed and which ones are to be discarded. That works like this:
  - Sort by explicit weight- **‘!important’** rules carry more weight than normal declarations.
  - Sort by origin: the author’s style sheets trump the browser default values.
  - Sort by specificity of selector. More specific selectors trump more general ones. The formula is as follows:
    - factor in any inline styles
    - count the number of ID attributes in the selector
    - the number of CLASS attributes in the selector
    - the number of tag names in the selector



# Specificity/The Cascade

## Some Examples

Selector	# of INLINE RULES	# of IDS	#of CLASSES	# of TAGS	Specificity
LI	0	0	0	1	0,0,0,1
UL LI	0	0	0	2	0,0,0,2
DIV UL LI	0	0	0	3	0,0,0,3
DIV UL .mLIClass	0	0	1	2	0,0,1,2
#myLI	0	1	0	0	0,1,0,0
<li style="color:blue">	1	0	0	0	1,0,0,0

# Specificity/The Cascade

- Sort by order specified: if two rules have the same weight, the latter specified wins. Rules in imported style sheets are considered to be before any rules in the style sheet itself.
- If two rules only impact one column, the higher number wins. If the selector cuts across more than one column, the biggest numbers in the farthest most left column wins. So, inline styles (**which you should avoid**) are more specific than an ID, which, in turn is more specific than a class, which itself will trump a tag. If you can wrap your head around these concepts, you'll go a long way towards making sense of CSS and how the rules are applied.

# CSS Versions

- **CSS 1**

- Font properties such as typeface and emphasis
- Color of text, backgrounds, and other elements
- Text attributes such as spacing between words, letters, and lines of text
- Alignment of text, images, tables and other elements
- Margin, border, padding, and positioning for most elements
- Unique identification and generic classification of groups of attributes

- **CSS2**

includes a number of new capabilities like

- absolute, relative, and fixed positioning of elements and z-index,
- the concept of media types
- support for aural style sheets and bidirectional text
- new font properties such as shadows.

# CSS Versions

- **CSS3**

Modules include:

- **Borders** (border-radius, box-shadow)
- **Backgrounds** (multiple backgrounds)
- **Color** (HSL colors, HSLA colors, opacity, RGBA colors)
- Also:
  - media queries
  - multi-column layout
  - Web fonts

Let's See it in Action



# Fonts/Backgrounds/Borders

<http://jsfiddle.net/JwsBn/>

# Layout



<http://jsfiddle.net/np43E/2/>

Drunken

# A Quick Aside on Floats

“A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property. “

w3c: <http://www.w3.org/TR/CSS2/visuren.html>



# It looks like this:

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

I'm floating, yo!

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

# Floated Content, Keeps on Floating

Sometimes, you have to “clear” it.



# This is what that looks like.

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

I'm floating, yo!

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the 'clear' property). Content flows down the right side of a left-floated box and down the left side of a right-floated box. The following is an introduction to float positioning and content flow; the exact rules governing float behavior are given in the description of the 'float' property.

# Do this enough, you need a system.

We messed around with this for a while. Eventually we found:

**“Simple Clearing of Floats”** (overflow:auto on the containing element. Learn it, love it, live it)

<http://blogs.sitepoint.com/2005/02/26/simple-clearing-of-floats/>

Also see:

**.clearfix**

<http://www.positioniseverything.net/easyclearing.html>

**:after**

<http://lists.w3.org/Archives/Public/www-style/2002Aug/0134.html>

Thank Adam for the right links: <http://www.amodernfable.com/>

# Positioning

<http://jsfiddle.net/3uNsN/>

Drunken

# CSS3



<http://jsfiddle.net/eSNqx/>

Also

<http://CSS3Please.com/>

# Frameworks/Abstractions

## Frameworks

Pre-built layout systems which allow for much easier layout construction. All of the hard stuff is figured out for you, you just need to learn/love the system.

See: <http://sethetter.com/web-design/css-framework-comparison/>

## Abstractions

Sits at a higher level than CSS. Allows for variables, functions and alternative syntax. See: SASS/COMPASS (<http://sass-lang.com/> and <http://compass-style.org/> )

# Reset Style Sheets

Level the playing field across browsers.

Up until now, there were never rules for how browsers should set defaults on how elements were styled. Resets allow us to level the playing field.



# Reset Options:

<http://meyerweb.com/eric/tools/css/reset/>

<http://html5doctor.com/html-5-reset-stylesheet/>

Also:

<http://html5boilerplate.com/>

# Meyer Reset

```
/* v1.0 | 20080212 */
```

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,  
blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, font,  
img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i,  
center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption,  
tbody, tfoot, thead, tr, th, td {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    outline: 0;  
    font-size: 100%;  
    vertical-align: baseline;  
    background: transparent;  
}
```

# Meyer Reset

```
body {  
    line-height: 1;  
}  
ol, ul {  
    list-style: none;  
}  
blockquote, q {  
    quotes: none;  
}  
blockquote:before, blockquote:after,  
q:before, q:after {  
    content: '';  
    content: none;  
}
```

# Meyer Reset

```
/* remember to define focus styles! */
:focus {
  outline: 0;
}

/* remember to highlight inserts somehow! */
ins {
  text-decoration: none;
}
del {
  text-decoration: line-through;
}

/* tables still need 'cellspacing="0"' in the markup */
table {
  border-collapse: collapse;
  border-spacing: 0;
}
```

# Testing



<http://browsercam.com/>

<http://browsershots.org/>

<http://browserlab.adobe.com/>

<http://litmus.com/>

# FireBug

## HTMLCSSJavaScript

Let's Push Things Forward

### [Recent Reading \(JS Natives Duke it Out in jQuery, Performance, a New Image Format\)](#)

October 7th, 2010 by Rob Larsen

Sorry, it's been a while. I've been busy at [work](#), I've been wringing every drop of summer on my [bike](#), and I've spent a lot of my free time on my upcoming [presentation](#), so I haven't been posting as much as I would like. Fall means I should have more time for writing. That's cool.

Anyway, to break the ice here are a few articles that have recently caught my attention.

[Read the rest of this entry »](#)

Categories: [Performance](#) Tagged: [browsers](#) | [chrome](#) | [dom](#) | [google](#) | [jquery](#)  
Short Link: <http://dfst.us/7vi>

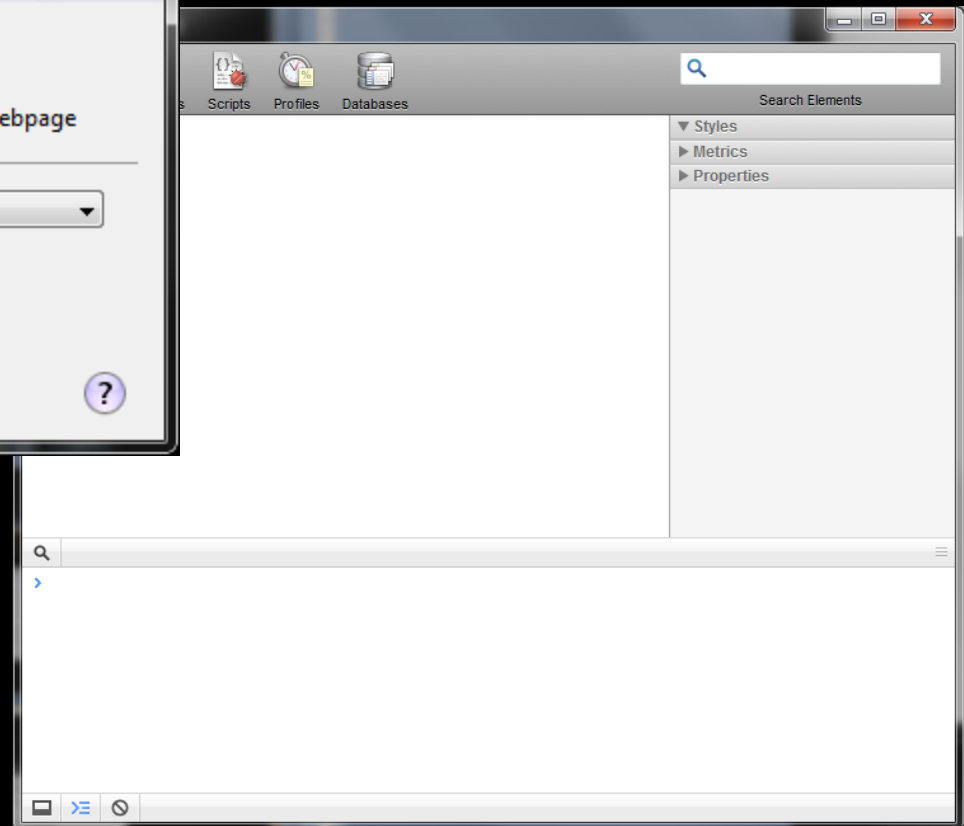
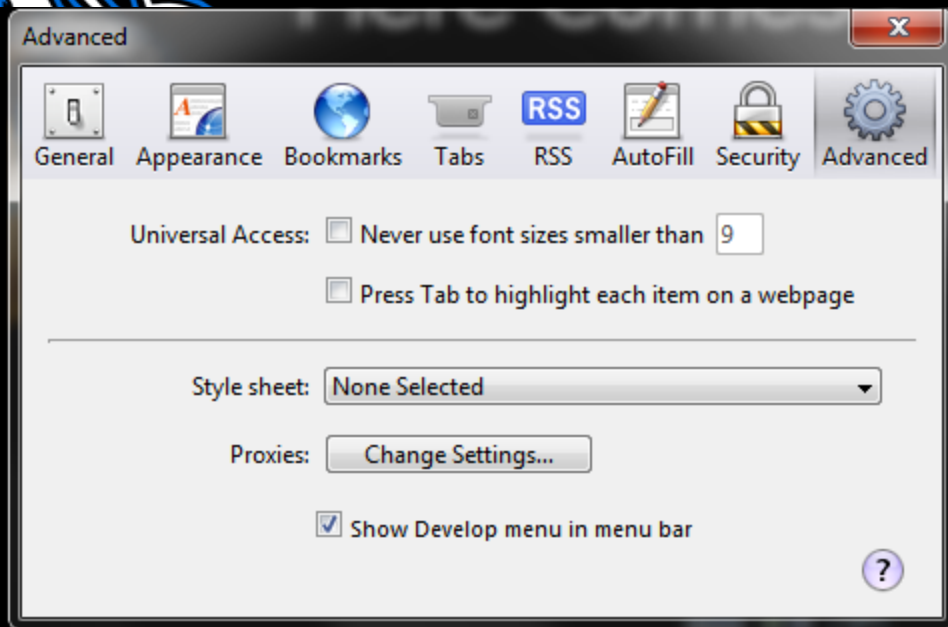
### [Ant. Ant? Really?](#)

September 8th, 2010 by Rob Larsen

Quick, if you were to guess a technology I'd be making commits on a project using, would Apache Ant be at the top of the list? I didn't think so. [Here I am committing an Ant Build script](#) to Paul's excellent [HTML5 Boilerplate](#) project. It makes sense since I've done work on concatenation, minification, and other performance enhancements using Ant, but it's still a weird technology to contribute to such a cool project with...

```
File View Help
Console HTML CSS Script DOM Net Page Speed Page Speed Activity YSlow
Edit style.css
font-style:italic;
}
/*end header*/
/*****main content area*****/
/*overflow:auto clears floats like magic*/
#content {
  overflow:auto;
  border-right:10px solid #fff;
  border-left:10px solid #fff;
}
/*main copy area*/
#container #copy {
  width:600px;
  border:1px solid #999;
  padding:16px 10px 16px 12px;
  float:left;
  -moz-border-radius: 12px; /* FF1+ */
  -webkit-border-radius: 12px; /* Saf3+, Chrome */
  border-radius: 12px; /* Opera 10.5, IE 9 */
  -moz-box-shadow: 0px 0px 8px #999; /* FF3.5+ */
  -webkit-box-shadow: 0px 0px 8px #999; /* Saf3.0+, Chrome */
  box-shadow: 0px 0px 8px #999; /* Opera 10.5, IE 9.0 */
  margin-bottom:20px;
}
#content .post h1 {
  font: 23px Consolas, Inconsolata, "Lucida Console", Monaco, "Lucida Sans Typewriter", monospace;
  font-weight:bold;
}
#comments, #content .post h1 a {
  color:hotpink;
}
#comments, #content .post h2 {
  font:bold 20px Consolas, Inconsolata, "Lucida Console", Monaco, "Lucida Sans Typewriter",
monospace;
  margin-top: 13px;
  margin-bottom: 6px;
}
>>>
```

# Safari (ctrl + alt + i)



# Internet Explorer 8 (f12)

The image shows a screenshot of Internet Explorer 8 with the Developer Tools window open. The browser window displays the URL `http://htmlcssjavascript.com/`. The Developer Tools window is split into three main panes: HTML, CSS, and Style.

- HTML Pane:** Shows the DOM tree. The selected element is an `<h1>` tag with the text "Permanent Link to Recent Reading". The parent element is a `<div id="copy">` which is inside an `<article class="post" id="post-7733">`.
- CSS Pane:** Shows the CSS rules applied to the selected element. The selected rule is `#content .post H1 A` with a color of `hotpink`.
- Style Pane:** Shows the computed styles for the selected element. The color is `hotpink`. Other visible styles include `font-size: 100%`, `font-weight: inherit`, `border-bottom: 0px`, `border-left: 0px`, `padding-bottom: 0px`, `font-style: inherit`, `margin: 0px`, `outline-style: none`, `outline-color: invert`, `padding-left: 0px`, `outline-width: 0px`, `padding-right: 0px`, `font-family: inherit`, `font-size: 100%`, `vertical-align: baseline`, `border-top: 0px`, `font-weight: inherit`, `border-right: 0px`, `padding-top: 0px`.

The background page content includes a header "HTMLCSSJava" and a section titled "Recent Reading in jQuery, Pe" dated "October 7th, 2010 by Ro".



# Chrome (ctrl + shft + j)

The screenshot shows a Chrome browser window with the address bar at `htmlcssjavascript.com`. The page content includes a header with the title "HTMLCSSJavaScript" and the tagline "Let's Push Things Forward". A featured article titled "Recent Reading (JS Natives Duke it Out, Regexp in jQuery, Performance, a New Image Format?)" is displayed, dated October 7th, 2010, by Rob Larsen. The article text discusses the author's busy schedule and mentions links to "work", "bike", and "CSS presentation".

The Chrome Developer Tools are open at the bottom. The "Elements" panel shows the DOM tree with the selected element being an `h1` tag with the text "Recent Reading (JS Natives Duke it Out, Regexp in jQuery, Performance, a New Image Format?)". The "Computed Style" panel shows the style rules for this element, including a `color: hotPink;` rule. The "Console" panel at the bottom shows several error messages, such as "Unmatched </p> encountered. Converting </p> into <p></p>."

# Targeting Browsers (\*cough\* Internet Explorer)

- Hacks?
- Body/HTML class
- HasLayout

# Targeting Internet Explorer- HACKS

Just say no.

But... if you must:

<http://paulirish.com/2009/browser-specific-css-hacks/>

# Targeting Internet Explorer-Use This

```
<!--[if lt IE 7 ]> <body class="ie6"> <![endif]-->
<!--[if IE 7 ]> <body class="ie7"> <![endif]-->
<!--[if IE 8 ]> <body class="ie8"> <![endif]-->
<!--[if IE 9 ]> <body class="ie9"> <![endif]-->
<!--[if gt IE 9]> <body> <![endif]-->
<!--[if !IE]><!--> <body> <!--<![endif]-->

<!--(or better- the HTML5 version) -- >
<!--[if lt IE 7 ]> <html lang="en" class="ie6"> <![endif]-->
<!--[if IE 7 ]> <html lang="en" class="ie7"> <![endif]-->
<!--[if IE 8 ]> <html lang="en" class="ie8"> <![endif]-->
<!--[if IE 9 ]> <html lang="en" class="ie9"> <![endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html lang="en"> <!--<![endif]-->
```

# Targeting Internet Explorer-Use This

```
#wplinks #searchsubmit {  
    margin-bottom:7px;  
}  
.ie7 #wplinks #searchsubmit {  
    margin-bottom:0px;  
}
```

# Internet Explorer- HasLayout

“Layout” is an IE/Win proprietary concept that determines how elements draw and bound their content, interact with and relate to other elements, and react on and transmit application/user events.

This quality can be irreversibly triggered by some CSS properties. Some HTML elements have “layout” by default.

Microsoft developers decided that elements should be able to acquire a “property” (in an object-oriented programming sense) they referred to as hasLayout, which is set to true when this rendering concept takes effect.

- - <http://www.satzansatz.de/cssd/onhavinglayout.html>

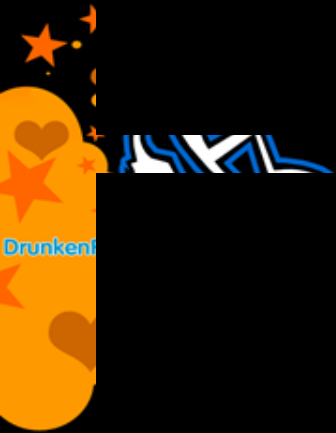
# Internet Explorer- Triggering HasLayout

- position: absolute
- float: left|right
- display: inline-block
- width: any value other than 'auto'
- height: any value other than 'auto'
- zoom: any value other than 'normal'

As of IE7, overflow became a layout-trigger.

- overflow: hidden|scroll|auto
- position: fixed
- min-width: any value
- max-width: any value other than 'none'
- min-height: any value
- max-height: any value other than 'none'

Any Questions?



Drunken



# Resources

- <http://www.w3.org/Style/CSS/>
- <http://www.csszengarden.com/>
- <http://meyerweb.com/eric/css/>
- <http://www.alistapart.com/topics/code/css/>
- <http://www.quirksmode.org/css/contents.html>
- <http://www.w3.org/Style/Examples/011/firstcss>
- <http://www.w3schools.com/css/default.asp>
- [http://www.westciv.com/style\\_master/academy/css\\_tutorial/](http://www.westciv.com/style_master/academy/css_tutorial/)
- <http://molecularvoices.molecular.com/standards/>
- <http://handcraftedcss.com/> (book)
- <http://www.zeldman.com/dwws/> (book)

# Thanks



[@deathbearbrown](#)

[@jayroh](#)

[@unruthless](#)

[@ryanobrooks](#)

[@adamjmcintyre](#)