CSS and Web Design

- Learning about CSS
- What sort of menus can I make with CSS?
- Top Ten Web Design Mistakes of 2005
- The importance of "!important" in CSS
- CSS Design Concerns for IE6, IE7, and Firefox
- Forcing a page break with CSS
- What's a solid starting point (global reset) for a CSS file?
- UX Team (UCLA Library - Digital Initiatives & Information Technology)
- Hi, are there any UCLA style resources or style guides for websites?
- UX Resources
- What to do when CSS stylesheets refuse to apply
- Web Accessibility Resources
- Sass versus LESS

Learning about CSS

CSS, or Cascading Style Sheets, have taken over the design end of websites, it seems. I'm not that good with them yet, so I'm starting to collect useful CSS Links.

- http://css.maxdesign.com.au/ – *Great site to learn CSS. Straightforward to find available topics.*
- http://www.csszengarden.com/ – *an amazing demonstration of what can be accomplished visually through CSS-based design. Each link on the side shows exactly the same text with a completely different CSS design and layout.*
- http://businesslogs.com/design_and_usability/my_5_css_tips.php
- http://del.icio.us/tag/css

Here is a list of Books that were recommended thru the CWP group.

1. Designing With Web Standards – Jeffrey Zeldman

2. CSS Definitive Guide OR http://proquest.safaribooksonline.com/0596527330
has a ton of info — better as a reference than to just read since it's information overload – it covers all sorts of minute details on CSS and is guaranteed to give you a headache.(Sue)

3. CSS Cookbook by Christopher Schmitt -

4. The Zen of CSS Design: Visual Enlightenment for the Web is a great book for design ideas from a more artistic than technical standpoint.

5.Web Developers Handbook

6.Bullet Proof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS (Paperback) by Dan Cederholm (Author)

7. CSS Mastery: Advanced Web Standards Solutions (Paperback)
by Andy Budd (Author), Simon Collison (Author), Cameron Moll (Author)

Also, don't forget that UCLA has access to Safari Online Proquest) which lists 5 or 6 books on CSS that you can read online for free from a UCLA computer (or using the VPN from off-campus).(Sue)

What sort of menus can I make with CSS?

There are practically no limits to what you can do with CSS, the trick is figuring out how to make it do what you want! Or another way of looking at it is: what sort of ideas can you come up with to use CSS in a new (and hopefully useful) way?

Here are just a few common things you'll see sites do with CSS:

- Drop-down menus that show up when you hover your mouse over them
- Pull-up menus, that go up instead of down
- Horizontal menus
- Nested hierarchical menus
- Menus with images in them
- Stylized borders

Tons of great examples can be found here: http://www.cssplay.co.uk/menus/index.html

Top Ten Web Design Mistakes of 2005

By Jakob Nielsen:

http://www.useit.com/alertbox/designmistakes.html

The importance of "!important" in CSS

Normally, CSS works by having the most recently-declared rule take precedence. However, this isn't always the case. Rules can be followed by the expression "!important" to give them precedence over later rules.

This can come up a lot in systems like Plone, where a large amount of CSS has already been written, to which you make your own additions. The existing CSS may use !important rules which will override any rules you declare, no matter where you place them. Needless to say, this can be very confusing, since it goes against what most of us have been taught about CSS.

Most of the time, you'll be using !important to get your new CSS to override some existing rules that have been declared !important. An example usage would be:

```
p { font-size: 10px !important; }
```

This makes sure that later CSS, including user stylesheets that browsers sometimes apply, does not override this rule. However, many times you'll want to override a rule that has already been declared !important. To do so, just make your new rule !important as well, and place it somewhere after the rule you want to replace.

Internet Explorer 6 and earlier versions do not recognize the !important rule. If you are trying to override someone else's !important rules, this does not matter much, since your later rules will override them no matter what. However, if you are trying to create new rules that user stylesheets won't override, you're out of luck. IE7 should fully support the rule.

The official W3C word on !important can be found at: http://www.w3.org/TR/REC-CSS2/cascade.html#important-rules

Number 4 out of Ten CSS tricks you may not know mentions the lack of !important support in IE6, and how it can be used to feed it browser-specific code.

CSS Design Concerns for IE6, IE7, and Firefox

Below is the beginning of (hopefully) an ongoing collection of design-related CSS issues concerning different browsers. For this initial posting, I focus on Firefox, IE6, and IE7.

The good news regarding IE7 is that a lot of bugs were fixed. See the following for a list of fixes.

Unfortunately, those improvements leave a fairly large gap between how IE6 and IE7 render the same CSS. See http://kimblim.dk/csstest/ for a good look at CSS testing of Selector and Pseudo-selectors for Firefox, IE6, IE7b3, and Safari. You will notice that IE7 and Firefox display more similarly than IE7 does with IE6.

If you looked through the above site, you noticed that IE6 did not properly display any of the Selectors and Pseudo-selectors that were tested. In an effort to demonstrate some of the steps that can be taken to make the same page (or element within a page) display similarly across IE6, IE7, and Firefox, I listed some **workarounds** (min/max-width, inline vs. block-level elements) for a few of the tested Selectors and Pseudo-selectors. You will find them at:

http://www.sscnet.ucla.edu/ssc/lederman/browser_design_issues_1.html.

Some general ways to deal with IE6:

In an effort to have your web pages validate, use Conditional Comments to write browser specific rules/code, etc. You can find more info at:

http://msdn.microsoft.com/workshop/author/dhtml/overview/ccomment_ovw.asp.

When using Conditional Comments, make sure that the links for them come later than the main CSS file link in the coding, AND that the rules within the Conditional Comment have at least the same or higher specificity than any similar rules in the main sheet. Otherwise, the rules in the Conditional Comments will not override the main CSS.

Though not an ideal solution, hacks can be helpful at times. A list of CSS hacks.

Two common hacks, the Star HTML hack and the Underscore hack, both IE-specific, (you can see a list of common hacks specific to which browsers at http://www.centricle.com/ref/css/filters), can be avoided using the following approach:

Write a rule, adding the IE-specific declaration.

```
.title h3 { height: 21px; }
```

After each rule, add a second rule using a child selector. This is key because IE6 will ignore this rule, but Firefox and IE7 will apply it.

```
.title h3 {height: 21px; }
.title > h3 {height: auto; min-height: 21px; }
```

Though this leaves your CSS a bit bloated, it is an approach that avoids using these two common hacks in favor of a CSS-based solution. I should note that I havenï¿½t tested this approach yet using really *complicated* coding. Iï¿½ve read that the above method requires you to be operating in strict mode, however, I found the method to work just fine (as far as I tested it) while in transitional mode. Test away!

Forcing a page break with CSS

Here is a simple style sheet method for making a web page have page breaks, or form feeds in it.

Example:

Try going to the following example page and printing it. It will come out on two separate pages.

http://www.sscnet.ucla.edu/test/formfeed.htm

Source Code:

```
1. <html><head><title>Form Feed Test</title>2. <STYLE TYPE="text/css"> P.newpage {page-break-bef
```

Explanation:

- Line 2 is where you set up your style sheet definition.
- Line 6 shows what you put wherever you want a page break.

What's a solid starting point (global reset) for a CSS file?

Browsers often have different ways of rendering the same element. For example we expect lists to be left-indented. Firefox does this by applying a left padding of 40 pixels to an unordered list (`<ul>`). Either Opera or Internet Explorer acheives the same effect by applying a *margin* of 40 pixels.

Now this is fine and dandy if you don't want to change that indent value.

Let's say you test in Firefox want to pull the list back a little so you:

```
ul {  padding-left: 30px;}
```

Looks good… until some other browser adds that 30 pixels of padding onto their 40 pixels of margin. Crap. The original solution proposed was simple and elegant:

- {
  padding:0;
  margin:0;
  }

It also broke forms and obviously is not inclusive of everything for a true "global whitespace reset."

You can read more on this implementation (which was revised) on the now classic leftjustified.net post.

Yahoo! have their own CSS Reset, which has a broader scope. Not sure setting h1 through h6 at 100% is an improvement over default behaviour though. ^^

An alternate (still in progress) reset is up at Eric Meyers website, and has evolved through a couple comment heavy blog posts.

UX Team (UCLA Library - Digital Initiatives & Information Technology)

The UX Team is charged by the UCLA Library Digital Initiatives & Information Technology leadership to improve the user experience of all library digital interfaces by

- Adopting an improved and systematic UX process on new projects/products
- Auditing and remediating legacy applications
- Maintain that capability by moving from Stage 3 to Stage 4 in the maturity model
- UX prior to coding
- Design as problem-solving process (driven by data, goals) not just style and form
- Increase repertoire of UX tools
- Expand UX team
- Achieve official designation
- add new roles and membership
- UX champion in upper management

UX Team
Joshua Gomez
Tinu Awopetu
Ashton Prigge
Sharon Shafer

Hi, are there any UCLA style resources or style guides for websites?

Edit: 9/7/2012
UCLA released Brand Guidelines 1.0 in March 2012. The document contains guidance on things like fonts, colors, tone, photography, and design. There are new colors and a lot of colors got discontinued, like all of the greys.

UCLA also released the templates for the web gateway in HTML only. They are available here: www.images.ucla.edu. UCLA employees will be able to log in via their Bruin Online ID. If you're not an employee (ie you are student or a volunteer), you will need to register for an account.

—Sirinya Tritipeskul Matute, UCLA Fund

Here is the website for the UCLA Graphic Identity: http://www.identity.ucla.edu/
It has information on the UCLA Logo and colors.

I would suggest you visit The UCLA Campus Web Publishers (CWP): http://cwp.ucla.edu/
There you will find many resources including the above site.

UX Resources

http://ux.stackexchange.com/

https://www.reddit.com/r/userexperience/

http://uxmyths.com/

What to do when CSS stylesheets refuse to apply

There are a number of common mistakes users make when writing CSS stylesheets that are difficult to debug. Because browsers differ in how picky they are about errors, pages may look fine in some browsers but wrong in others. A common symptom of these problems is large sections of your page that seem to have no styles applied to them at all.

One mistake when writing CSS is to comment your stylesheets using the hash sign (#), which is used in many languages such as Perl and Python to denote comments. In CSS, however, it's used as an ID selector, meaning your "comments" will be interpreted as strange CSS. The proper way to comment CSS code is to use C-style comments, which begin with `/*` and end with `*/`. They do not have to begin and end on the same line, but take care not to nest comments.

Another mistake is omitting the semicolon (;) after each attribute. You don't need semicolons after selectors or brackets, just after assigning attribute values such as `border: none;`. Some browsers are fine without semicolons, others are not. To be safe, always use them.

Finally, try adding the phrase `!important` at the end of a rule that refuses to apply, right before the semicolon. (For users of IE 6 and below, don't bother, this keyword isn't supported.) See the importance of !important in CSS for an explanation why.

A great resource when debugging tricky CSS problems is the W3C CSS validator: http://jigsaw.w3.org/css-validator/

Here's another: Will the browser apply the rule(s)?

Web Accessibility Resources

This article is a resource list to help Web developers design sites that are accessible to persons with disabilities. The list will evolve as technology changes, and also in response to issues raised by UCLA Web developers. It will be further supplemented by other materials produced by the UCLA Disabilities and Computing Program .

See also other articles in this Knowledge Base under the “accessibility” tag.

UC System Accessibility

“Accessible Web Design Resources ":http://www.ucop.edu/irc/itaccessibility/resources/ This will collect best practices for Web designers across the UC’s. For those already familiar with basic accessibility concepts, the Design Tips and Technical Topics pages will be the most immediately useful.

WebAIM

Web Accessibility In Mind (WebAIM) at Utah State University has emerged as a leading reference site for many accessibility topics.:

Introduction to Web Accessibility from WebAIM

Resources from WebAIM , including Evaluating Web Sites for Accessibility with the Firefox Web Developer Toolbar ;

The WAVE Accessibility Tool , and many more.

Others

World Wide Access: Accessible Web Design (University of Washington, DO-IT Program)

“Accessibility tagged articles from 456Berea Street blog ":

http://www.456bereastreet.com/archive/categories/accessibility/ (current issues in accessibility and usability)

Sass versus LESS

Quoting, "One of the hot new trends in web design is CSS pre-processed languages and there's two big ones vying for your attention—LESS and Sass. LESS and Sass are both ways of writing CSS code with a syntax that allows you to use features not yet available with Cascading Style Sheets (CSS), such as variables, nesting, conditionals, and more.

Pre-processed CSS languages add features to CSS that aren't there yet—like variables, conditionals, and functions. They're called pre-processed, because their final step is a processing, also called compiling, that converts the pre-processed language to regular CSS. In a nutshell, what you use on your site ends up being plain vanilla CSS, but comes as a result of processing the LESS, Sass, or other pre-processed language files you create." *via Lynda.com blog*

Review links—

- http://coding.smashingmagazine.com/2011/09/09/an-introduction-to-less-and-comparison-to-sass/
- http://css-tricks.com/sass-vs-less/
- http://www.hongkiat.com/blog/sass-vs-less/
- http://blog.lynda.com/2012/07/20/an-introduction-to-less-and-sass-pre-processed-css-languages/

Lynda.com has modules on the topic—

- **CSS with LESS and Sass**, http://www.lynda.com/CSS-tutorials/CSS-LESS-SASS/107921-2.html
- **Responsive CSS with Sass and Compass**, http://www.lynda.com/CSS-tutorials/Responsive-CSS-Sass-Compass/140777-2.html

Resources—

- http://sass-lang.com/
- http://lesscss.org/
- http://incident57.com/codekit/
- http://gruntjs.com/
- http://compass-style.com/
- http://css-tricks.com/semantic-class-names/
- http://codepen.io/