Digital Evolution

The current state of transferring print design principles to the web

by Kate Burkett

Introduction

As the internet continues its march toward ubiquity in human society, web design tools continue to allow for more and more easy implementation of print design principles. This makes sense; after all, the goal of print design principles, especially those promoted by typographer Robert Bringhurst in *The Elements of Typography*, are to create a comfortable, pleasing reading experience. Today, web pages and web designs look extraordinarily more interesting and dynamic than those that came just a decade before. The question is not whether there is a desire to transfer print principles to the web, the question is which principles are currently possible to transfer, and because of the very different experience of reading on a screen versus a page, which principles *should* be transferred from print to the web.

The tools of web design: css & you

Even though websites are more sophisticated than ten years ago because the tools to design them are more sophisticated, the ability to significantly alter the presentation of web pages is a relatively new phenomenon. The most important catalyst in transferring print design principles to the web was the standardization of css1 as a web design language in 1996. css, or Cascading Styles Sheets, is what controls the presentation of elements on a website and separates that presentation from the content. Just like paragraph styles in professional text editing software, css styles change the appearance of a document's content without changing the content itself. Before 1996, web pages were styled primarily with html—a tedious and clunky procedure that made consistency in layout difficult. This is why early web designs were limited and often simplistic.

Additions to the css language, called *specifications*, have to be "recommended" by the World Wide Web Consortium (w3c) in order to be considered stable for use and adopted by all major web browsers. The w3c is an unofficial community that works to develop the standardization of web guidelines—they're basically artificial selection in action; the "breeders" of a continually advancing web. For this reason, they can be said to control the evolution of web design.

The current css iteration, css3, works differently than previous specifications. It's divided into several specification parts called *modules* that are organized by similar features and can be recommended separately from one another. Even though early drafts of css3 were published in 1999, the first module was not recommended until June 2011. Only three other css3 modules have been recommended to date, the last occurring just six months ago in June 2012.[1] Because css3 isn't technically "finished" yet, only the modules recommended by the w3c are adopted by a majority of web browser and can be safely used in web design today.[2]

The capabilities of css3 are slowly but surely shaping the web to more visually resemble print media, but because of the limited scope of css3 in relation to professional text editing programs like Adobe InDesign, elaborate coding hacks are more common today than simple css rules. Some principles are made easier with the web language, but if css doesn't have a feature, called a *declaration*, built in, you have to get creative. For instance, in the future, css4 may contain a declaration for easily creating drop caps, but for now, there is no simple line of code that makes foolproof drop caps possible in web design.

Letting go of control

Web design is intrinsically different from print design in several ways. While printed text is not easily separated from its presentation, in best practice, presentation and content on the web are *fully* separate. Unlike the fixed layout of a printed page, the screen size, resolution, or browser used by the reader of a website can't be controlled by the designer. Additionally, while the entirety of a page is viewed all at once in printed media, only a fraction of the design is typically visible when you first visit a webpage. As a result, important information must be put "above the fold" or it risks being overlooked. These and other considerations greatly affect the way web pages are designed and must be accounted for.

Designer Khoi Vinh proposes that the difference between graphic design and web design can be boiled down to designer control. Every element of a webpage cannot be managed exactly; this results in the transfer of control from the designer of the page to the user. Web page layouts and typography become simply suggestions.

Fortunately, Vinh looks at this as a challenge. In web design, "there are more states to design," he says, so the author must pick and choose battles while he or she patiently waits for the new features of css3 to be adopted and standardized across all browsers.[3]

Different browsers

When you look at your website on different web browsers, you might find that things don't look as you expect them to. This is because there are many different types of browsers—Chrome, Firefox, Safari, Internet Explorer—created by different companies, and each has its own basic style sheet built in to the application. Because of this, each browser interprets computer code slightly differently than its competitors. In order to truly start with a blank "page" when designing for the web, overrides to the default styles must be added to the css file.

The loss of control due to these nuances between web browsers is very frustrating for web designers. With each new version of a browser, more css3 properties are supported, but often times users do not update their software and designers must still cater to them, resulting in clunky rendering of the code. Browsers that are still used by a significant fraction of people, such as Internet Explorer 8, for example, do not support the border-radius declaration. In order to use rounded corners in your design, extra attention and time must be spent to create a fix for this out of date software.

Designing for multiple screens

Before the surge of internet-browsing devices of all shapes and sizes—from smart phones, to tablets, to laptops, to 27-inch iMacs—web pages, like print, were designed with a fixed-width. As previously discussed, today the canvas of possible device screens means that web designs have to be adaptable to the changing environment around them. Web designs can no longer be absolute, or they will struggle with surviving amongst the adaptable designs now conveniently used across several devices.

Because of the influx of aspect ratios in internet-capable devices, not to mention the ability for the user to change the size of the browser window, web designers have little control over the "trim size" of their layout. The only certainty is that the page height is boundless. Very recently, with the rise of css-based grid systems and the standardization of the Media Queries css3 module, designers have gained adequate control over how their designs adapt to different screen sizes. This is called responsive design.

Grid Systems

Grid systems use columns and rows to organize text and make sure line lengths aren't too many characters, just like in print design. They impose order on the chaos of different browser sizes. Constructing makeshift grids used to involve HTML elements for tables and was extremely tedious, but many Open Source grid systems are now available to make fluid layouts more accessible to every web designer.

W3.org has this to say about css-based grids:

> "...the absence of content structure in the Grid helps to manage changes to layout by using fluid and source order independent layout techniques. By combining media queries with the css properties that control layout of the Grid and its children, authors can adapt their layout to changes in device form factors, orientation, and available space, without needing to alter the semantic nature of their content."[4]

Now, instead of a single design having to account for every possible aspect ratio, multiple design iterations can be created to affect one HTML file, resulting in an optimized display across all devices.

Media queries

Grid systems allow for fluid layouts, but media queries make responsive design possible. Recommended by the w3c just six months ago in June 2012, media queries specify parameters for when each design should be used. For example, if the width of a screen dips below 480px, like on a smart phone, the media query tells the browser to use a specified design catered to the smaller width.

Along with the fluid layout allotted by grid systems, this allows popular design ratios like 20th-century typographer Jan Tschihold's golden section ratio to be more easily adhered to when viewed at different sizes.[5] Responsive web design allows ratios used in web design to transmit higher meaning, adheres more closely to print principles, gives more control back to the designer, and heightens user experience.

A brief history of digital type design

Digital screens are made up of pixels. When Apple was adding typefaces to its Macintosh computers in 1984, their screens had a modest 72 pixels-per-inch (PPI) and 512×384 resolution. These statistics account for the appearance of their font Chicago, which was created specifically for their user interface. The font was a bitmap, which means its construction, mixed with the low PPI and resolution of the screen, allowed it to only be displayed at one size.[6] This was the height of possibility for typography on the screen in 1984.

Over time, PPI and screen resolution grew larger, but never large enough to reach the 1200 dots-per-inch that magazine and book typography are printed in.[7] This severely restricted the fonts that displayed well on computer screens and ushered forward more typefaces commissioned with digital intent in mind. Arial, Verdana, Georgia, and Trebuchet MS were all created for use digitally, and are designed to take into account the limitations of digital display. These fonts were commonly found in web browsers because they had more open counters and large x-heights that made them easier to read on screens.

A possible resurgence for serif fonts on the web

Previously, opposite to print design principles, web designers considered san-serif fonts easier to read in long passages of text. This is because low screen resolutions and pixel densities caused the subtle detail in serif fonts to be muddied and less readable in small sizes. Georgia, for example, a font created by Matthew Carter for Microsoft in 1996, was designed with sharp serifs and a large x-height in order to optimize it for screen reading.

With advances in screen manufacturing that allow for a much higher pixel density and resolution, restrictions on what typefaces are legible on screens may soon be a thing of the past. Mobile phones now have pixel densities as high as 326 PPI, and desktop computers look to be doubling their pixel densities in the near future.

Why web safe fonts are old news

In the recent past, if a font wasn't located on a user's computer they were not able to see it implemented in a web page design. Fonts had to be available on the device accessing a web document, significantly limiting the control designers had over the typefaces they used in designs. It was highly recommended that fonts deemed to be "web-safe," or fonts that were likely to be included on all computers, be used in web typography.[8] Not surprisingly, these fonts included Arial, Helvetica, Georgia, Verdana, Tahoma, and Times New Roman.

With the addition of the @font-face CSS property and server-hosted web font directories like Typekit and Google Web Fonts, this is no longer a concern. The Google Font API in particular, released two years ago in 2010, has opened up hundreds of new Open Source fonts for designers to use free of charge. Because the fonts are located on Google's servers, all that's needed is a few lines of CSS and almost every web user can access and see the font used in a website's design.

While this opens up the pool of fonts available for web design significantly, only fonts licensed for use by web font directories can be accessed this way. Classic fonts belonging to foundries, like Baskerville, would still only be visible to those who have the font embedded in their digital device.

Real & fake bold

Though many new typefaces created for digital display are now more easily accessed for web design, a number of these faces don't contain different font weights and font styles. If this is the case, tagging these fonts as <b> or <i> in HTML will output what Robert Bringhurst calls "fake bold" and "fake italic" letterforms. Fake bold refers to the outline a computer draws around each letterform in order to thicken its appearance, but in typography, this is not how bold letters are created. Separate sets of letterforms are specifically drawn to appear with heavier weight. A similar situation is true for italic letterforms. Fake italic is outputted when the computer slants the letterforms slightly, while real italic letterforms are crafted with more complex subtlety when they are created.

Additionally, using the declaration font-variant: small caps; will almost always create fake small caps. In order to circumvent this typographic discretion, use fonts that come with the desired letter sets and use the @font-face CSS property for each weight or style of characters.

Css declarations & typography

CSS declarations are available to control many of the elements of typography used in print design. While some declarations have received recommendation from the W3C and are in widespread use, other declarations exist but have not yet been implemented in all major web browsers.

Css declarations ready for use include:

background-color: specifies the background color of an element. Parallel to the color of print material, white and off-white are the most prevalent colors used for backgrounds in web design.[9]

color: specifies the color of element's text. Similar to print, web designers are discouraged from using pure black for a text's hue.

font-family: specifies the typeface used in the element, typically in conjection with @font-face. Like print, typically two separate typefaces, one serif and one san-serif, are enough for an individual design.

font-size: specifies the size of an element's font. A typical web font-size can go as low as 12px, but many modern web designers such as Patrick Winfield and Oliver Reichenstein advocate that a larger size of 14–16px would better suit web typography. It depends on the website. Sophisticated websites that rely on typography to create a visual mood look better with larger text and more generous line spacing, but on a busy e-commerce site, for example, this would not be as feasible.

Using ems for specifying font-size allows hierarchies between textual design elements to be precisely scaled. Scaling with ems can be specified to three decimal places. Richard Rutter, creator of *The Elements of Typographic Style Applied to the Web*, offers this example:

```
body { font-size: 100%; }
h1 { font-size: 2.25em;  /* 16×2.25=36 */ }
h2 { font-size: 1.5em;   /* 16×1.5=24 */ }
h3 { font-size: 1.125em; /* 16×1.125=18 */ }
h4 { font-size: 0.875em; /* 16×0.875=14 */ }
p  { font-size: 0.75em;  /* 16×0.75=12 */ }
```

letter-spacing: specifies the distance between letters. For extremely detailed web typography, the spacing between capitals and small capitals, as suggested in *The Elements of Typography*, can be altered. Letter-spacing equally distances all the letters surrounded by an HTML tag and should not be confused with kerning.

line-height: specifies the leading of a textual block element. Like in print, David Kadavy suggests web line-height be 120 percent to 140 percent the font-size. However, typographer John Boardly proposes that it should be *at least* 140 percent. The choice, again, depends on the practicality of implementation.

text-align: specifies the alignment of text. It's widely considered that justified text should rarely, if ever, be used for web design. This is because browsers do not yet have good hyphenation dictionaries, so rivers in the text are easily formed when justified. This may change, however, as hyphenation dictionaries improve and more CSS3 declarations gain browser-support. For now, text-align: left; is the standard.

text-indent: specifies the indent of the first line in a block of text. Though block paragraphs are currently the default style and the norm in web design, text-indent makes both options possible. A 1 em text-indent is recommended.

Css properties that exist but are not yet widely supported include:

hyphens: specifies how hyphens should be used in a block of text. Browser support started for this property within the past year, but it is not yet supported in Chrome. Vendor prefixes allow hyphenation in Firefox and Safari, but the hyphenation dictionaries for these browsers are spotty at best.

hyphenate-before & hyphenate-after: specifies the minimum number of characters in a hyphenated word before or after the hyphenation character. Not yet implemented by any web browser.

hanging-punctuation: specifies whether punctuation should be placed outside the element's box at the start and/or end of a block of text. Not yet implemented by any web browser.

widows: specifies the minimum number of lines that should be left at the top of a page. Only supported by IE8 and later.

orphans: specifies the minimum number of lines that should be left at the bottom of a page. Only supported by IE8 and later.

Typographic principles with no CSS equivalent include:

Kerning: it would be possible to surround duos of letters with the letter-spacing property to create specific kerning, but it's doubtful this would work across all browsers to output the desired result. According to Richard Rutter, similar to fonts in Adobe InDesign, "text on the Web does not generally need manual kerning, as all digital fonts have kerning tables built-in."

Baseline grid: fluid columns and rows can be created with CSS-grid systems, but no specific declaration exists to control this. Composing all elements with a precise scale is possible to create the effect of a baseline grid, but only with creative hacks and a lot of math can this be achieved.

Text-wrap: Because of the nature of CSS Box Model, which wraps a box around each HTML element, the computer sees every image as a rectangular shape. Even if your image is not a rectangle, like a PNG file of the branches on a tree, the computer will see the image as a box and surround it with the appropriate white space in relation to other elements. Due to this, text elements don't conform to the shape of an image, they conform to the box around it and can't be easily wrapped.

The details: HTML entities

HTML entities are special character codes that are assigned to symbols not easily produced on a keyboard. To ensure that the correct characters are being produced in web design and that print typography principles are being upheld, designers must use these codes.

Helpful HTML entities for typography include:

Open quotation: &ldquo;
Closing quotation: &rdquo;
Opening single quotation: &lsquo;
Closing single quotation: &rsquo;
Feet: &prime;
Inches: &Prime;
Copyright: &copy;
Ampersand: &amp;

If desired, HTML entities exist for all types of dashes and spaces, but browser support and typeface inclusion for these characters vary.

Helpful HTML entities for spaces and dashes include:

Em dash: &mdash;
En dash: &ndash;
Hair space: &#8202;

En space: &ensp;
Em space: &emsp;

HTML entities exist for common typeface ligatures, but because of the conflicts they create with the Find command in web browsers, using them can be problematic. Additionally, if a typeface is changed using CSS to one that does not contain a particular ligature within its set, this will cause problems in the display of the text. A CSS3 declaration called font-variant-ligatures proposes to solve these problems, but is only currently supported by Firefox.[10]

Conclusion

Because the ability to significantly and easily alter the presentation of web pages is still relatively new, and the adoption of critical css3 properties are still in flux, there are many limitations to what can and can't be transferred from print to digital design. With the continued advancement allowed by css3, however, more varied and sophisticated design is continually evolving.

Endnotes

1. Lie, Håkon W., and Bert Bos. "The CSS Saga." *Cascading Style Sheets, Designing for the Web*. 02 Dec. 2012. http://www.w3.org/Style/LieBos2e/history/Overview.html.

2. Weyl, Estelle. *What's New in CSS3*. Sebastopol: O'Reilly Media, 2012.

3. Vinh, Khoi. "Control." *Subtraction.com*. 13 Oct. 2007. 2 Dec. 2012. http://www.slideshare.net/khoiv/control-annotated#btnNext.

4. "CSS Grid Layout." *W3C.org*. Ed. Alex Mogilevsky, Phil Cupp, Markus Mielke, and Daniel Glazman. W3C, 6 Nov. 2012. 02 Dec. 2012. http://www.w3.org/TR/css3-grid-layout/.

5. Kadavy, David. *Design for Hackers: Reverse-engineering Beauty*. 142. West Sussex, UK: John Wiley & Sons, 2011.

6. Kadavy, David. *Design for Hackers: Reverse-engineering Beauty*. 85. West Sussex, UK: John Wiley & Sons, 2011.

7. Lynch, Patrick J., and Sarah Horton. *Web Style Guide: Basic Design Principles for Creating Web Sites*. 3rd ed. 205. New Haven: Yale UP, 1999.

8. Kadavy, David. *Design for Hackers: Reverse-engineering Beauty*. 83. West Sussex, UK: John Wiley & Sons, 2011.

9. Kadavy, David. *Design for Hackers: Reverse-engineering Beauty*. 265. West Sussex, UK: John Wiley & Sons, 2011.

10. Kadavy, David. *Design for Hackers: Reverse-engineering Beauty*. 342. West Sussex, UK: John Wiley & Sons, 2011.

Sources

Boardly, John. “A Guide to Web Typography.” *I Love Typography.* 28 Feb. 2008. 02 Dec. 2012. http://ilovetypography.com/2008/02/28/a-guide-to-web-typography/.

“CSS Grid Layout.” *W3C.org.* Ed. Alex Mogilevsky, Phil Cupp, Markus Mielke, and Daniel Glazman. W3C, 6 Nov. 2012. 02 Dec. 2012. http://www.w3.org/TR/css3-grid-layout/.

Kadavy, David. *Design for Hackers: Reverse-engineering Beauty.* West Sussex, UK: John Wiley & Sons, 2011.

Lie, Håkon W., and Bert Bos. “The CSS Saga.” *Cascading Style Sheets, Designing for the Web*. 02 Dec. 2012. http://www.w3.org/Style/LieBos2e/history/Overview.html.

Lynch, Patrick J., and Sarah Horton. *Web Style Guide: Basic Design Principles for Creating Web Sites*. 3rd ed. New Haven: Yale UP, 1999.

Reichenstein, Oliver. “Responsive Typography: The Basics.” *Information Architects.* 1 June 2012. Web. 02 Dec. 2012. http://informationarchitects.net/blog/responsive-typography-the-basics/.

Rutter, Richard. *The Elements of Typographic Style Applied to the Web: a Practical Guide to Web Typography*. 07 Nov. 2012. http://webtypography.net/.

Vinh, Khoi. “Control.” *Subtraction.com*. 13 Oct. 2007. 2 Dec. 2012. http://www.slideshare.net/khoiv/control-annotated#btnNext.

Weyl, Estelle. *What’s New in CSS3*. Sebastopol: O’Reilly Media, 2012.

Winfield, Patrick. “Design Trends: What’s Hot in Web Typography Right Now.” *BlueGlass*. 30 Aug. 2012. 07 Nov. 2012. http://www.blueglass.com/blog/web-typography-trends/.

Colophon

Digital Evolution is set in Adobe Caslon Pro at 11 points, with 14 points of lead. Web safe fonts were consciously avoided when choosing a typeface in order to portray the print and web design convergence of the future, rather than the limitiations of the past.

No table of contents was created due to the brevity of the manual, and Adobe Caslon Pro italic ampersands were not used for HTML entities in order to prevent confusion.

A simplistic design and colors of white, black, and red—the typographer's colors—were chosen to reflect the manual's subject matter.