SIDE BAR

MAIN CONTENT

PROFESSIONAL
WEB
DESIGN

VOLUME
2

SMASHING
MAGAZINE

Imprint

Published in March 2011

Smashing Media GmbH, Freiburg, Germany

Book Cover Design: Ricardo Gimenes

Editing: Thomas Burkert

Proofreading: Brian Goessling

Concept: Sven Lennartz, Vitaly Friedman

Founded in September 2006, [Smashing Magazine](#) delivers useful and innovative information to Web designers and developers. Smashing Magazine is a well-respected international online publication for professional Web designers and developers. Our main goal is to support the Web design community with useful and valuable articles and resources, written and created by experienced designers and developers.

Version: July 26, 2011

Table of Contents

Preface

We're seeing better interaction design and more aesthetically pleasing designs. And we're seeing more personal, engaging and memorable sites, too. But what exactly is making the difference? What new directions is Web design heading in today? What new techniques, concepts and ideas are becoming important?

This eBook describes existing and upcoming trends and explains how the Web design might evolve in the coming months and years. It also touches on what Web designers should be ready for to keep abreast of new challenges and opportunities.

The articles are the best about professional Web design that have been published on Smashing Magazine in 2010 and 2011. They have been carefully edited and prepared for this eBook. Some screenshots and links were removed to make the book easier to read and print out.

— Thomas Burkert, Smashing eBook Editor

The Current State of Web Design

Vitaly Friedman

Web design is a fickle industry. Just like every other form of artistic expression, Web design has undergone a continuous and surprisingly fast evolution. Once a playground for enthusiasts, it has now become a mature rich medium with strong aesthetic and functional appeal. In fact, we are experiencing what could be the golden era of Web design — or at least the best period thus far. We have powerful new tools at our disposal (CSS3, HTML5, font-embedding, etc.), a plethora of freely available resources, a strong design community and also (if you needed any more!) reliable support of Web standards in the major browsers.

We're seeing better interaction design and more aesthetically pleasing designs. And we're seeing more personal, engaging and memorable sites, too. But what exactly is making the difference? What new directions is Web design heading in today? What new techniques, concepts and ideas are becoming important? In this article, we present some observations on the current state of Web design. We describe existing and upcoming trends and explain how Web design might evolve in the coming months and years. We'll also touch on what we as Web designers should be ready for to keep abreast of new challenges and opportunities.

Design For Delight

As designers, our job is to communicate ideas effectively. For every particular message, we create a context in which the message would work best, guiding users to achieving their tasks, gaining their trust or convincing them of whatever we're communicating. Of course, there are endless ways to create this context. One of them is to design for visual aesthetics, surprise, joy, happiness — design for delight; design to be memorable and remarkable.

Attractive things work better and help focus and keep the user's attention. Memorable design increases excitement for products and brands, leading to increased engagement. In fact, a strong, reliable emotional relationship between your clients and their audience could be the best thing that ever happens to your career.

Although the vast majority of brands are still silent, passive and impersonal, we've observed more websites trying hard to engage our senses, whether through a strong aesthetic appeal, through witty animations in the content block or simply through a little extra attention to small design elements on the "About" page. Such designs are beautiful to look at, fun to navigate but, most importantly, memorable — for the simple reason that they are different. By adding delightful personal touches to your designs, you stand out from the crowd and give visitors something to talk about and share with friends and colleagues. And that's a good start.

You can elicit delight in a variety of settings: on your maintenance mode page, on the 404 error page, in your pre-loader, and everywhere else. The idea is to surprise visitors by giving them something pleasant to talk about.

Bounty Bev

Bounty Bev is a beverage company with a beautiful one-page design. Apart from its subtle hover effects and animations, the website has some nice extras: if you scroll down the page manually with the mouse wheel, a small pop-up appears asking you if you need a lift. The typography is strong and memorable, and the design is playful. Simple, clear and personal, the website leaves a strong positive impression.
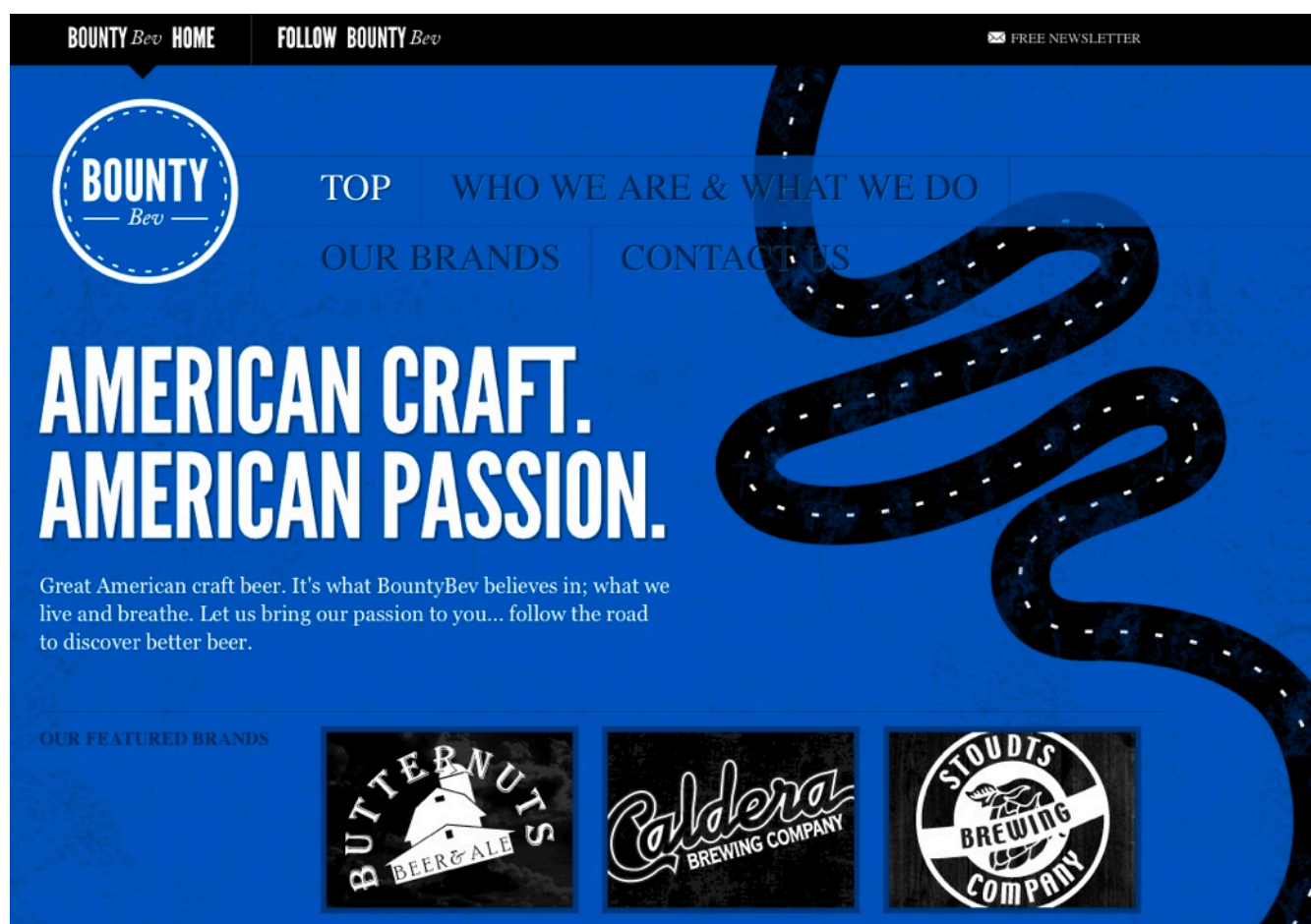
www.bountybev.com

Analog.coop

Analog provides a very personal experience to visitors. When you visit the page, it displays where you are located and tells you the members of the team who are closest to you (in our case, Alan and Jon, who are about 500 miles away in Bristol). The website has a couple of nice Easter eggs that are not visible at first glance. You might want to play around with the header and the photos of team. The page is just fun to explore.


analog.coop

Billy Tamplin

On his blog, Billy Tamplin focuses on the small victories in his life. Each post records a personal achievement, displaying a custom-designed merit badge and an explanation of the conquest. Billy uses this metaphor throughout the website, speaking of "super Web abilities" (Agile CSS, PHP-prepared, IE6-reinforced, etc.) and "heroic design strengths" (human-friendly aim, keen creative detail, etc.). He also has a personal portfolio on the website. Notice how well the color scheme fits the theme. The design is simple and beautiful, and the "achievement" twist is unusual and memorable.
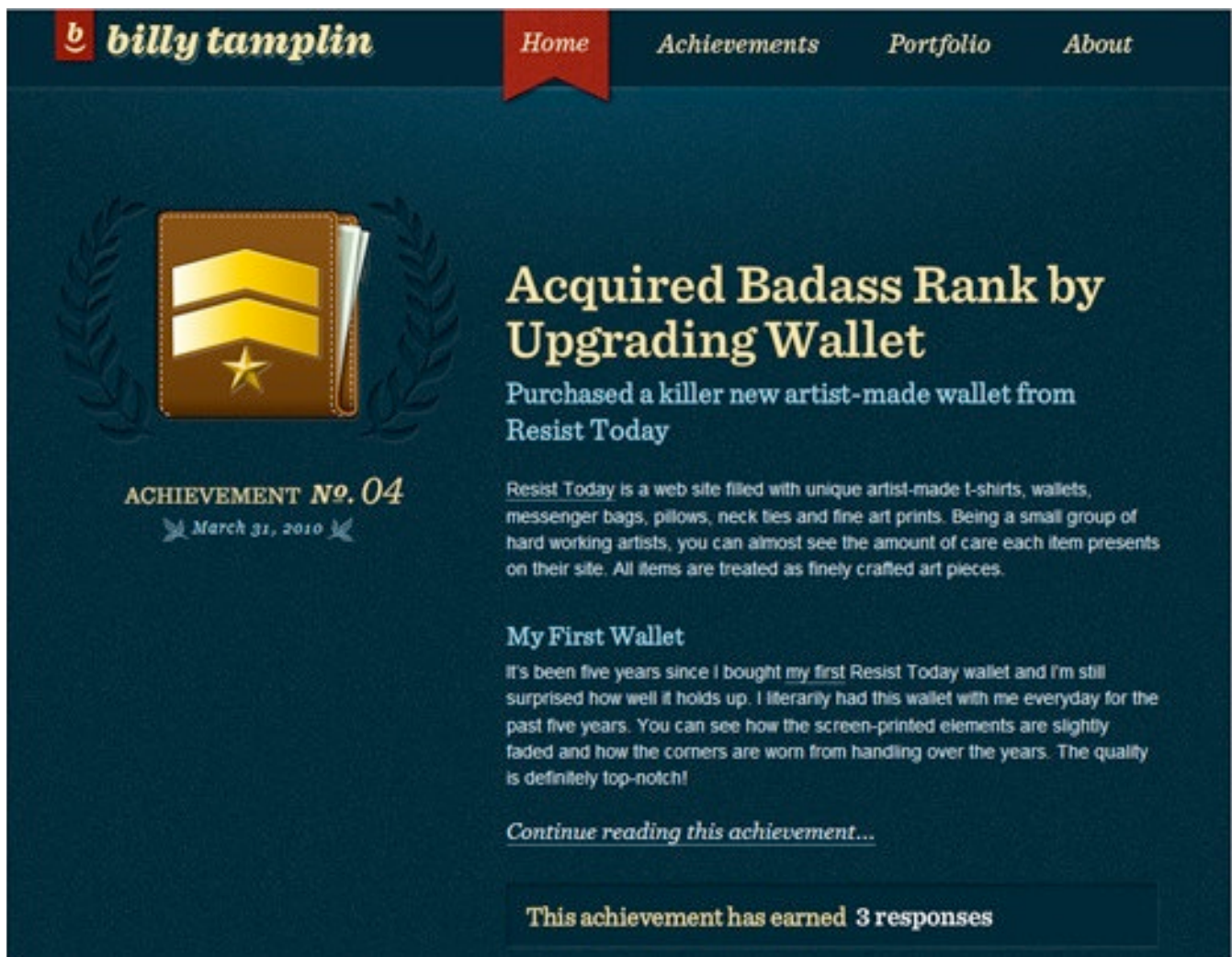
billytamplin.com

MIX

MIX labs, a community blog for designers and Web developers, doesn't have hidden features, appealing animations or striking hover effects. Instead, it has a consistent, visually appealing design: can you spot where and how often colorful circles are repeated throughout the website? The design emphasizes the content and has a personal touch. Simply beautiful.
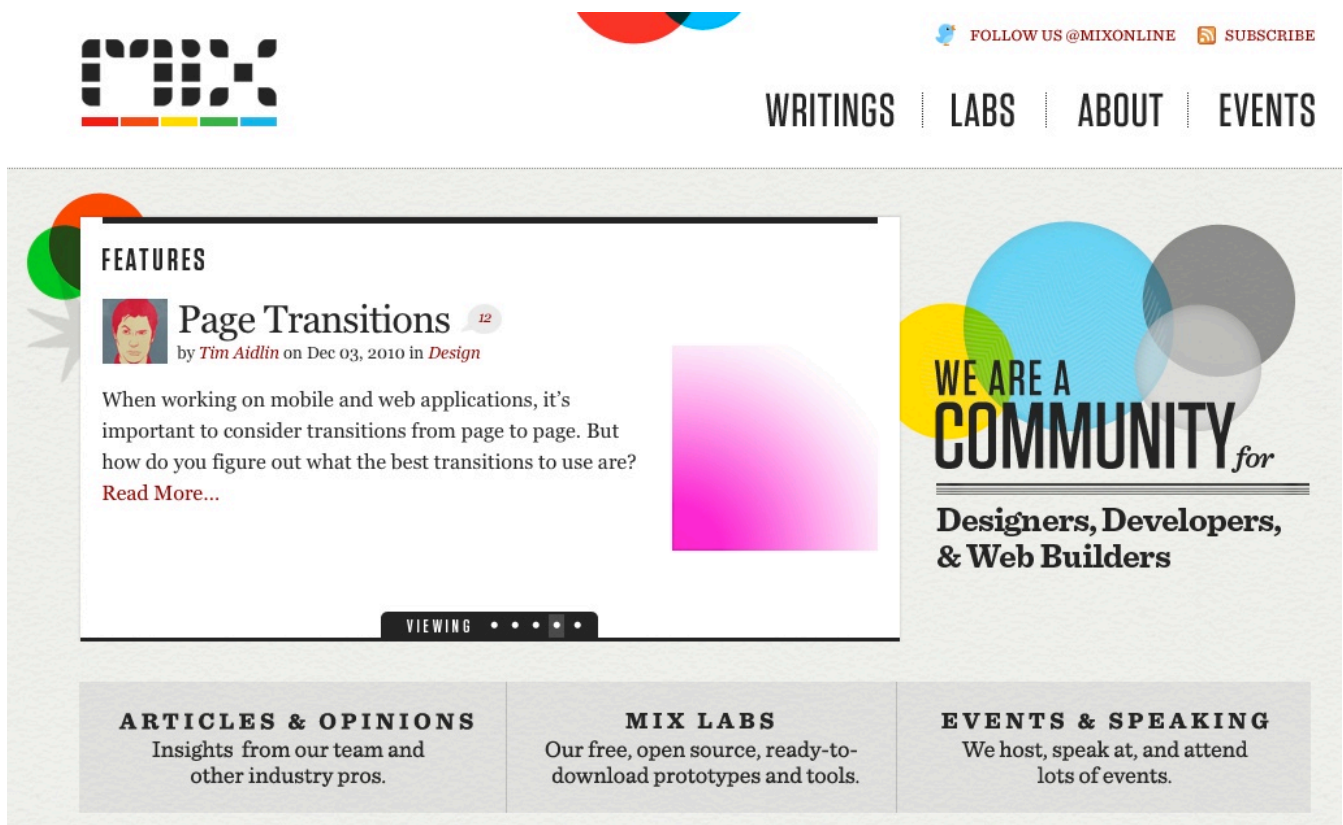
visitmix.com

Mailchimp

MailChimp heavily incorporates the monkey metaphor in all aspects of its design. To inform customers of recent updates, Mailchimp present an ASCII animation that tells the user something is happening in the background; this nice detail is surprising yet unobtrusive. The company also uses personal, friendly and perhaps occasionally geeky language when addressing user needs. This is the part of the image that MailChimp thoughtfully preserves in its Web application.
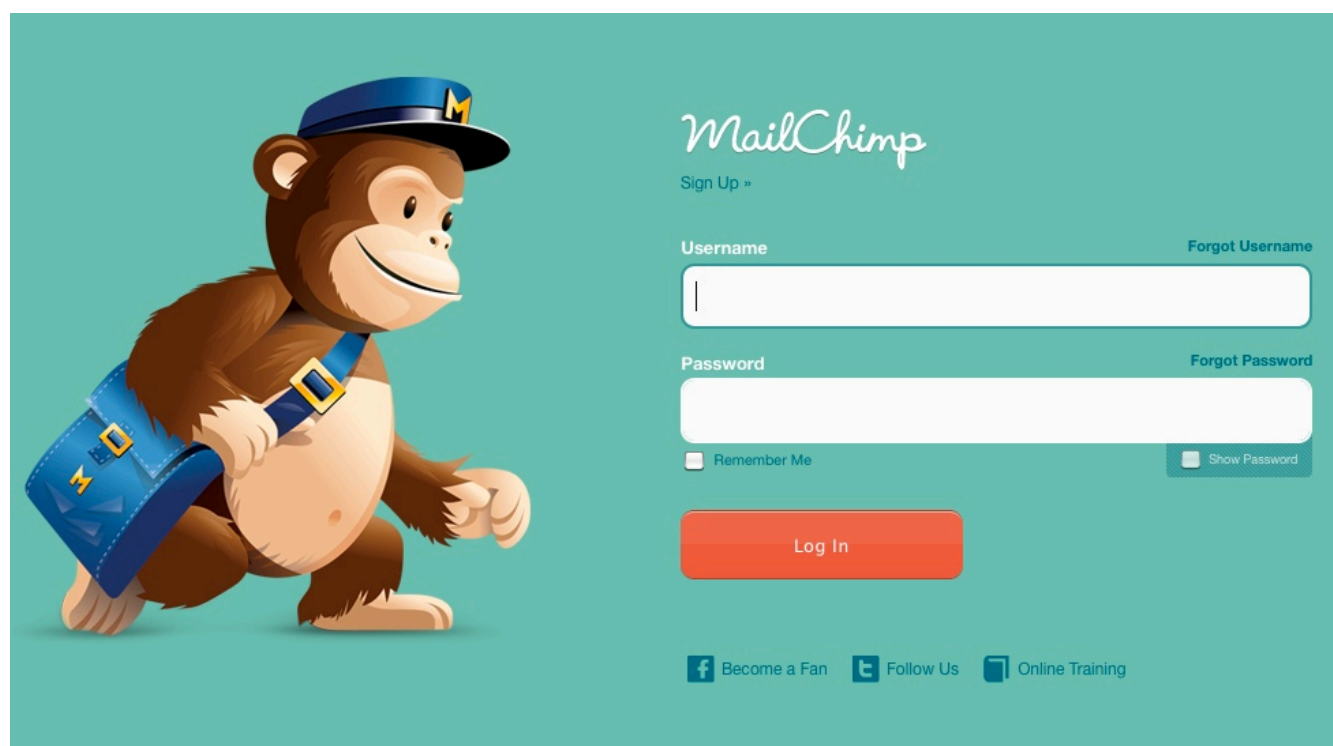

www.mailchimp.com

Keypress Navigation

As designers try to make their designs more intuitive, it is no surprise that websites are becoming more responsive. Not only does this apply to user interfaces in modern Web applications (which are becoming as robust as desktop applications — and often smarter), but with the wide adoption of JavaScript libraries, "classic" websites are becoming more robust and interactive, too. One way to make websites more responsive is through "keypress navigation," which hasn't been widely adopted so far. But lately we've observed more designs implementing this effectively. The most popular setting for such navigation is on photo websites such as Flickr or FFFFound.

The general idea is to give users keyboard shortcuts that help them perform tedious tasks, such as navigating between blog posts, moving through images in a slideshow, changing the current view (e.g. from a horizontal to vertical grid), liking articles and navigating between sections of a website. Keypress navigation is common in Flash-based designs, but we are now seeing it applied to CSS-based designs, too. [Google Reader](#) is a prime example of advanced keypress navigation, but other websites have good implementations, too.

They Make Apps

Last year, [They Make Apps](#) began offering users smooth and advanced keyboard navigation as an alternative to classic scrolling. Users could switch between both modes using a drop-down menu in the main navigation of the page. In "keyboard navigation mode," users used the arrow keys to navigate between content blocks; the "Return" key triggered the detailed view and "Escape" returned to the main page. For some reason, this navigation isn't available any longer.
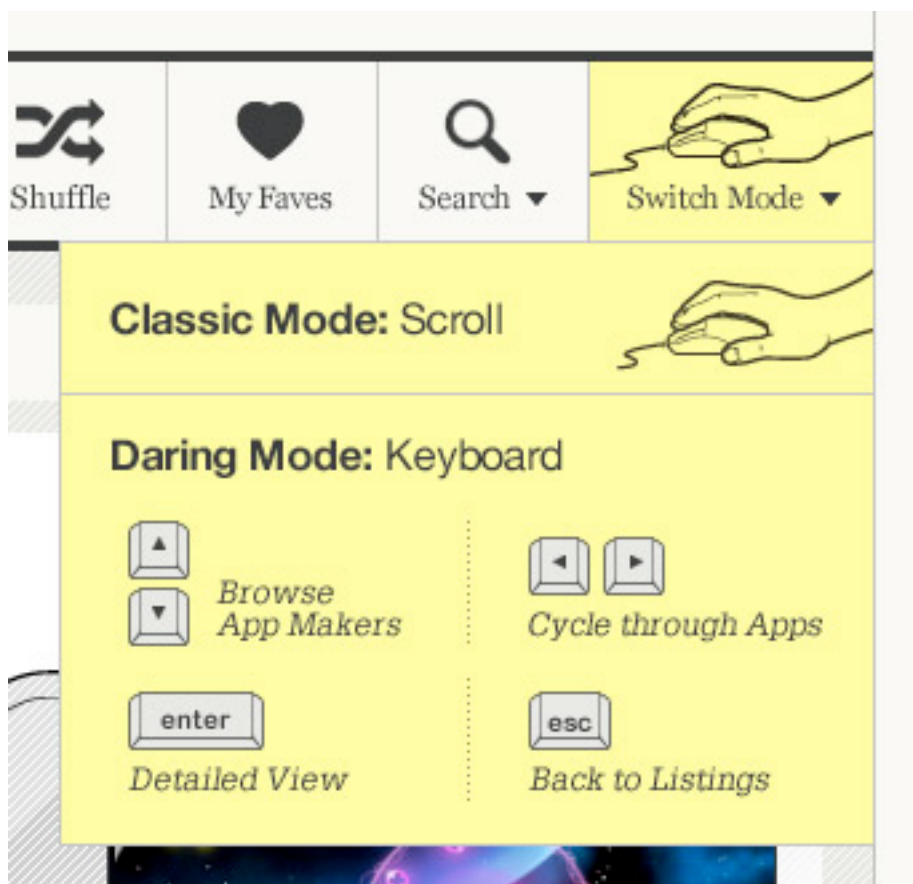

patterntap.com

Mad-ar.ch

Marc Anton Dahmen's website is Flash-based, and its navigation is quite
advanced: users can jump to the contact form with "c," scale images with "-"
and "+," and then navigate and sort images and scroll through text with the
vertical arrow keys.
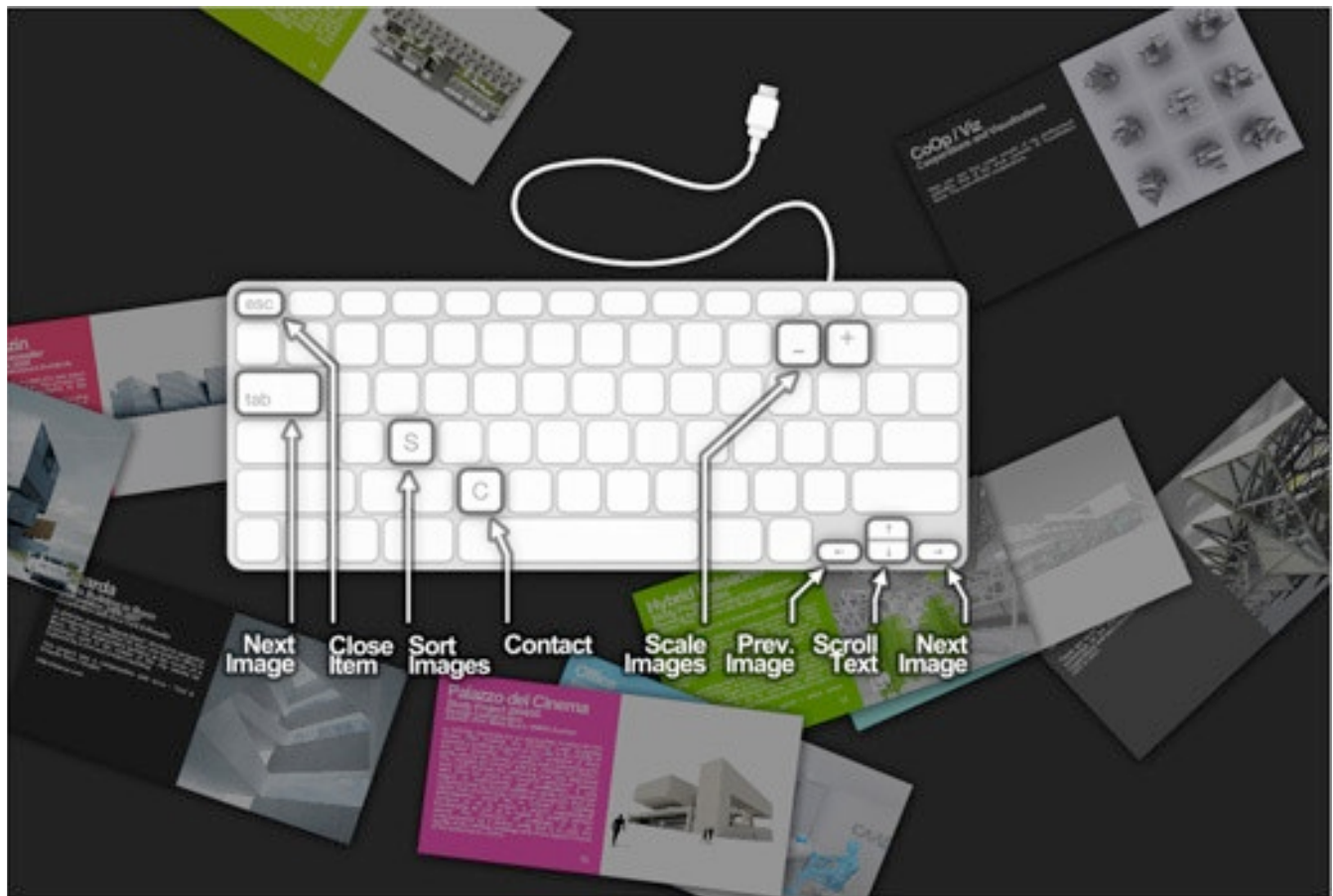
[mad-ar.ch](mad-ar.ch)

9GAG

9GAG is a social image bookmarking website. Users can navigate to the next and previous image using "j" and "k," respectively. The current image can be voted up using "l" (for love): no mouse scrolling necessary. In this case, a shortcut to the grid view would be useful, too.
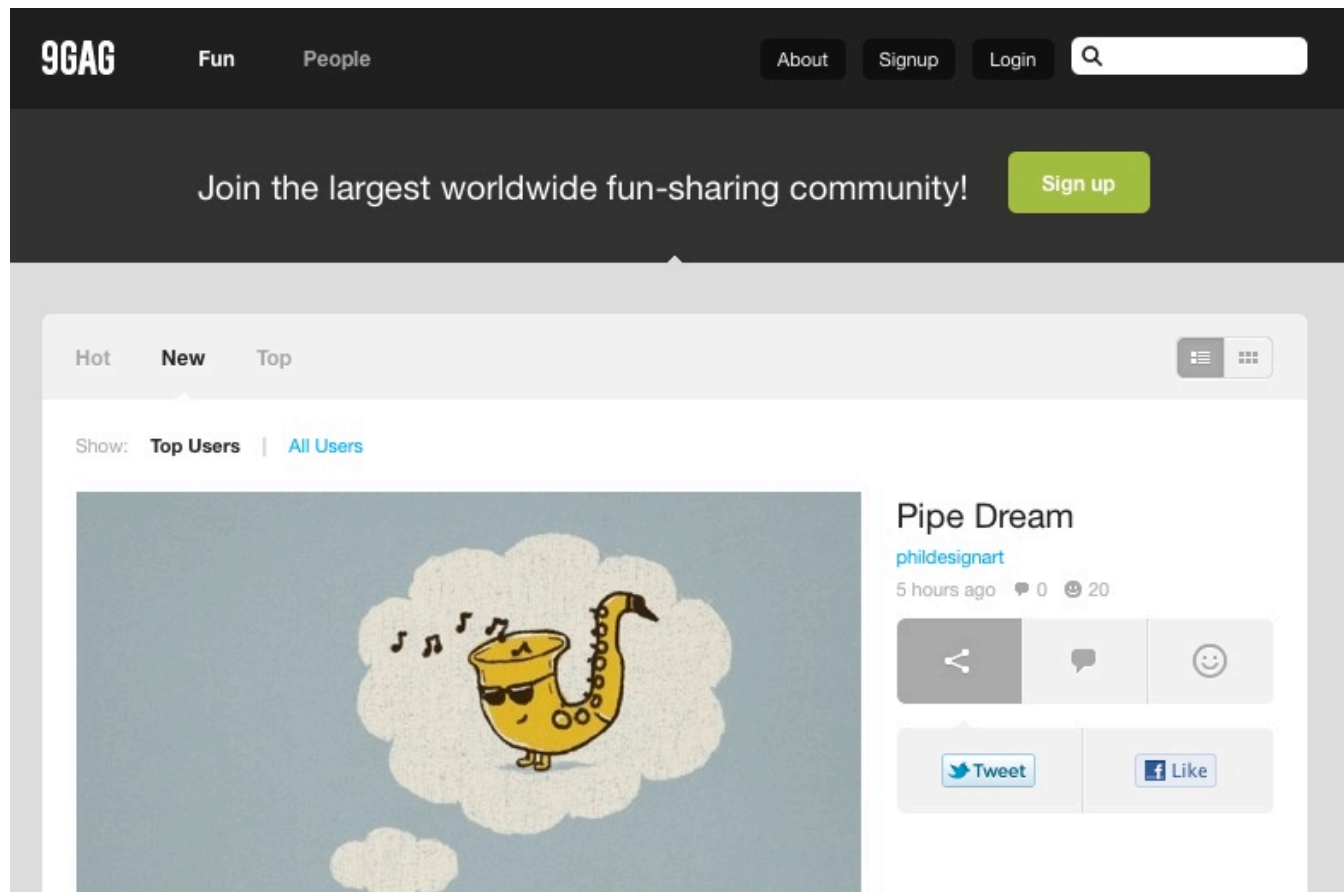
9gag.com

Feta

Yet another Flash-based website that lets you use the left and right arrow keys to browse items of a section, the down key to select and the up key to go back.

www.feta.pl

NY Times: Times Skimmer

The New York Times' quick overview page has very advanced keypress navigation. Users can use the arrows to navigate sections, zoom in using "Shift," return to the top with "t," refresh the current section with "r" and select article using "a" and the arrows. Learning the keys is a bit time-consuming, but once you've got them, navigating the page is much easier.

www.nytimes.com

Pictory

PictoryMag, a magazine dedicated to photo stories, also has "j" and "k" navigation to browse images.

www.pictorymag.com

CrushLovely

CrushLovely, a single-page portfolio, lets you use the arrow keys to navigate sections of the page.
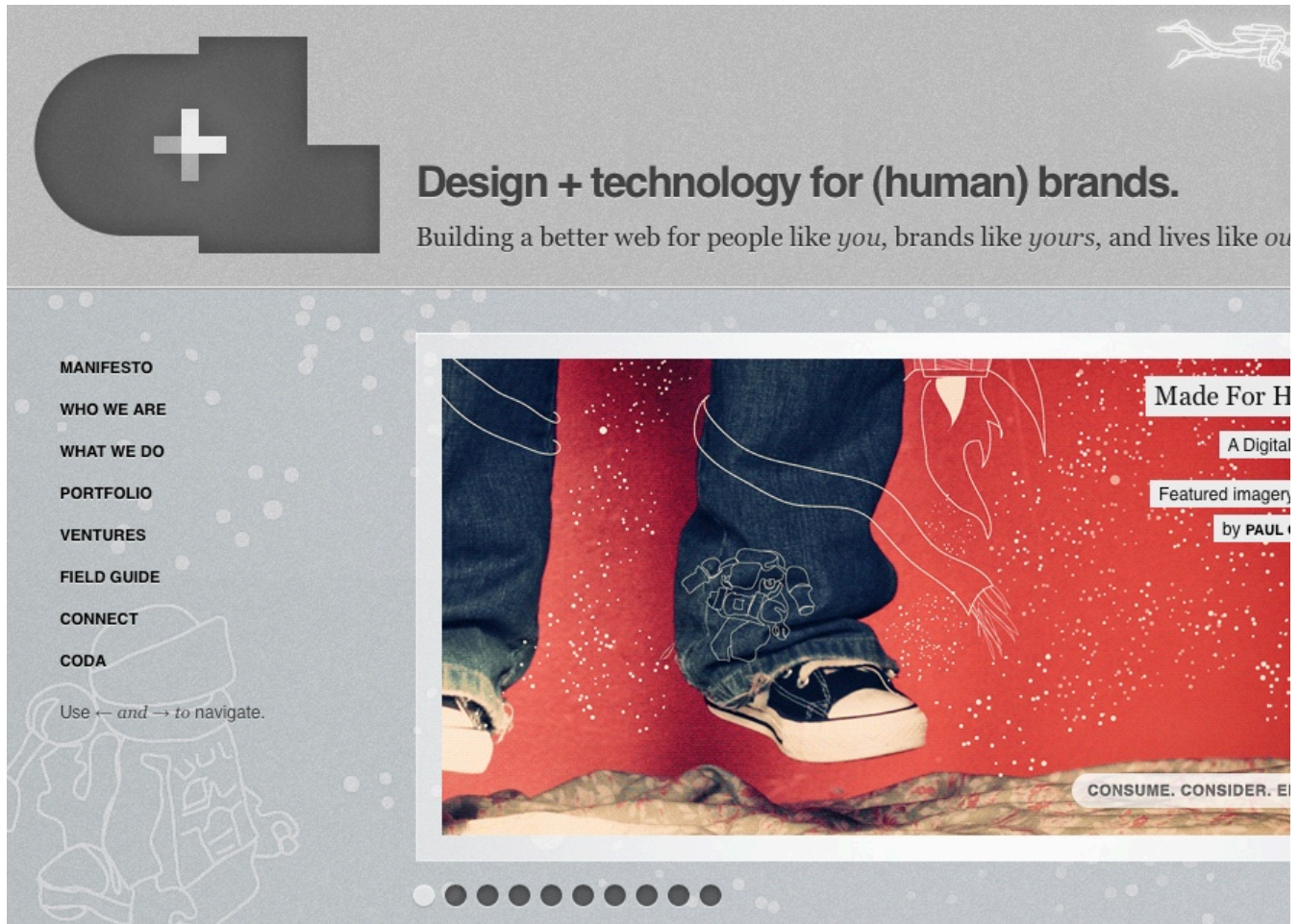
crushlovely.com

Picnic Extraterrestre

Aside from being one of the most unusual designs we've seen so far, Iván Ferreiro's Picnic Extraterrestre has quite advanced keypress navigation. The design imitates Teletext and does a pretty good job. All navigation items can be loaded using the digits shortcuts. Now that's fun!

www.ivanferreiro.es

Coding Techniques and Tutorials

Note that when implementing keypress navigation in your design, make sure that the shortcuts you define do not conflict with common browser shortcuts, OS shortcuts, screen-reader shortcuts or user-defined shortcuts. This may sound simpler than it is. As usual, extensive testing (with savvy and novice users) before implementation will help you find issues with your shortcuts. It's safe to assume that the arrow keys, the "j" and "k" combination and the "Escape" key are safe. On the other hand, using the "Control," "Alt" and "Shift" keys is not recommended.

Also, regard keypress navigation as an additional (and therefore optional) feature that will not be available to users who have disabled JavaScript in their browsers. Therefore, it is highly recommended that you offer keyboard navigation as a secondary, not primary, layer of navigation. Below, you'll find some helpful techniques, tutorials and references for implementing keypress navigation in your designs.

- Adding Keyboard Navigation with jQuery
  This screencast describes how to implement keyboard navigation to move a slider backwards and forwards. The demo and code are available as well.

- How to Create Keypress Navigation Using jQuery
  This tutorial describes how to implement keypress navigation to browse sections of the website.

- Advanced Keypress Navigation with jQuery
  You could use your mouse to select links, but you can also use the arrow keys (i.e. up and down) to navigate the list. This script is a bit advanced because of the extra functionality when the user combines the mouse hover and key presses.

- [Using Keyboard Shortcuts in JavaScript](#)
  In this article, you'll learn how to use JavaScript keyboard shortcuts, with and without the JQuery framework.

- [How to Build a Site With Keyboard Navigation: PSD to HTML](#)
  This article looks at how to add keyboard navigation to a website using a few simple lines of JavaScript. First, you'll create a simple theme in Photoshop and then transform it into a working website that offers keyboard functions to jump pages.

Print Design Influence

While designing for delight is primarily about impressing visitors with unexpected and pleasing touches to a design, modern Web designers often go one step further and experiment with the underlying details of their work, producing more creative and unique layouts. In fact, one doesn't have to be an expert to see the growing influence of traditional print design techniques on the Web. They are often manifested in so-called "art-directed" blog posts, whereby every blog post has a unique and carefully crafted design.

The layouts of these websites often resemble those of print magazines or posters, with striking headlines, multi-column text, highlighted quotations, indented text, supporting imagery, sidenotes and footnotes. The designs usually adhere to grids and have strong, vivid typography.

GRID-BASED WEB DESIGN, SIMPLIFIED

A holistic approach for the typical workflow

by CHRIS BRAUCKMULLER for DESIGN INFORMER

A grid at its barest is nothing more than a series of intersecting horizontal and vertical lines spaced at regular intervals, but its innate propensity for creating order out of chaos makes it one of the most powerful tools at a designer's disposal. If you want to reap their benefits of grids on your next project but are unsure of the specifics, this article is for you.
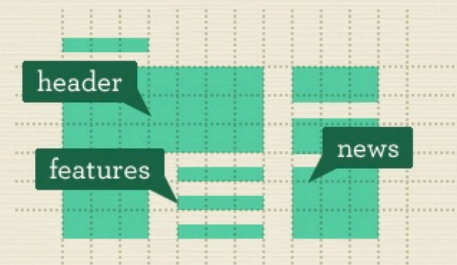
Introduction

If you're even vaguely acquainted with the fundamentals of graphic design, you've probably worked on some kind of a grid or at the very least seen examples of grid-based layouts. Grids are an established design tool, and a wealth of knowledge exists in the literature discussing the theory of grids and extolling their benefits. I will make no attempt to summarize them here (if you want a good primer on grid theory, have a look at this piece by Mark Boulton).
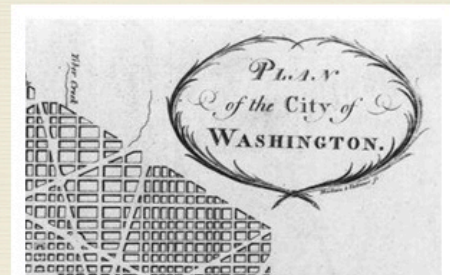
Design Informer: Grid-Based Web Design, Simplified has a simple clean two-column layout that clearly separates text from illustrations. Notice the capital letters in the author's name under the header, also visible in the quote design on the page. The content here dictates the layout.

In most cases, art-directed designs are fueled purely by the ambition and determination of their creators. Such designs are predominantly found on freelance websites (being the fruit of personal projects) and rarely found in corporate settings. The main obstacle to wider adoption of these techniques is that the creation of such designs (or rather their implementation with (X)HTML and CSS) is time-consuming. Art-directed layouts are quite difficult to code and maintain, and they often require inline CSS styling, or else designers would end up with dozens of un-semantic classes in their style sheets. Also, integrating advertisements on these pages is difficult because they put constraints on the designer's layout. So, at the moment, these designs are more appropriate for less frequently updated websites because of the overhead.

If you decide to experiment with art-directed design, be aware that the layout of an article should be secondary and always support the content itself, not dominate it. The problem is that once you start designing a blog post, it's easy to overdesign page elements just because you can, not because the content dictates it. In fact, the design community is having an ongoing debate on whether art-directed designs are merely "over-Photoshopped articles," designed purely for the sake of design.

Good design is about effective communication, not decoration at the expense of legibility. As Francisco Inchauste puts it, "I think it's a 'pick two' sort of scenario. The choices are: great content, great art direction and regular schedule. If you try to hit all three, one of those will begin to fall short." Bottom line: Web designs that are heavily influenced by print design are beautiful, but only when the techniques support your article.

A Lesson on How to Be a Villain

A colorful and nicely illustrated article in a unique layout. Notice something unusual? The design has a CSS-layout with tabular data for the actual info-graphic bits. Sometimes that's necessary for art-directed designs.
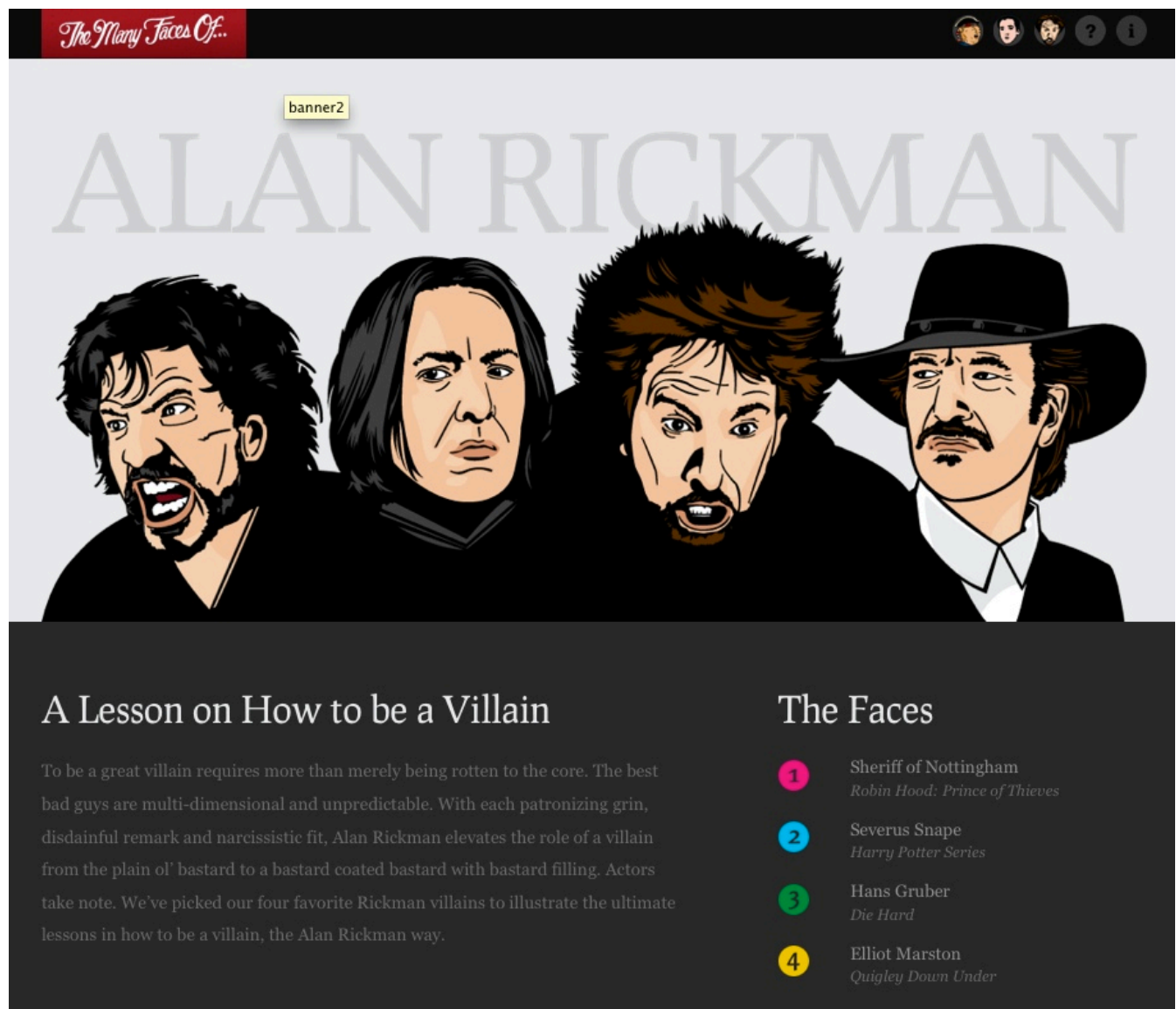
themanyfacesof.com

Evan Dinsmore: 21

A poster design for the Web. This blog post is simple, and it replaces tired plain text with vivid images. But that can be a disadvantage, too: a text-based version would be more user-friendly here.
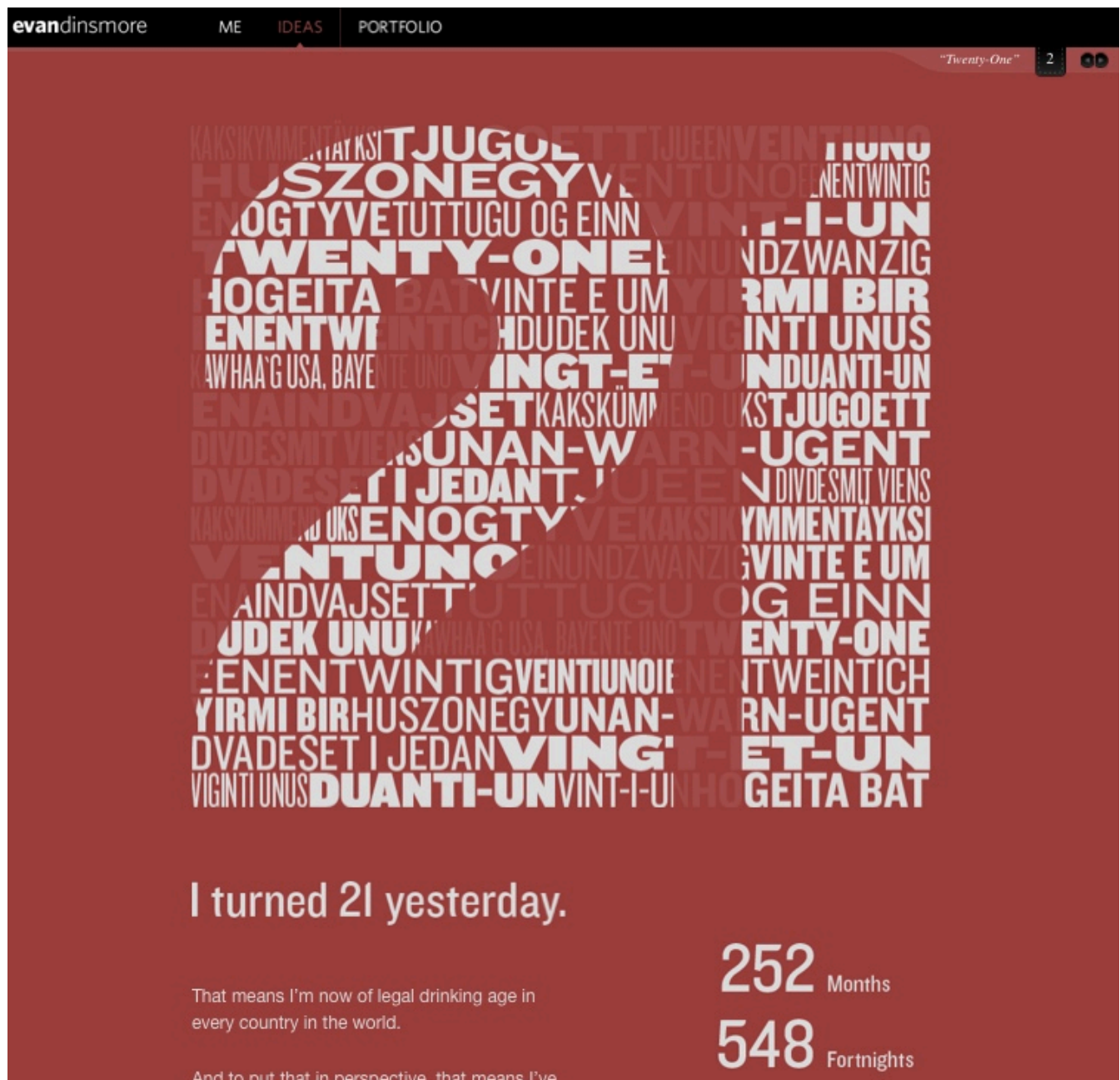
evandinsmore.ca

A Way Back: Revised Font Stack

A very long, detailed and elaborate design. In art-directed designs, including this one, large images are often used to push the boundaries of the layout. Such images are often 800 to 1000 pixels wide, filling the width of the entire layout.

www.awayback.com

Chris Coyier: The Safari Challenge

Here is a more subtle design, with big margins, multiple columns of text, footnotes and indented headings. From an aesthetic point of view, it could be a page from a book.

chriscoyier.net

Kyle Fielder: Keeping Curious

A classic. Do you remember those old magazines that used big quotes and visuals to create text flow? Notice how well this headline and colophon are positioned in the question mark. A nice, simple, original design.

kylefiedler.com

Sleepover: A Critical Analysis of my Shoes

A simple grid-based design with justified text, serif typography and nice shoe illustrations. Unfortunately, justified text still doesn't look very good on the Web.

www.sleepoversf.com

Yaron Schoen: Too Many Buttons

Sometimes art-directed blog posts require something slightly more: like a background image and background color, as well as a bit of CSS styling. This examples demonstrates exactly that.

yaronschoen.com

The Bold Italic: Keep Off the Grass

Another remarkable example of multi-column-layouts...

thebolditalic.com

The Bold Italic: Cinderella Story

... and another one. Print-design inspiration at its best.

thebolditalic.com

Travis Neilson: Default Switch

A calm, simple, clean design with custom headings.

travisneilson.com

Horizontalism

Over the last year, we've observed a slow transformation in the orientation of text-heavy Web designs. Not only are designs gaining depth and realism, but navigation is changing as well. Some designers are augmenting traditional vertical scrolling with sliding navigation, which usually scrolls in both a vertical and horizontal direction, or even pure horizontal scrolling. This is called "horizontalism."

Websites with horizontal scroll bars have been more difficult to navigate because the mouse was designed for vertical scrolling. But the emergence of multi-touch devices forces us to rethink the usability concerns of such designs. After all, whether the user browses vertically or horizontally on such a device doesn't really make a difference. And some plug-ins (like Scrollable and jScrollHorizontalPane) simplify the action by enabling users to navigate horizontally by using the standard vertical scroll wheel on the mouse, thus shrinking the learning curve.

Horizontal scroll bars have been out there for a decade, but today it feels that they are gaining a new context. The move to horizontal scroll bars is probably an attempt among some designers to provide a more distinct user experience. Such designs are usually carefully crafted and found primarily on portfolio websites and elaborate e-commmerce websites. Whether horizontalism will expand to more types of websites remains to be seen in the months to come.

Thinking for a Living

Not only does this article discuss the advantages and disadvantages of horizontalism with regard to readability, but it also has a nice horizontal layout itself, with multiple text columns. While the orientation is unusual at the first sight, reading the post is quite pleasing and comfortable.

www.thinkingforaliving.org

Jung v. Matt

This website has a horizontal timeline for navigation. Notice that there is no horizontal scroll bar; visitors use the vertical scroll wheel to navigate horizontally.
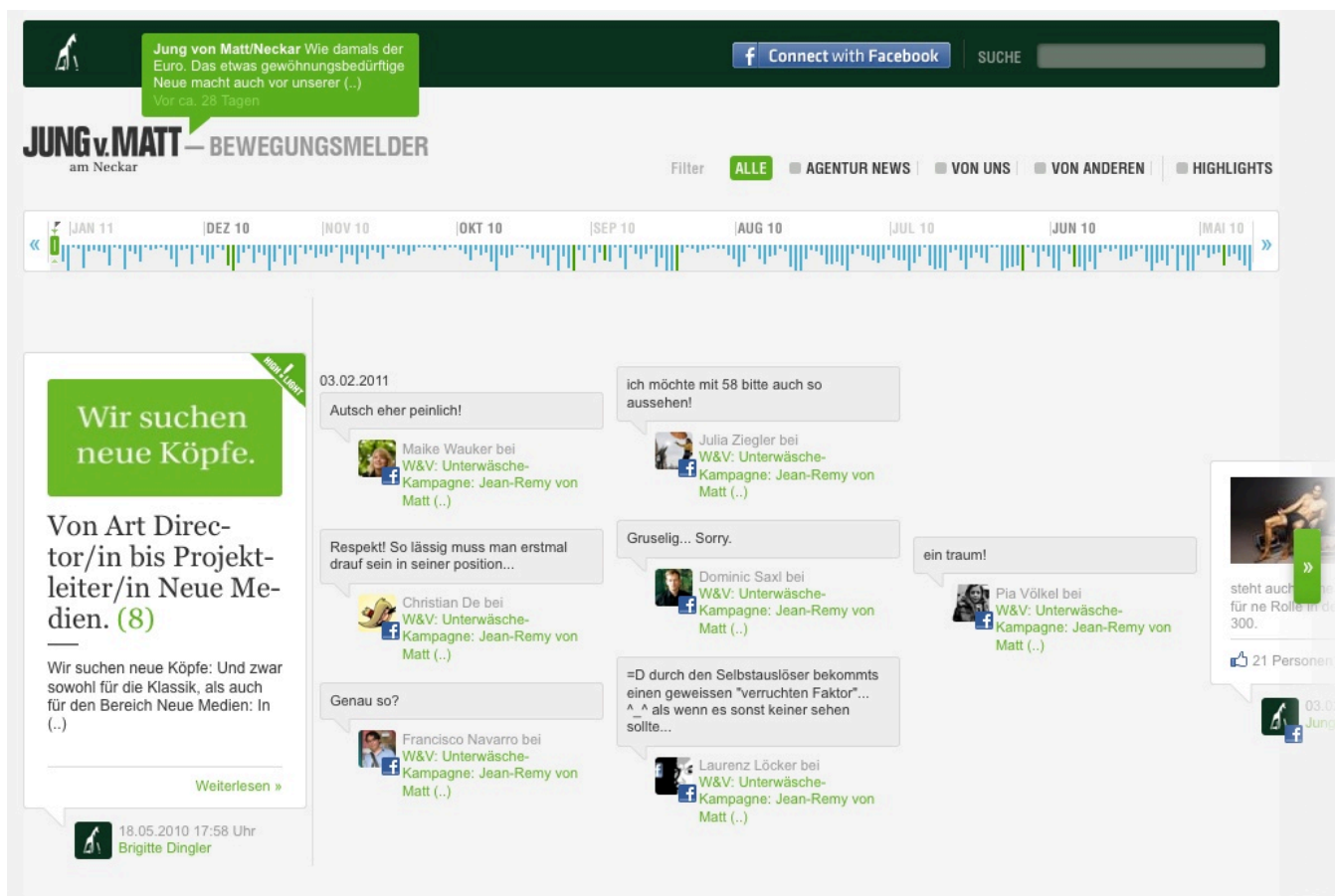
www.jvm-neckar.de

Your Auxillary

One of many so-called "single-page layouts." The full content of these websites is on a single page, which is navigated using either the keyboard, the mouse or a menu (this website uses the third option). Here we have a good (and common) combination of vertical and horizontal navigation (showing the jQuery ScrollTo plug-in in action).
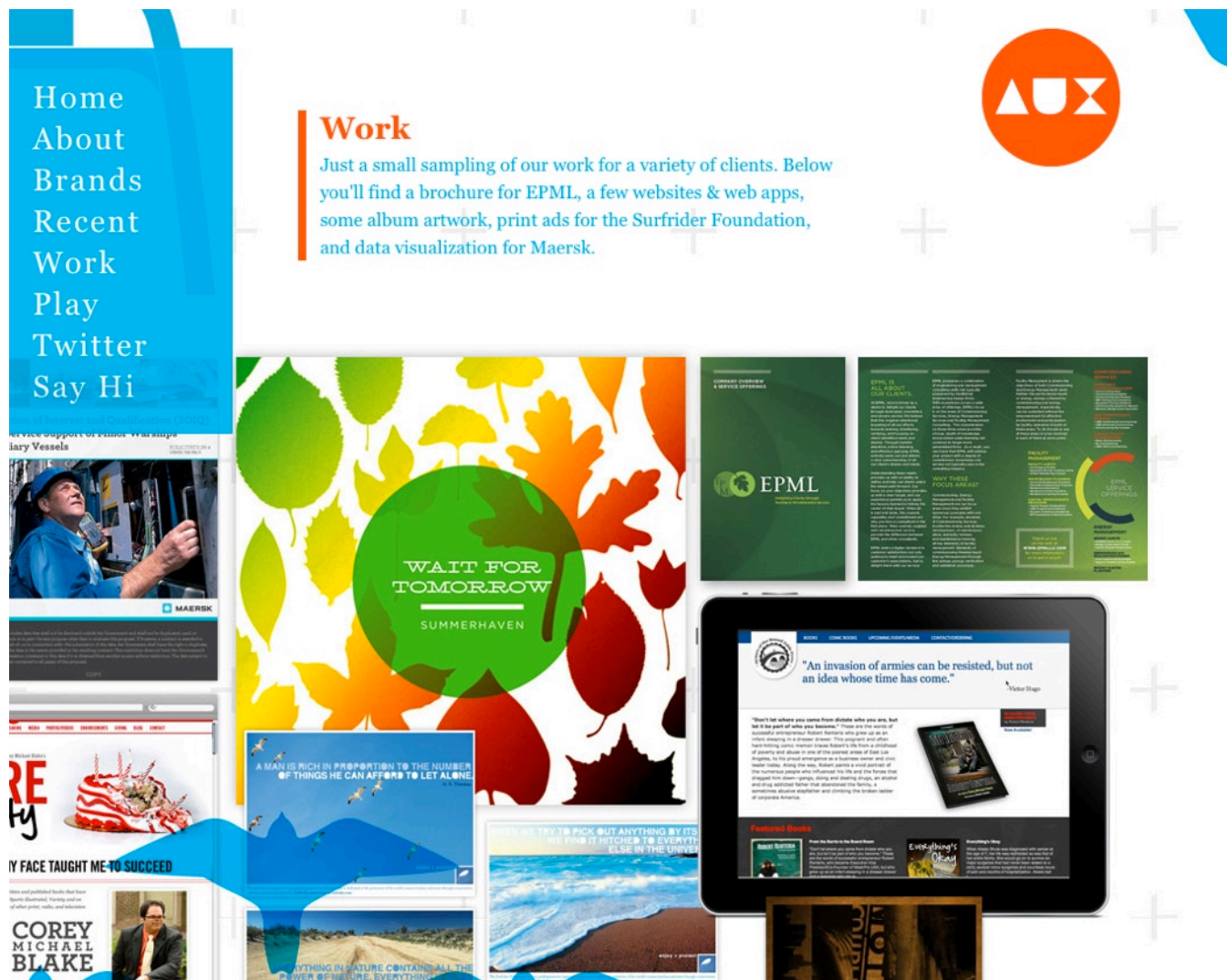
www.yourauxiliary.com

One Twenty Six

This portfolio has a different kind of horizontal navigation. Apart from "Previous" and "Next" buttons, the user also gets an overview of selected content in a drop-down menu. Once they select an option, the page scrolls horizontally. Horizontal navigation with the mouse wheel would probably improve this design's usability.
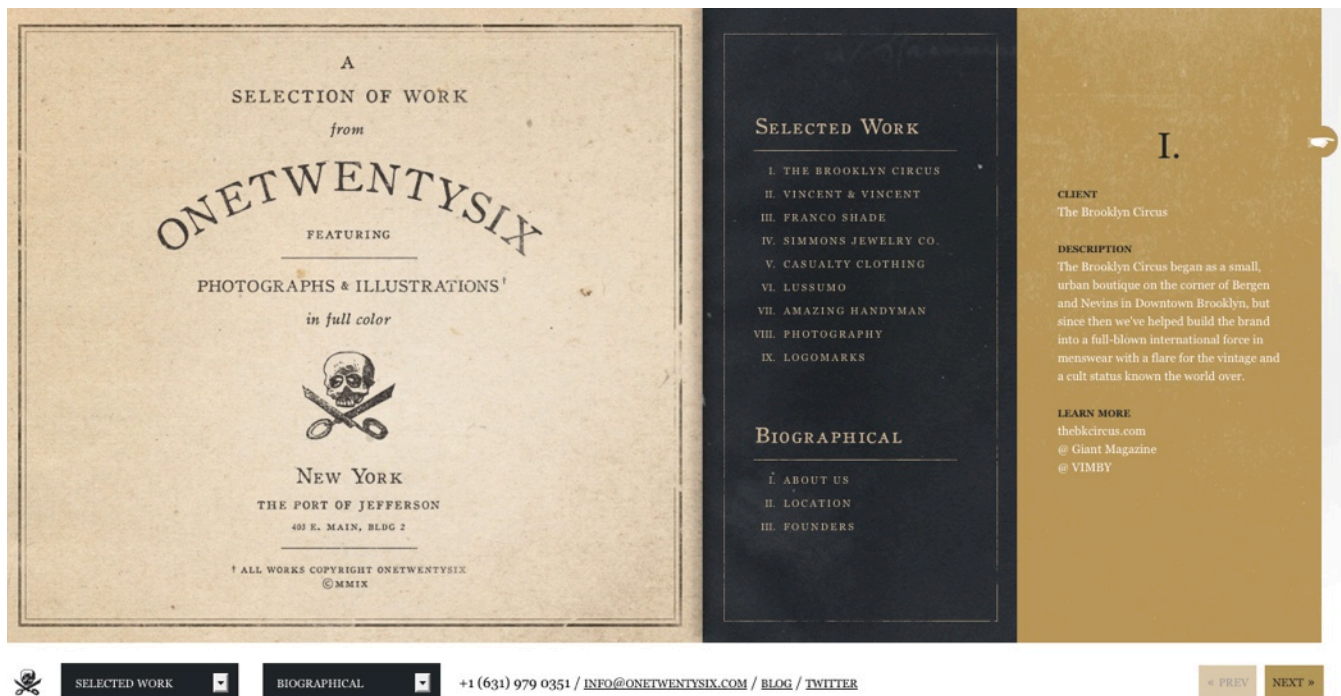
www.onetwentysix.com

C. L. Holloway

Candice Holloway's portfolio has a nice take of horizontal layout. Her artwork is placed on a "wall"; horizontal navigation is used as a metaphor for strolling an art gallery. Also interesting: scrolling is triggered when your mouse hovers over the horizontal arrows; no clicking necessary.
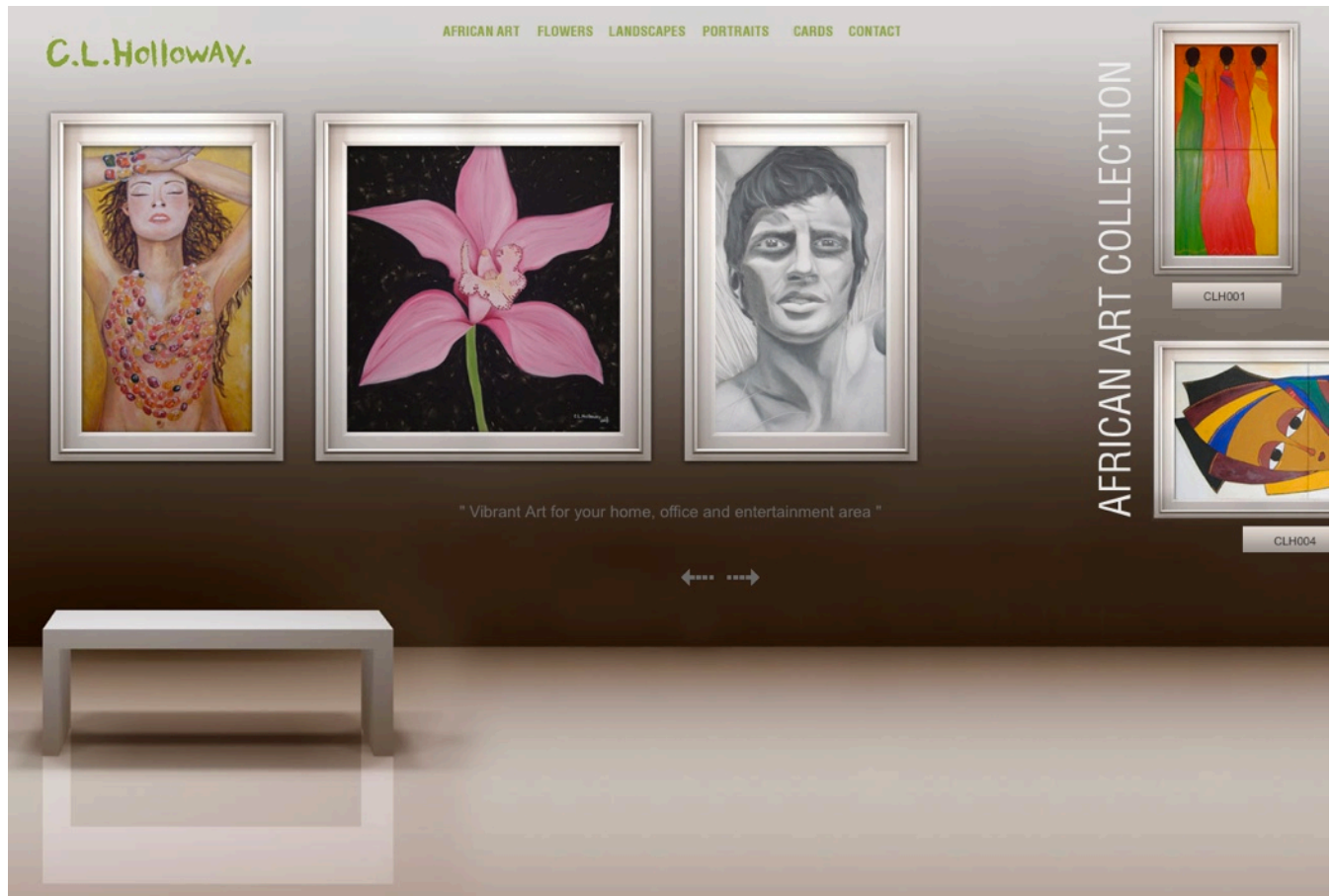
www.clholloway.co.za

Yamaha Ginza

You'll find that designers experiment with perspective. Sometimes the orientation is diagonal…
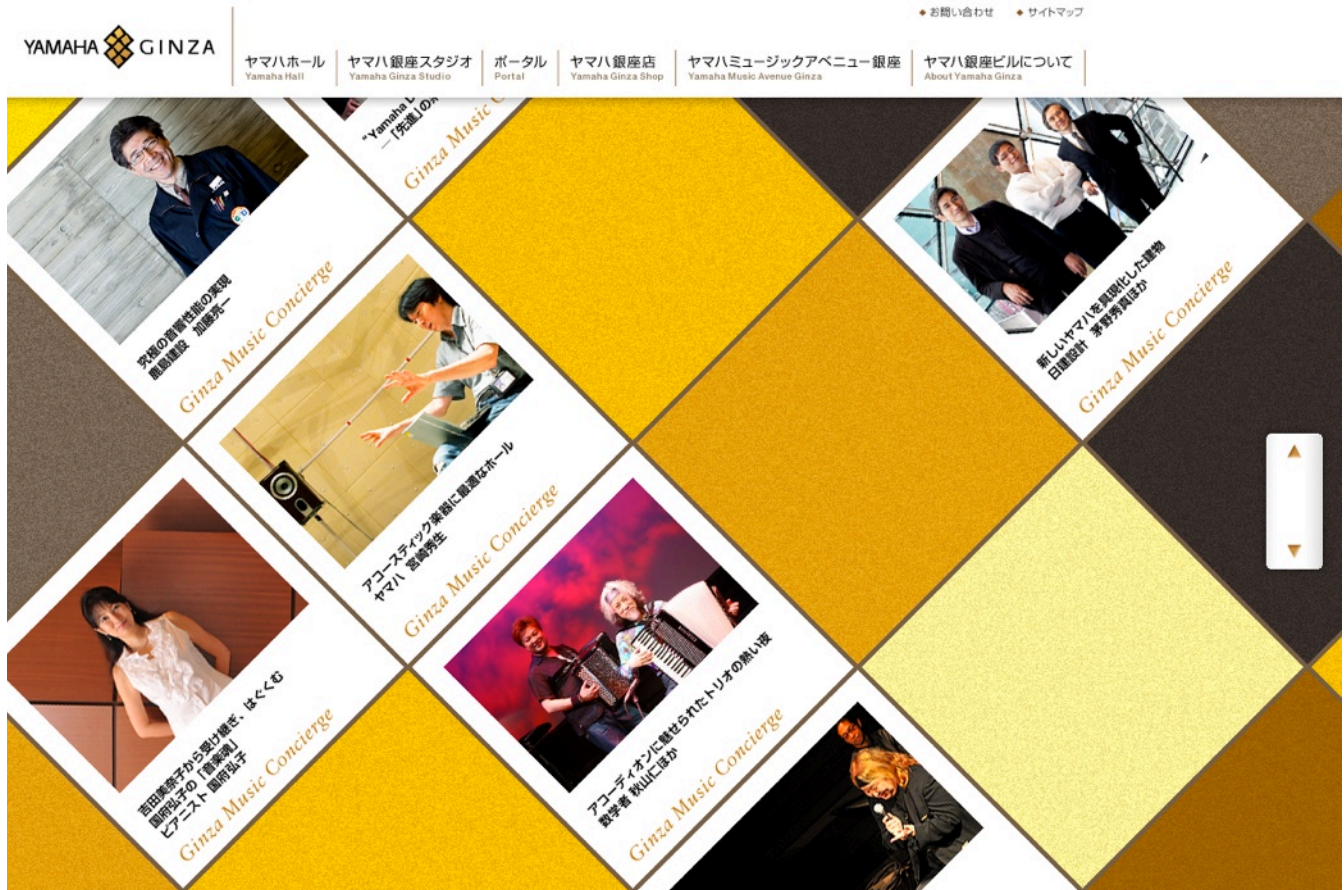
www.yamaha.co.jp

Edpeixoto

... and sometimes the layout just hangs in the air...
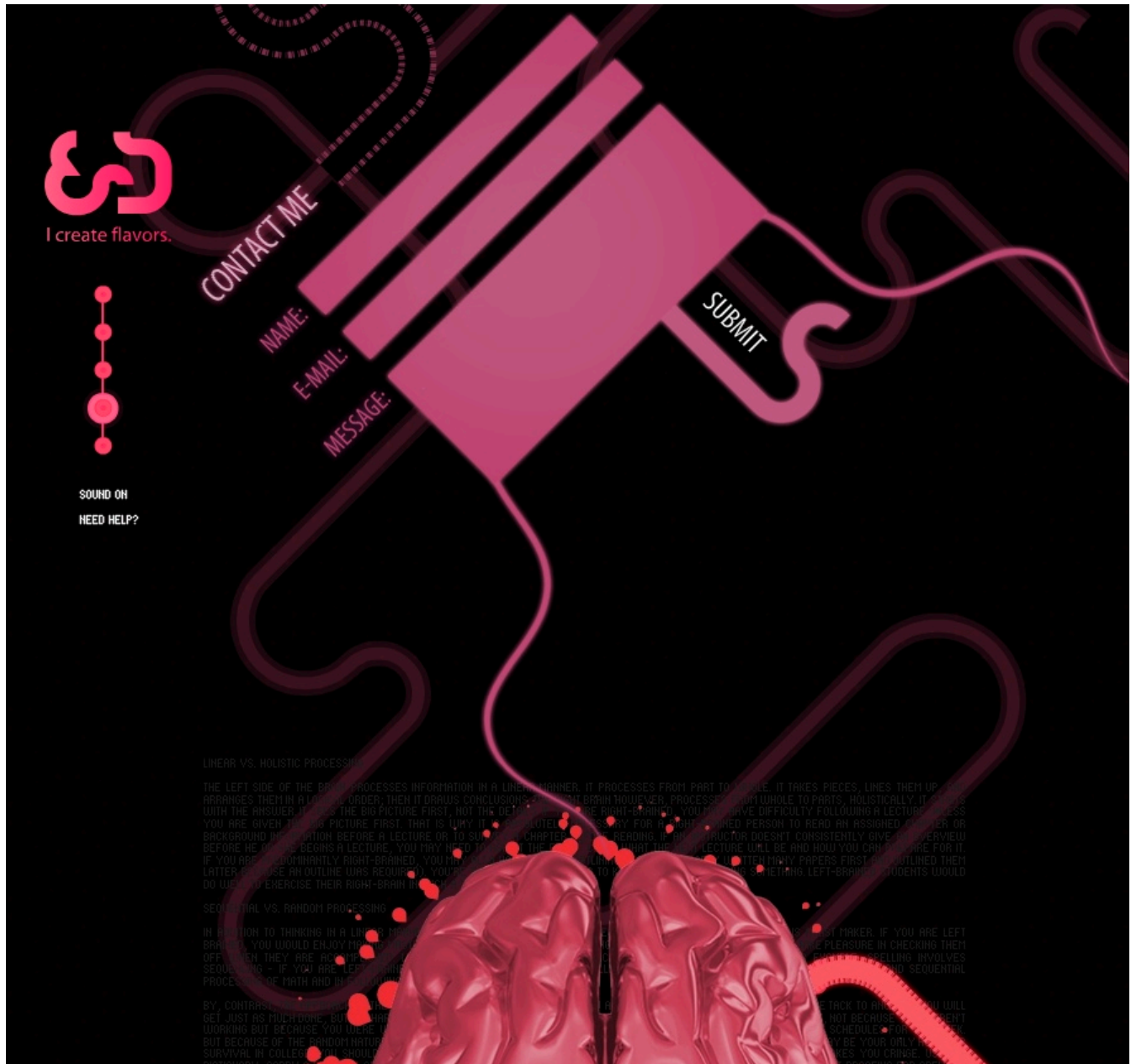
www.edpeixoto.com

5. Rich, Strong Typography

Typography has played a major role in Web design for years now. Bold, strong, heavy headlines can effectively convey the purpose of an e-commerce website or portfolio, while more subtle headings help structure content and improve legibility. Obviously, the big change we're seeing today is richer, more versatile typography, partly made possible by the `@font-face` attribute and the emergence of font-embedding services such as TypeKit. Rich typographic elements can now be selected and copied from the browser, which wasn't that easy a couple of years ago.

The future is big, bold and typographic. Rich font families will be used not only for headlines but for body copy as well, bringing typographic practices from print over to the Web. Also, designers will experiment more with rich, sophisticated serif fonts and bold, imposing slab fonts, supported by subtle imagery. Web designers are also adding more depth to typography with the `text-shadow` attribute in CSS3. Naturally, such subtleties are closely tied to the choice of layout. These typographic designs are often grid-based and borrow techniques from print design, such as sidenotes and footnotes.

We've further noticed that designers are extending their font stacks, adding increasingly more fall-back fonts in case a specified font is not available. That's fine, as long as the aspect ratios (or weights) of the fonts are not too different; some screen fonts will appear wider or taller than other fonts and hence have a larger aspect ratio, which means that some users would see your pages at a much smaller font size than others would.

Kilian Muster

Kilian Muster uses quite an extended serif font stack for his design: `font-family: Palatino, "Palatino Linotype", "Book Antiqua", Constantia, Times, "Times New Roman", serif;`. The posts in Kilian's blog also have sidenotes.

kilianmuster.com

extrapolish

Notice that the text on this website of a Polish Web design agency is set mostly in capitals: the navigation menu, introductory text and even contact address are in full capitals. Yet the design is calm, clean and polished.

www.fajnechlopaki.com

DNA to Darwin

This website has only serif fonts throughout its design: `font-family:` `"skolar-1","skolar-2", Georgia, Times, serif;`. Notice that the text is split into columns; we didn't see this last year.
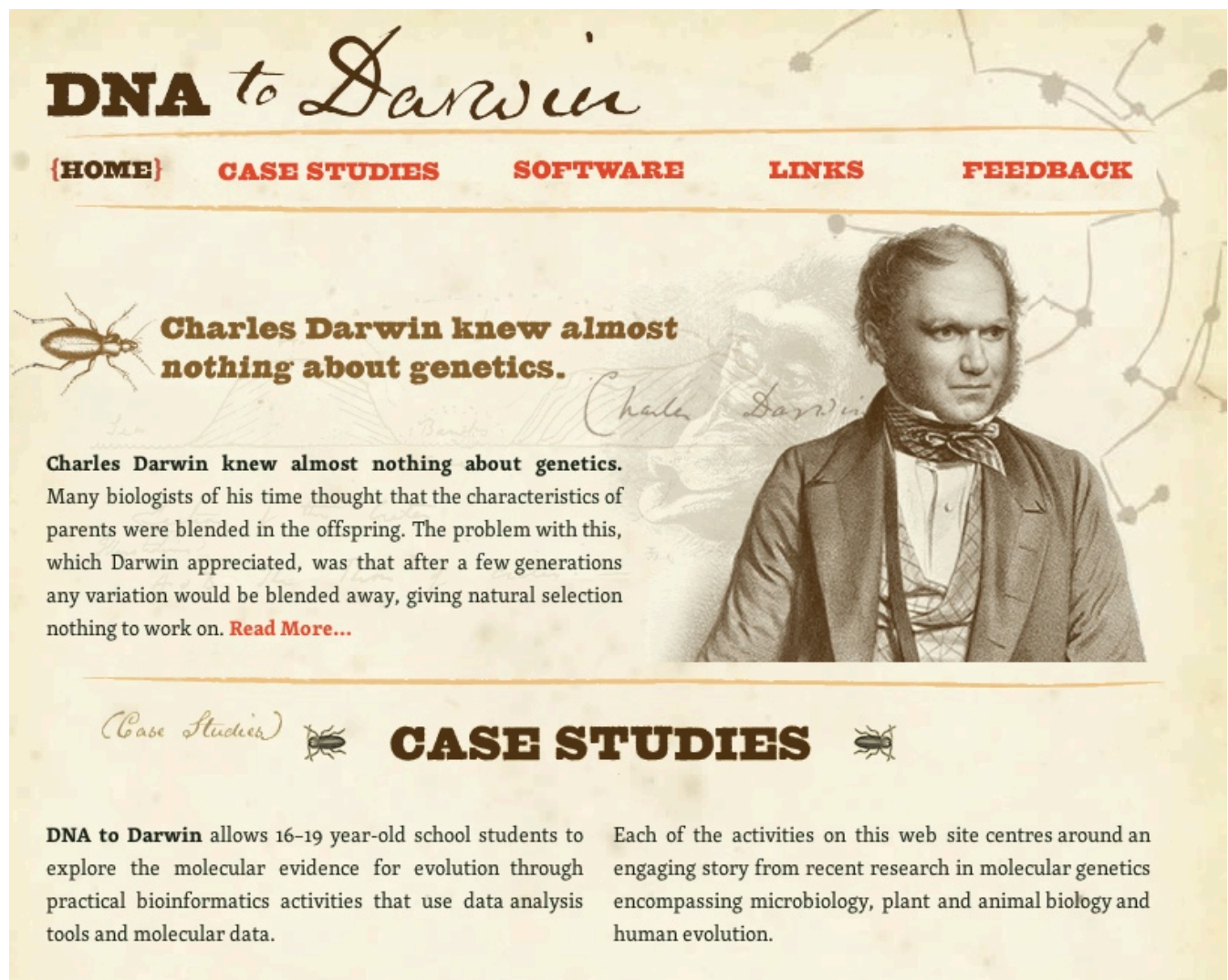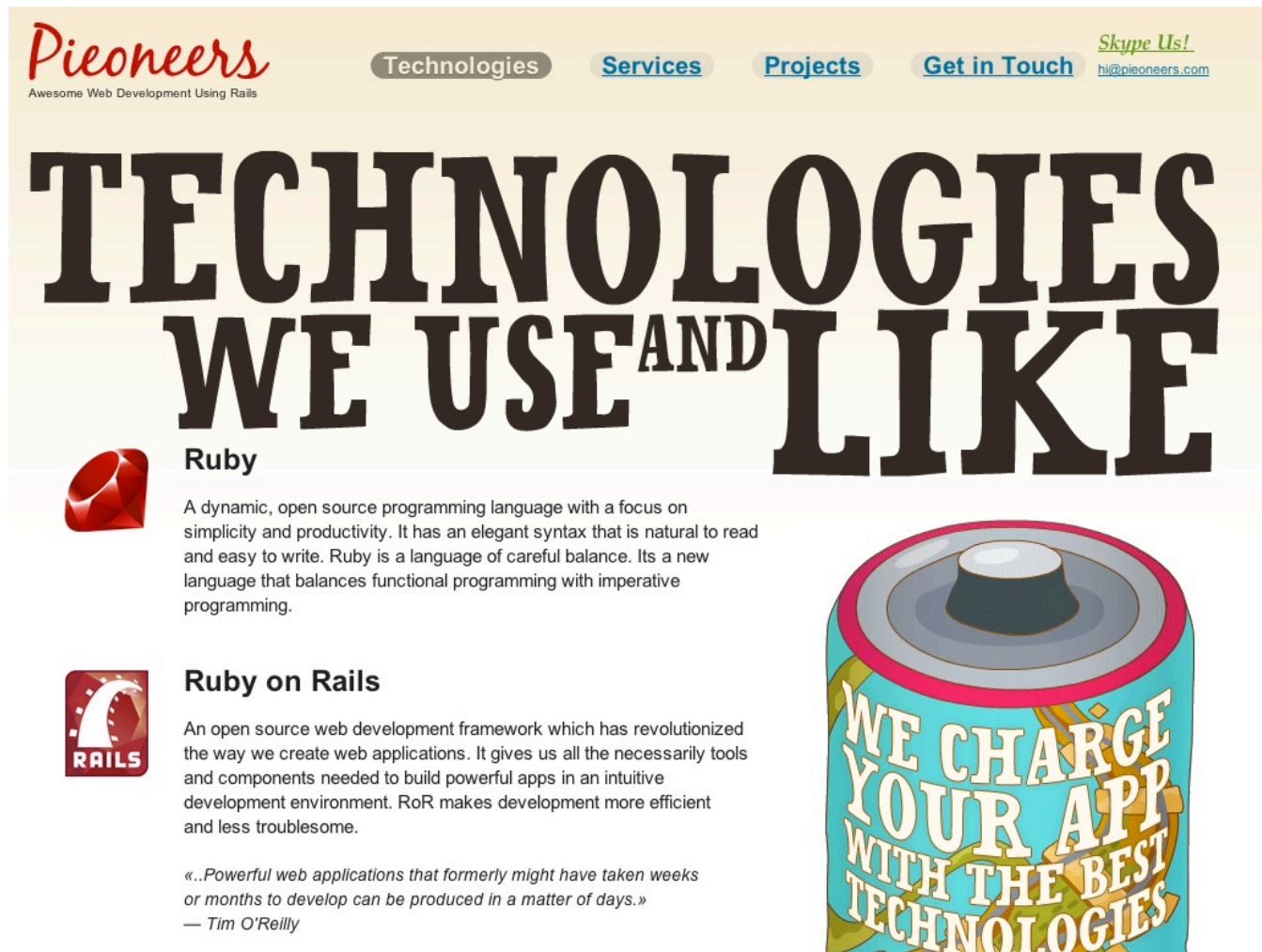
www.dnadarwin.org

Pioneers

This website combines vivid imagery and playful typography. The design looks more like a brochure or poster than a "classic" Web page.

www.pieoneers.com

Colly

Simon Collison's subtle attention to the tiniest details make the typography literally stand out. No bold, screaming typography here; just legible, aesthetically pleasing design.
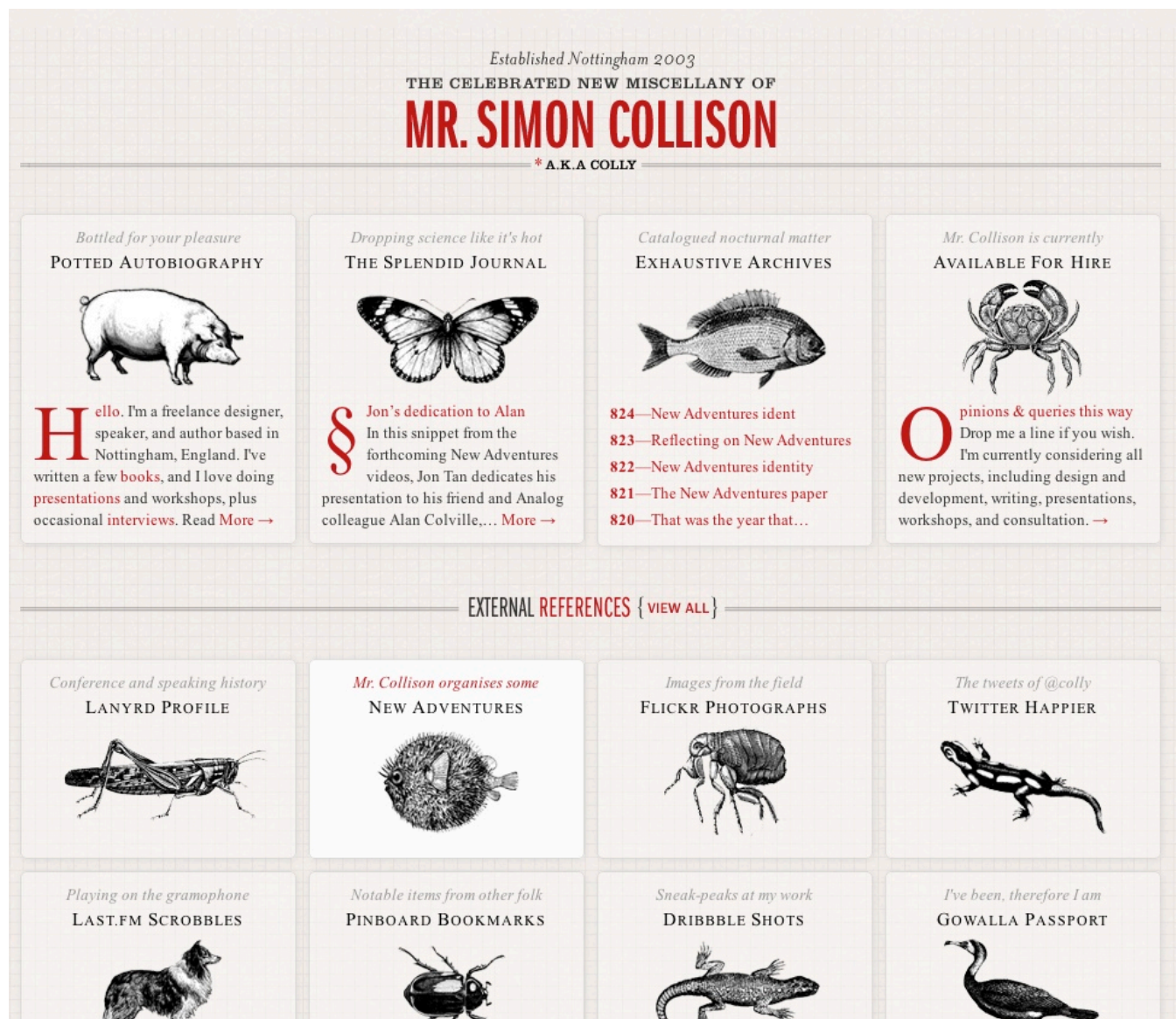
colly.com

The Saint John's Bible

This website shows serif fonts at their best. The fonts complement the theme and fit the layout perfectly. Notice how well a beautiful visual design and classic typography can work together.
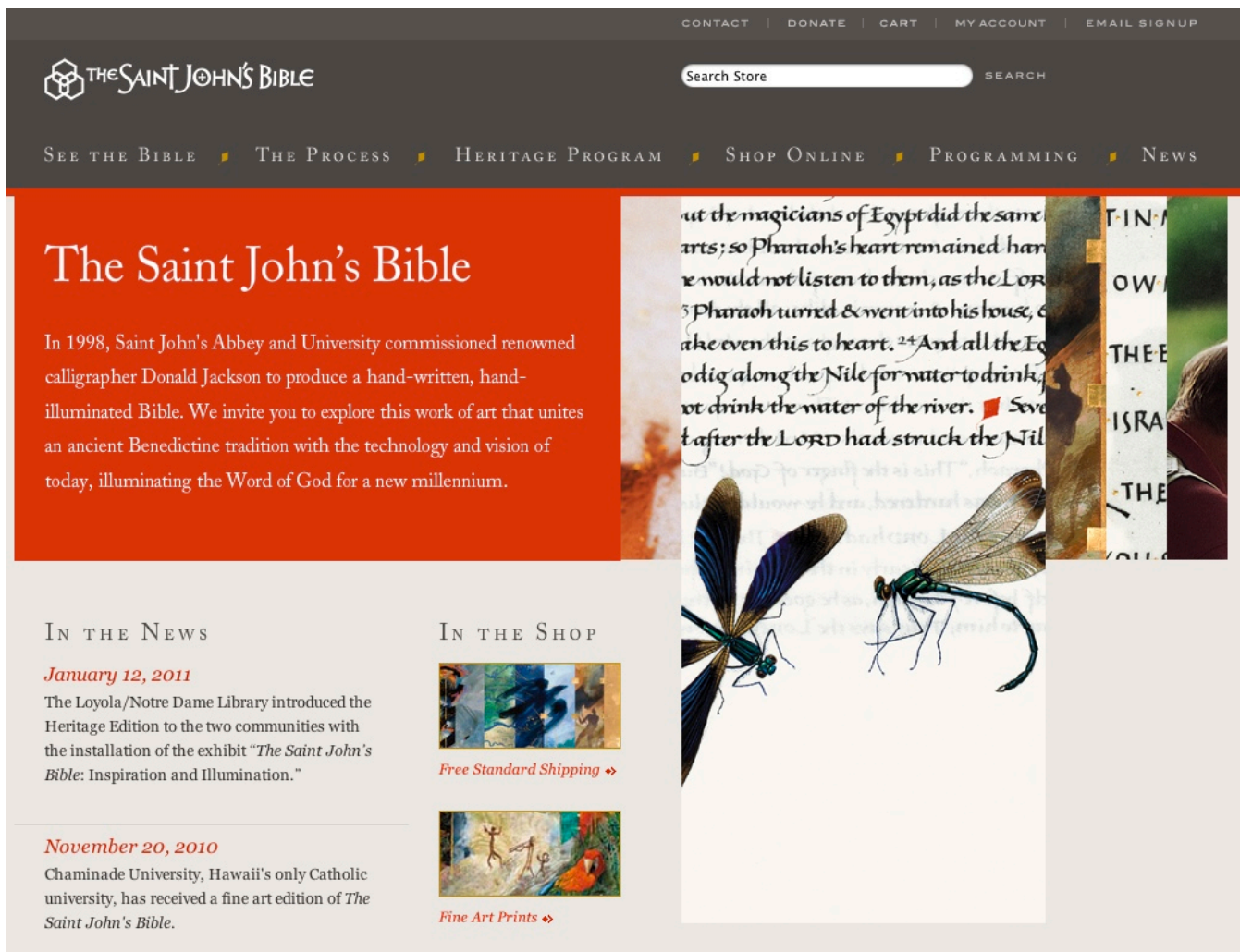
www.saintjohnsbible.org

Brewhouse

A nice combination of type and visuals make this page remarkable. But it's not clear why the page has three different typefaces for the headings; two would be enough.

terminalbrewhouse.com

Tick Talk

Can this get any bolder? Big bold typography, with capital letters spread across the whole page. When scrolling the page, notice the nice background effect. A very simple and strong design.

www.chris-armstrong.com

Conclusion

Modern Web design is better, richer and more user-friendly. We're seeing better use of visual design for the sake of aesthetics and a pleasing user experience. Traditional techniques from print design are increasingly being applied to the Web, be they layout techniques or rich versatile typography. Horizontal and even diagonal orientations bring a fresh perspective to the flat 2-D designs we've seen for years (with their text-heavy, Flash-based pages).

These developments are a sign of the upcoming era of Web design, in which designers can use new tools and techniques to their fullest potential. Web designers should look forward to the exciting and promising years to come.

Real-Life Metaphors

In the chapter before we covered the strong influence of print design techniques, keypress navigation, horizontal layouts, rich typography and aesthetically pleasing designs. Web design today is developing rapidly, breaking the limitations of conventional approaches and exploring the possibilities of upcoming technologies. Designers are not only experimenting with new techniques and design approaches, but they are improving the quality of their designs in both technical and conceptual respects.

Modern websites have great strengths, such as flexibility, cross-browser compatibility and personalization, but they are also becoming increasingly simple and intuitive. This is being done through the application of subtle

usability enhancements, drawn either from the Web itself or from offline interactive systems. The Web today is increasingly complex, while usage of the Web is becoming increasingly simple.

In this article, we'll explore some new design approaches and techniques that you may want to develop for your own projects. We'll present rather unusual or unconventional design approaches and try to understand what's actually interesting about them and how we can apply them to modern Web design. More specifically, we will discuss the following: real-life metaphors that are applied to the Web, hidden complexity, subtle interactivity, context-sensitive navigation and rapid CSS3 adaptation.

Real-Life Metaphors And Hidden Complexity

Probably one of the most striking features of modern Web applications and services is that they provide rich user experiences. User interaction in such systems is becoming very straightforward and intuitive, even "native." But this sense of intuition does not just come from the application of traditional principles of good visual design or better information architectures. We have observed more Web designs in which design elements either imitate real-life metaphors or hide the complexity of choice by letting users make more limited and simpler decisions.

Buttons are good example of the first development. In Web design today, buttons look and interact with users just like buttons in the real world. Buttons often look very realistic. They stand out, have vibrant colors and are very responsive: when the user hovers over or clicks on a button, they are often given visual feedback. If you compare these buttons with ATM buttons (in terms of shape, color and responsiveness), you will find some striking similarities.

Affordance is a key quality of native design elements. It refers to the aspect of an object or environment that allows a user to perform some action. As Donald Norman states in his book "The Design of Everyday Things," affordances provide strong clues to the operation of things. When affordances are taken advantage of, the user knows what to do just by looking: no picture, label or instruction required. The more familiar a design element looks on the page, the easier it is for the user to associate some concrete practical application or task with it.

This approach doesn't relate only to call-to-action buttons: as you will see below, it's about using appropriate metaphors to convey the meaning and purpose of certain design elements in general. Obviously, this approach reduces the complexity of user interaction. Another way to achieve this is by minimizing the appearance of elements or, to be more precise, giving them a different context or just decreasing their visual weight on the page.

More and more websites are getting rid of outdated, awkward, unusable interfaces with steep learning curves. Look out for intuitive, native metaphors. For example, sliders will sometimes work better than regular selector fields (like to show budgets and time frames); toggle on-off knobs may fit better than check boxes; and complexity can be hidden to simplify user interaction. Modern Web applications are attractive, intuitive and straightforward.

Real-Life Metaphors

Dripping in Fat

This t-shirt store uses the real-life metaphor of clothes on hangars. That's a nice idea if you don't have too many items in your store, but difficult if you have hundreds of products.
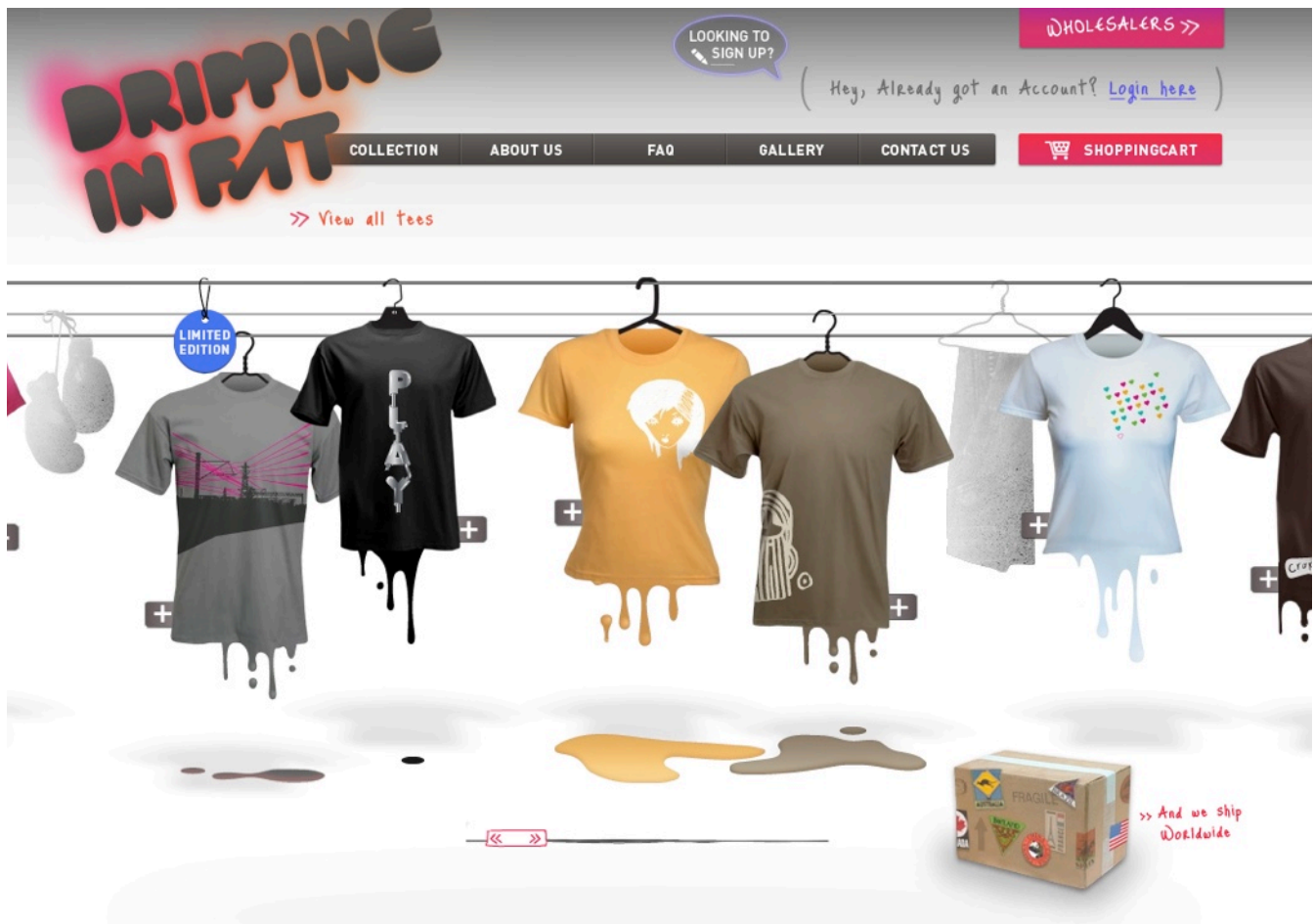
www.drippinginfat.com

XHTML Cafe

XHTML Cafe does not bore its visitors with lifeless promo content blocks: instead, it adapts a nice real-life metaphor to its profile. Attractive and memorable.
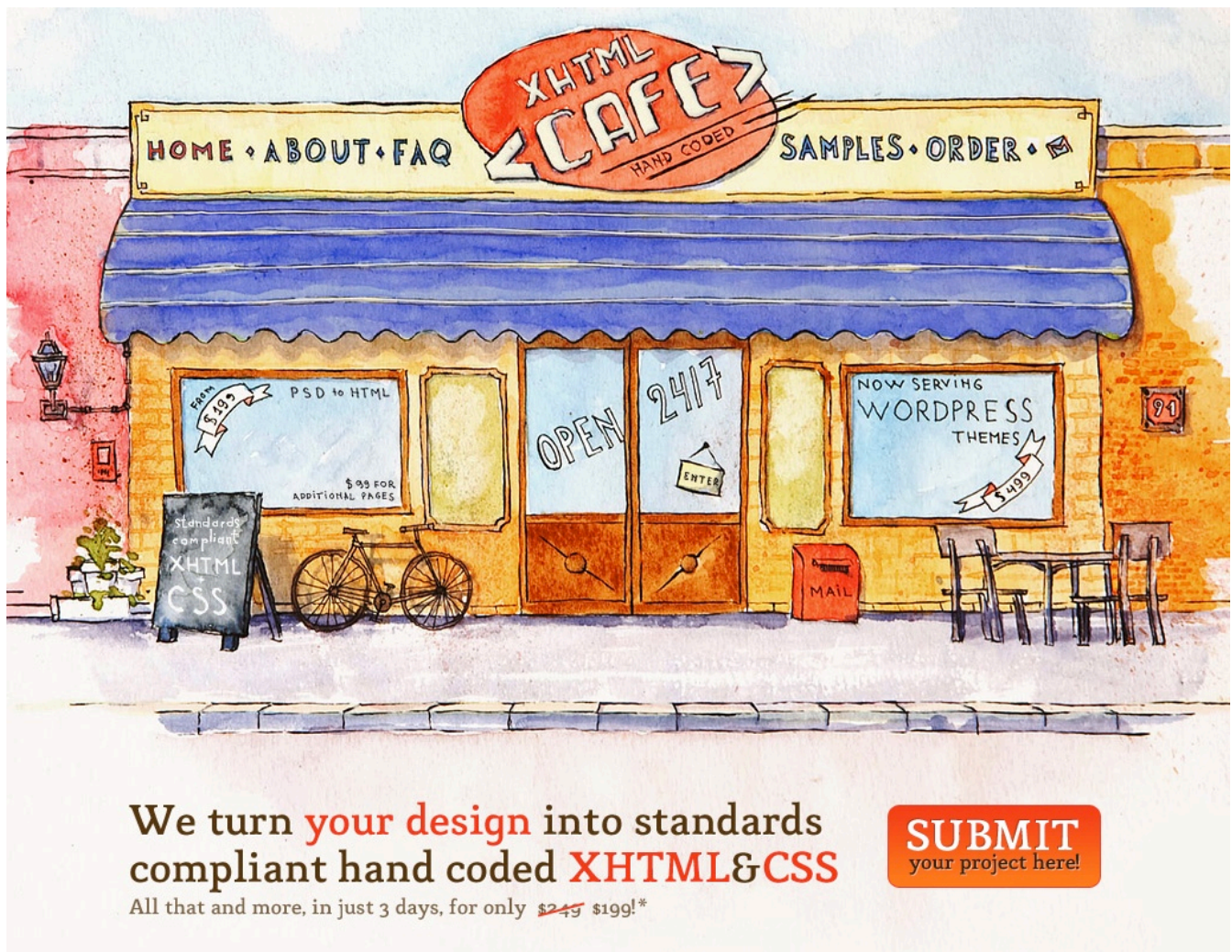
xhtmlcafe.net

CulturedCode: Status

The developers of the Culture Code list their current tasks as Flight Arrivals on their status page. Each item in the list has a description, assigned members and status. The design resembles those classic flight tables one often sees in airports. An interesting way to use a real-life metaphor for a Web design. It wouldn't work in every context, but it fits well here.

culturedcode.com

Builditwith.me

While this website may look a bit ordinary at first, what is interesting is the toggle switch in the upper-right area. Basically, it is a check box that allows users to select whether the person they are looking for should already have a good idea. But instead of using a traditional check box, the designers decided on a toggle on-off knob. While this may be a good idea in general, it's not clear if this particular implementation works. It might actually irritate users who assume it is a slider. The interaction experience is different than what you get with a slider, and rather resembles a switch.

www.builditwith.me

Oh! Media

One of the many examples of an attractive, prominent, shiny and clickable button. The small icon on the right side of the button is a very subtle metaphor for action or progress; hardly noticeable, but it works. Simple yet eye-catching.

www.ohmedia.ca

Apple

iPhone's user interface has a simple on-off switch for its preferences. This metaphor is similar to many real-life objects whose buttons and settings are turned on and off with a simple switch. Straightforward.

www.apple.com

Hidden Complexity

SlideDeck

Slideshows are by nature a nice way to hide complexity because they reorganize multiple information points in a compact interactive area. SlideDeck is a good example of a design solution that integrates both horizontal and vertical sliding navigation into one element. The layout is clean and provides a good overview of available features.
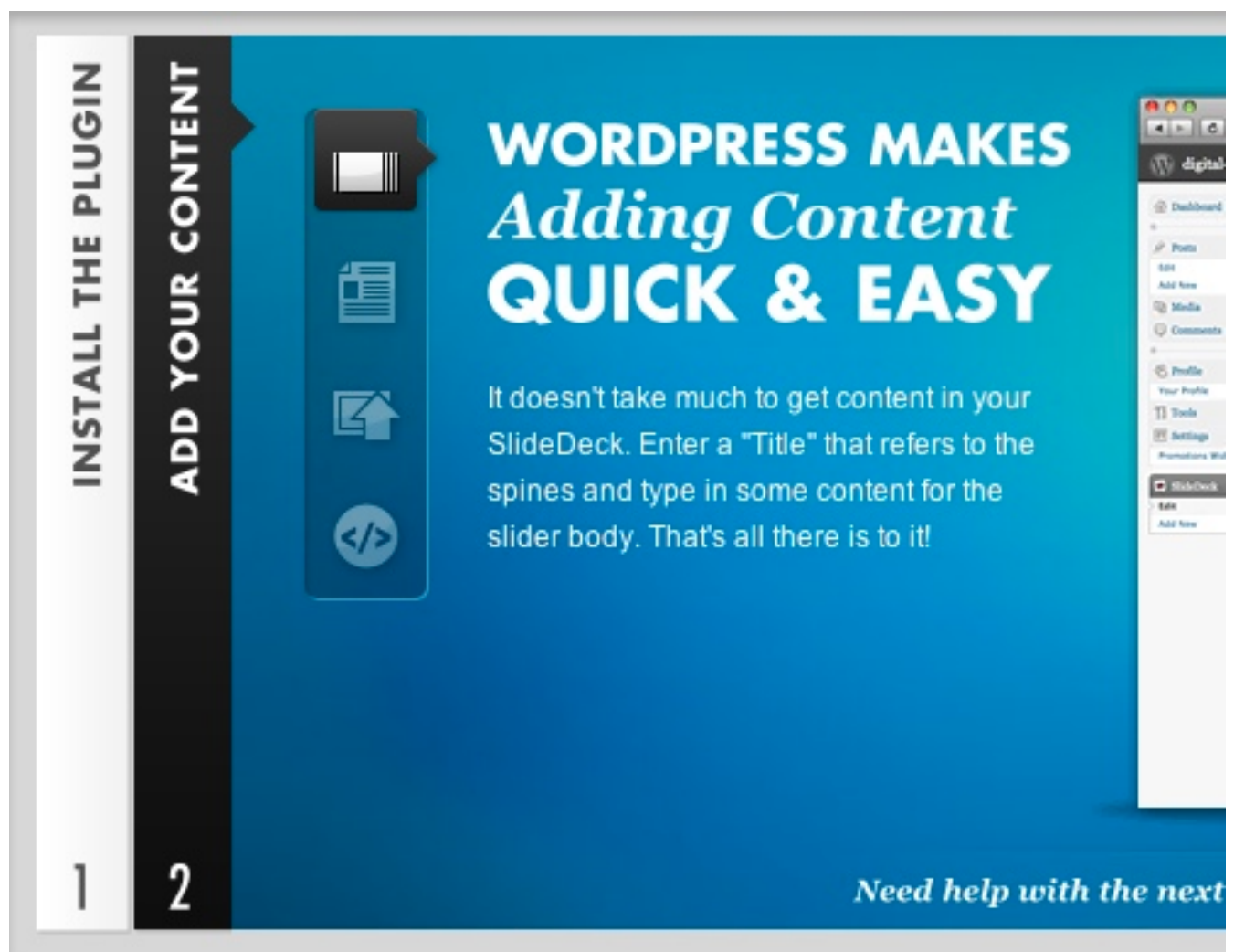
www.slidedeck.com

Nosotros

Nosotros has an interesting and unconventional navigational approach. It very cleverly minimizes complexity by reducing navigation to only meaningful icons. The blog page has six icons that serve as the navigation options for the blog. Once the user clicks on an icon to select that category, the icons for the other categories fade away. It would be interesting to see whether this approach would work on a larger scale. Calicott takes a similar approach, albeit with a different design.
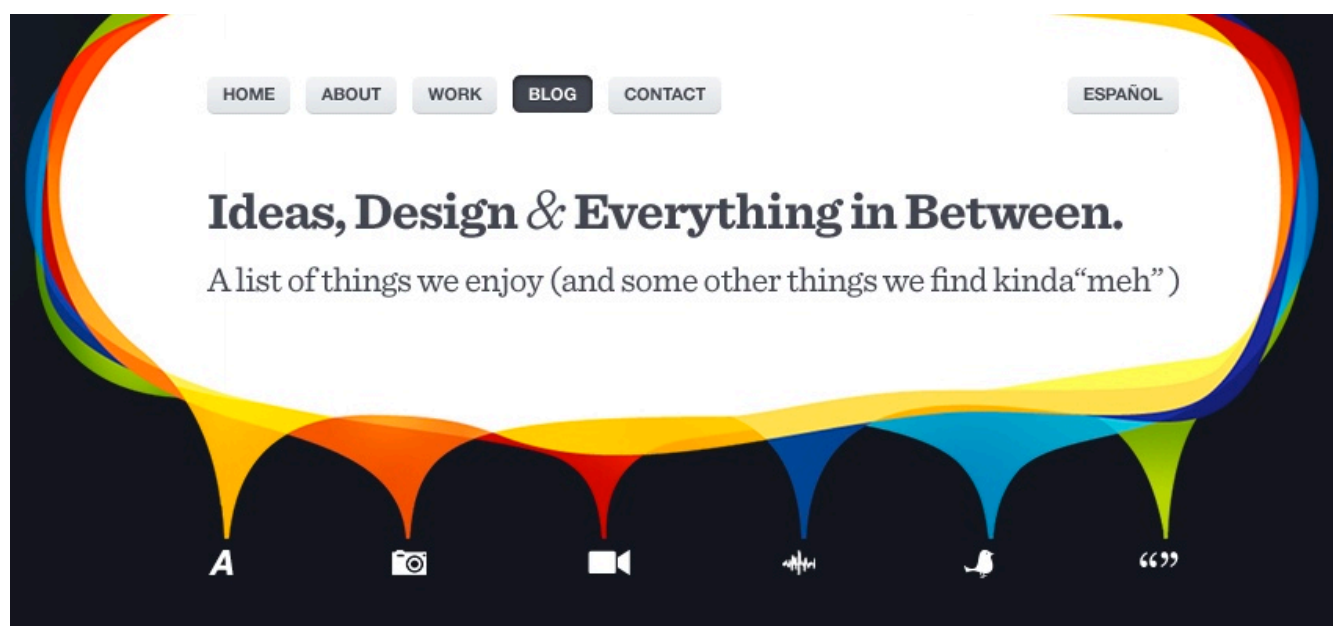
www.nosotroshq.com

Gmail

Paying close attention to the little things can make for a more intuitive and native user experience. If you are writing an email in Google's mail service and mention that you are attaching a file to your message but forget to attach it, the system points that out to you with a friendly warning message. This is a great example of an application having the user's best

interest at heart; the difference between a good and great Web application. If you would permit a slight exaggeration, you could compare this to a customer walking into a postal office with a package, and after a short distraction the friendly worker reminds the customer to send the package off.
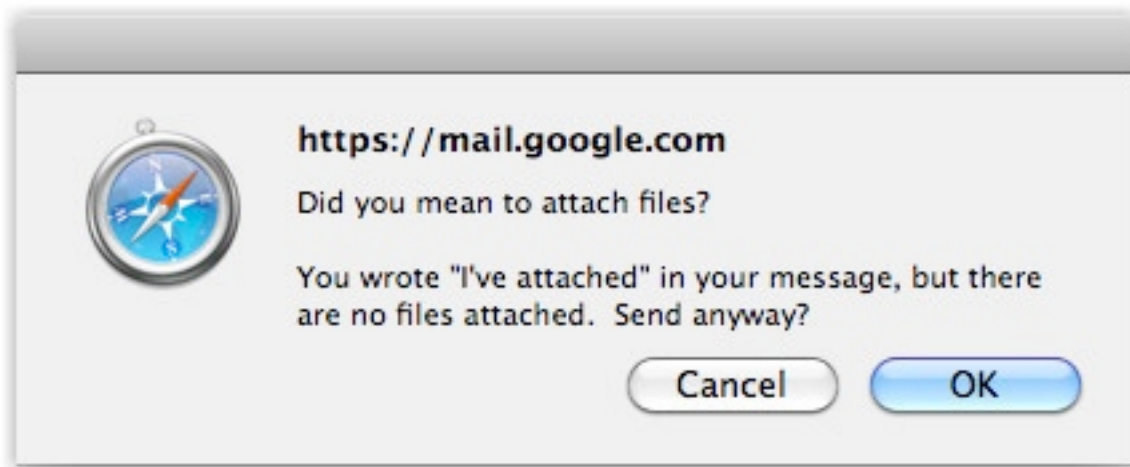
matthewlyle.com

KBB

KBB, a resource for finding and selling new and used cars, provides retail owners with an original pre-written email template. If the user is interested, they can just type their first and last name, street address and phone number and send the email in one click. Of course, the message can be customized. It's a good example of a Web application that carefully takes care of tedious tasks by providing users with a simple one-click solution.

✏ I'm Interested

To: 👤 ▸ **East Bay BMW**
1-866-439-3703
... More Info

From: [my email address]

Your message:

Hello, my name is [first name] [last name] and I'm writing you today to learn more about the **2007 BMW 6 SERIES 650I** listed for **$44,895**. I live at [my street address (optional)] in the 94114 [edit] area and I would like to hear back from you soon and learn more about this vehicle. Please call me at [my phone number] at your earliest convenience.

[personalize this message]

Thank you.

☑ Yes, I'm interested in receiving news and special offers from Kelley Blue Book.

▸ Privacy Policy   **SEND THIS MESSAGE**
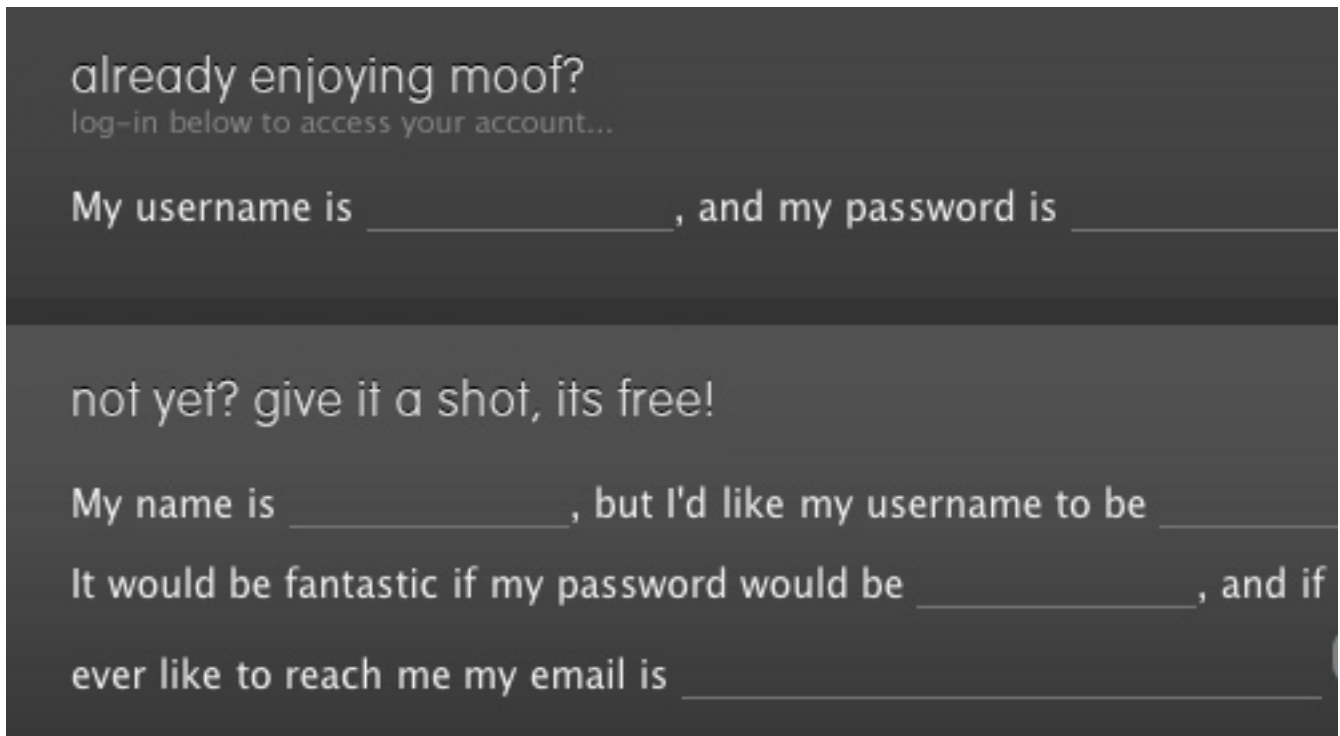
usedcars.kbb.com

Moof

Moof takes a similar approach with its contact form. It's not clear whether its approach increases or decreases conversion rates; it would be interesting to conduct a usability test on it.

moof.com

Livestream

Many Web applications have a variety of pricing plans and features, as does Livestream. This design, though, clearly communicates which plan would suit a particular category of users best. Notice the help tooltip accompanying the "Streaming" feature and the red ribbons at the bottom of the table. The yellow buttons are responsive, too, providing visual feedback. A simple, attractive design.

www.livestream.com

Notable App: Pricing Table

Highlighting the most popular plan is a good idea, because it helps users make a decision without having to inspect the whole table. This is a good example of using the metaphor of social validation: when users are in a social situation, they will look to others to see how to behave. It's not a conscious process, but it helps us to come to grips with difficult decisions.
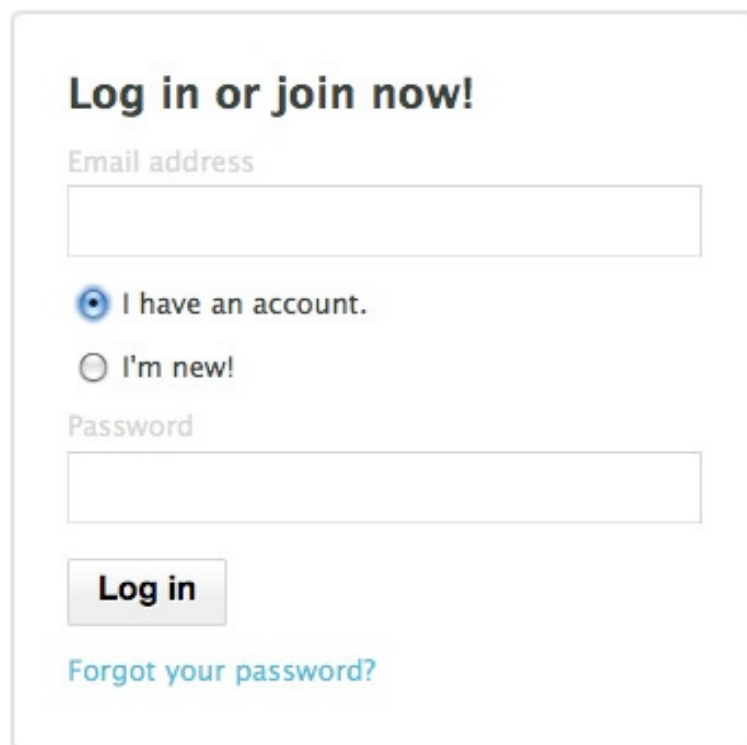
www.notableapp.com

Single Log-In and Sign-Up

A straightforward solution for reducing complexity is to combine multiple related features into one single feature. For instance, instead of having two separate links and pages for logging in and signing up, you could offer users a single "Log in or sign up" button that serves both purposes. The form would update via JavaScript after the user has selected a radio button. For this log-in form, notice that the "Submit" button says "Log in," and there is a link to retrieve lost passwords. For the sign-up form, the label for the password field prompts users to "Choose a password" and "Sign up" and to accept the terms of service.

An alternative method would be to provide two input fields, labeled "Email" and "Password," and then put a "Don't have an account yet? Sign up!" link next to it. Both approaches would work fine for new and returning visitors.

blog.leahculver.com

Web Design Trends

Vitaly Friedman

Subtle Interactivity

Although the opposite is often asserted, we believe that Flash has its place in the modern Web design. Today, we observe a clear separation of concerns between HTML/CSS websites and Flash websites. While HTML/CSS has become the de facto standard for text-based websites, focusing on information consumption, Flash is dominating entertainment and multimedia websites with its extensive visual and audio effects and rich user interaction.

But that doesn't mean that HTML/CSS websites are restricted to plain, unsophisticated user interaction. In fact, the opposite seems to be true. We're seeing more HTML/CSS websites getting interactive, playful and engaging. More animation is being used for visual feedback (for example, on mouse hovers and clicks), and visual effects are being used for a more responsive user experience. Of course, these effects rely on JavaScript libraries, which fuel the interactive layer in the background.

Still, designers of CSS-based websites tend to avoid extreme interactivity and instead use subtle, refined effects sparingly. Their designs use interactivity to support user interaction and shirk overwhelming users with striking 3D-effects and inconvenient navigation structures.

CoTweet

CoTweet has a good example of clean, attractive and responsive buttons. While the hover effect is subtle, clicking on the button is striking. Notice the difference in size and text color of the "Sign up" and "Learn more" buttons. Also notice the eye-catching ribbon on the "Sign Up" form, which stands out but isn't obtrusive. It may be a very subtle change, but adding small icons to the call-to-action button can be very helpful: for instance, think of adding a lock icon (metaphor for security) to communicate that the log in is secure.
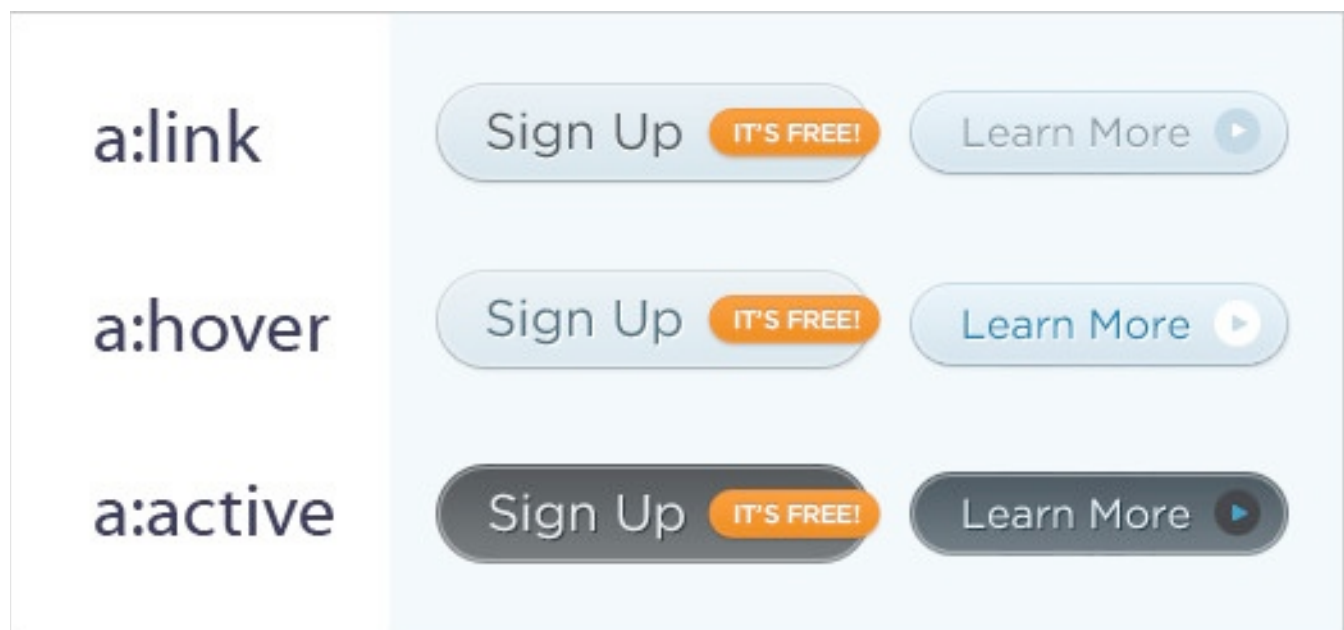
cotweet.com

Veerle Pieters

Veerle's recent redesign is very sexy. Links, content blocks and illustrations subtly react to mouse hovers in different ways: varying the transparency, incorporating animation, highlighting the link, changing the background image, fading to another color, displaying comments or modifying the navigation. The design is very inviting, engaging and responsive. Veerle's design also uses keyboard navigation, another common development in the Web design industry. Hands down, a beautiful first-class design.

veerle.duoh.com

Fuel Brand Inc.

The official page of Fuel Brand Inc. has a variety of nifty animations: images increase in size on hover, background images change, slideshows are used for navigation, and interactive tooltips and other effects are apparent. Interactivity seems to be popular with official company websites, too.

fuelbrandinc.com

Malcolm Reading Consultants

Another interesting design technique for subtle interactivity: the three images displayed at the top automatically slide vertically, updating the background images for the horizontal navigation at the top, too. No Flash is used here.
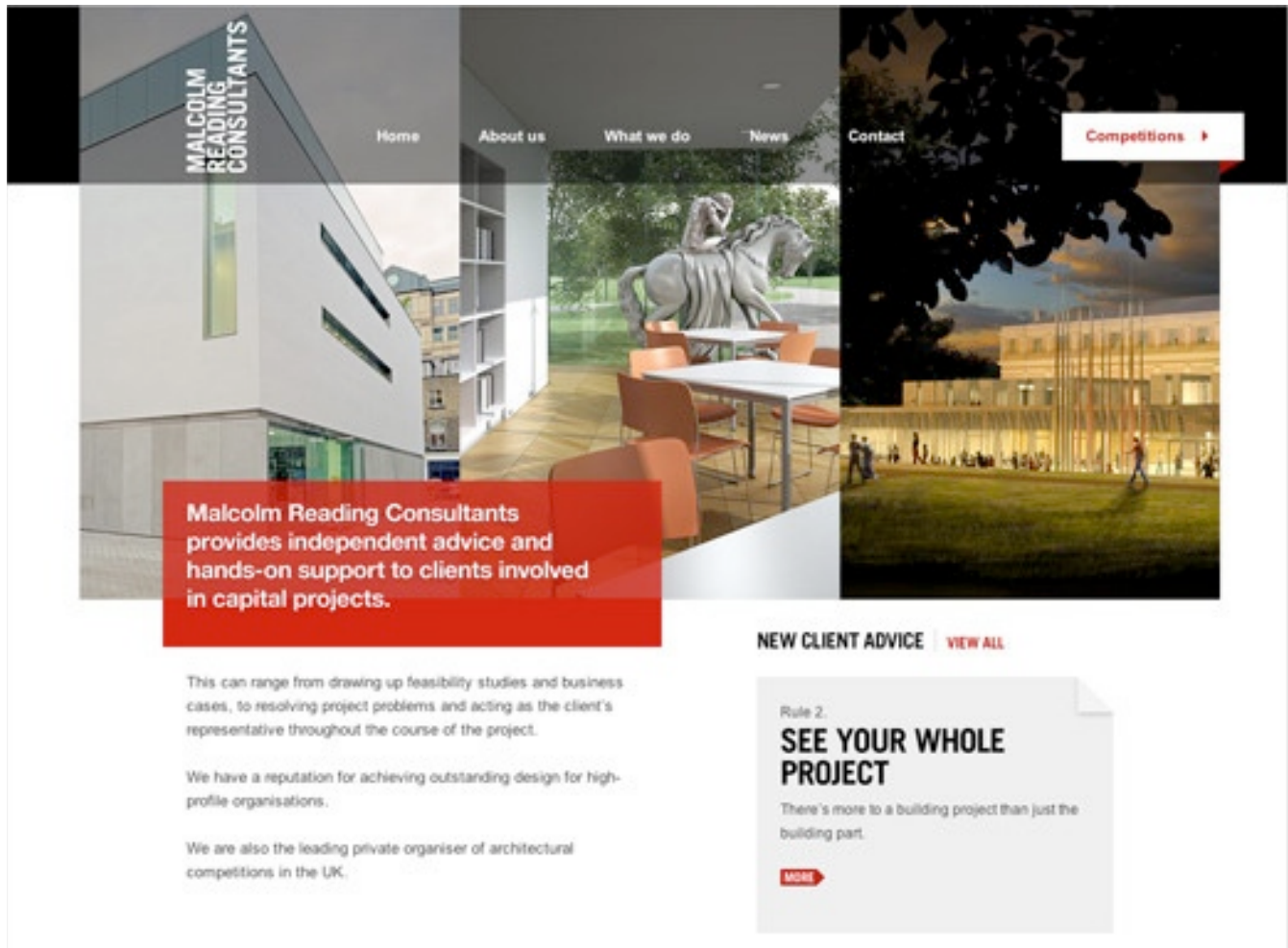
www.malcolmreading.co.uk

LifeGear Mockup

A mock-up for an interactive product page. Many e-commerce pages are being given more interactive product presentations, too; for example, 360-degree views or interactive instructions such as this one.
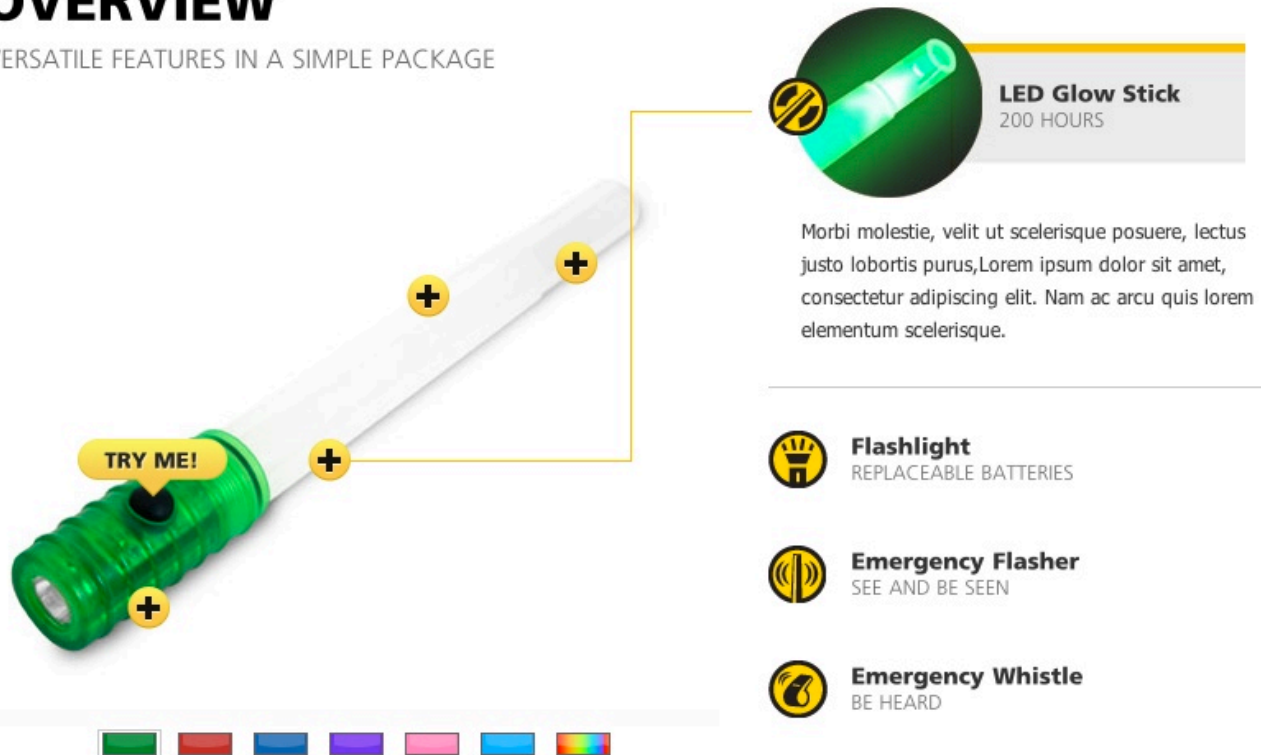
addyawards.dtelepathy.com

Carsonified: 100 Things

Carsonified integrates a subtle color transition and fade in-out effect to quickly display information about the speakers at this conference.

carsonified.com

Harmony Republic

Here we have interactive hover effects and rich typography, but also clumsy text management and long loading time because of the Flash nature of the website. This site could just as easily have been designed with pure JavaScript, allowing for better copying and pasting of text, flexible font sizing and native mouse interaction (i.e. changing from the pointer to hand-on-link hovering).
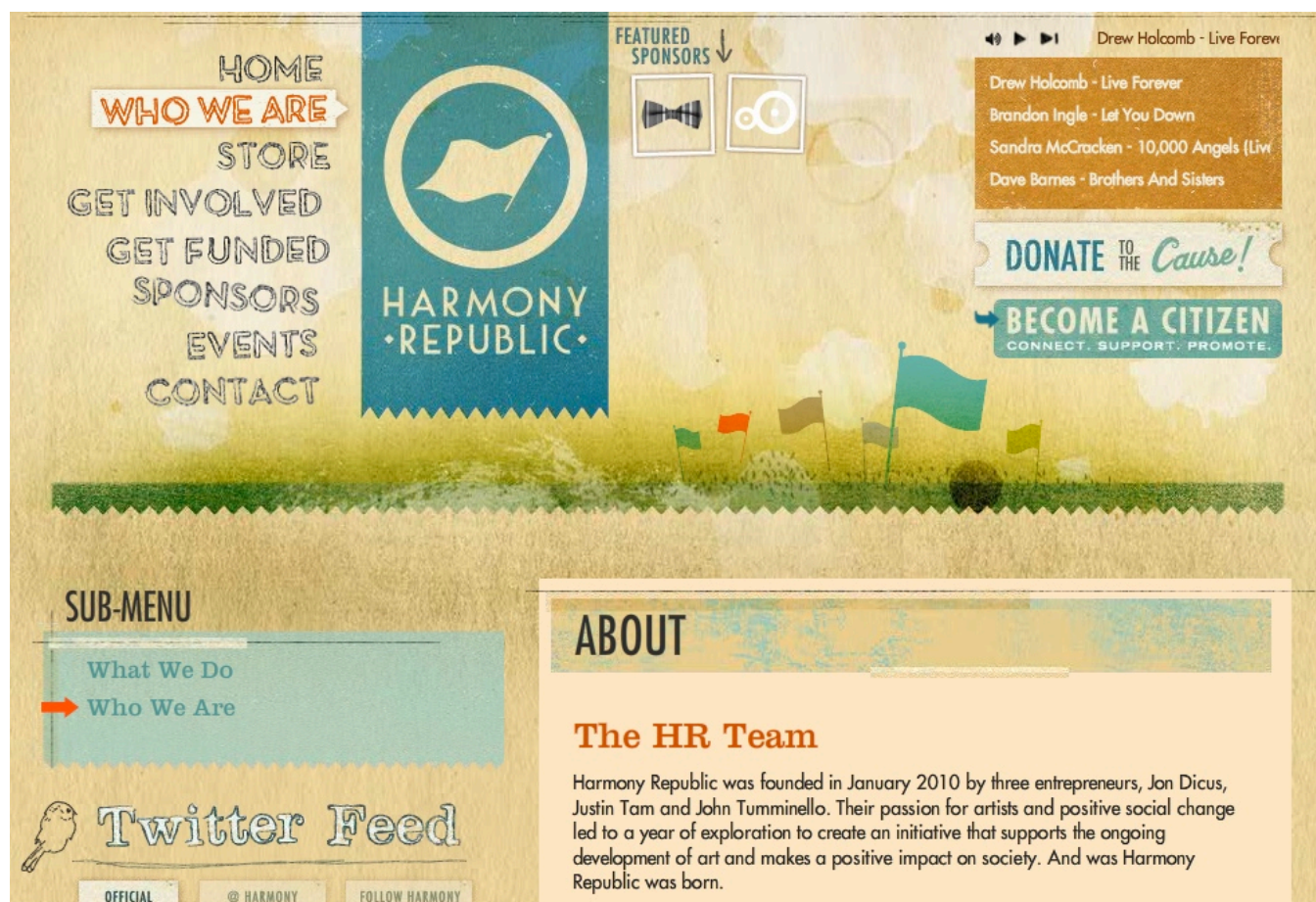
www.harmonyrepublic.com

arc 90: Meet the Team

Yet another example of an "About" page with interactive design elements. Photos are replaced on hover: but visitors can also easily filter members of the team by their occupation. Images filtered out fade to grayscale.
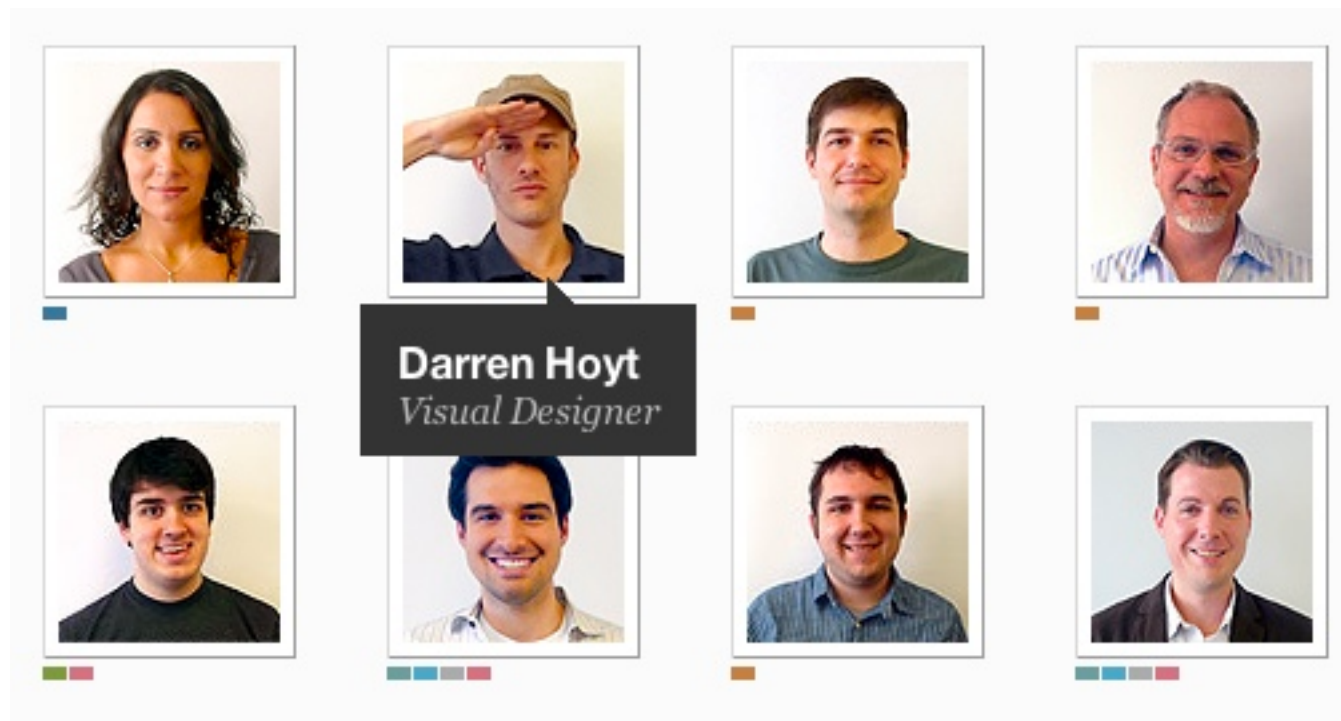
arc90.com

S P D

Another nice subtle hover effect. On this portfolio page, each item fills one cell. The header has 18 cells in all. When the user hovers over one, it expands to fill the area of the six adjacent cells, showing additional information about the selected item. No Flash in use here. And again, the subtle interaction is rather helpful.

The 401(k) Company
Via nautical metaphors, this corporate capabilities brochure guides the way for companies outsourcing large 401(k) programs for their employees. Thus allowing their clients to keep focused on their core businesses.
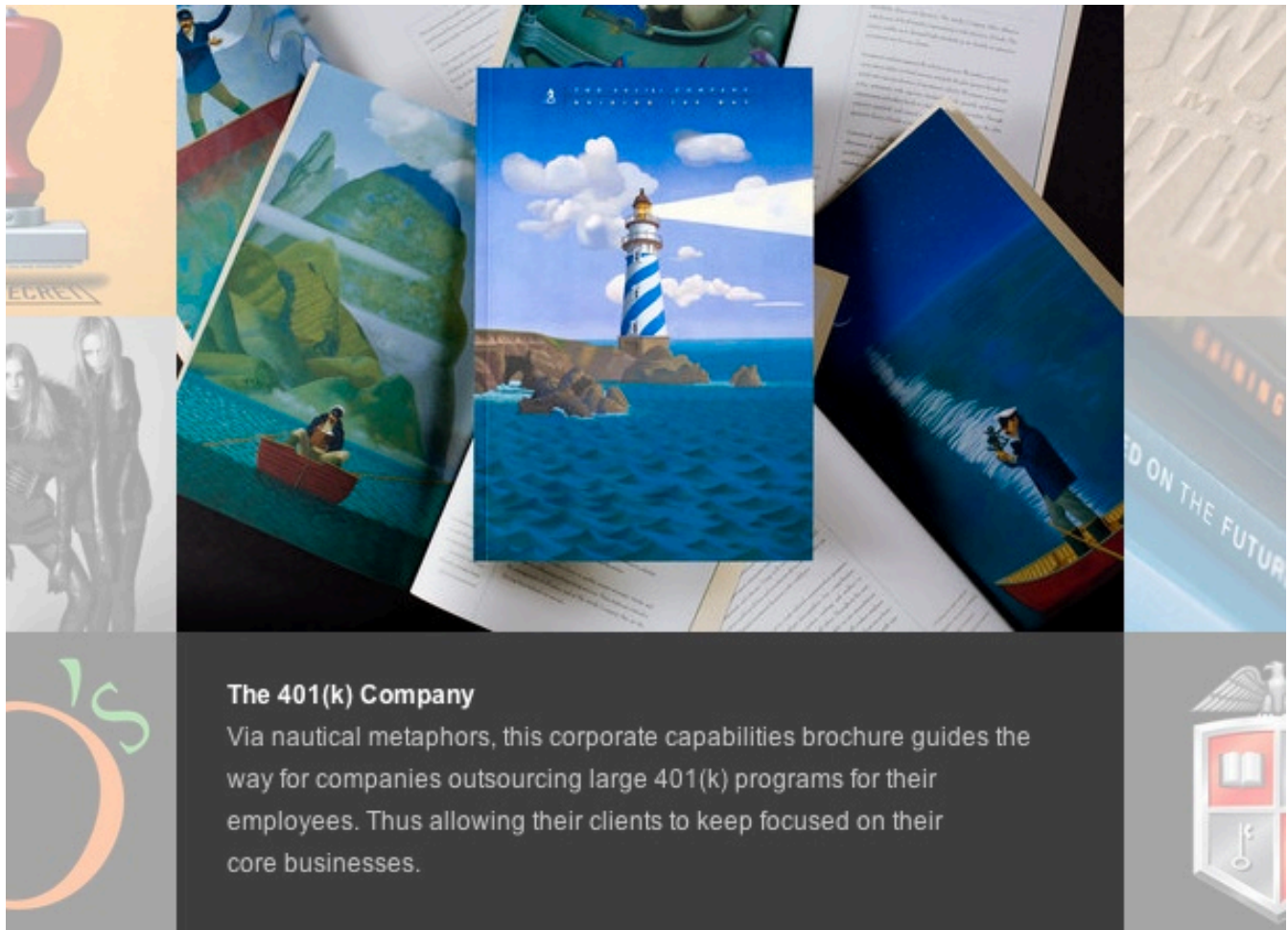
www.spdaustin.com

Duplos

One can take interactivity a bit further, too. In Ricardo Mestre's portfolio, the design elements slowly float across the page. Can you spot the dangerous Twitter-birdie monster? And of course, no Flash is used here.

duplos.org

Context-Sensitive Navigation

Context-sensitive navigation is essentially navigation (often additional to the primary navigation) that depends on the context of the user on the page (for example, what they're doing). Its biggest advantage is that it can remove distractions and controls that are irrelevant to the user's current context. Context-sensitive navigation displays options only when the user actually needs them. Obviously, this paradigm would be particularly useful

to user interface designers: it can help de-clutter interfaces and focus the user's attention on the task they are performing.

This approach has been used in Web development for years now, but from our observations, it is appearing on more websites and Web applications today. Vimeo is a classic example of context-sensitive navigation: playback controls fade away once the video starts playing and appear again on demand (i.e. when the user hovers over the video area).

When implementing context-sensitive navigation, make sure the navigation is actually available when users need it. Keep in mind various settings and user scenarios, and test it in worst-case scenarios. For video playback controls, users of touch interfaces (e.g. iPhone or iPad users) will not be able to toggle the navigation because the hover effect won't be available. In this case, you may want to consider touch gestures instead.

Vimeo

Vimeo, a video-sharing platform, displays playback controls until the user starts watching a video. Once the video is playing, the controls fade away and appear again only when the user hovers over the video. A classic example of context-sensitive navigation. Such navigation is being used increasingly on other playback websites, too — for example, Blip.fm.

www.vimeo.com

New York Times

The New York Times cleverly displays related and featured stories in its articles. Rather than promoting them in the actual content of articles, the designers have moved them to an appropriate context: a block slides in from the bottom-right corner when the user reaches the end of the article: convenient and helpful. Usability studies will determine whether users find this behavior annoying in the long run. Also, notice that whenever you click on a word in an article (except for links), a "?" tooltip appears allowing you to look up the selected word. This service makes much more sense presented in this context than if it was hidden in the navigation menu at the top or in the sidebar. Context-sensitive navigation at its best.

Uncertainty about the stability of assets in money market funds signal
that accelerated the downward spiral of the credit crisis in 2008, and u
banks to briefly halt lending to one other.

Now, as Europe teeters, the dangers to the A**?**erican economy — and t
financial system — are becoming increasingly evident. "It seems like o
European policy makers were gleefully watching the U.S. get its econo
not appreciating the massive tidal wave coming at them across the Atl
Rogoff, a H
economist c

MORE IN BUSINESS (1 OF 30 ARTICLES)

Europe Strives to Catch Up With Digital Hollywood

Read More »

James Kant

www.nytimes.com

Reuters

Drop-down navigation is another common place for context-sensitive navigation. Reuters' drop-down-navigation is fairly advanced. It doesn't just provide access to navigation options, but also displays useful information, such as current stocks and hot topics, which are revealed immediately. Also notice the "Open" button in the right sidebar (not displayed in the screenshots below). When it is clicked, a detailed information box (initially hidden but instantly available on demand) overlays the current page and displays the relevant information (see the second screenshot below). Reuters also allows users to follow their favorite topics and save articles for later reading. Personalization is indeed coming to the Web.

REUTERS

EDITION: U.S.

News & Markets ▾

Sectors & Industries ▾

Markets

U.S. Markets   Stocks
Global Markets   Hot Stocks
Indices   Funds
Deals   Currencies
M&A   Commodities
Private Capital   Futures
Regulation   Bonds
Small Business   Analyst
Green Business   Research
Economy   Alerts
                      Portfolio

» All Markets

U.S. & World News

U.S.   Science
World   Lifestyl
Technology   Enterta
Politics   Oddly E
Health   Sports
Housing Market   Film
Healthcare   Music
Reform   Televisi
Media   Fashior
G20

» All U.S. & World News

Today's Hot Topics

UK Election   Pakistan   Latin America Investment Summit   Greece   Oil

DOW   NASDAQ   S&P 500
-139.89   -54.00   -17.27
10,380.43   2,265.64   1,110.88

Russia offers olive branch as NATO joins parade

Full A
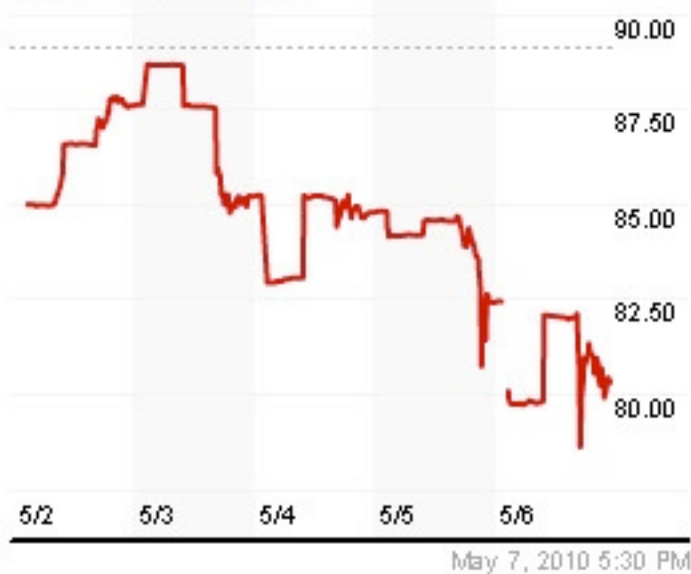
www.reuters.com

AllTop

Instead of displaying dozens of categories by default, Alltop displays them on demand, when a letter in the upper-right corner is triggered. Another example of hiding secondary information in a convenient, user-friendly way.
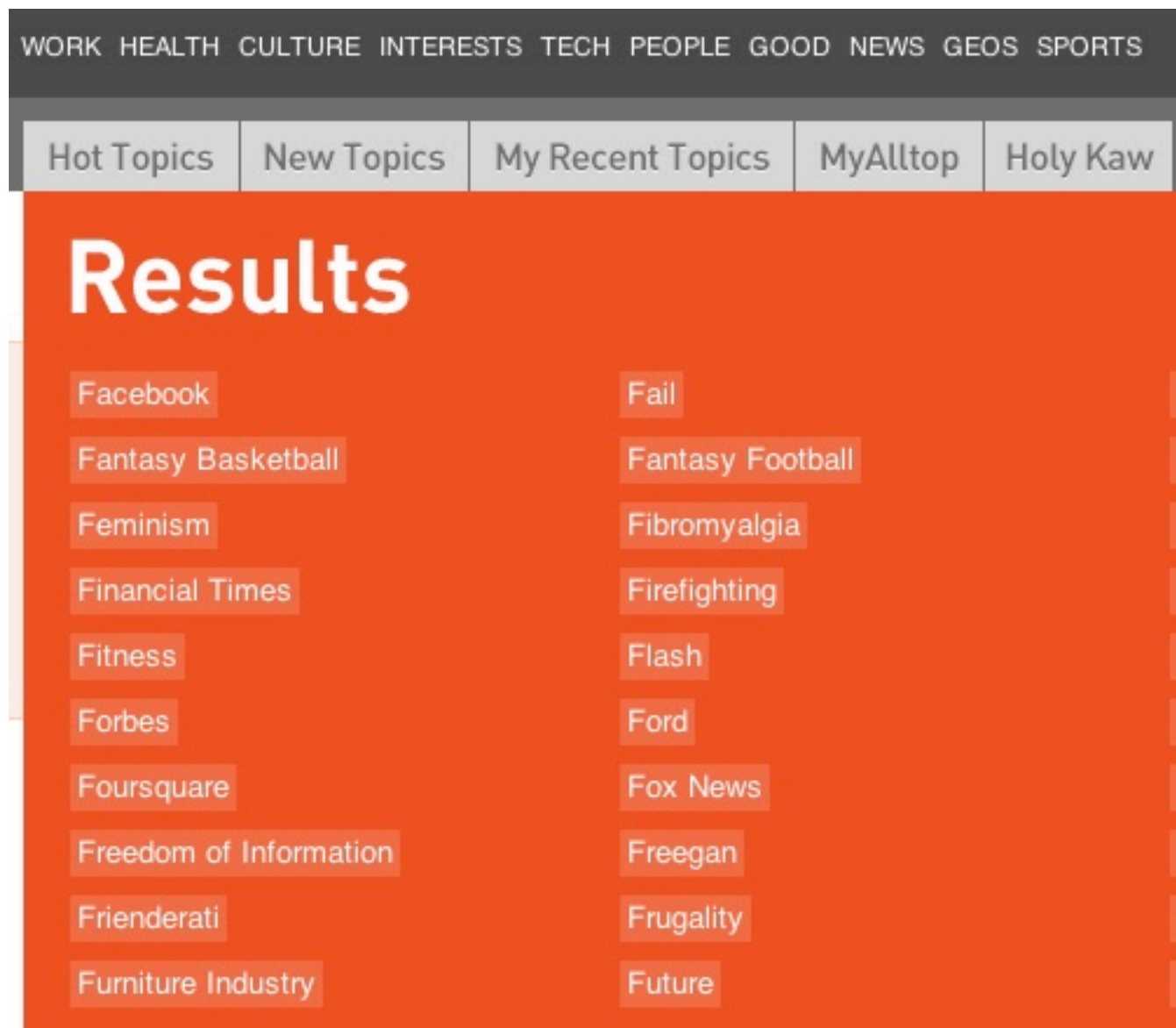
user-interface.alltop.com

DailyMile

DailyMile has a clean and attractive user interface. The major navigation options are displayed in the left sidebar. When the user hovers over one, a hint is displayed, explaining the link. In other Web applications, it may be useful to underline the navigation options and display the secondary options on hover. Also notice the "More" switch at the bottom of the sidebar, offering secondary navigation options; it is available only if the user is actively searching for those options, but is hidden otherwise.
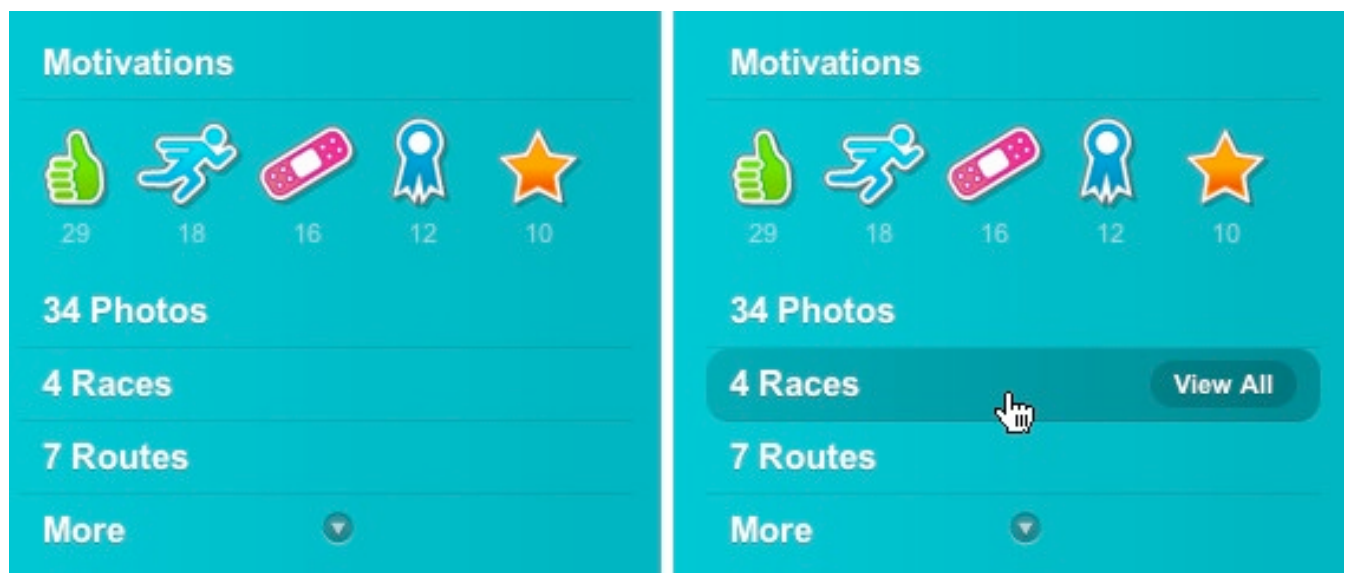
www.dailymile.com

The Black Harbor

By default, the images section here doesn't display any content other than the actual image. But when the user hovers over the image, the title, description and number of comments appear. Notice the "Share" balloon in the upper-right corner. On hover (and only on hover), it slides in links to Twitter, Facebook, Digg and StumbleUpon. All of this information is displayed when the mouse hovers over the specific area (in the specific context), and it fades away automatically when the mouse moves away.
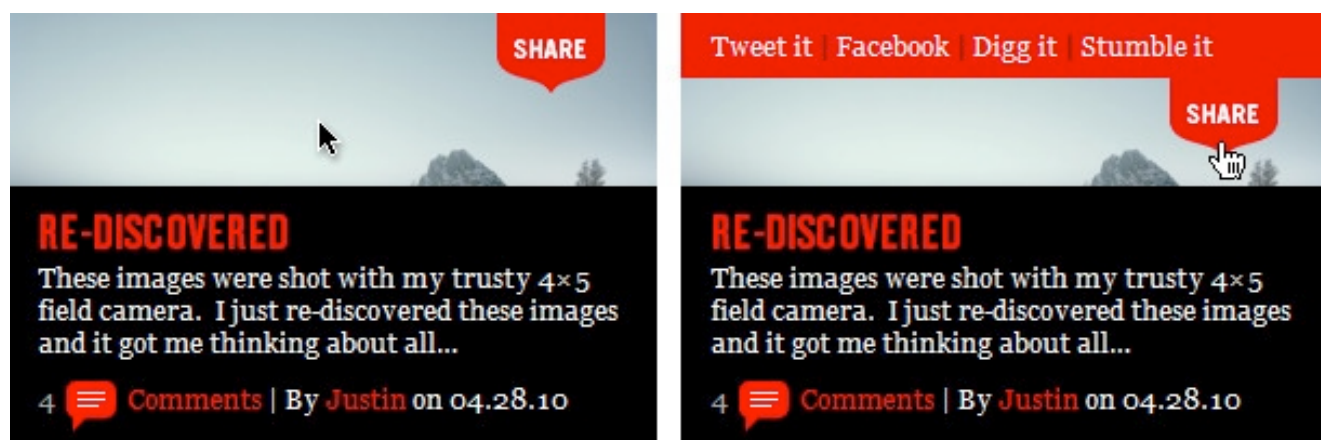
theblackharbor.com

Rapid CSS3 Adaptation

With improving support of CSS3 features in major modern browsers, the design community seems very excited about the new advancements and possibilities. Everyone seems to be experimenting with CSS3; whether on RGBa transparency, advanced CSS3 selectors, CSS3 transforms or other attributes. The result is truly remarkable: we're seeing many designs that are more beautiful, advanced, versatile and robust than before, with fewer nasty browser hacks.

The most popular CSS3 features so far seem to be `border-radius`, animations, multiple backgrounds and `box-shadow`. This is understandable because these can easily be used to replace quirky earlier solutions that required workarounds and third-party tools to perform the task. Also, `@font-face` is rapidly gaining popularity, but keep in mind that it is not a CSS3 feature because it was proposed for CSS2 and was luckily implemented in early versions of Internet Explorer (and in other browsers later — FF 3.5+, Safari 3.2+, Chrome 4.0+, Opera 10.1+). This is why it's actually safe to use `@font-face` in your designs today.

The bad news with CSS3 is that in order for the features to be easily applied to all major browsers, we will have to wait for Internet Explorer 8 to expire. The most popular browsers today (i.e. the family of Internet Explorer versions) support only the CSS3 box-sizing attribute. Internet Explorer 7/8 have broad support for CSS3 attribute selectors (`[att$=val]`, `[att^=val]`, etc.), combinators (`E~F`), but no support for CSS3 pseudo-classes (`:nth-child()`, `:last-child`, etc.). Internet Explorer 9 will support at least CSS3 media queries, CSS3 colors (HSL, HSLa, RGBa), CSS3 selectors and `border-radius`. In the meantime, we will have to use CSS3 for modern browsers and integrate workarounds for IE, and there are [plenty of them](#).

Things that seemed impossible a year ago are being implemented with pure CSS3, fueled by the creative minds of the design community. We may have entered a fertile new era of Web design, when crucial CSS techniques are being born, similar to the 2003 to 2005 period when CSS Sprites and CSS Sliding Doors were developed. In fact, we have a couple of exciting creative years to look forward to.

SxSW Beercamp

For the design of SxSW Beercamp, the creators made some rather bold decisions. For each letter at the top of the page, they used the `text-shadow` property 49 times. Also, various animations and transformations were applied, as were `box-shadow` and HSLA. By the way, did you notice the sunrise at the top? Well, the designers use 20 `<hr>`-tags to achieve this effect.

sxsw.beercamp.com

Tapbots

CSS3 features work best when applied subtly, with fine details added to a clean, simple and meaningful design. Tapbots uses CSS3 for rounded corners, hover animations and some transparency effects, which are difficult to find at the first glance. This is a good example of CSS3 features being properly applied: no bold headlines with `text-shadow`, no exaggerated gradients, no excessive animation — just solid, good design.
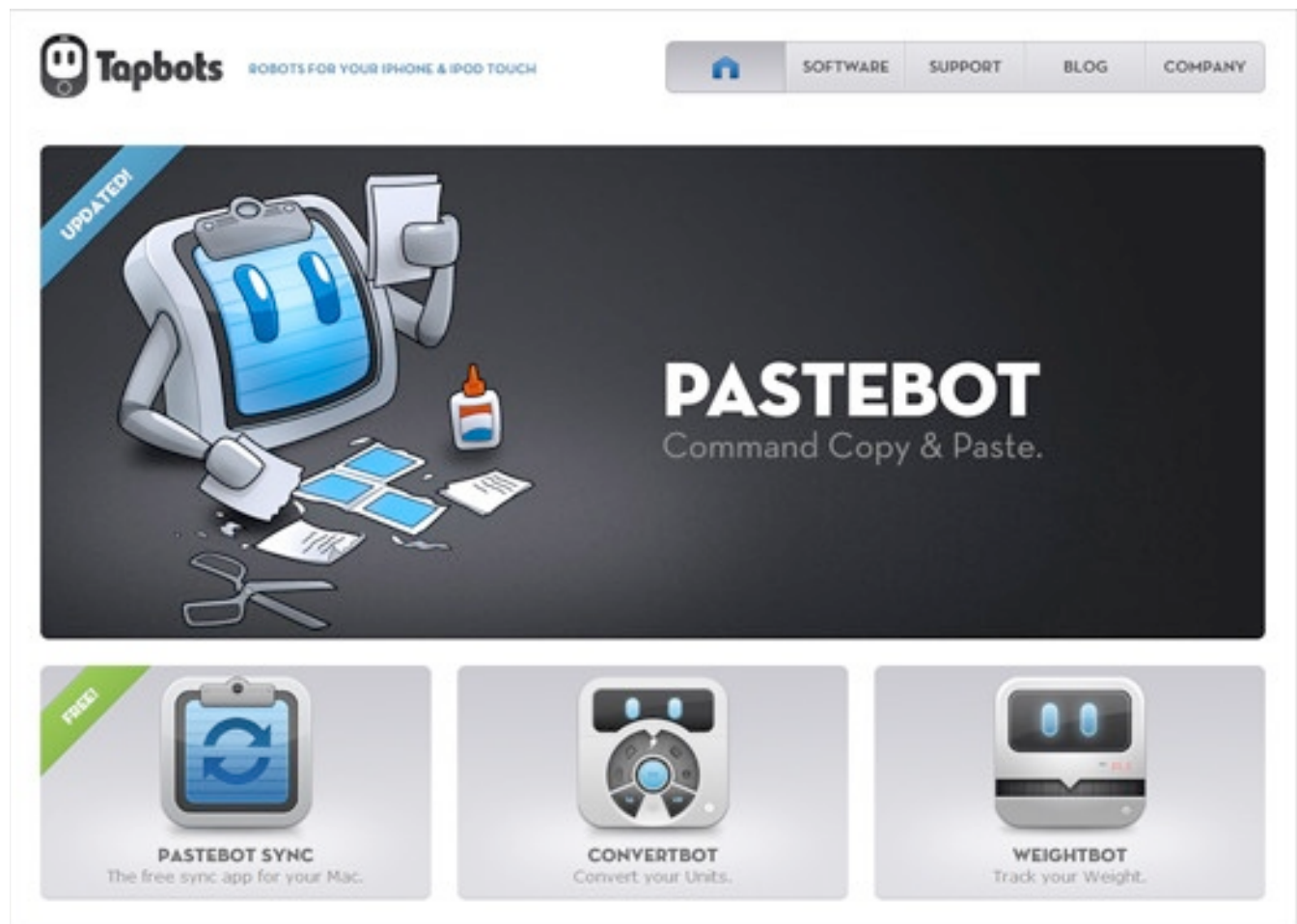
tapbots.com

full frontal

The `text-shadow` property seems to be particularly effective when a light text shadow is used against a dark background. Here is another simple example of a dark text shadow (a bit darker than the background) used on white headlines. Also, the page uses `border-radius` and `box-shadow` for the quick links listed under the description of each speaker.

full-frontal.org

going steady with stacey

But sometimes applying a dark `text-shadow` against a light background works well, too. But in such cases you may want to add a "stronger" text shadow.

desandro.com

CSS3 Drop Down Menu

A more practical approach to CSS3 features, using `border-radius`, `box-shadow`, gradients and `text-shadow` on a multi-level drop-down menu to good effect. No images used here.
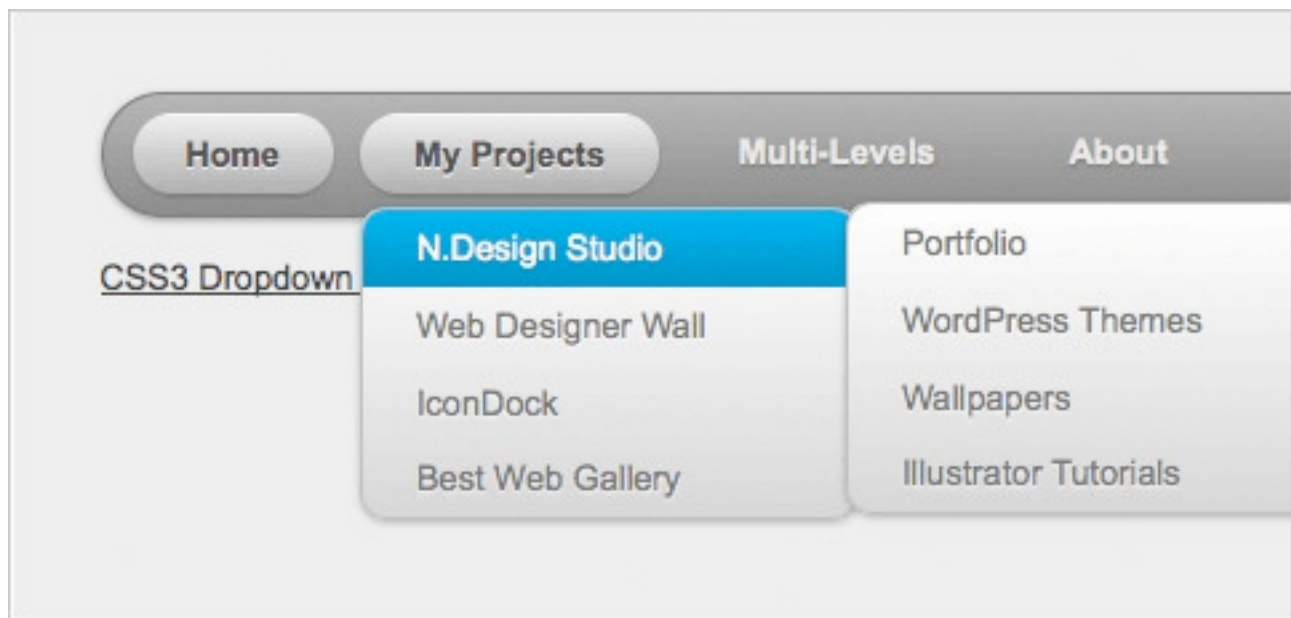
www.webdesignerwall.com

Kristof Orts

Kristof Orts uses the `text-shadow` property to add depth to his headlines. The embossing effect is clean and sharp and helps the typography stand out.

adifferentdesign.be

CSS3 Experiments: Moon

Dan Cederholm has prepared a nice CSS3 demo to illustrate some of the new features designers can use in their designs today. In particular, notice subtle animations in the sidebar. CSS-transitions and CSS-animations in use. The design provides users with nice visual rewards, creating a more responsive and engaging interaction.
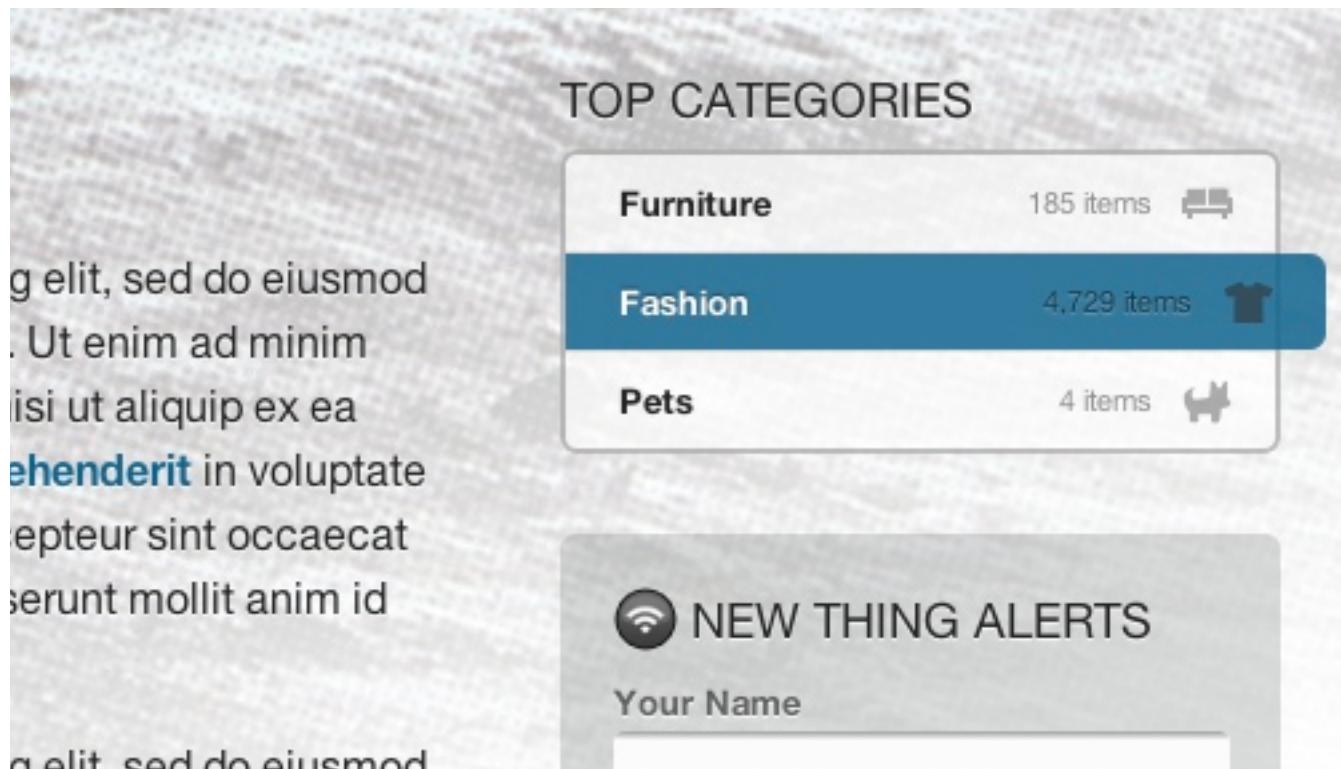
[css3exp.com](css3exp.com)

Faruk Ateş

Faruk Ateş' blog is a practical example of using CSS3 animations and transitions for a nice hover-effect. Notice the effective use of `text-shadow` on a vibrant blue background.
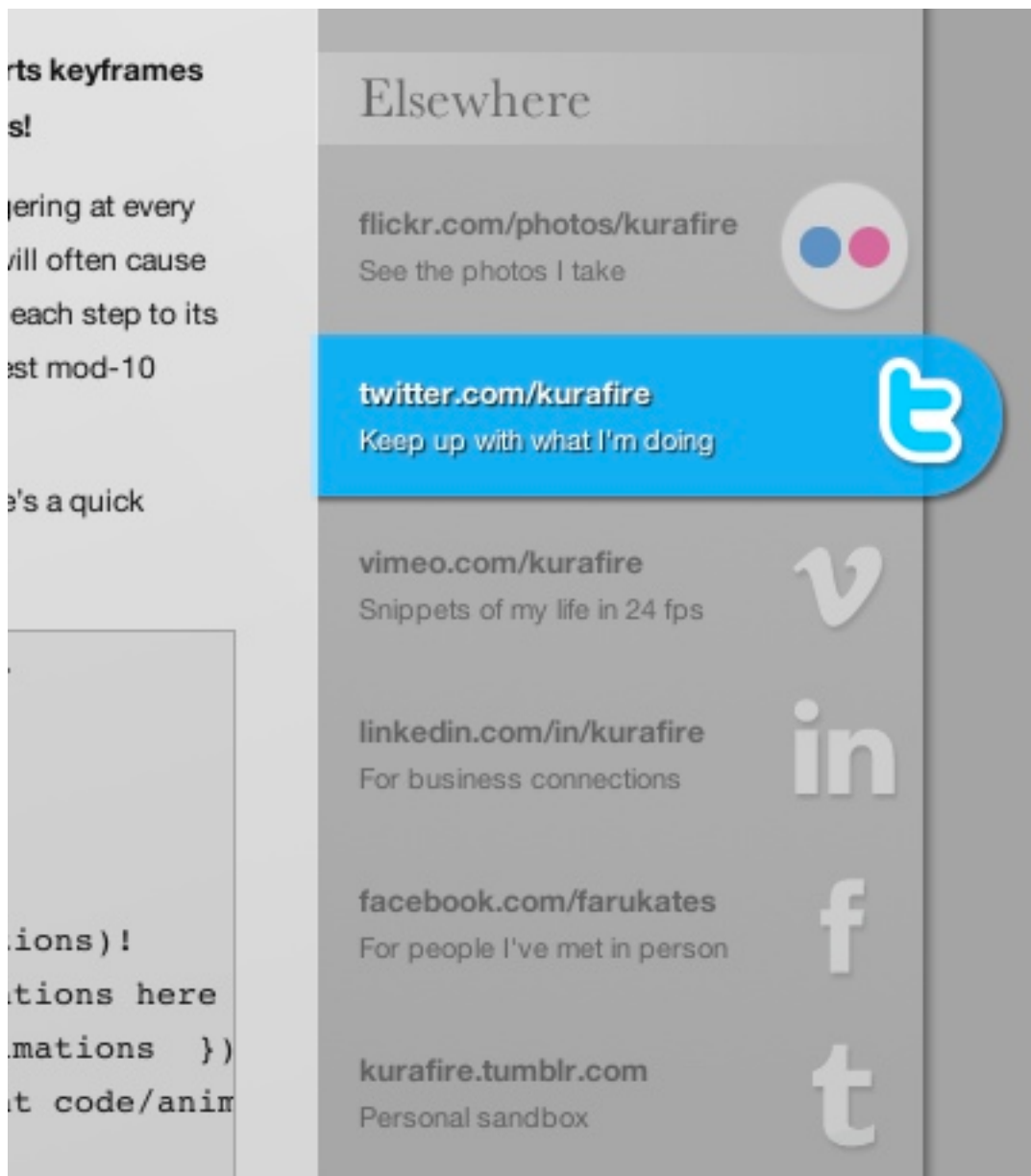
farukat.es

Mindgarden

Mindgarden uses CSS3 media queries to display the layout differently in different resolutions. Users with a wide browser window size get a multi-columns layout (two or three columns, see the first screenshot), and users with a smaller browser window size see only one column (second screenshot). Also, the dimensions of images are scaled automatically depending on user's screen size. Web designers have been using JavaScript for the same purpose for years, but now we can create adaptive layouts using pure CSS-based solutions. Media queries are a CSS3 extension to media types that gives designers more control over rendering of the site layout. They are currently supported in Firefox 3.6+, Safari 4+, Opera 10+ and Chrome. Internet Explorer 8 does not support them.

www.mindgarden.de

Chirp

The Chirp conference website uses CSS animation for the rotation and animation of the images of speakers.

chirp.twitter.com

Bello and Proxima Nova: Type Experiment

For his experiment, Tim Brown uses gradients, masking, rounded corners, transitions and multiple background images. On hover, the heading at the top seems to stand out.
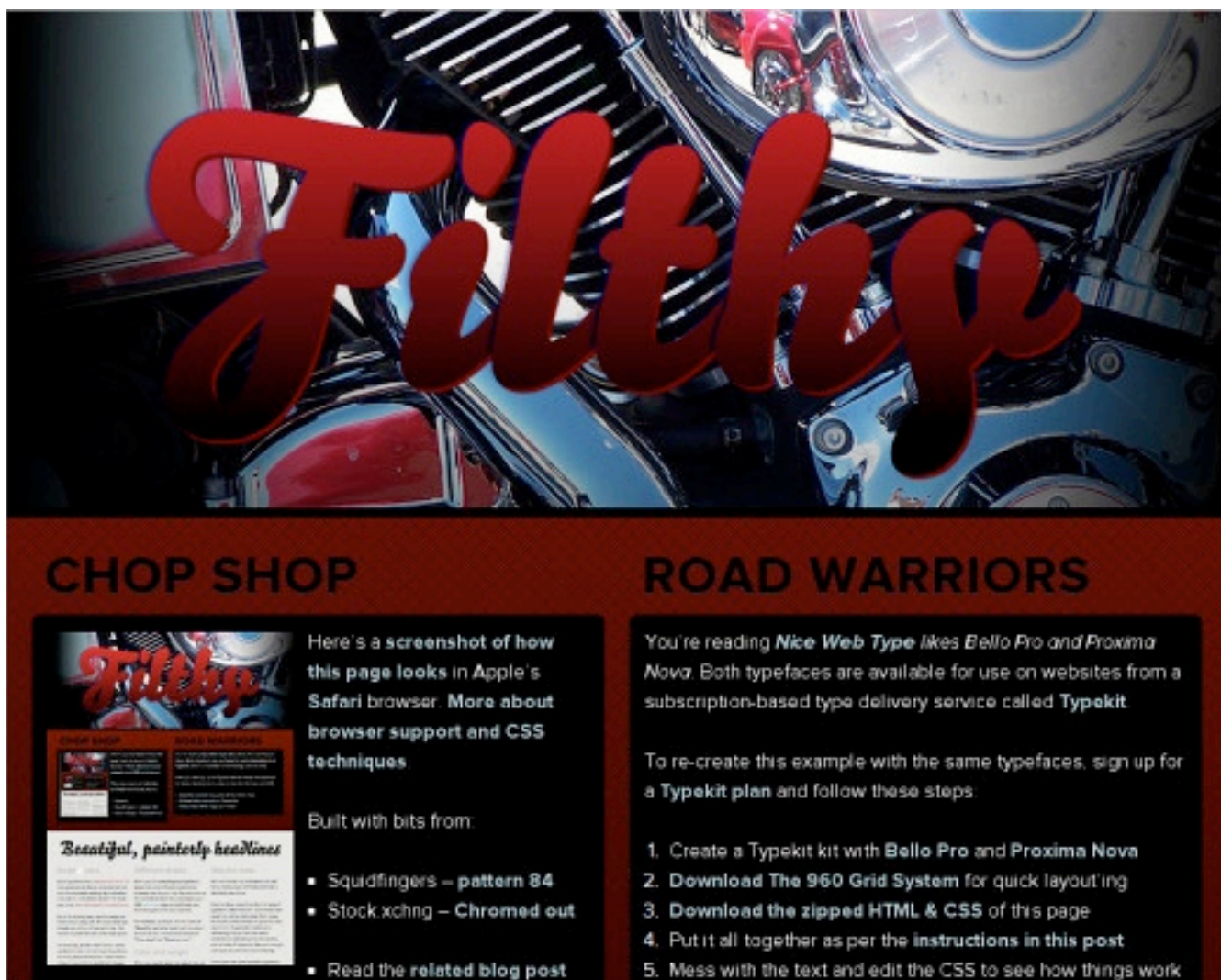
nicewebtype.com

Neography CSS3 Type Experiment

No images are used here. Pure CSS3 transitions, rotations, shadows and other tweaks.
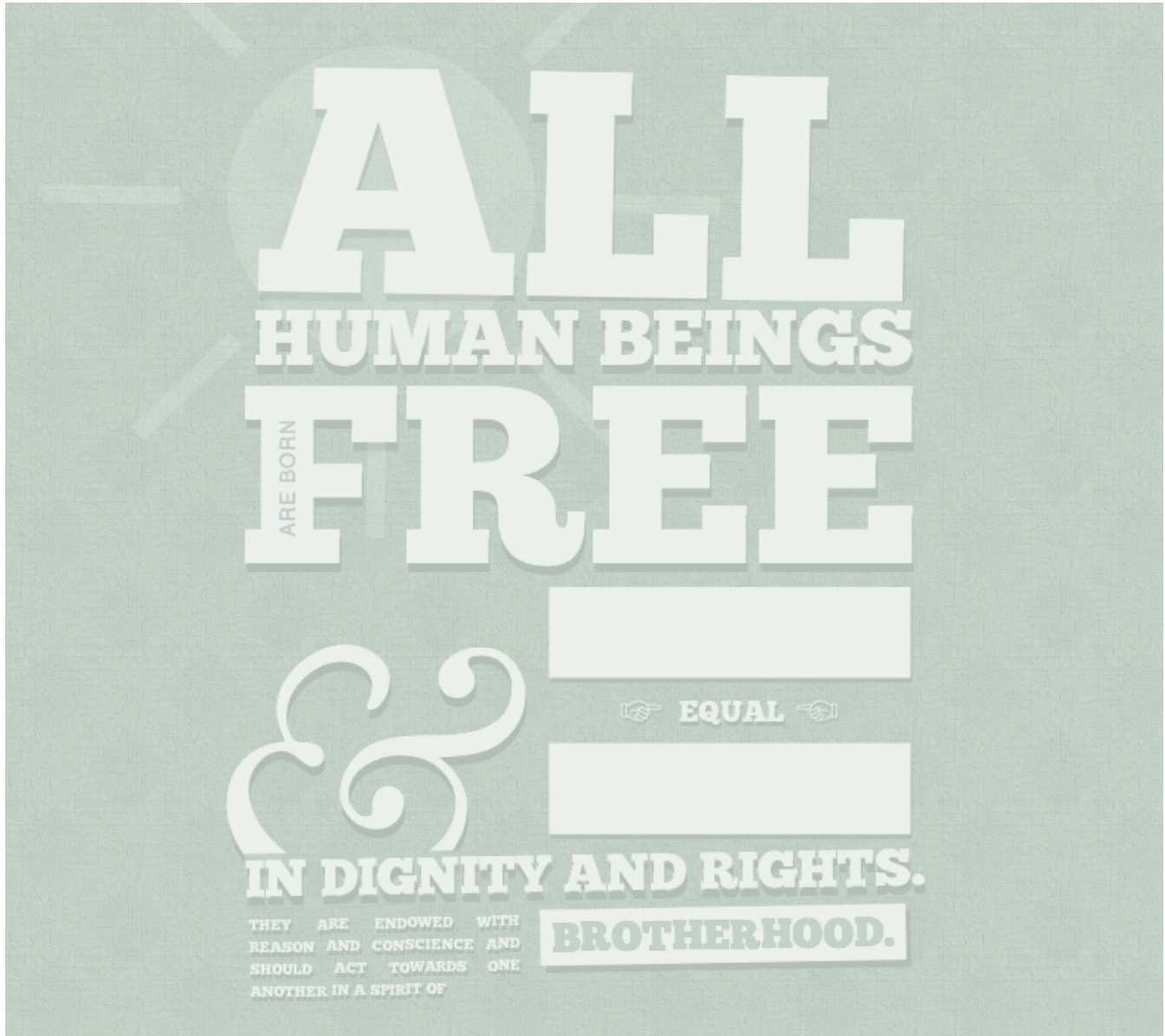
neography.com

CSS Posters

Designers are experimenting with different layout techniques and CSS3 techniques. Here is an example of a CSS poster, designed with simple CSS rules. Actually, a very simple technique that uses `@font-face` for font embedding. But it just looks great and serves its purpose, too.

designinformer.com

David Desandro: Footer

CSS3 gone mad: David Desandro uses various CSS3 rules in the footer of his website. Shadows, animations and transitions are very vivid, colorful and appealing. This is the opposite to subtle CSS3 effects used on most sites with CSS3-snippets. However, it works perfectly fine in this case as it sets the design apart from other portfolio websites.
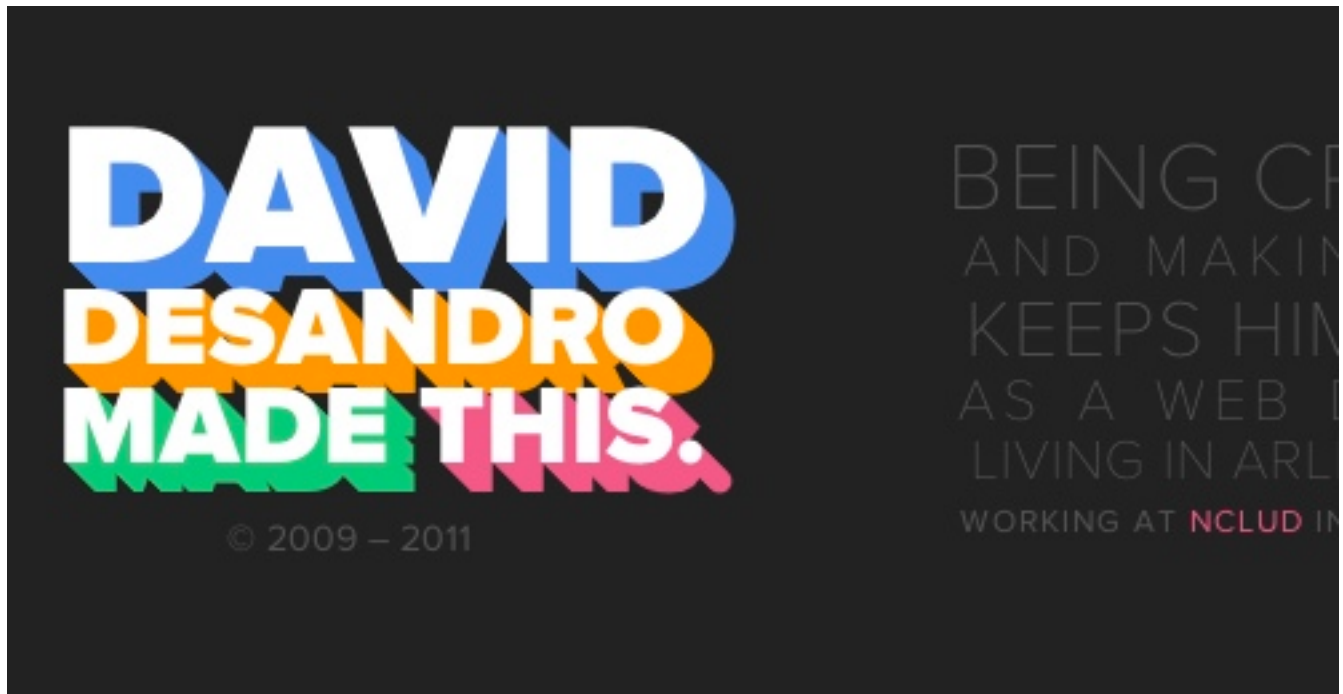
desandro.com

Opera Logo with CSS3

Various visual effects are possible with CSS3. With a bit of artistic creativity, you can create truly extraordinary things. For example, the Opera logo here is designed purely with CSS3: gradients, `box-shadow` and `border-radius`. Are CSS shapes something that we should be expecting to come soon?

The Opera logo, rendered only with CSS, no images. Compare it with the real deal.

Original Logo   Firefox 3.6   Chrome 5.0   Opera 10.5   IE 6, 7, & 8

desandro.com

Rain Drop

Another interesting example of a visual effect achieved purely with CSS3.
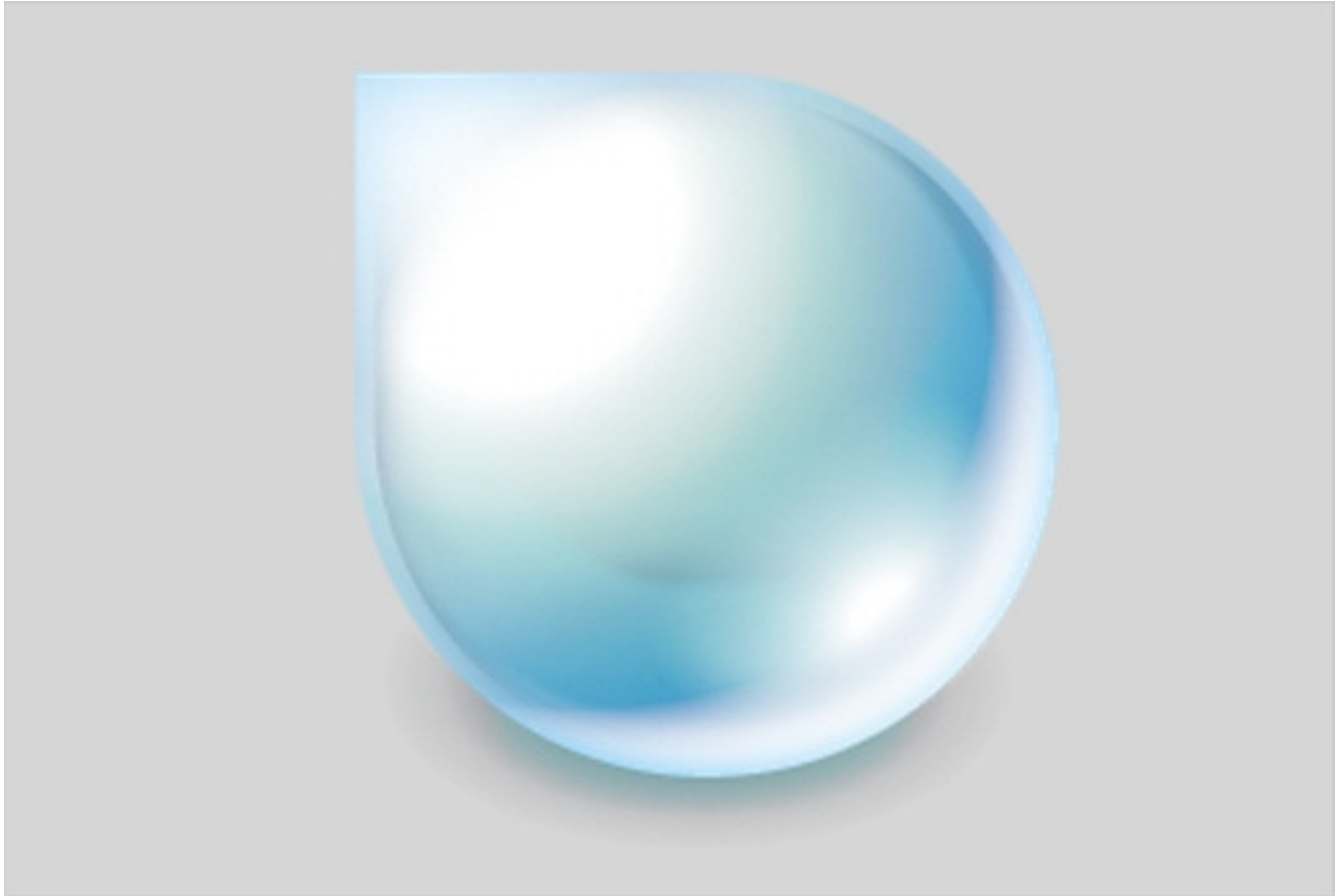
mozilla.seanmartell.com

A Sad Note About the Text-Shadow Property

A general problem we have observed with CSS3 adoption today is the emergence of designs crafted merely for the sake of CSS3. A perfect illustration of this is the overuse of the `text-shadow` property, which is sometimes heavily applied to headings and body copy. Bear in mind that CSS3 is a powerful tool, and as effective as it can be, it can also be easily misused, leading to a new round of usability problems and legibility issues. It seems like a very obvious advice, but it is still worth mentioning: before adding a CSS3 feature to your website, make sure it is actually an enhancement, added for the purpose of aesthetics *and* usability, and not aesthetics at the cost of usability.

An obvious yet important rule relates to this: your design shouldn't rely on CSS3 improvements, but should rather integrate them as a rich additional layer for users with modern browsers. The drawback of advanced CSS3 properties is that, because they are not yet standardized, and because Gecko and Webkit require proprietary attributes to implement certain ones, in a couple of years we may end up with quite bloated style sheets that contain useless styles. So, you may want to extract all proprietary CSS3 enhancements to a separate style sheet in order to simplify maintenance later on. Also, keep in mind that your CSS3 stack should always end with the standardized CSS3 rule (e.g. `border-radius`, not `-moz-border-radius`). By doing this, you make sure that in the next years, with a better support for CSS3 rules in newer versions of Web browsers, your rules will actually still apply and work in major browsers.

Media Twitter

On some Windows machines, the navigation, headlines and body copy set with `text-shadow` can be quite difficult to read with the default text settings. Sometimes it is really not necessary to use `text-shadow` for body copy and navigation, thus compromising the quality of the design for users with older browsers.
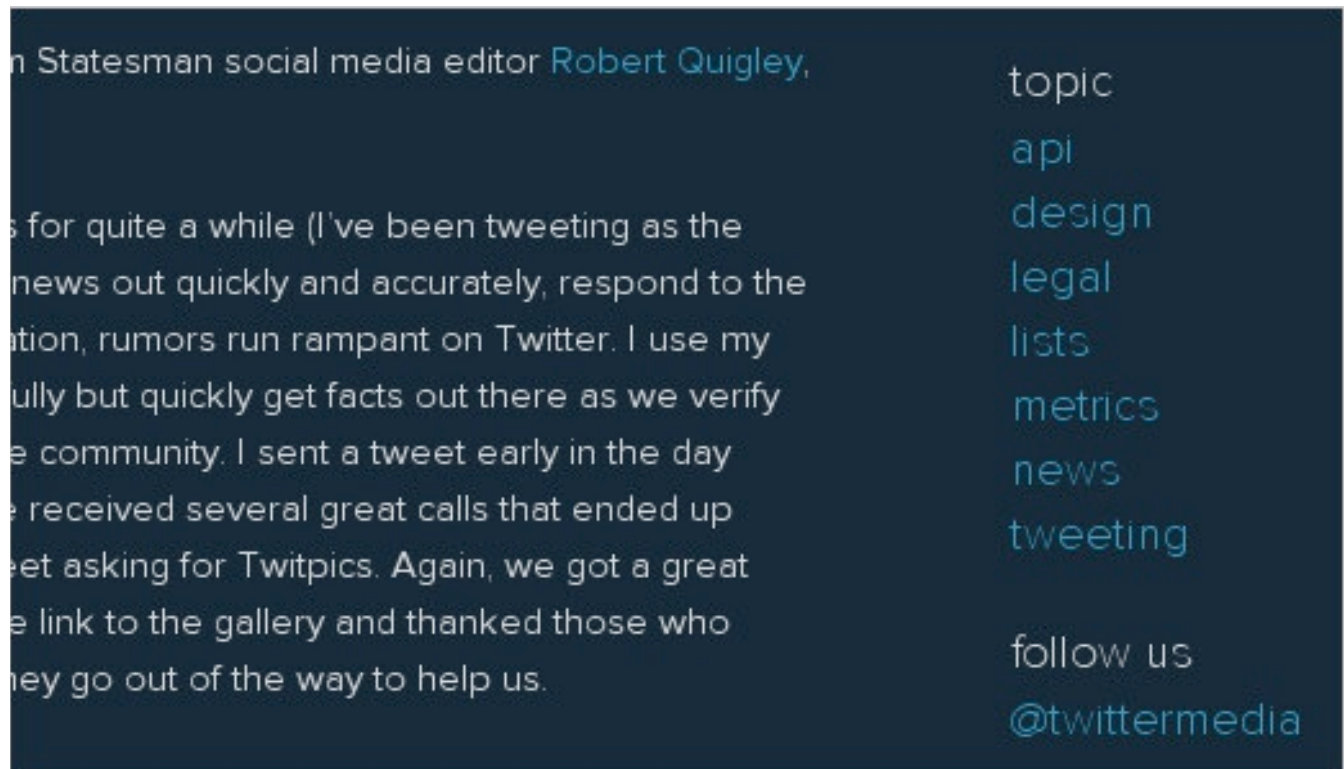
media.twitter.com

Brandon Cash

Brandon Cash uses the `text-shadow` property for his body copy. While the text is legible, it is more difficult to read than if it were plain copy. Compare the view in Chrome (left) and Safari (right): the difference is remarkable. It is very likely that many users do not use ClearType or any other kind of font-smoothing for text. As a designer and advocate of user's needs, you have to keep that in mind.
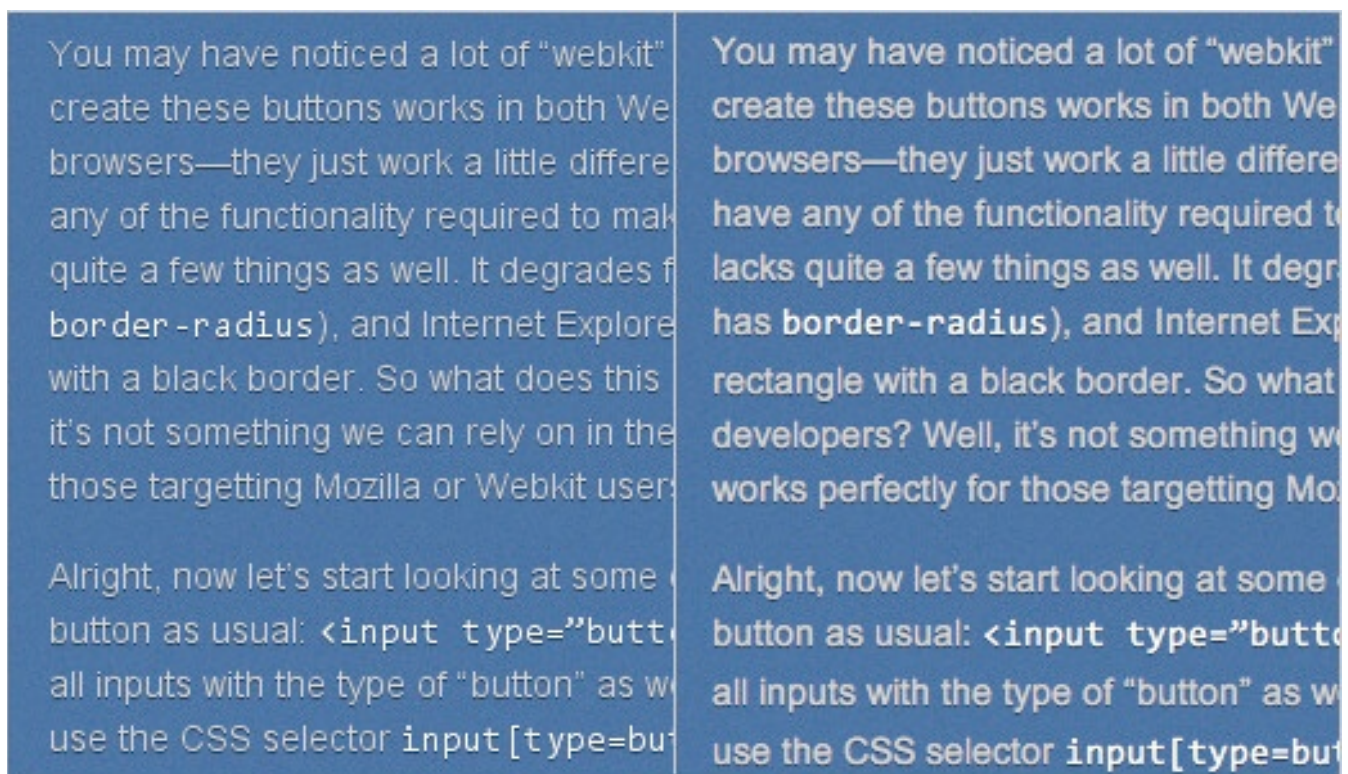
blog.brandoncash.net

Conclusion

New possibilities are always implemented creatively, but being so trendy, they are also abused. While good design practices are being incorporated into Web designs rapidly (through real-life metaphors, responsive interfaces and context-sensitive navigation), we have observed many "over-designed" websites that use new CSS3 features for their own sake.

Design trends do not exist to be followed. They exist to be broken, reshaped and abolished by the spirit of creativity rippling through the design community. By nature, trends are anything but good: they distract from the main purpose of communication design and disrupt the balance between aesthetics and function. Rather than follow trends, use the appropriate tools in the appropriate context for the right purpose. That's where the beauty of information design lies, and it makes the huge difference between a nice design and a nice and *usable* design.

Responsive Web Design: What It Is and How to Use It

Kayla Knight

Almost every new client these days wants a mobile version of their website. It's practically essential after all: one design for the BlackBerry, another for the iPhone, the iPad, netbook, Kindle — and all screen resolutions must be compatible, too. In the next five years, we'll likely need to design for a number of additional inventions. When will the madness stop? It won't, of course.

In the field of Web design and development, we're quickly getting to the point of being unable to keep up with the endless new resolutions and devices. For many websites, creating a website version for each resolution and new device would be impossible, or at least impractical. Should we just suffer the consequences of losing visitors from one device, for the benefit of gaining visitors from another? Or is there another option?

Responsive Web design is the approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform and orientation. The practice consists of a mix of flexible grids and layouts, images and an intelligent use of CSS media queries. As the user switches from their laptop to iPad, the website should automatically switch to accommodate for resolution, image size and scripting abilities. In other words, the website should have the technology to automatically *respond* to the user's preferences. This would eliminate the need for a different design and development phase for each new gadget on the market.

The Concept Of Responsive Web Design

[Ethan Marcotte](#) wrote an introductory article about the approach, "[Responsive Web Design](#)," for A List Apart. It stems from the notion of responsive architectural design, whereby a room or space automatically adjusts to the number and flow of people within it:

> *"Recently, an emergent discipline called "responsive architecture" has begun asking how physical spaces can respond to the presence of people passing through them. Through a combination of embedded robotics and tensile materials, architects are experimenting with art installations and wall structures that bend, flex, and expand as crowds approach them. Motion sensors can be paired with climate control systems to adjust a room's temperature and ambient lighting as it fills with people. Companies have already produced "smart glass technology" that can automatically become opaque when a room's occupants reach a certain density threshold, giving them an additional layer of privacy."*

Transplant this discipline onto Web design, and we have a similar yet whole new idea. Why should we create a custom Web design for each group of users; after all, architects don't design another building for each group size and type that passes through it? Like responsive architecture, Web design should automatically adjust. It shouldn't require countless custom-made solutions for each new category of users.

Obviously, we can't use motion sensors and robotics to accomplish this the way a building would. Responsive Web design requires a more abstract way of thinking. However, some ideas are already being practiced: fluid layouts, media queries and scripts that can reformat Web pages and mark-up effortlessly (or *automatically*).

But responsive Web design is not only about adjustable screen resolutions and automatically resizable images, but rather about a whole new way of thinking about design. Let's talk about all of these features, plus additional ideas in the making.

Adjusting Screen Resolution

With more devices come varying screen resolutions, definitions and orientations. New devices with new screen sizes are being developed every day, and each of these devices may be able to handle variations in size, functionality and even color. Some are in landscape, others in portrait, still others even completely square. As we know from the rising popularity of the iPhone, iPad and advanced smartphones, many new devices are able to switch from portrait to landscape at the user's whim. How is one to design for these situations?
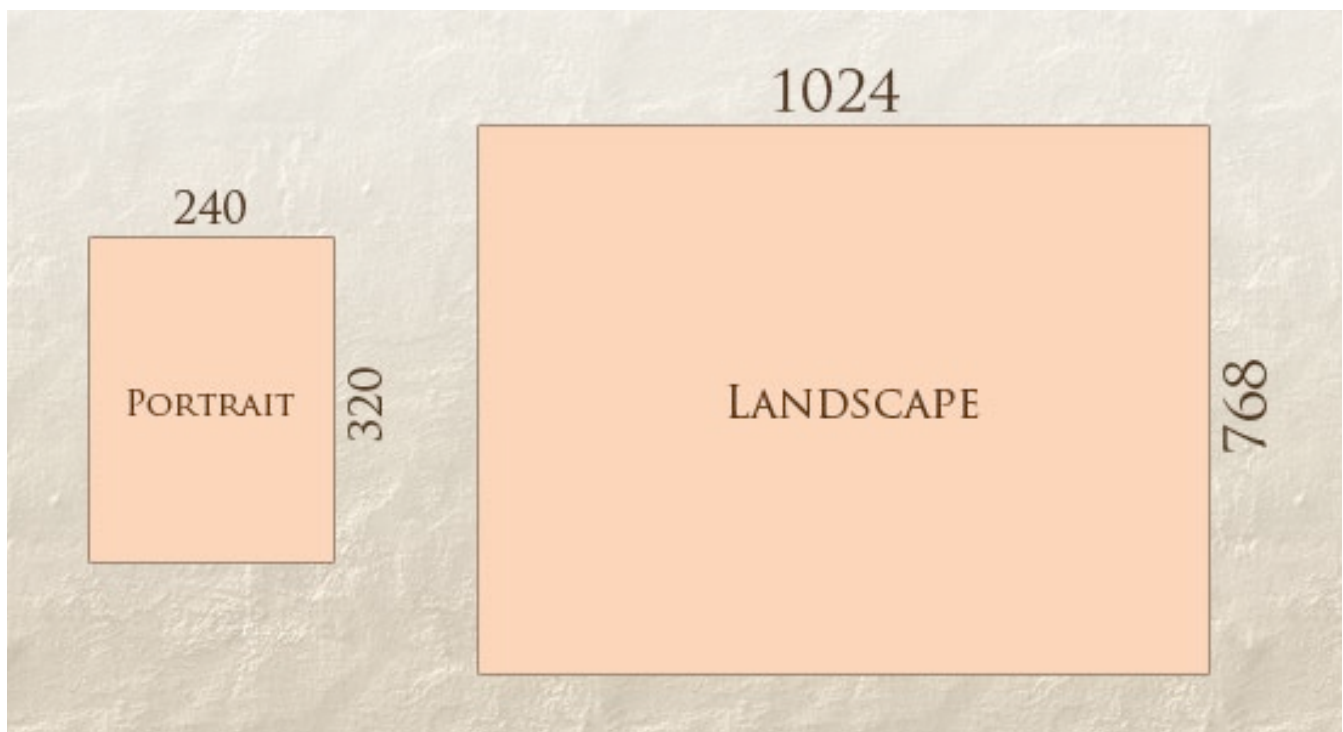
In addition to designing for both landscape and portrait (and enabling those orientations to possibly switch in an instant upon page load), we must consider the hundreds of different screen sizes. Yes, it is possible to group them into major categories, design for each of them, and make each design as flexible as necessary. But that can be overwhelming, and who knows what the usage figures will be in five years? Besides, many users do not maximize their browsers, which itself leaves far too much room for variety among screen sizes.

Morten Hjerde and a few of his colleagues identified statistics on about 400 devices sold between 2005 and 2008. Below are some of the most common:
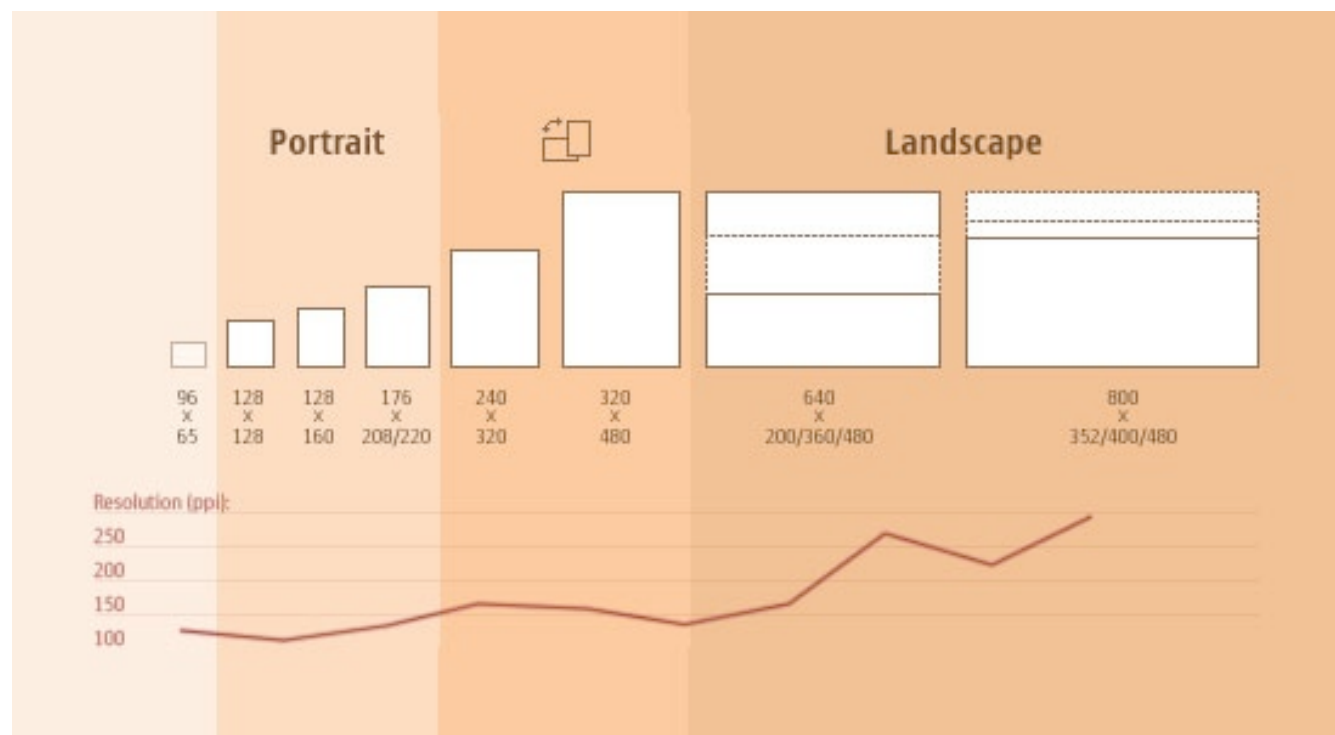
Since then even [more devices have come out](). It's obvious that we can't keep creating custom solutions for each one. So, how do we deal with the situation?

Part of the Solution: Flexible Everything

A few years ago, when flexible layouts were almost a "luxury" for websites, the only things that were flexible in a design were the layout columns (structural elements) and the text. Images could easily break layouts, and even flexible structural elements broke a layout's form when pushed enough. Flexible designs weren't really that flexible; they could give or take a few hundred pixels, but they often couldn't adjust from a large computer screen to a netbook.

Now we can make things more flexible. Images can be automatically adjusted, and we have workarounds so that layouts never break (although they may become squished and illegible in the process). While it's not a complete fix, the solution gives us far more options. It's perfect for devices that switch from portrait orientation to landscape in an instant or for when users switch from a large computer screen to an iPad.

In Ethan Marcotte's article, he created a sample Web design that features this better flexible layout:
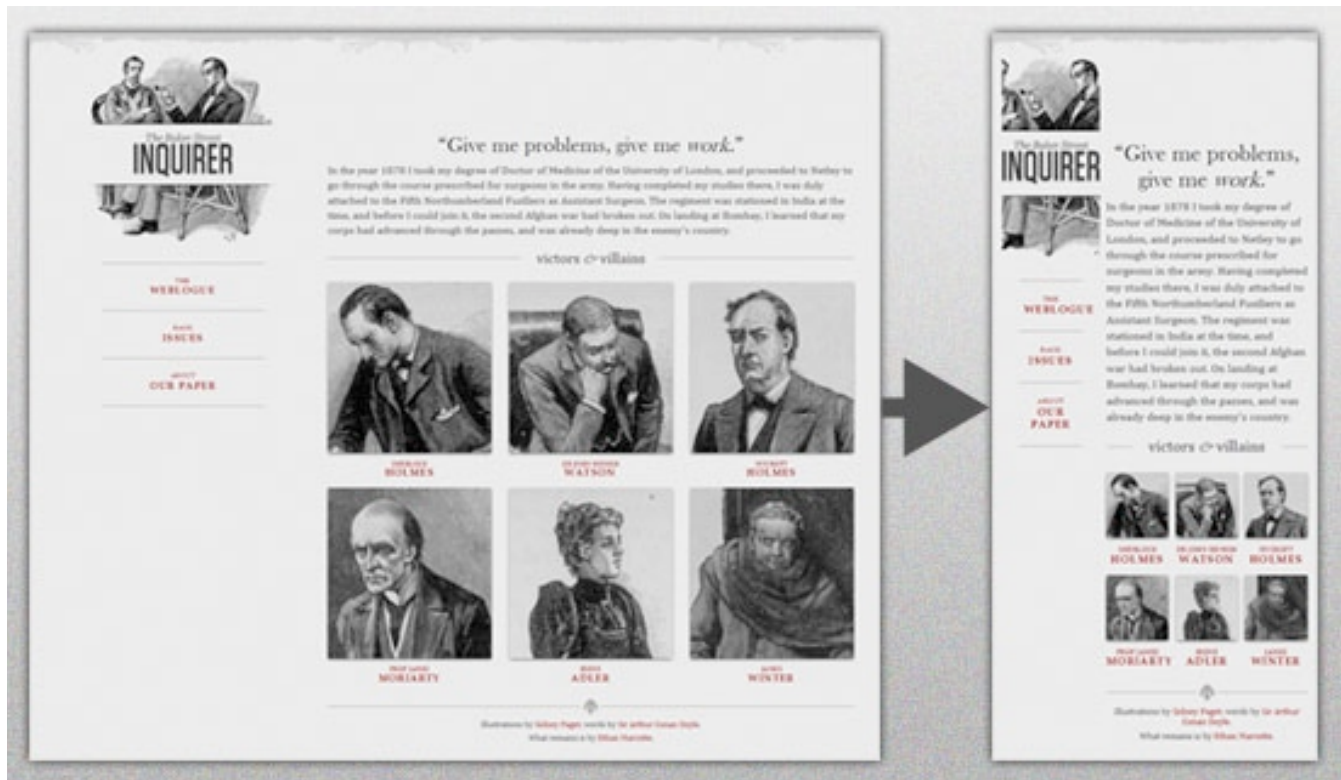
www.alistapart.com

The entire design is a lovely mix of [fluid grids](#), [fluid images](#) and smart mark-up where needed. Creating fluid grids is fairly common practice, and there are a number of techniques for creating fluid images:

- [Hiding and Revealing Portions of Images](#)

- [Creating Sliding Composite Images](#)

- [Foreground Images That Scale With the Layout](#)

For more information on creating fluid websites, be sure to look at the book "Flexible Web Design: Creating Liquid and Elastic Layouts with CSS" by Zoe Mickley Gillenwater, and download the sample chapter "[Creating Flexible Images](#)." In addition, Zoe provides the following extensive list of

tutorials, resources, inspiration and best practices on creating flexible grids and layouts: "Essential Resources for Creating Liquid and Elastic Layouts."

While from a technical perspective this is all easily possible, it's not just about plugging these features in and being done. Look at the logo in this design, for example:
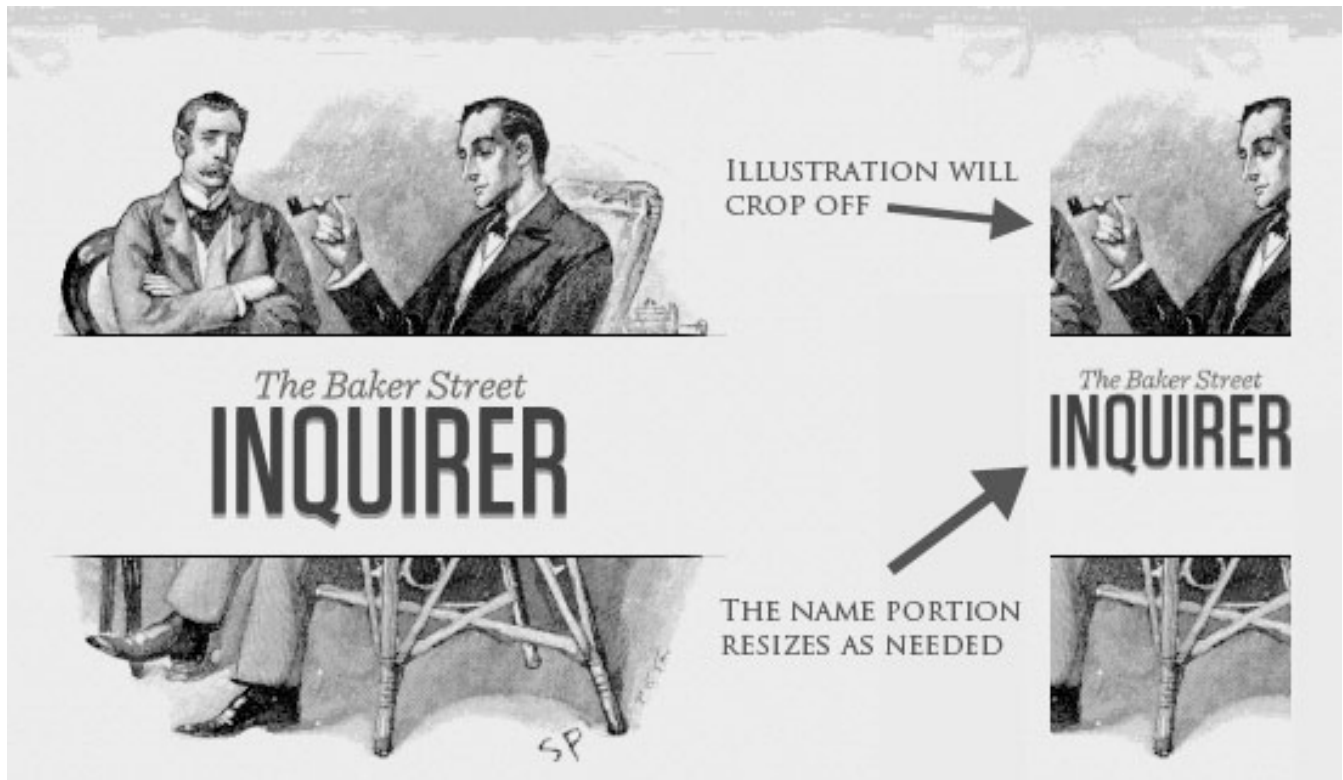
www.alistapart.com

If resized too small, the image would appear to be of low quality, but keeping the name of the website visible and not cropping it off was important. So, the image is divided into two: one (of the illustration) set as a background, to be cropped and to maintain its size, and the other (of the name) resized proportionally.

```
1 <h1 id="logo"><a href="#"><img src="site/logo.png" alt="The
  Baker Street Inquirer" /></a></h1>
```

Above, the `h1` element holds the illustration as a background, and the image is aligned according to the container's background (the heading).

This is just one example of the kind of thinking that makes responsive Web design truly effective. But even with smart fixes like this, a layout can become too narrow or short to look right. In the logo example above (although it works), the ideal situation would be to not crop half of the illustration or to keep the logo from being so small that it becomes illegible and "floats" up.

Flexible Images

One major problem that needs to be solved with responsive Web design is working with images. There are a number of techniques to resize images proportionately, and many are easily done. The most popular option, noted in Ethan Marcotte's article on [fluid images](#) but first experimented with by [Richard Rutter](#), is to use CSS's `max-width` for an easy fix.

```
1 img { max-width: 100%; }
```

As long as no other width-based image styles override this rule, every image will load in its original size, unless the viewing area becomes narrower than the image's original width. The **maximum width** of the image is set to 100% of the screen or browser width, so when that 100% becomes narrower, so does the image. Essentially, as Jason Grigsby [noted](#),:

> *"The idea behind fluid images is that you deliver images at the maximum size they will be used at. You don't declare the height and width in your code, but instead let the browser resize the images as needed while using CSS to guide their relative size." It's a great and simple technique to resize images beautifully.*

Note that `max-width` is **not supported in IE**, but a good use of `width:` `100%` would solve the problem neatly in an IE-specific style sheet. One more issue is that when an image is resized too small in some older browsers in Windows, the rendering isn't as clear as it ought to be. There is a JavaScript to fix this issue, though, found in <u>Ethan Marcotte's article</u>.

While the above is a great quick fix and good start to responsive images, image resolution and download times should be the primary considerations. While resizing an image for mobile devices can be very simple, if the original image size is meant for large devices, it could significantly slow download times and take up space unnecessarily.

Filament Group's Responsive Images

This technique, presented by the Filament Group, takes this issue into consideration and not only resizes images proportionately, but shrinks image resolution on smaller devices, so very large images don't waste space unnecessarily on small screens. Check out <u>the demo page here.</u>
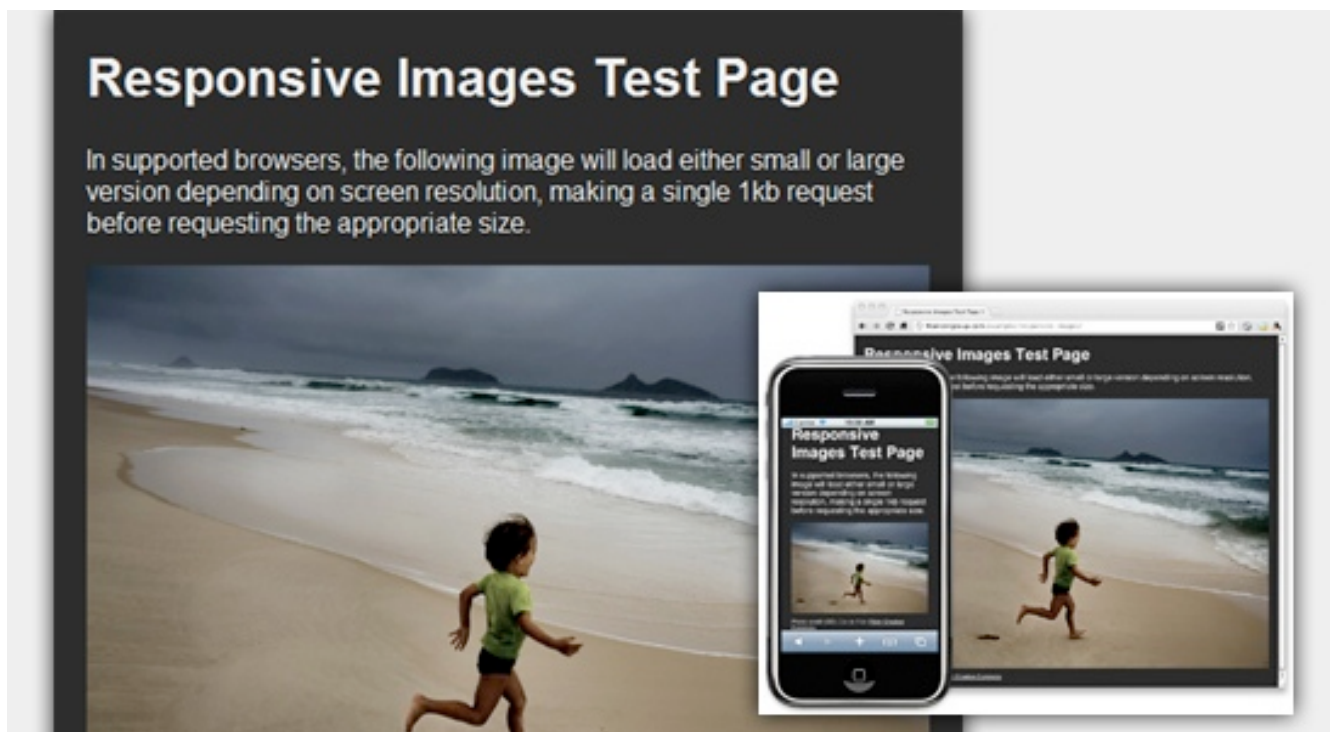
filamentgroup.com

This technique requires a few files, all of which are available on [Github](http://github.com). First, a JavaScript file (*rwd-images.js*), the *.htaccess* file and an image file (*rwd.gif*). Then, we can use just a bit of HTML to reference both the larger and smaller resolution images: first, the small image, with a *.r* prefix to clarify that it should be responsive, and then a reference to the bigger image using `data-fullsrc`.

```
1  <img src="smallRes.jpg" data-fullsrc="largeRes.jpg">
```

The `data-fullsrc` is a custom HTML5 attribute, defined in the files linked to above. For any screen that is wider than 480 pixels, the larger-resolution image (*largeRes.jpg*) will load; smaller screens wouldn't need to load the bigger image, and so the smaller image (*smallRes.jpg*) will load.

The JavaScript file inserts a base element that allows the page to separate responsive images from others and redirects them as necessary. When the

page loads, all files are rewritten to their original forms, and only the large or small images are loaded as necessary. With other techniques, all higher-resolution images would have had to be downloaded, even if the larger versions would never be used. Particularly for websites with a lot of images, this technique can be a great saver of bandwidth and loading time.

This technique is fully supported in modern browsers, such as IE8+, Safari, Chrome and Opera, as well as mobile devices that use these same browsers (iPad, iPhone, etc.). Older browsers and Firefox degrade nicely and still resize as one would expect of a responsive image, except that both resolutions are downloaded together, so the end benefit of saving space with this technique is void.

Stop iPhone Simulator Image Resizing

One nice thing about the iPhone and iPod Touch is that Web designs automatically rescale to fit the tiny screen. A full-sized design, unless specified otherwise, would just shrink proportionally for the tiny browser, with no need for scrolling or a mobile version. Then, the user could easily zoom in and out as necessary.

There was, however, one issue this simulator created. When responsive Web design took off, many noticed that images were still changing proportionally with the page even if they were specifically made for (or could otherwise fit) the tiny screen. This in turn scaled down text and other elements.

Because this works only with Apple's simulator, we can use an Apple-specific meta tag to fix the problem, placing it *below* the website's `<head>` section. Thanks to Think Vitamin's article on image resizing, we have the meta tag below:

```
1 <meta name="viewport" content="width=device-width; initial-
  scale=1.0">
```

Setting the `initial-scale` to 1 overrides the default to resize images proportionally, while leaving them as is if their width is the same as the device's width (in either portrait or landscape mode). Apple's documentation has a lot more information on the [viewport meta tag](#).

Custom Layout Structure

For extreme size changes, we may want to change the layout altogether, either through a separate style sheet or, more efficiently, through a CSS media query. This does not have to be troublesome; most of the styles can remain the same, while specific style sheets can inherit these styles and move elements around with floats, widths, heights and so on.

For example, we could have one main style sheet (which would also be the default) that would define all of the main structural elements, such as `#wrapper`, `#content`, `#sidebar`, `#nav`, along with colors, backgrounds and typography. Default flexible widths and floats could also be defined.

If a style sheet made the layout too narrow, short, wide or tall, we could then detect that and switch to a new style sheet. This new child style sheet would adopt everything from the default style sheet and then just redefine the layout's structure.

Here is the **style.css (default) content:**

```
1 /* Default styles that will carry to the child style sheet */
2
3 html,body{
4   background...
```

```css
 5   font...
 6   color...
 7 }
 8
 9 h1,h2,h3{}
10 p, blockquote, pre, code, ol, ul{}
11
12 /* Structural elements */
13 #wrapper{
14   width: 80%;
15   margin: 0 auto;
16
17   background: #fff;
18   padding: 20px;
19 }
20
21 #content{
22   width: 54%;
23   float: left;
24   margin-right: 3%;
25 }
26
27 #sidebar-left{
28   width: 20%;
29   float: left;
30   margin-right: 3%;
31 }
32
33 #sidebar-right{
34   width: 20%;
35   float: left;
36 }
```

Here is the *mobile.css* **(child)** content:

```css
1 #wrapper{
2   width: 90%;
3 }
4
5 #content{
6   width: 100%;
7 }
8
9 #sidebar-left{
10   width: 100%;
11   clear: both;
12
13   /* Additional styling for our new layout */
14   border-top: 1px solid #ccc;
15   margin-top: 20px;
16 }
17
18 #sidebar-right{
19   width: 100%;
20   clear: both;
21
22   /* Additional styling for our new layout */
23   border-top: 1px solid #ccc;
24   margin-top: 20px;
25 }
```

Content

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi feugiat, nisl in lacinia egestas, leo est ultricies ipsum, nec posuere felis quam nec ligula. Aliquam erat volutpat. Vestibulum a dolor at nisl tristique pharetra. Maecenas convallis aliquam massa, et porttitor nisi hendrerit vel. Proin justo massa, facilisis nec suscipit in, aliquam vel neque. Pellentesque orci enim, convallis sed faucibus sit amet, semper vitae diam. Mauris lacus mauris, semper id mollis vitae, bibendum et metus. Proin tempor velit eu mi ultricies sed pulvina turpis ultricies. Integer egestas sodales porttitor.

Aenean quis metus arcu. Nam adipiscing suscipit nulla ut dapibus. Nullam at metus massa. Suspendisse id lectus vitae nisi congue hendrerit vitae sed dolor. Aliquam erat volutpat. Aenean elementum ante at nibh ornare sit amet volutpat massa aliquet. Nam gravida fringilla elementum. Nulla facilisi. Proin tellus quam, tincidunt in aliquet sed, lobortis a risus. Vestibulum in mi augue. Suspendisse potenti. Donec sit amet condimentum leo.

Left Sidebar

Nulla in nisl eros. Donec fermentum urna ut nibh rutrum non aliquet purus euismod. Praesent tempor gravida massa eget vulputate. Morbi lobortis tincidunt ante sit amet placerat. Fusce leo nisl, pulvinar suscipit molestie sed, lobortis quis purus. Quisque porta enim eu ante consequat varius. Curabitur sit amet arcu ante. Etiam quis fermentum risus. Sed id volutpat elit. Vivamus molestie dictum nisl, in venenatis nulla malesuada

Right Sidebar

Nulla in nisl eros. Donec fermentum urna ut nibh rutrum non aliquet purus euismod. Praesent tempor gravida massa eget vulputate. Morbi lobortis tincidunt ante sit amet placerat. Fusce leo nisi, pulvinar suscipit molestie sed, lobortis quis purus. Quisque porta enim eu ante consequat varius. Curabitur sit amet arcu ante. Etiam quis fermentum risus. Sed id volutpat elit. Vivamus molestie dictum nisl, in venenatis nulla malesuada

Content

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi feugiat, nisl in lacinia egestas, leo est ultricies ipsum, nec posuere felis quam nec ligula. Aliquam erat volutpat. Vestibulum a dolor at nisl tristique pharetra. Maecenas convallis aliquam massa, et porttitor nisi hendrerit vel. Proin justo massa, facilisis nec suscipit in, aliquam vel neque. Pellentesque orci enim, convallis sed faucibus sit amet, semper vitae diam. Mauris lacus mauris, semper id mollis vitae, bibendum et metus. Proin tempor velit eu mi ultricies sed pulvinar turpis ultricies. Integer egestas sodales porttitor.

Aenean quis metus arcu. Nam adipiscing suscipit nulla ut dapibus. Nullam at metus massa. Suspendisse id lectus vitae nisi congue hendrerit vitae sed dolor. Aliquam erat volutpat. Aenean elementum ante at nibh ornare sit amet volutpat massa aliquet. Nam gravida fringilla elementum. Nulla facilisi. Proin tellus quam, tincidunt in aliquet sed, lobortis a risus. Vestibulum in mi augue. Suspendisse potenti. Donec sit amet condimentum leo.

Left Sidebar

Nulla in nisl eros. Donec fermentum urna ut nibh rutrum non aliquet purus euismod. Praesent tempor gravida massa eget vulputate. Morbi lobortis tincidunt ante sit amet placerat. Fusce leo nisi, pulvinar suscipit molestie sed, lobortis quis purus. Quisque porta enim eu ante consequat varius. Curabitur sit amet arcu ante. Etiam quis fermentum risus. Sed id volutpat elit. Vivamus molestie dictum nisl, in venenatis nulla malesuada vitae.

Right Sidebar

Nulla in nisl eros. Donec fermentum urna ut nibh rutrum non aliquet purus euismod. Praesent tempor gravida massa eget vulputate. Morbi lobortis tincidunt ante sit amet placerat. Fusce leo nisi, pulvinar suscipit molestie sed, lobortis quis purus. Quisque porta enim eu ante consequat varius. Curabitur sit amet arcu ante. Etiam quis fermentum risus. Sed id volutpat elit. Vivamus molestie dictum nisl, in venenatis nulla malesuada vitae.

ON A WIDER SCREEN, THE LEFT AND RIGHT SIDEBAR CONTENT FIT NICELY TO THE SIDE. FOR THINNER SCREENS, WE MOVE THIS CONTENT BELOW FOR BETTER USABILITY.

Media Queries

CSS3 supports all of the same media types as CSS 2.1, such as `screen`, `print` and `handheld`, but has added dozens of new media features, including `max-width`, `device-width`, `orientation` and `color`. New devices made after the release of CSS3 (such as the iPad and Android devices) will definitely support media features. So, calling a media query using CSS3 features to target these devices would work just fine, and it will be ignored if accessed by an older computer browser that does not support CSS3.

In Ethan Marcotte's article, we see an example of a media query in action:

```
1 <link rel="stylesheet" type="text/css"
2   media="screen and (max-device-width: 480px)"
3   href="shetland.css" />
```

This media query is fairly self-explanatory: if the browser displays this page on a screen (rather than print, etc.), and if the width of the screen (not necessarily the viewport) is 480 pixels or less, then load *shetland.css*.

New CSS3 features also include `orientation` (portrait vs. landscape), `device-width`, `min-device-width` and more. Look at "The Orientation Media Query" for more information on setting and restricting widths based on these media query features.

One can create multiple style sheets, as well as basic layout alterations defined to fit ranges of widths — even for landscape vs. portrait orientations. Be sure to look at the section of Ethan Marcotte's article entitled "Meet the media query" for more examples and a more thorough explanation.

Multiple media queries can also be dropped right into a single style sheet, which is the most efficient option when used:

```
 1 /* Smartphones (portrait and landscape) ---------- */
 2 @media only screen
 3 and (min-device-width : 320px)
 4 and (max-device-width : 480px) {
 5 /* Styles */
 6 }
 7
 8 /* Smartphones (landscape) ---------- */
 9 @media only screen
10 and (min-width : 321px) {
11 /* Styles */
12 }
13
14 /* Smartphones (portrait) ---------- */
15 @media only screen
16 and (max-width : 320px) {
17 /* Styles */
18 }
```

The code above is from a free template for multiple media queries between popular devices by Andy Clark. See the differences between this approach and including different style sheet files in the mark-up as shown in the post "Hardboiled CSS3 Media Queries."

CSS3 Media Queries

Above are a few examples of how media queries, both from CSS 2.1 and CSS3 could work. Let's now look at some specific how-to's for using CSS3 media queries to create responsive Web designs. Many of these uses are relevant today, and all will definitely be usable in the near future.

The **min-width and max-width** properties do exactly what they suggest. The `min-width` property sets a minimum browser or screen width that a certain set of styles (or separate style sheet) would apply to. If anything is below this limit, the style sheet link or styles will be ignored. The `max-width` property does just the opposite. Anything above the maximum browser or screen width specified would not apply to the respective media query.

Note in the examples below that we're using the syntax for media queries that could be used all in one style sheet. As mentioned above, the most efficient way to use media queries is to place them all in one CSS style sheet, with the rest of the styles for the website. This way, multiple requests don't have to be made for multiple style sheets.

```
1 @media screen and (min-width: 600px) {
2   .hereIsMyClass {
3     width: 30%;
4     float: right;
5   }
6 }
```

The class specified in the media query above (`hereIsMyClass`) will work only if the browser or screen width is above 600 pixels. In other words, this media query will run only if the **minimum width is 600 pixels** (therefore, 600 pixels or wider).

```
1 @media screen and (max-width: 600px) {
2   .aClassforSmallScreens {
3     clear: both;
4     font-size: 1.3em;
5   }
6 }
```

Now, with the use of `max-width`, this media query will apply only to browser or screen widths with a maximum width of 600 pixels or narrower.

While the above `min-width` and `max-width` can apply to either screen size or browser width, sometimes we'd like a media query that is relevant to device width specifically. This means that even if a browser or other viewing area is minimized to something smaller, the media query would still apply to the size of the actual device. The **min-device-width and max-device-width** media query properties are great for targeting certain devices with set dimensions, without applying the same styles to other screen sizes in a browser that mimics the device's size.

```
1 @media screen and (max-device-width: 480px) {
2   .classForiPhoneDisplay {
3     font-size: 1.2em;
4   }
5 }
```

```
1 @media screen and (min-device-width: 768px) {
2   .minimumiPadWidth {
3     clear: both;
4     margin-bottom: 2px solid #ccc;
5   }
6 }
```

There are also other tricks with media queries to target specific devices. Thomas Maier has written two short snippets and explanations for targeting the iPhone and iPad only:

- CSS for iPhone 4 (Retina display)

- How To: CSS for the iPad

For the iPad specifically, there is also a media query property called **orientation**. The value can be either landscape (horizontal orientation) or portrait (vertical orientation).

```
1 @media screen and (orientation: landscape) {
2   .iPadLandscape {
3     width: 30%;
4     float: right;
5   }
6 }
```

```
1 @media screen and (orientation: portrait) {
2   .iPadPortrait {
3     clear: both;
4   }
5 }
```

Unfortunately, this property works only on the iPad. When [determining the orientation for the iPhone](#) and other devices, the use of `max-device-width` and `min-device-width` should do the trick.

There are also many media queries that **make sense when combined**. For example, the `min-width` and `max-width` media queries are combined all the time to set a style specific to a certain range.

```
1 @media screen and (min-width: 800px) and (max-width: 1200px) {
2   .classForaMediumScreen {
3     background: #cc0000;
4     width: 30%;
5     float: right;
6   }
7 }
```

The above code in this media query applies only to screen and browser widths between 800 and 1200 pixels. A good use of this technique is to show certain content or entire sidebars in a layout depending on how much horizontal space is available.

Some designers would also prefer to **link to a separate style sheet** for certain media queries, which is perfectly fine if the organizational benefits outweigh the efficiency lost. For devices that do not switch orientation or for screens whose browser width cannot be changed manually, using a separate style sheet should be fine.

You might want, for example, to place media queries all in one style sheet (as above) for devices like the iPad. Because such a device can switch from portrait to landscape in an instant, if these two media queries were placed in separate style sheets, the website would have to call each style sheet file every time the user switched orientations. Placing a media query for both the horizontal and vertical orientations of the iPad in the same style sheet file would be far more efficient.

Another example is a flexible design meant for a standard computer screen with a resizable browser. If the browser can be manually resized, placing all variable media queries in one style sheet would be best.

Nevertheless, organization can be key, and a designer may wish to define media queries in a standard HTML link tag:

```
1 <link rel="stylesheet" media="screen and (max-width: 600px)"
  href="small.css" />
2 <link rel="stylesheet" media="screen and (min-width: 600px)"
  href="large.css" />
3 <link rel="stylesheet" media="print" href="print.css" />
```

JavaScript

Another method that can be used is JavaScript, especially as a back-up to devices that don't support all of the CSS3 media query options. Fortunately, there is already a pre-made JavaScript library that makes older browsers (IE 5+, Firefox 1+, Safari 2) support CSS3 media queries. If you're already using these queries, just grab a copy of the library, and include it in the mark-up: *css3-mediaqueries.js*.

In addition, below is a sample jQuery snippet that detects browser width and changes the style sheet accordingly — if one prefers a more hands-on approach:

```
1  <script type="text/javascript" src="http://ajax.googleapis.com/
   ajax/libs/jquery/1.4.4/jquery.min.js "></script>
2
3  <script type="text/javascript">
4    $(document).ready(function(){
5      $(window).bind("resize", resizeWindow);
6      function resizeWindow(e){
7        var newWindowWidth = $(window).width();
8
9        // If width is below 600px, switch to the mobile
   stylesheet
10       if(newWindowWidth < 600){                     $
   ("link[rel=stylesheet]").attr({href :
   "mobile.css"});                   }            // Else if width is
   above 600px, switch to the large stylesheet else
   if(newWindowWidth > 600){
11         $("link[rel=stylesheet]").attr({href : "style.css"});
12       }
13    }
```

```
14   });
15 </script>
```

There are many solutions for pairing up JavaScript with CSS media queries. Remember that media queries are not an absolute answer, but rather are fantastic options for responsive Web design when it comes to pure CSS-based solutions. With the addition of JavaScript, we can accommodate far more variations. For detailed information on using JavaScript to mimic or work with media queries, look at "[Combining Media Queries and JavaScript](#)."

Showing or Hiding Content

It is possible to shrink things proportionally and rearrange elements as necessary to make everything fit (reasonably well) as a screen gets smaller. It's great that that's possible, but making every piece of content from a large screen available on a smaller screen or mobile device isn't always the best answer. We have best practices for mobile environments: simpler navigation, more focused content, lists or rows instead of multiple columns.

Responsive Web design shouldn't be just about how to create a flexible layout on a wide range of platforms and screen sizes. It should also be about the user being able to pick and choose content. Fortunately, CSS has been allowing us to show and hide content with ease for years!

```
1 display: none;
```

Either declare `display: none` for the HTML block element that needs to be hidden in a specific style sheet or detect the browser width and do it through JavaScript. In addition to hiding content on smaller screens, we can also hide content in our default style sheet (for bigger screens) that should

be available only in mobile versions or on smaller devices. For example, as we hide major pieces of content, we could replace them with navigation to that content, or with a different navigation structure altogether.

Note that we haven't used `visibility: hidden` here; this just hides the content (although it is still there), whereas the `display` property gets rid of it altogether. For smaller devices, there is no need to keep the mark-up on the page — it just takes up resources and might even cause unnecessary scrolling or break the layout.

CONTENT DISPLAYED ON THE WIDER SCREEN IS HIDDEN WHEN MINIMIZED, AND THEN REPLACED WITH LINKS.
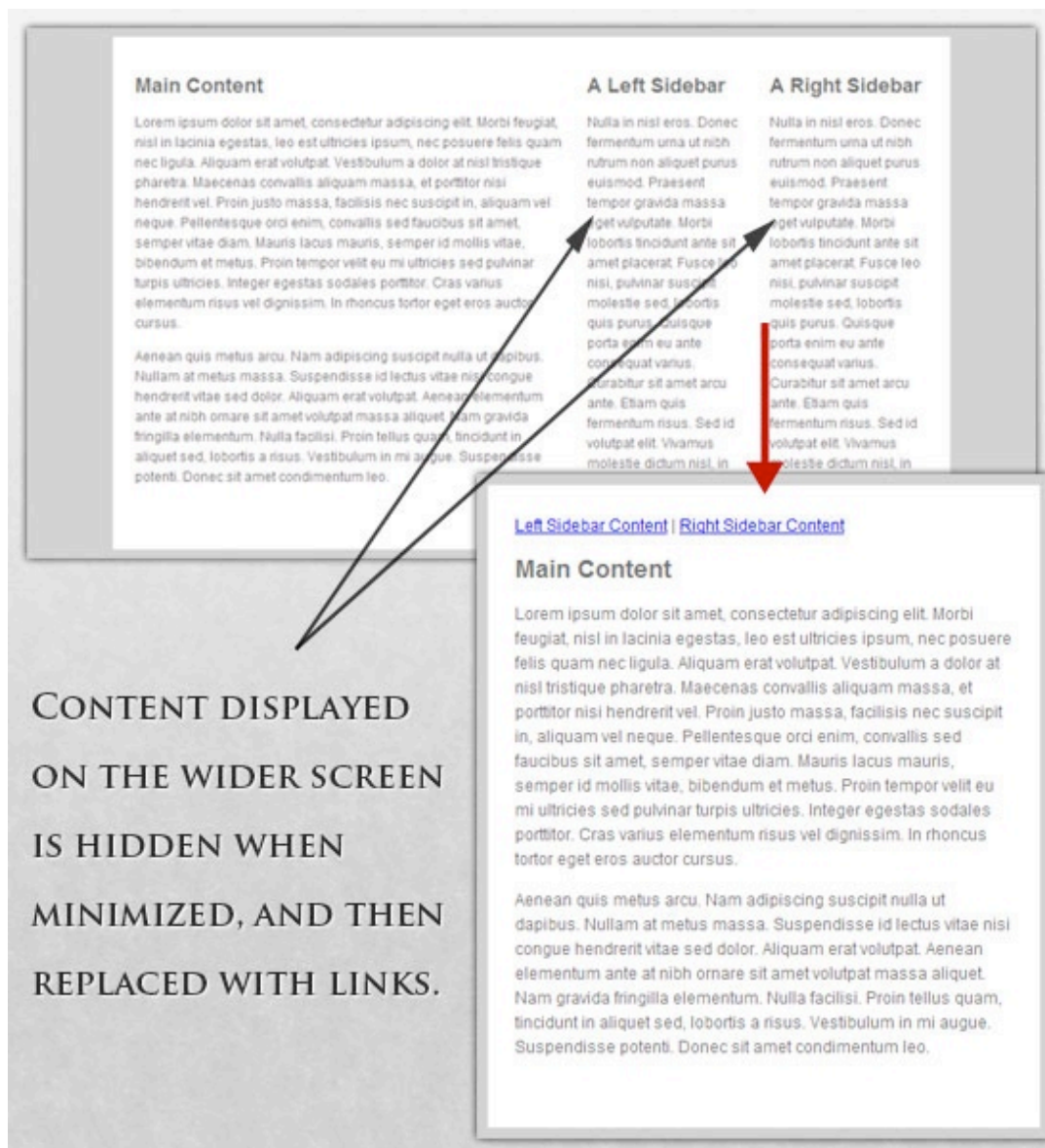
Here is **our mark-up:**

```
 1 <p class="sidebar-nav"><a href="#">Left Sidebar Content</a> | <a
   href="#">Right Sidebar Content</a></p>
 2
 3 <div id="content">
 4   <h2>Main Content</h2>
 5 </div>
 6
 7 <div id="sidebar-left">
 8   <h2>A Left Sidebar</h2>
 9
10 </div>
11
12 <div id="sidebar-right">
13   <h2>A Right Sidebar</h2>
14 </div>
```

In our default style sheet below, we have hidden the links to the sidebar content. Because our screen is large enough, we can display this content on page load.

Here is the *style.css* **(default)** content:

```
 1 #content{
 2   width: 54%;
 3   float: left;
 4   margin-right: 3%;
 5 }
 6
 7 #sidebar-left{
 8   width: 20%;
 9   float: left;
```

```
10    margin-right: 3%;
11 }
12
13 #sidebar-right{
14    width: 20%;
15    float: left;
16 }
17 .sidebar-nav{display: none;}
```

Now, we hide the two sidebars (below) and show the links to these pieces of content. As an alternative, the links could call to JavaScript to just cancel out the `display: none` when clicked, and the sidebars could be realigned in the CSS to float below the content (or in another reasonable way).

Here is the ***mobile.css* (simpler)** content:

```
1 #content{
2    width: 100%;
3 }
4
5 #sidebar-left{
6    display: none;
7 }
8
9 #sidebar-right{
10    display: none;
11 }
12 .sidebar-nav{display: inline;}
```

With the ability to easily show and hide content, rearrange layout elements and automatically resize images, form elements and more, a design can be transformed to fit a huge variety of screen sizes and device types. As the

screen gets smaller, rearrange elements to fit mobile guidelines; for example, use a script or alternate style sheet to increase white space or to replace image navigation sources on mobile devices for better usability (icons would be more beneficial on smaller screens).

Touchscreens vs. Cursors

Touchscreens are becoming increasingly popular. Assuming that smaller devices are more likely to be given touchscreen functionality is easy, but don't be so quick. Right now touchscreens are mainly on smaller devices, but many laptops and desktops on the market also have touchscreen capability. For example, the *HP Touchsmart tm2t* is a basic touchscreen laptop with traditional keyboard and mouse that can transform into a tablet.

Touchscreens obviously come with different design guidelines than purely cursor-based interaction, and the two have different capabilities as well. Fortunately, making a design work for both doesn't take a lot of effort. Touchscreens have no capability to display CSS hovers because there is no cursor; once the user touches the screen, they click. So, don't rely on CSS hovers for link definition; they should be considered an additional feature only for cursor-based devices.

Look at the article "Designing for Touchscreen" for more ideas. Many of the design suggestions in it are best for touchscreens, but they would not necessarily impair cursor-based usability either. For example, sub-navigation on the right side of the page would be more user-friendly for touchscreen users, because most people are right-handed; they would therefore not bump or brush the navigation accidentally when holding the device in their left hand. This would make no difference to cursor users, so

we might as well follow the touchscreen design guideline in this instance. Many more guidelines of this kind can be drawn from touchscreen-based usability.

A Showcase Of Responsive Web Design

Kayla Knight

Below we have a few examples of responsive Web design in practice today. For many of these websites, there is more variation in structure and style than is shown in the pairs of screenshots provided. Many have several solutions for a variety of browsers, and some even adjust elements dynamically in size without the need for specific browser dimensions. Visit each of these, and adjust your browser size or change devices to see them in action.

[Art Equals Work](#)
Art Equals Work is a simple yet great example of responsive Web design. The first screenshot below is the view from a standard computer screen dimension. The website is flexible with browser widths by traditional standards, but once the browser gets too narrow or is otherwise switched to a device with a smaller screen, then the layout switches to a more readable and user-friendly format. The sidebar disappears, navigation goes to the top, and text is enlarged for easy and simple vertical reading.

How to Become an Expert

a post on Critical Thinking

I have been aspiring to be an expert for just under 30 years now, creeping ever forward with each bit of knowledge and experience. "Experts" are people with efficient answers and deep explanations. Malcolm Gladwell calls them "mavens" in the *The Tipping Point*, and correctly observes that one cannot just amass an expertise – a maven must compulsively share.

My 3 Rules of Expertise are...

The Form of the Book, Digested

a post on Typography

The Form of the Book, by Jan Tschichold, is the authority on

2/17/11 · 6:55PM

@nathansmith hmm... yes, you got me there. The rev. or dr. might suffice to narrow it down though ;)

follow nathan_ford

recently

How to Become an Expert

The Form of the Book, Digested

In Case of Zombie Apocalypse

WRITINGS   WORK   ABOUT

How to Become an Expert

a post on Critical Thinking

I have been aspiring to be an expert for just under 30 years now, creeping ever forward with each bit of knowledge and experience. "Experts" are

[Think Vitamin](#)

With Think Vitamin, we see a similar approach. When on a smaller screen or browser, the sidebar and top bar are removed, the navigation simplifies and moves directly above the content, as does the logo. The logo keeps its general look yet is modified for a more vertical orientation, with the tagline below the main icon. The white space around the content on larger screens is also more spacious and interesting, whereas it is simplified for practical purposes on smaller screens.
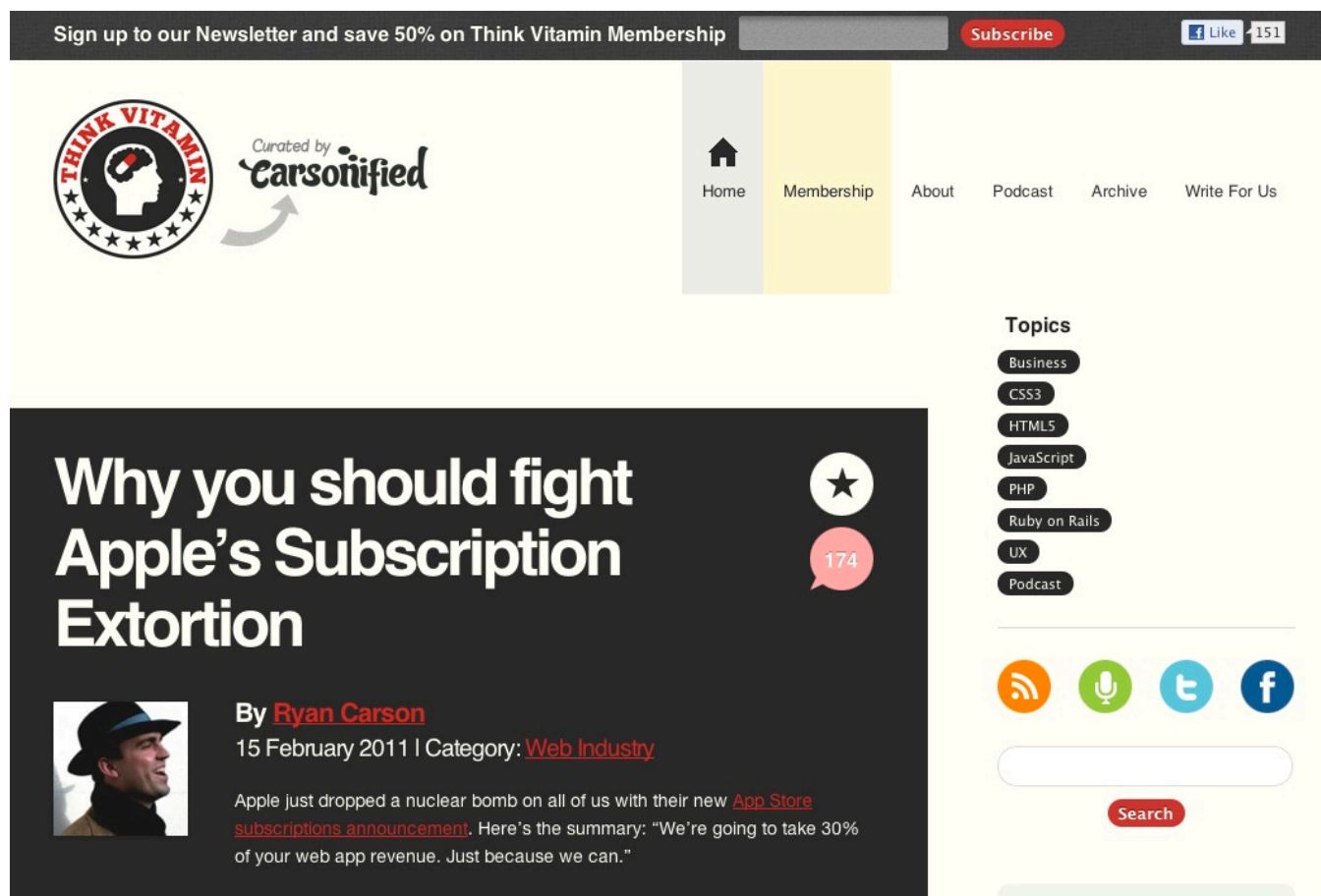
[8 Faces](#)

8 Faces' website design is flexible, right down to a standard netbook or tablet device, and expands in content quantity and layout width when viewed on wider screens or expanded browsers. When viewed on narrower screens, the featured issue on the right is cut out, and the content below is shortened and rearranged in layout, leaving only the essential information.

[Information Architects](#)

Here is a great example of a flexible image. The image in this design automatically resizes after certain "break" points, but in between those width changes, only the side margins and excess white space are altered. On smaller screens and minimized browsers, the navigation simplifies and the columns of navigation at the top fall off. At the design's smallest version, the navigation simplifies to just a drop-down menu, perfect for saving space without sacrificing critical navigation links.

November 18, 2010

News on iPad, the Obvious Way

Today our first news project for iPad went online and we are proud like kids. Technically, it's "just" an HTML5 optimization, but it has been a demanding design process to get to the point of simplicity where it's at right now. Unlike most news launches on iPad the response from its readers so far has been overwhelmingly positive. This does not come as a surprise: While many like our surface design—which, of course, makes us happy—, the main reason why people love our iPad design for ZEIT ONLINE is that it's optimized to run on the killer content app No1: The browser. What is somewhat astonishing is that, so far, only very few newspapers took the time to optimize their website. Read more

ABOUT   **KNOW HOW**   **STORE**

Profile   Articles   Overview

Projects   Speaking   Template

Contact   Tweets   Writer for iPad

➡ **iA**

November 18, 2010

News on iPad, the Obvious Way

Today our first news project for iPad went online and we are proud like kids. Technically, it's "just" an HTML5 optimization, but it has been a demanding design process to get to the point of simplicity where it's at right now. Unlike most news launches on iPad the response from its readers so far has been overwhelmingly positive. This does not come as a surprise: While many like our surface design—which, of course, makes us happy—, the main reason why

[Simon Collison](#)

With four relatively content-heavy columns, it's easy to see how the content here could easily be squished when viewed on smaller devices. Because of the easy organized columns, though, we can also collapse them quite simply when needed, and we can stack them vertically when the space doesn't allow for a reasonable horizontal span. When the browser is minimized or the user is on a smaller device, the columns first collapse into two and then into one. Likewise, the horizontal lines for break points also change in width, without changing the size or style of each line's title text.
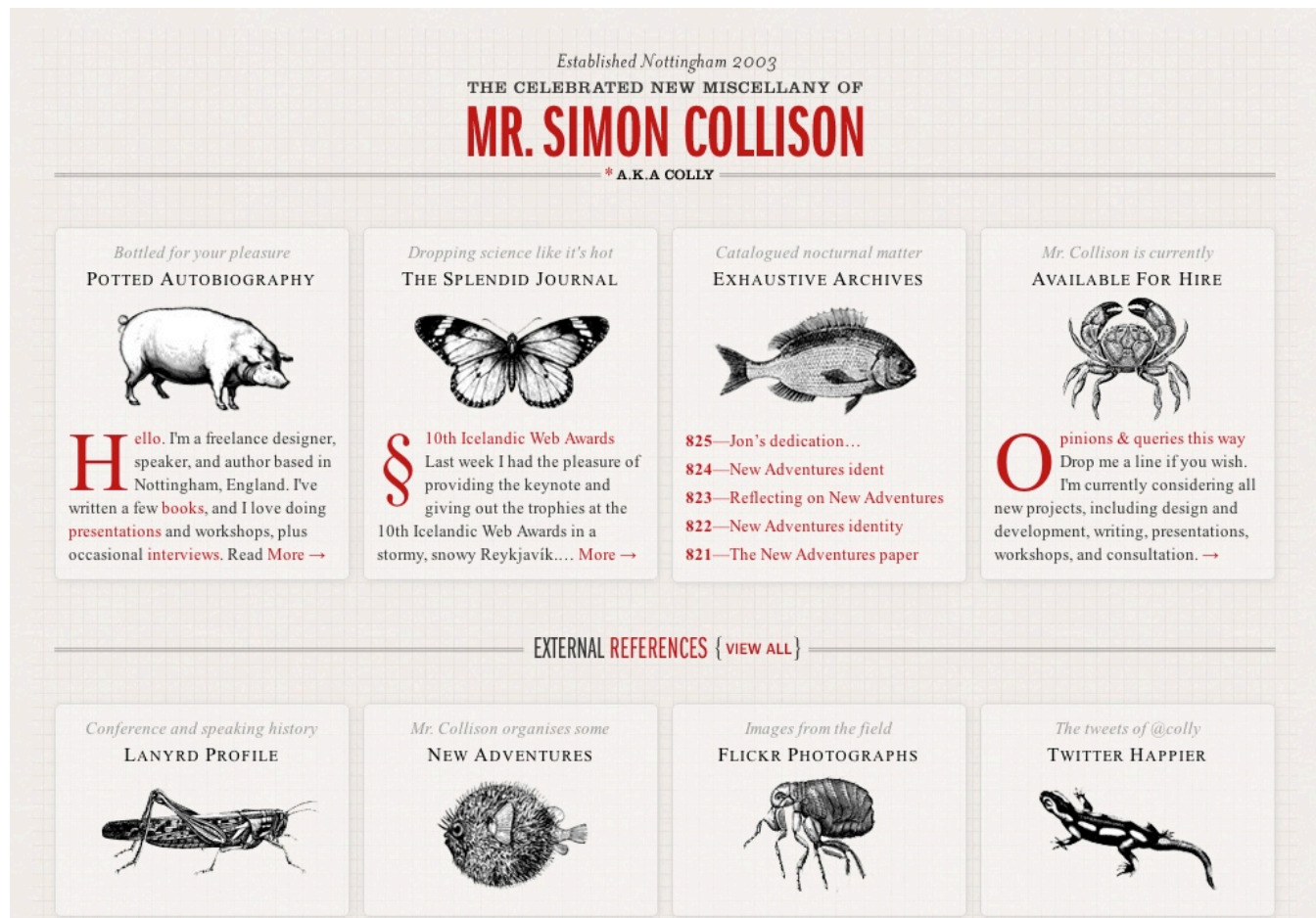
Established Nottingham 2003

THE CELEBRATED NEW MISCELLANY OF

MR. SIMON COLLISON

— * A.K.A COLLY —

Bottled for your pleasure

POTTED AUTOBIOGRAPHY

Hello. I'm a freelance designer, speaker, and author based in Nottingham, England. I've written a few books, and I love doing presentations and workshops, plus occasional interviews. Read More →

Dropping science like it's hot

THE SPLENDID JOURNAL

§ 10th Icelandic Web Awards Last week I had the pleasure of providing the keynote and giving out the trophies at the 10th Icelandic Web Awards in a stormy, snowy Reykjavík.... More →

Catalogued nocturnal matter

EXHAUSTIVE ARCHIVES

825—Jon's dedication…

824—New Adventures ident

823—Reflecting on New Adventures

822—New Adventures identity

821—The New Adventures paper

Mr. Collison is currently

AVAILABLE FOR HIRE

Opinions & queries this way Drop me a line if you wish. I'm currently considering all new projects, including design and development, writing, presentations, workshops, and consultation. →

CSS Tricks

On the CSS Tricks website, like many other collapsible Web designs, the sidebars with excess content are the first to fall off when the screen or browser gets too narrow. On this particular website, the middle column or first sidebar to the left was the first to disappear; and the sidebar with the ads and website extras did the same when the browser got even narrower. Eventually, the design leaves the posts, uses less white space around the navigation and logo and moves the search bar to below the navigation. The remaining layout and design is as flexible as can be because of its simplicity.

CSS-TRICKS*

A Web Design Community curated by Chris Coyier
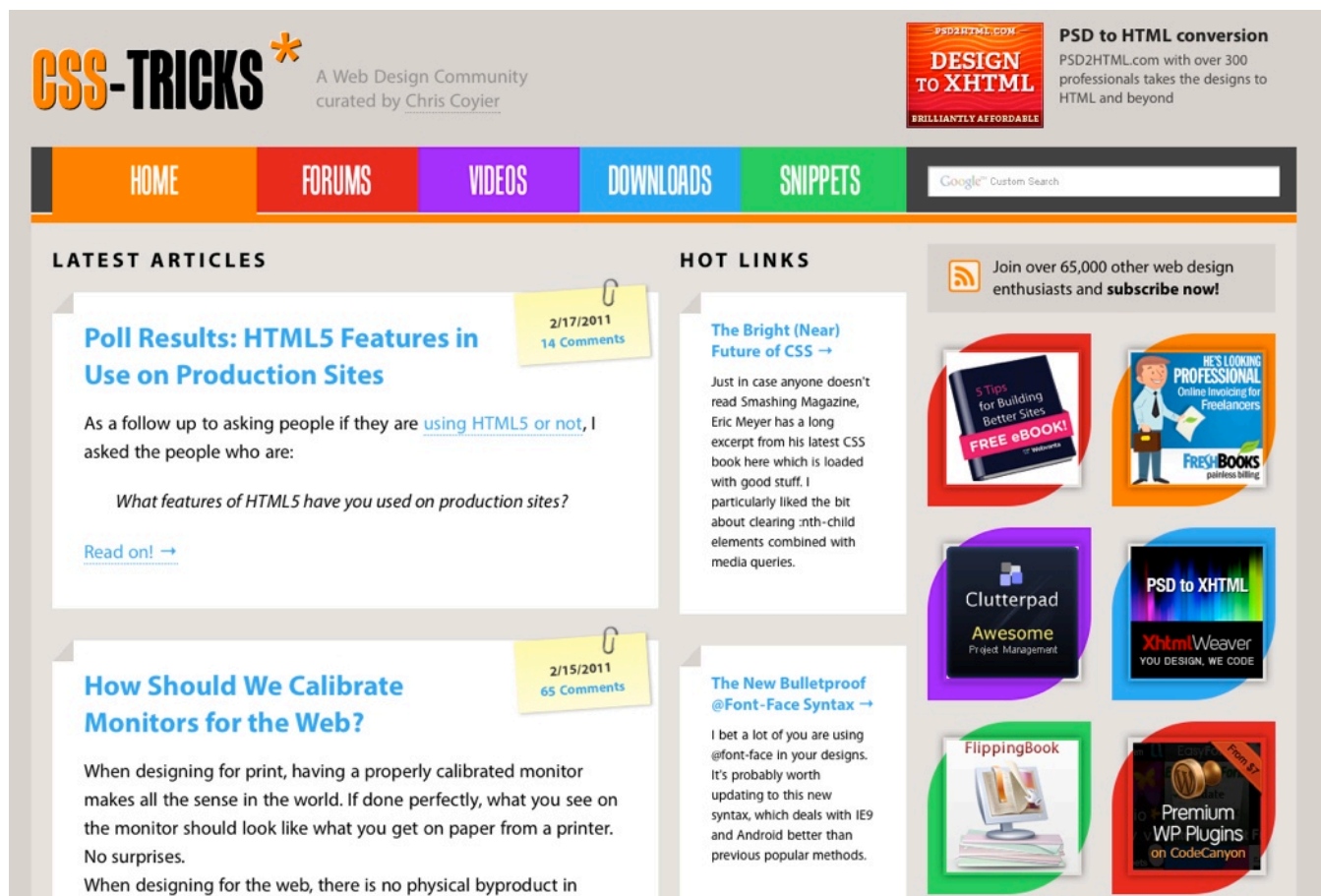
HOME **FORUMS** **VIDEOS** **DOWNLOADS** **SNIPPETS**

Google™ Custom Search

LATEST ARTICLES

Poll Results: HTML5 Features in Use on Production Sites

2/17/2011
14 Comments

As a follow up to asking people if they are using HTML5 or not, I asked the people who are:

What features of HTML5 have you used on production sites?

Read on! →

[Tee Gallery](#)

As one can see, the main navigation here is the simple layout of t-shirt designs, spanning both vertically and horizontally across the screen. As the browser or screen gets smaller, the columns collapse and move below. This happens at each break point when the layout is stressed, but in between the break points, the images just change proportionally in size. This maintains balance in the design, while ensuring that images (which are essential to the website) don't get so small that they become unusable.
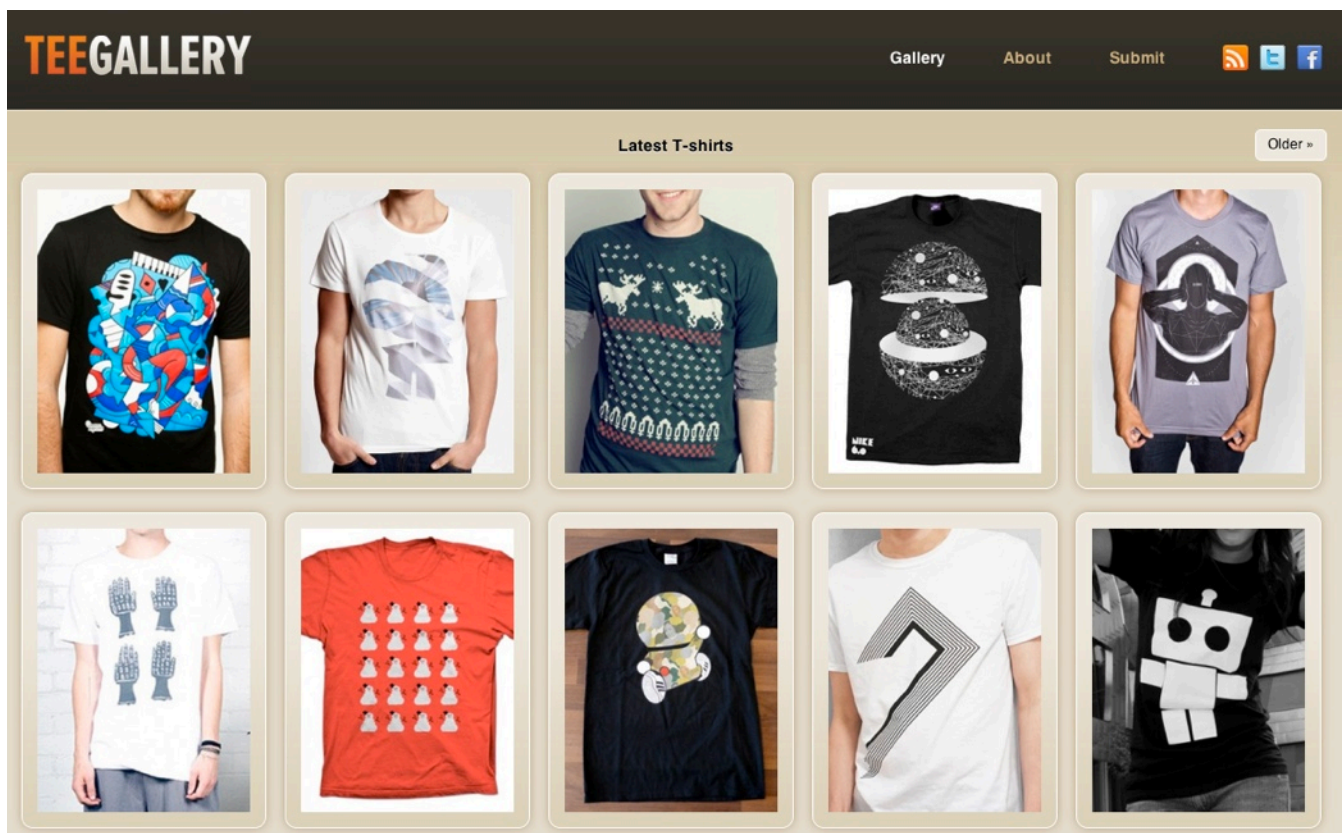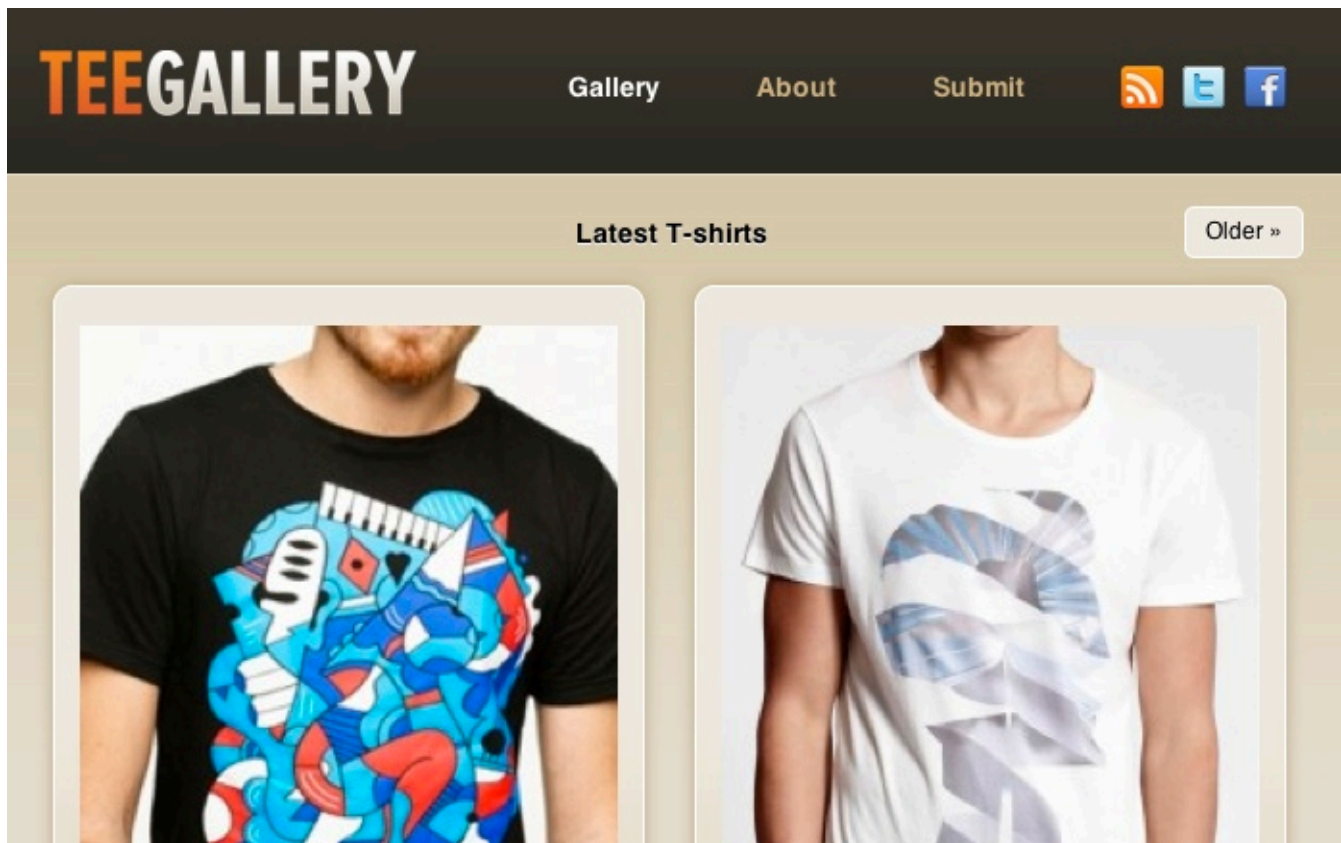
Ten by Twenty

Ten by Twenty is another design that does not resort to changing layout structure at all after certain break points, but rather simplifies responsive Web design by making everything fully flexible and automatically resizing, no matter what the screen or browser width. After a while, the design does stress a bit and could benefit from some rearrangement of content. But overall, the image resizing and flexible content spaces allow for a fairly simple solution that accommodates a wide range of screen sizes.

Teixido

This design features a complex layout that looks inspired by a print style. When viewed on a standard wide computer screen, more portfolio pieces are featured and spanned horizontally across the page. As one moves down the page, more graphics and imagery span the space. On a smaller screen,

the portfolio piece is cut down to one, and then eventually left out altogether for very small screens and narrow browsers. The visualizations below collapse into fewer columns and more rows, and again, some drop off entirely for very small screens. This design shows a creative and intelligent way to make a not-so-common layout work responsively.

[Stephen Caver](#)

This design has three main stages at which the design and layout collapse into a more user-friendly form, depending on how wide the screen or browser is. The main image (featuring type) is scaled proportionally via a flexible image method. Each "layout structure" is fully flexible until it reaches a breaking point, at which point the layout switches to something more usable with less horizontal space. The bottom four columns eventually collapse into two, the logo moves above the navigation, and the columns of navigation below are moved on top or below each other. At the design's narrowest stage, the navigation is super-simplified, and some inessential content is cut out altogether.

[Bryan James](#)

This last design by Bryan James shows that responsive Web design need not apply only to static HTML and CSS websites. Done in Flash, this one features a full-sized background image and is flexible up to a certain width and height. As a result of the design style, on screens that are too small, the background image gets mostly hidden and the content can become illegible and squished. Instead of just letting it be, though, a message pops up informing the user that the screen is too small to adequately view the website. It then prompts the user to switch to a bigger screen. One can discuss if the design solution is good or bad in terms of usability, but the example shows that Flash websites can respond to user's viewport, too.

CSS Wizardry

Harry Roberts shows that responsive design can also have quite humble uses. If the user has a large viewport, the website displays three columns with a navigation menu floating on the left. For users with a viewport between 481px and 800px, a narrow version is displayed: the navigation jumps to the top of the site leaving the area for the content column and the sidebar. Finally, the iPhone view displays the sidebar under the content area. Harry also wrote a detailed article about the CSS styles he added to the stylesheet in his article "Media queries, handier than you think". A nice example of how a couple of simple CSS adjustments can improve the website's appearance across various devices.

Hicksdesign

The Hicksdesign website has three columns when viewed on a conventional computer screen with a maximized browser. When minimized in width, the design takes on a new layout: the third column to the right is rearranged above the second, and the logo moves next to the introductory text. Thus, no content needs to be removed for the smaller size. For even narrower screens and browser widths, the side content is removed completely and a simplified version is moved up top. Finally, the font size changes with the screen and browser width; as the browser gets narrower, the font size throughout gets smaller and remains proportional.

Conclusion

We are indeed entering a new age of Web design and development. Far too many options are available now, and there will be far too many in the future to continue adjusting and creating custom solutions for each screen size, device and advancement in technology. We should rather start a new era today: creating websites that are future-ready right now. Understanding

how to make a design responsive to the user doesn't require too much learning, and it can definitely be a lot less stressful and more productive than learning how to design and code properly for every single device available.

Responsive Web design and the techniques discussed above are not the final answer to the ever-changing mobile world. Responsive Web design is a mere concept that when implemented correctly can *improve* the user experience, but not completely solve it for every user, device and platform. We will need to constantly work with new devices, resolutions and technologies to continually improve the user experience as technology evolves in the coming years.

Besides saving us from frustration, responsive Web design is also best for the user. Every custom solution makes for a better user experience. With responsive Web design, we can create custom solutions for a wider range of users, on a wider range of devices. A website can be tailored as well for someone on an old laptop or device as it can for the vast majority of people on the trendiest gadgets around, and likewise as much for the few users who own the most advanced gadgets now and in the years to come. Responsive Web design creates a great custom experience for everyone. As Web designers, we all strive for that every day on every project anyway, right?

Persuasion Triggers in Web Design

David Travis

How do you make decisions? If you're like most people, you'll probably answer that you pride yourself on weighing the pros and cons of a situation carefully and then make a decision based on logic. You know that other people have weak personalities and are easily swayed by their emotions, but this rarely happens to you.

You've just experienced the [fundamental attribution error](#) — the tendency to believe that other people's behavior is due to their personality ("Josh is late because he's a disorganized person") whereas our behavior is due to external circumstances ("I'm late because the directions were useless").

Cognitive biases like these play a significant role in the way we make decisions so it's not surprising that people are now examining these biases to see how to exploit them in the design of Web sites. I'm going to use the term 'persuasion architects' to describe designers who knowingly use these techniques to influence the behavior of users. (Many skilled designers already use some of these psychological techniques intuitively — but they wouldn't be able to articulate why they have made a particular design choice. The difference between these designers and persuasion architects is that persuasion architects use these techniques intentionally).

There are 7 main weapons of influence in the persuasion architect's arsenal:

- Reciprocation
- Commitment
- Social Proof

- Authority

- Scarcity

- Framing

- Salience

How do persuasion architects apply these principles to influence our behavior on the web?

Reciprocation

"I like to return favors."

This principle tells us that if we feel we have been done a favor, we will want to return it. If somebody gives you a gift, invites you to a party or does you a good turn, you feel obliged to do the same at some future date.

Persuasion architects exploit this principle by giving users small gifts — a sample chapter from a book, a regular newsletter or just useful information — in the knowledge that users will feel a commitment to offer something in return.
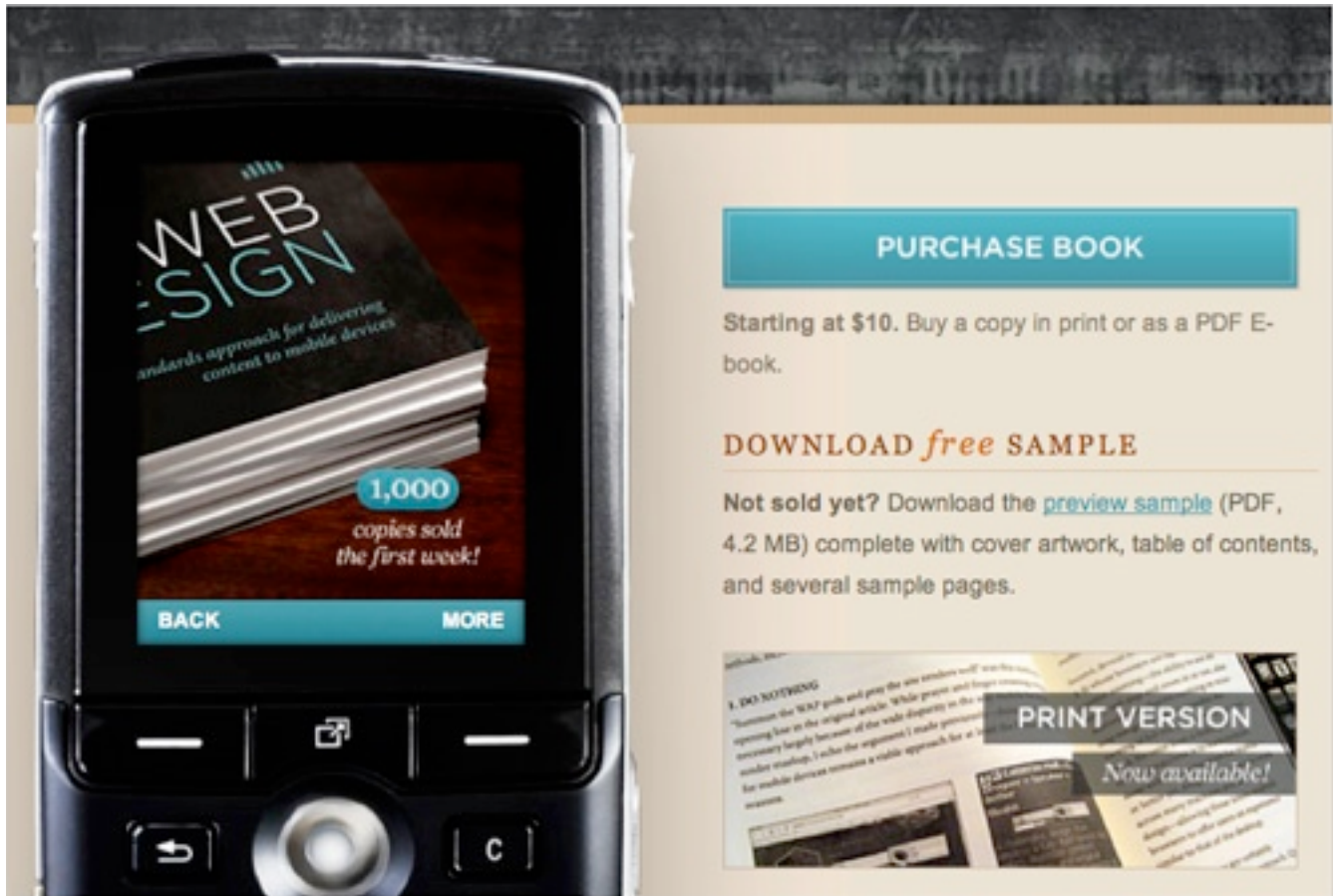
Book publishers offer free sample chapters in the hope that you'll reciprocate the favor and buy the book.

That 'something in return' need not be a purchase (not yet, anyway). Persuasion architects know that they need to contact prospective customers on several occasions before they become an actual customer — this is why regular newsletters are a staple offering in the persuasion architect's toolkit. So in return they may simply *ask* for a referral, or a link to a Web site, or a comment on a blog. And note the emphasis on 'ask'. Persuasion architects are not shy of asking for the favor that you 'owe' them.

> This ebook is available for free by visiting
> http://www.sethgodin.com. Click on my head to find my blog. If you bought it, you paid too much.
>
> In return, I'd consider it a mutual favor if you'd click here:
> http://feeds.feedburner.com/typepad/sethsmainblog
>
> and subscribe to the RSS feed of my blog. You get the latest on my doings, and I get to find you when I've got something neat to share. Like my new ebooks or the latest on my new secret project...

Seth Godin knows how to leverage the principle of reciprocation. This comes from one of Seth's free PDFs and you'll notice he's not shy of asking you to return the favor.

Commitment

"I like to do what I say."

This principle tells us that we like to believe that our behavior is consistent with our beliefs. Once you take a stand on something that is visible to other people, you suddenly feel a drive to maintain that point of view to appear reliable and constant.

A familiar example of this in action is when comments on a blog degrade into a flame war. Commentators are driven to justify their earlier comments and often become even more polarized in their positions.

Flamewars.net contains many examples of people justifying their commitment to comments they have made on a blog posting.

Persuasion architects apply this principle by asking for a relatively minor, but visible, commitment from you. They know that if they can get you to act in a particular way, you'll soon start believing it. For example, an organization may ask you to 'Like' one of their products on Facebook to watch a video or get access to particular content. Once this appears in your NewsFeed, you have made a public commitment to the product and feel more inclined to support it.

[Oxfam](#) uses the principle of commitment in the knowledge that a small change in behavior will lead to larger changes later on.

Social Proof

"I go with the flow."

This principle tells us that we like to observe other people's behavior to judge what's normal, and then we copy it.

Persuasion architects apply this principle by showing us what other people are doing on their Web sites. For example, [researchers at Columbia University](#) set up a Web site that asked people to listen to, rate and download songs by unsigned bands. Some people just saw the names of the songs and bands, while others — the "social influence" group — also saw how many times the songs had been downloaded by other people.

In this second group, the most popular songs were much more popular (and the least popular songs were less popular) than in the independent condition, showing that people's behavior was influenced by the crowd. Even more surprisingly, when they ran the experiment again, the particular songs that became "hits" were different, showing that social influence didn't just make the hits bigger but also made them more unpredictable.
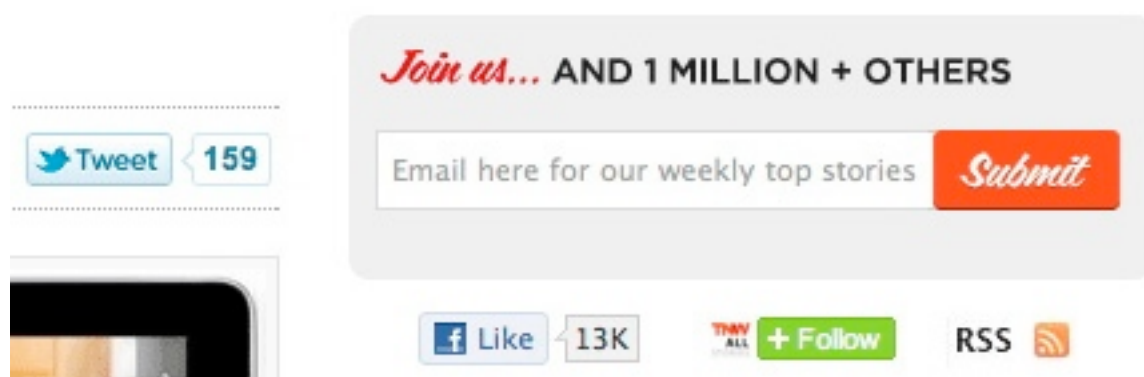
1 million people can't be wrong (from thenextweb.com).

Some familiar examples of social proof on the Web are, "People who shopped for this product also looked at..." feature and Amazon's, "What do customers ultimately buy after viewing this item?".

Persuasion architects also exploit this principle in the power of defaults. They know that the default setting of a user interface control has a powerful influence over people's behavior. We tend to see the default setting as a 'recommended' option — the option that most other people would choose in our situation. There are many examples of this being used as a black hat usability technique, where additional items (like insurance) are sneaked into the user's basket.

Authority

"I'm more likely to act on information if it's communicated by an expert."

This principle is about influencing behavior through credibility. People are more likely to take action if the message comes from a credible and authoritative source. That's why you'll hear people name dropping and it's also what drives retweets on Twitter.

A tweet from @smashingmag is likely to be retweeted because the brand has such authority.

For design guidance, we can turn to the [Stanford Persuasive Technology Lab](#) (founded by B.J. Fogg) as they have developed a number of guidelines for the credibility of Web sites. These guidelines are based on research with over 4,500 people and are based on peer-reviewed, scientific research. Thanks to their research, we know that you should highlight the expertise in your organization and in the content and services you provide; show that honest and trustworthy people stand behind your site; and avoid errors of all types, no matter how small they seem.

Persuasion architects exploit this principle by providing glowing testimonials on their Web site. If it's an e-commerce site they will have highly visible icons showing the site is secure and can be trusted. If the site includes a forum, they'll give people the opportunity to rate their peers: for example, some Web forums (like [Yahoo! Answers](#)) let users vote up (or down) answers to posted questions. The top ranked answer is then perceived to be the most authoritative.

UXExchange allows users to vote up and vote down answers to questions, ensuring that the most authoritative answer rises to the top.

Scarcity

"If it's running out, I want it."

This principle tells us that people are more likely to want something if they think it is available only for a limited time or if it is in short supply. Intriguingly, this isn't just about the fear of missing out (a kind of reverse social proof). Scarcity actually makes stuff appear more valuable. For example, psychologists have shown that if you give people a chocolate

biscuit from a jar, they rate the biscuit as more enjoyable if it comes from a jar with just 2 biscuits than from a jar with 10.

Persuasion architects exploit this by revealing scarcity in the design of the interface. This could be an item of clothing that is running short in your size, theatre tickets that are running out, or invitations to a beta launch. They know that perceived scarcity will generate demand.

Related to this is the 'closing down' sale. One of the artists at my friend's art co-op recently decided to quit the co-op and announced this with a sign in-store. She had a big rush on sales of her art. Then she decided not to quit after all. So pretending to go out of business might be a ploy!

Phrases like 'only 4 left in stock' seem to stimulate a primal urge not to miss out.

Framing

"I'm strongly influenced by the way prices are framed."

This principle acknowledges that people aren't very good at estimating the absolute value of what they are buying. People make comparisons, either against the alternatives you show them or some external benchmark.

One example is the way [a restaurant uses an "anchor" dish on its menu](): this is an overpriced dish whose sole aim is to make everything else near it look like a relative bargain. Another example is the [Goldilocks effect]() where you provide users with three alternative choices. However, two of the choices are decoys: one is an overpriced, gold plated version of your product; another is a barely functional base version. The third choice — the one you want people to choose — sits midway between the other two and so feels "just right."

Salience

"My attention is drawn to what's relevant to me right now."

This principle tells us that people are more likely to pay attention to elements in your user interface that are novel (such as a colored 'submit' button) and that are relevant to where there are in their task. For example, there are specific times during a purchase when shoppers are more likely to investigate a promotion or a special offer. By identifying these [seducible moments]() you'll learn when to offer a customer an accessory for a product they have bought.

| 1. Your basket | 2. Delivery/collection | 3. Your details |

Please review your Basket

Item		Item price	Home delivery
1	SONY KDL32EX301U 32" LCD TV ☒ Remove this item	£299.00 Quantity: 1 ▼	✔ Delivery from 1 to 5 days

Need help with installation or recycling?

3 Year ExtraCare Plan ⑦	+ £79.99	Add to order
5 Year ExtraCare Plan ⑦	+ £99.99	Add to order
Recycling service for old appliance ⑦	+ £9.95	Add to order
TV installation - 24" - 39" ⑦	+ £40.00	Add to order
TV Wall Mounting (does not include bracket) ⑦	+ £110.00	Add to order
Digital Aerial Installation to 1 TV point ⑦	+ £149.95	Add to order

Product total   £299.00   HOME DELIVERY ›

After placing an order for a TV at the Comet Web site, the designers encourage you to add other relevant items to your basket. This is exactly the right time to make the offer: once you've ordered the TV they remind you that you'll need to install it.

Why We Should Start Using CSS3 and HTML5 Today

Vitaly Friedman

For a while now, here on Smashing Magazine, we have taken notice of how many designers are reluctant to embrace the new technologies such as CSS3 or HTML5 because of the lack of full cross-browser support for these technologies. Many designers are complaining about the numerous ways how the lack of cross-browser compatibility is effectively holding us back and tying our hands — keeping us from completely being able to shine and show off the full scope of our abilities in our work. Many are holding on to the notion that once this push is made, we will wake to a whole new Web — full of exciting opportunities just waiting on the other side. So they wait for this day. When in reality, they are effectively waiting for Godot.

Just like the elusive character from Beckett's classic play, this day of full cross-browser support is not ever truly going to find its dawn and deliver us this wonderful new Web where our work looks the same within the window of any and *every* Web browser. Which means that many of us in the online reaches, from clients to designers to developers and on, are going to need to adjust our thinking so that we can realistically approach the Web as it is now, and more than likely how it will be in the future.

Sometimes it feels that we are hiding behind the lack of cross-browser compatibility to avoid learning new techniques that would actually dramatically improve our workflow. And that's just wrong. Without an adjustment, we will continue to undersell the Web we have, and the

landscape will remain unexcitingly stale and bound by this underestimation and mindset.

Adjustment in Progress

Sorry if any bubbles are bursting here, but we have to wake up to the fact that full cross-browser support of new technologies is just not going to happen. Some users will still use older browsers and some users will still have browsers with deactivated JavaScript or images; some users will be having weird view port sizes and some will not have certain plugins installed.

But that's OK, really.

The Web is a damn flexible medium, and rightly so. We should embrace its flexibility rather than trying to set boundaries for the available technologies in our mindset and in our designs. The earlier we start designing with the new technologies, the quicker their wide adoption will progress and the quicker we will get by the incompatibility caused by legacy browsers. More and more users are using more advanced browsers every single day, and by using new technologies, we actually encourage them to switch (if they can). Some users will not be able to upgrade, which is why our designs should have a basic fallback for older browsers, but it can't be the reason to design only the fallback version and call it a night.

Select[ivizr] is one of the many tools that make it possible to use CSS3 today.

There are so many remarkable things that we, designers and developers, can do today: be it responsive designs with CSS3 media queries, rich Web typography (with full support today!) or HTML5 video and audio. And there are so many useful tools and resources that we can use right away to incorporate new technologies in our designs while still supporting older browsers. There is just no reason *not* to use them.

We are the ones who can push the cross-browser support of these new technologies, encouraging and demanding the new features in future browsers. We have this power, and passing on it just because we don't feel

like there is no full support of them yet, should not be an option. We need to realize that we are the ones putting the wheels in motion and it's up to us to decide what will be supported in the future browsers and what will not.

More exciting things will be coming in the future. We should design for the future and we should design for today — making sure that our progressive designs work well in modern browsers and work fine in older browsers. The crucial mistake would be clinging to the past, trying to work with the old nasty hacks and workarounds that will become obsolete very soon.

We can continue to cling to this notion and wait for older browsers to become outdated, thereby selling ourselves and our potential short, or we can adjust our way of thinking about this and come at the Web from a whole new perspective. One where we understand the truth of the situation we are faced with. That our designs are not going to look the same in every browser and our code will not render the same in every browser. And that's the bottom line.

Yaili's beautiful piece [My CSS Wishlist on 24ways](#). Articles like these are the ones that push the boundaries of web design and encourage more innovation in the industry.

Andy Clarke spoke about this at the DIBI Conference earlier this year (you can check his presentation [Hardboiled Web Design on Vimeo](#)). He really struck a nerve with his presentation, yet still we find so many stalling in this dream of complete Web standardization. So we wanted to address this issue here and keep this important idea being discussed and circulated. Because this waiting is not only hurting those of us working with the Web,

but all of those who use the Web as well. Mainly through this plethora of untapped potential which could improve the overall experience across the spectrum for businesses, users and those with the skills to bring this sophisticated, rich, powerful new Web into existence.

For Our Clients

Now this will mean different things for different players in the game. For example, for our clients this means a much more developed and uniquely crafted design that is not bound by the boxes we have allowed our thinking to be contained in. However, this does come with a bit of a compromise that is expected on the parts of our clients as well. At least it does for this to work in the balanced and idealized way these things should play out. But this should be expected. Most change does not come without its compromises.

In this case, our clients have to accept the same truism that we do and concede that their projects will not look the same across various browsers. This is getting easier to convince them of in these times of the expanding mobile market, but they may still not be ready to concede this inch on the desktop side of the coin. Prices might be adjusted in some cases too, and that may be another area that the clients are not willing to accept. But with new doors being opened and more innovation, comes more time and dedicated efforts. These are a few of the implications for our clients, though the expanded innovation is where we should help them focus.

In short:

- Conceding to the idea that the project will not be able to look the same across various browsers,

- This means more developed and unfettered imaginative designs for our clients

- This could lead to increased costs for clients as well, but with higher levels of innovation

- Client's visions for what they want will be less hindered by these limitations

For the Users

The users are the ones who have the least amount invested in most of what is going on behind the scenes. They only see the end result, and they often do not think too much about the process that is involved which brings it to the screens before them. Again, with the mobile market, they have already come across the concept of varying interfaces throughout their varied devices. They only care about the functionality and most probably the style that appeals to them — but this is where their interest tends to end. Unless of course, they too are within the industry, and they may give it a second thought or more. So all this talk of cross-browser compatibility really doesn't concern them, they really leave all that up to us to worry about.

Users only ever tend to notice anything if and when something does not work the way they expect it to from one place to the next. In most cases, they are willing to show something to a relative, friend or colleague, and suddenly from one device to the next, something is different that disrupts their ability to do so. That is when they actually take notice. But if we have done our jobs correctly, these transitions will remain smooth — even with the pushing of the envelopes that we are doing. So there is not much more that is going to change for the users other than a better experience. An average user is not going to check if a given site has the same rounded

corners and drop-shadow in two different browsers installed on the user's machine.

In short:

- Potentially less disruptions of experience from one device to another

- An overall improved user experience

For Designers/Developers

We, the designers and developers of the Web, too have to make the same concession our clients do and surrender the effort to craft the same exact presentation and experience across the vast spectrum of platforms and devices. This is not an easy idea to give up for a lot of those playing in these fields, but as has been already mentioned, we are allowing so much potential to be wasted. We could be taking the Web to new heights, but we allow ourselves to get hung up on who gets left behind in the process — and as a result we all end up getting left behind. Rather than viewing them as separate audiences and approaching them individually, so to speak, we allow the limitations of one group to limit us all.
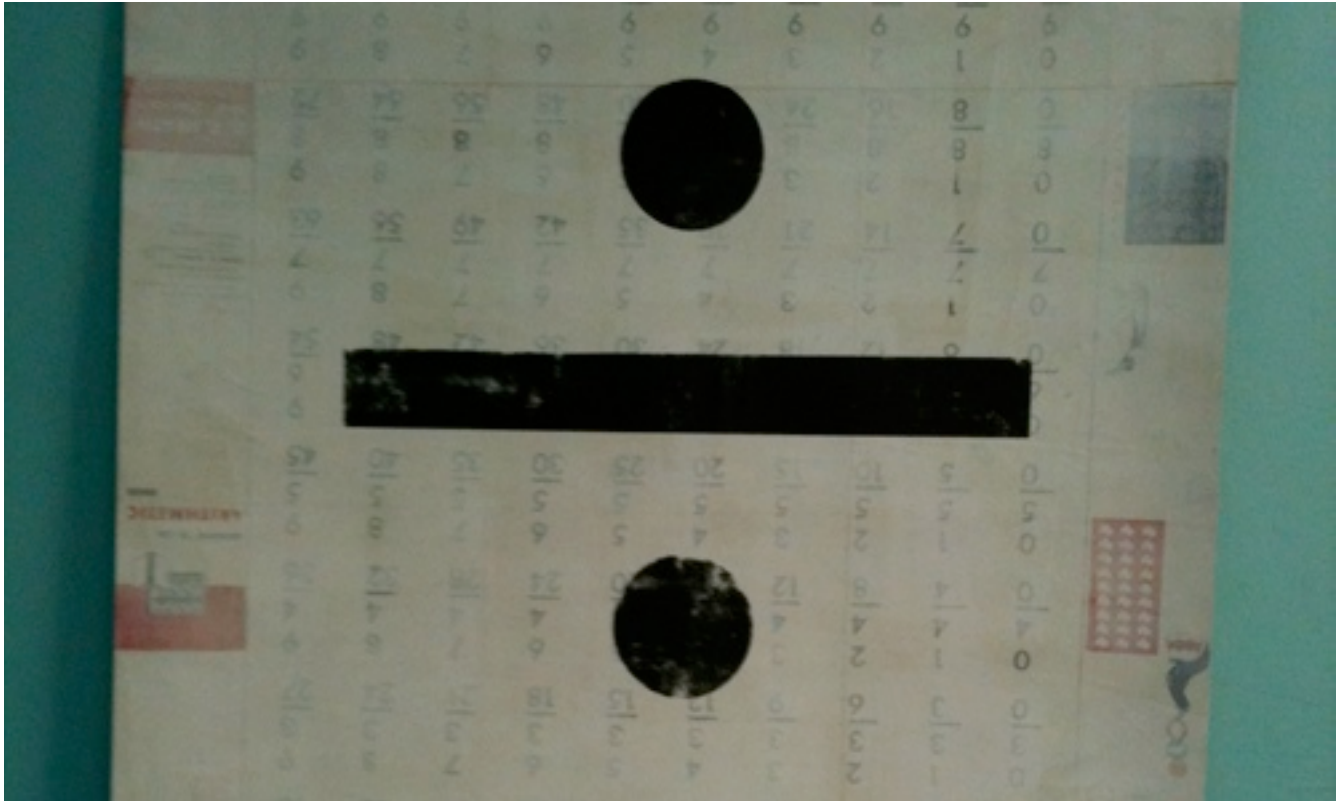
Perhaps a divide and conquer mentality should be employed. [Image Credit](#)

So this could mean a bit more thought for the desired follow through, and we are not suggesting that we strive to appease one group here and damn the rest. Instead, we should just take a unified approach, designing for those who can see and experience the latest, and another for those who cannot. It wouldn't mean more work if we design with those users in mind and produce meaningful and clean code up front and then just adjust it for older browsers. Having to remember that not everyone is afforded the privilege of choosing which browser they are using. And if necessary, this approach can be charged for. So it could lead to more revenue along with exciting new opportunities — by bringing some of the fun back into the work that being boxed in with limitations has robbed us of.

In short:

- Conceding to the idea that the project will not be able to look the same across various browsers

- A more open playing field for designers and developers all around; less restricted by this holding pattern

- More exciting and innovative landscape to attract new clientele

- Division of project audience into separate presentational approaches

- Probably less work involved because we don't need the many hacks and workarounds we've used before

So What Are We Waiting For?

So if this new approach, or adjusted way of thinking, can yield positive results across the browsers for everyone involved, then why are we still holding back? What is it that we are waiting for? Why not cast off these limitations thrown upon our fields and break out of these boxes? The next part of the discussion tries to figure out some of the contributing factors that could be responsible for keeping us restrained.

Fear Factor
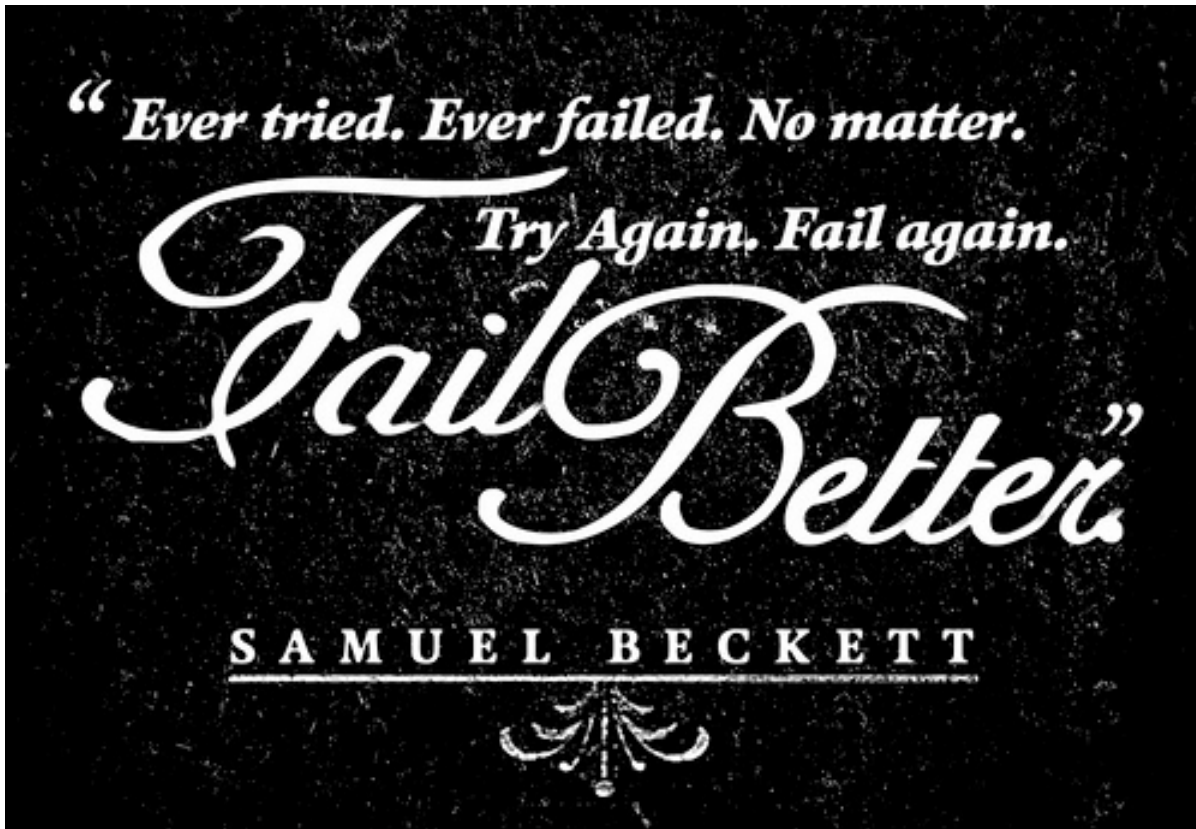
The fail awaits, and so some of us opt to stay back. Image by [Ben Didier](#)

One contributing factor that has to be considered, is perhaps that we are being held back out of fear. This might be a fear of trying something new, now that we have gotten so comfortable waiting for that magic day of compatibility to come. This fear could also stem from not wanting to stand up to some particular clients and try to make them understand this truism of the Web and the concessions that need to be made — with regards to consistent presentation across the browsers. We get intimated, so to speak, into playing along with these unrealistic expectations, rather than trusting that we can make them see the truth of the situation. Whatever the cause is that drives this factor, we need to face our fears and move on.

It's our responsibility of professionals to deliver high-quality work to our clients and advocate on and protect user's interests. It's our responsibility to confront clients when we have to, and we will have to do it at some point anyway, because 100% cross-browser compatibility is just not going to happen.

Comfortable Factor

A possible contributing factor that we should also look into is that some people in the community are just too comfortable with how we design today and are not willing to learn new technology. There are those of us who already tire of the extra work involved in the testing and coding to make everything work as it is, so we have little to no interest at all in an approach that seemingly calls for more thought and time. But really, if we start using new technologies today, we will have to master a learning curve first, but the advantages are certainly worth our efforts. We should see it as the challenge that will save us time and deliver better and cleaner code.

To some extent, today we are in the situation in which we were in the beginning of 2000s; at those times when the emergence and growing support of CSS in browsers made many developers question their approach to designing web sites with tables. If the majority of designers passed on CSS back then and if the whole design community didn't push the Web standards forward, we probably still would be designing with tables.

Doubt Factor

Doubt is another thing we must consider when it comes to our being in hold mode, and this could be a major contributor to this issue. We begin to doubt ourselves and our ability to pull off this innovative, boundary

pushing-kind-of-work, or to master these new techniques and specs, so we sink into the comfort of playing the waiting game and playing it safe with our designs and code. We just accept the limitations and quietly work around them, railing on against the various vendors and the W3C. We should take the new technologies as the challenge to conquer; we've learned HTML and CSS 2.1 and we can learn HTML5 and CSS3, too.

Faith Factor

Faith can be a good thing, but in this case, it can hold you back. Image by [fotologic](#)

Undoubtedly, some of us are holding off on moving forward into these new areas because we are faithfully clinging to the belief that the cross-browser support push will eventually happen. There are those saying that we will be

better off as a community if we allowed the Web to evolve, and that this evolution should not be forced.

But this is not forcing evolution, it is just evolution. Just like with Darwin's theory, the Web evolves in stages, it does not happen for the entire population at once. It is a gradual change over time. And that is what we should be allowing to happen with the Web, gradually using and implementing features for Web community here and there. This way forward progress is happening, and nobody should be held back from these evolutionary steps until we all can take them.

"It's Too Early" Factor

Another possible contributor is the ever mocking "It's too early" factor. Some members of the online community faithfully fear that if they go ahead and accept this new way forward and begin designing or developing in accordance, then as soon as they begin completing projects, the support might be dropped and they would need to update the projects they already completed in the past. It's common to think that it's just too early to work with new standards until they are fully implemented in many browsers; because it's just not safe to assume that they will be implemented at all.

However, one needs to understand the difference between two groups of new features: the widely accepted ones (CSS3's media queries, border-radius. drop-shadows and HTML5 canvas are not going to disappear) and the experimental ones (e.g. some OpenType features are currently supported only in Firefox 4 Beta). The widely accepted features are safe to use and they will not disappear for certain; the experimental features can always be extracted in a separate stylesheet and be easily updated and maintained when necessary. It might be a good idea not to use

experimental, unsupported features in large corporate designs unless they are not affecting the critical design elements of the design.

Validation Factor

We cannot forget to mention that there are also many of us who are refusing to dabble in these new waters simply due to the fact that implementing some of these techniques or styles would cause a plethora of vendor-specific pefixes to appear in the stylesheet, thus impeding the validation we as professionals strive for.

Many of us would never put forth any project that does not fully validate with the W3C, and until these new specs are fully standardized and valid, we are unwilling to include them in their work. And because using CSS3 usually means using vendor-specific prefixes, we shouldn't be using CSS3. Right?

Jeffrey Way's article [*But It Doesn't Validate*](#)

Well, not quite. As Jeffrey Way perfectly explains in his article [But it Doesn't Validate](#), validation is not irrelevant, but the final score of the CSS validator might be. As Jeffrey says,

> *"This score serves no higher purpose than to provide you with feedback. It neither contributes to accessibility, nor points out best-practices. In fact, the validator can be misleading, as it signals errors that aren't errors, by any stretch of the imagination.*
>
> *[...] Validation isn't a game, and, while it might be fun to test your skills to determine how high you can get your score, always keep in mind: it doesn't matter. And never, ever, ever compromise the use of the latest doctype, CSS3 techniques and selectors for the sake of validation."*
>
> *— Jeffrey Way*

Having our work validate 100% is not always the best for the project. If we make sure that our code is clean and accessible, and that it validates without the CSS3/HTML5-properties, then we should take our work to the next level, meanwhile sacrificing part of the validation test results. We should not let this factor keep us back. If we have a chance for true innovation, then we shouldn't allow ourselves to be restrained by unnecessary boundaries.

All in All...

Whatever the factors that keep us from daring into these new CSS3 styles or new HTML5 coding techniques, just for a tangible example, need to be gotten over. Plain and simple. We need to move on and start using CSS3 and HTML5 today. The community will become a much more exciting and innovative playground, which in turn will improve experiences for as well as

draw in more users to this dynamic new Web, which in turn will attract more clientele — effectively expanding the market. This is what could potentially be waiting on the other side of this fence that we are timidly facing — refusing to climb over it. Instead, waiting for a gate to be installed.

Only once we get past we get passed this limited way of looking at the situation, only then will we finally stop falling short of the full potential of ourselves and our field. Are there any areas that you would love to be venturing into, but you are not because of the lack of complete cross browser compatibility? Admittedly, I was a faith factor member of the community myself — how about you? And what CSS3 or HTML5 feature are you going to incorporate into your next design?

HTML5: The Facts And The Myths

Bruce Lawson, Remy Sharp

You can't escape it. Everyone's talking about HTML5. it's perhaps the most hyped technology since people started putting rounded corners on everything and using unnecessary gradients. In fact, a lot of what people call HTML5 is actually just old-fashioned DHTML or AJAX. Mixed in with all the information is a lot of misinformation, so here, JavaScript expert Remy Sharp and Opera's Bruce Lawson look at some of the myths and sort the truth from the common misconceptions.

First, Some Facts

Once upon a time, there was a lovely language called HTML, which was so simple that writing websites with it was very easy. So, everyone did, and the Web transformed from a linked collection of physics papers to what we know and love today.

Most pages didn't conform to the simple rules of the language (because their authors were rightly concerned more with the message than the medium), so every browser had to be forgiving with bad code and do its best to work out what its author wanted to display.

In 1999, the W3C decided to discontinue work on HTML and move the world toward XHTML. This was all good, until a few people noticed that the work to upgrade the language to XHTML2 had very little to do with the real Web. Being XML, the spec required a browser to stop rendering if it encountered an error. And because the W3C was writing a new language

that was better than simple old HTML, it deprecated elements such as `<img>` and `<a>`.

A group of developers at Opera and Mozilla disagreed with this approach and presented a [paper to the W3C in 2004](#) arguing that, "We consider Web Applications to be an important area that has not been adequately served by existing technologies... There is a rising threat of single-vendor solutions addressing this problem before jointly-developed specifications."

The paper suggested seven design principles:

1. Backwards compatibility, and a clear migration path.

2. Well-defined error handling, like CSS (i.e. ignore unknown stuff and move on), compared to XML's "draconian" error handling.

3. Users should not be exposed to authoring errors.

4. Practical use: every feature that goes into the Web-applications specifications must be justified by a practical use case. The reverse is not necessarily true: every use case does not necessarily warrant a new feature.

5. Scripting is here to stay (but should be avoided where more convenient declarative mark-up can be used).

6. Avoid device-specific profiling.

7. Make the process open. (The Web has benefited from being developed in the open. Mailing lists, archives and draft specifications should continuously be visible to the public.)

The paper was rejected by the W3C, and so Opera and Mozilla, later joined by Apple, continued a mailing list called Web Hypertext Application

Technology Working Group (WHATWG), working on their proof-of-concept specification. The spec [extended HTML4 forms](#), until it grew into a spec called Web Applications 1.0, under the continued editorship of Ian Hickson, who left Opera for Google.

In 2006, the W3C realized its mistake and decided to resurrect HTML, asking WHATWG for its spec to use as the basis of what is now called HTML5.

Those are the historical facts. Now, let's look at some hysterical myths.

The Myths

"I Can't Use HTML5 Until 2012 (or 2022)"

This is a misconception based on the projected date that HTML5 will reach the stage in the W3C process known as Candidate Recommendation (REC). The [WHATWG wiki](#) says this:

> For a spec to become a REC today, it requires two 100% complete and fully interoperable implementations, which is proven by each successfully passing literally thousands of test cases (20,000 tests for the whole spec would probably be a conservative estimate). When you consider how long it takes to write that many test cases and how long it takes to implement each feature, you'll begin to understand why the time frame seems so long.

So, by definition, the spec won't be finished until you can use *all of it*, and in two browsers.

Of course, what really matters is the bits of HTML5 that are already supported in the browsers. Any list will be out of date within about a week

because the browser makers are innovating so quickly. Also, much of the new functionality can be [replicated with JavaScript](#) in browsers that don't yet have support. The `<canvas>` property is in all modern browsers and will be in Internet Explorer 9, but it can be faked in old versions of IE with the [excanvas library](#). The `<video>` and `<audio>` properties can be faked with Flash in old browsers.

HTML5 is designed to degrade gracefully, so with clever JavaScript and some thought, all content should be available on older browsers.

"My Browser Supports HTML5, but Yours Doesn't"

There's a myth that HTML5 is some monolithic, indivisible thing. It's not. It's a collection of features, as we've seen above. So, in the short term, you cannot say that a browser supports everything in the spec. And when some browser or other does, it won't matter because we'll all be much too excited about the next iteration of HTML by then.

What a terrible mess, you're thinking? But consider that CSS 2.1 is not yet a finished spec, and yet we all use it each and every day. We use CSS3, happily adding `border-radius`, which will soon be supported everywhere, while other aspects of CSS3 aren't supported anywhere at all.

Be wary of browser "scoring" websites. They often test for things that have nothing to do with HTML5, such as CSS, SVG and even Web fonts. What matters is what you need to do, what's supported by the browsers your client's audience will be using and how much you can fake with JavaScript.

HTML5 Legalizes Tag Soup

HTML5 is a lot more forgiving in its syntax than XHTML: you can write tags in uppercase, lowercase or a mixture of the two. You don't need to self-close tags such as `img`, so the following are both legal:

```
1 <img src="nice.jpg" />
2 <img src="nice.jpg">
```

You don't need to wrap attributes in quotation marks, so the following are both legal:

```
1 <img src="nice.jpg">
2 <img src=nice.jpg>
```

You can use uppercase or lowercase (or mix them), so all of these are legal:

```
1 <IMG SRC=nice.jpg>
2 <img src=nice.jpg>
3 <iMg SrC=nice.jpg>
```

This isn't any different from HTML4, but it probably comes as quite a shock if you're used to XHTML. In reality, if you were serving your pages as a combination of text and HTML, rather than XML (and you probably were, because Internet Explorer 8 and below couldn't render true XHTML), then it never mattered anyway: the browser never cared about trailing slashes, quoted attributes or case—only the validator did.

So, while the syntax appears to be looser, the actual parsing rules are much tighter. The difference is that there is no more tag soup; the specification describes exactly what to do with invalid mark-up so that all conforming browsers produce the same DOM. If you've ever written JavaScript that has to walk the DOM, then you're aware of the horrors that inconsistent DOMs can bring.

This error correction is no reason to churn out invalid code, though. The DOM that HTML5 creates for you might not be the DOM you want, so ensuring that your HTML5 validates is still essential. With all this new stuff, overlooking a small syntax error that stops your script from working or that makes your CSS unstylish is easy, which is why we have HTML5 validators.

Far from legitimizing tag soup, HTML5 consigns it to history. Souper.

"I Need to Convert My XHTML Website to HTML5"

Is HTML5's tolerance of looser syntax the death knell for XHTML? After all, the working group to develop XHTML 2 was disbanded, right?

True, the XHTML 2 group was disbanded at the end of 2009; it was working on an unimplemented spec that competed with HTML5, so having two groups was a waste of W3C resources. But XHTML 1 was a finished spec that is widely supported in all browsers and that will continue to work in browsers for as long as needed. Your XHTML websites are therefore safe.

HTML5 Kills XML

Not at all. If you need to use XML rather than HTML, you can use XHTML5, which includes all the wonders of HTML5 but which must be in well-formed XHTML syntax (i.e. quoted attributes, trailing slashes to close some elements, lowercase elements and the like.)

Actually, you can't use *all* the wonders of HTML5 in XHTML5: `<noscript>` won't work. But you're not still using that, are you?

HTML5 Will Kill Flash and Plug-Ins

The `<canvas>` tag allows scripted images and animations that react to the keyboard and that therefore can compete with some simpler uses of Adobe Flash. HTML5 has native capability for playing video and audio.

Just as when CSS Web fonts weren't widely supported and Flash was used in sIFR to fill the gaps, Flash also saves the day by making HTML5 video backwards-compatible. Because HTML5 is designed to be "fake-able" in older browsers, the mark-up between the video tags is ignored by browsers that understand HTML5 and is rendered by older browsers. Therefore, embedding fall-back video with Flash is possible using the old-school `<object>` or `<embed>` tags, as pioneered by Kroc Camen is his article "Video for Everybody!" (see the screenshot below).
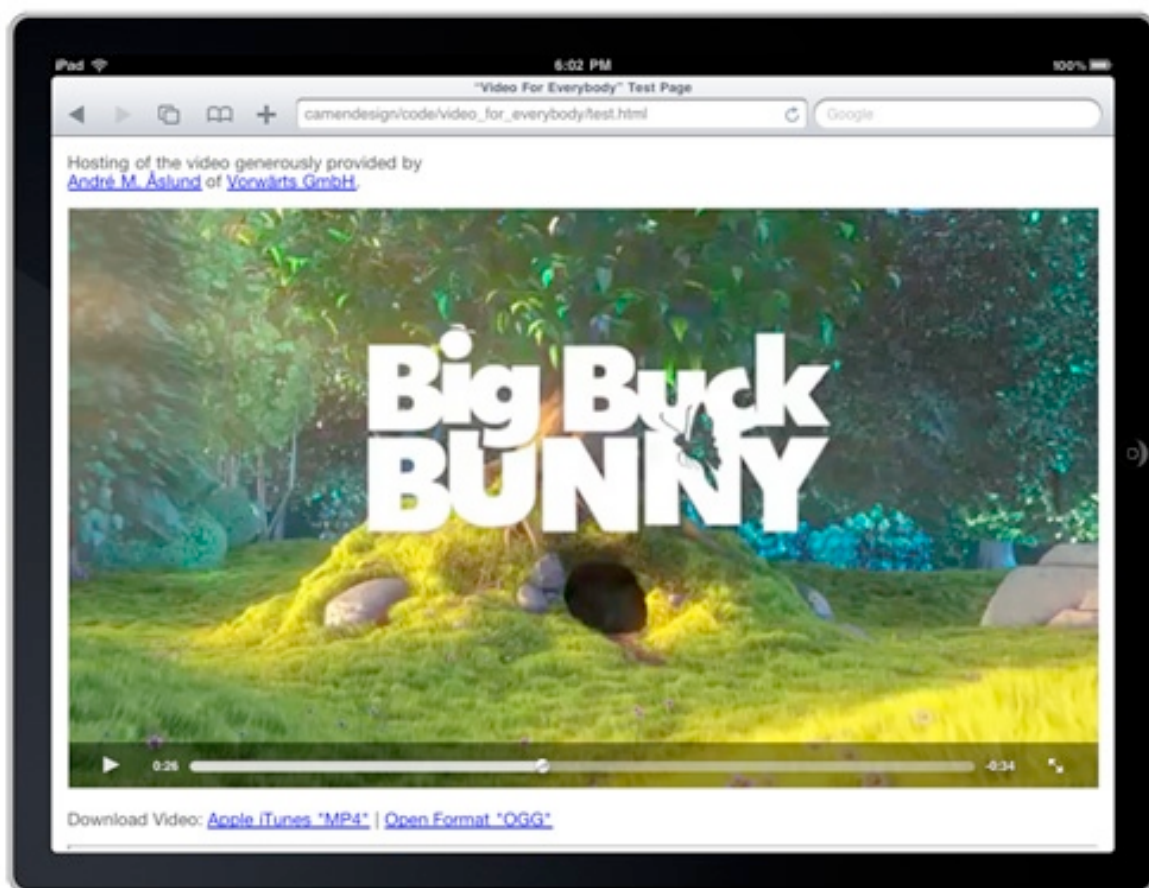
But not all of Flash's use cases are usurped by HTML5. There is no way to do digital rights management in HTML5; browsers such as Opera, Firefox and Chrome allow visitors to save video to their machines with a click of the context menu. If you need to prevent video from being saved, you'll need to use plug-ins. Capturing input from a user's microphone or camera is currently only possible with Flash (although a <code><device></code> element is being specified for "post-5" HTML), so if you're keen to write a Chatroulette killer, HTML5 isn't for you.

HTML5 Is Bad for Accessibility

A lot of discussion is going on about the accessibility of HTML5. This is good and to be welcomed: with so many changes to the basic language of the Web, ensuring that the Web is accessible to people who cannot see or use a mouse is vital. Also vital is building in the solution, rather than bolting it on as an afterthought: after all, many (most?) authors don't even add alternate text to images, so out-of-the-box accessibility is much more likely to succeed than relying on people to add it.

This is why it's great that HTML5 adds native controls for things like sliders (<code><input type=range></code>, currently supported in Opera and Webkit browsers) and date pickers (<code><input type=date></code>, Opera only)—see Bruce's HTML5 forms demo)—because previously we had to fake these with JavaScript and images and then add keyboard support and WAI-ARIA roles and attributes.

The <code><canvas></code> tag is a different story. It is an Apple invention that was reverse-engineered by other browser makers and then retrospectively specified as part of HTML5, so there is no built-in accessibility. If you're just using it for eye-candy, that's fine; think of it as an image, but without any

possibility of alternate text (some additions to the spec have been suggested, but nothing is implemented yet). So, ensure that any information you deliver via `<canvas>` supplements more accessible information elsewhere.

Text in a `<canvas>` becomes simply pixels, just like text in images, and so is invisible to assistive technology and screen readers. Consider using the W3C graphics technology [Scalable Vector Graphics](#) (SVG) instead, especially for things such as dynamic graphs and animating text. SVG is supported in all the major browsers, including IE9 (but not IE8 or below, although the [SVGweb](#) library can fake SVG with Flash in older browsers).

The situation with `<video>` and `<audio>` is promising. Although not fully specified (and so not yet implemented in any browsers), a new [`<track>` element](#) has been included in the HTML5 spec that allows timed transcripts (or karaoke lyrics or captions for the deaf or subtitles for foreign-language media) to be associated with multimedia. It [can be faked in JavaScript](#). Alternatively (and better for search engines), you could include transcripts directly on the page below the video and use [JavaScript to overlay captions](#), synchronized with the video.

"An HTML5 Guru Will Hold My Hand as I Do It the First Time"

If only this were true. However, the charming Paul Irish and lovely Divya Manian will be as good as there for you, with their [HTML5 Boilerplate](#), which is a set of files you can use as templates for your projects. Boilerplate brings in the JavaScript you need to style the new elements in IE; pulls in jQuery from the Google Content Distribution Network (CDN), but with fall-back links to your server in case the CDN server is down.

It adds mark-up that is adaptable to iOS, Android and Opera Mobile; and adds a CSS skeleton with a comprehensive reset style sheet. There's even an *.htaccess* file that serves your HTML5 video with the right MIME types. You won't need all of it, and you're encouraged to delete the stuff that's unnecessary to your project to avoid bloat.

The User Is The Anonymous Web Designer

Alexander Dawson

We invest time, money and effort into understanding our audience, and the movement toward a more socially networked Web has made us realize the power that visitors have over how our designs are engineered; and we try to meet their ever-growing needs. Community is integral to the evolution and functioning of a website, and visitors and website owners have become dependent on each other. This reflects a change in the industry: the user has turned into an anonymous designer.

This article will explore the influential figure 'user' and uncover the power of your community. Whether you boast social applications, interactivity or a stream of regular visitors, your audience might be a powerful untapped resource at your fingertips. You, the website owner, have the power to make decisions and override them (for better or worse), but the user deserves to be recognized as more than a statistic.

Traditional Roles

Back in the Web's infancy, the roles of the Web professional and visitor were clearly laid out. Technologies such as email, bulletin boards and Internet Relay Chat (IRC) rooms existed, but most Web users were technologically inclined — cost and complexity of computers and Internet access were the biggest obstacles for many in the early days. Additionally, website were simple and the power of users was restricted; users had little sway with designers and developers.

The Professionals

The role of the Web professional used to be — famously, in the '90s — to build websites that delivered certain information. Asking the user for feedback was an afterthought; this was evidence of the commonly held belief that we, the experts, knew what was best for users. **Usability and accessibility were a luxury**, and many website providers ignored them in an attempt to control the way the medium was used.
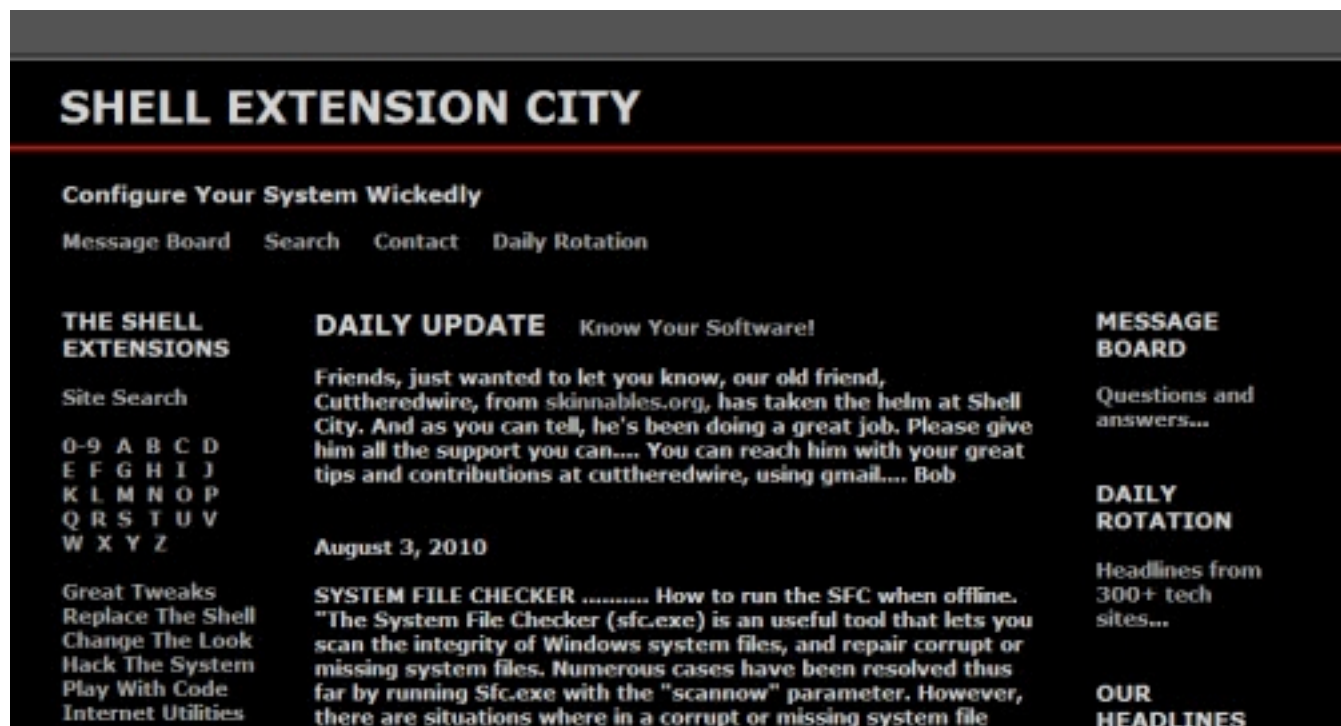
Historically, the Web professional's role was to maintain websites.

The following were considered the primary tasks of the Web professional:

- Providing content and services to the client

- Creating a website to hold the information

- Dealing with technical requests about the information

The Audience

Traditionally, the role of the audience was to give feedback, but it had — and still sometimes has — no influence on the website. The romanticized notion that a website owner should do whatever they want is to blame. In the past, a website's popularity was determined mostly by content; marketing was restricted to directories and search engines; empowering the user and improving the feedback process was far from the designer's mind.

The following were considered the primary tasks of the user:

- Give the owner financial incentive to continue offering the service

- Popularize or endorse the website

- Put the professional's creation, free or not, to good use

Those times have (mostly) come and gone. Without visitors, our websites would hardly exist, and when we realize that, we will abandon our old-fashioned ways and quit imposing our assumptions on users. Visitor interaction has become vital to the experience of a website, and the need for positive feedback — now visible everywhere and uncontrollable — has reversed the roles: our audience now has the voice and tools it needs to be satisfied.

Playing The Part

It has become important for us to define what roles the Web professional and user play in the dynamic process of building and improving websites, especially given the new-found focus on user experience and the development of tools that allow visitors to engage with and become

attached to our services. Furthermore, ensuring that these roles are, in fact, played is essential to the process.

The Professionals

Professionals hold the keys to the castle. While many designers focus on the result, understanding what audiences need, what problems need swift solutions, which visitors require these solutions and how best to implement them is paramount. You might have some control over your users, but they can still dampen your popularity.
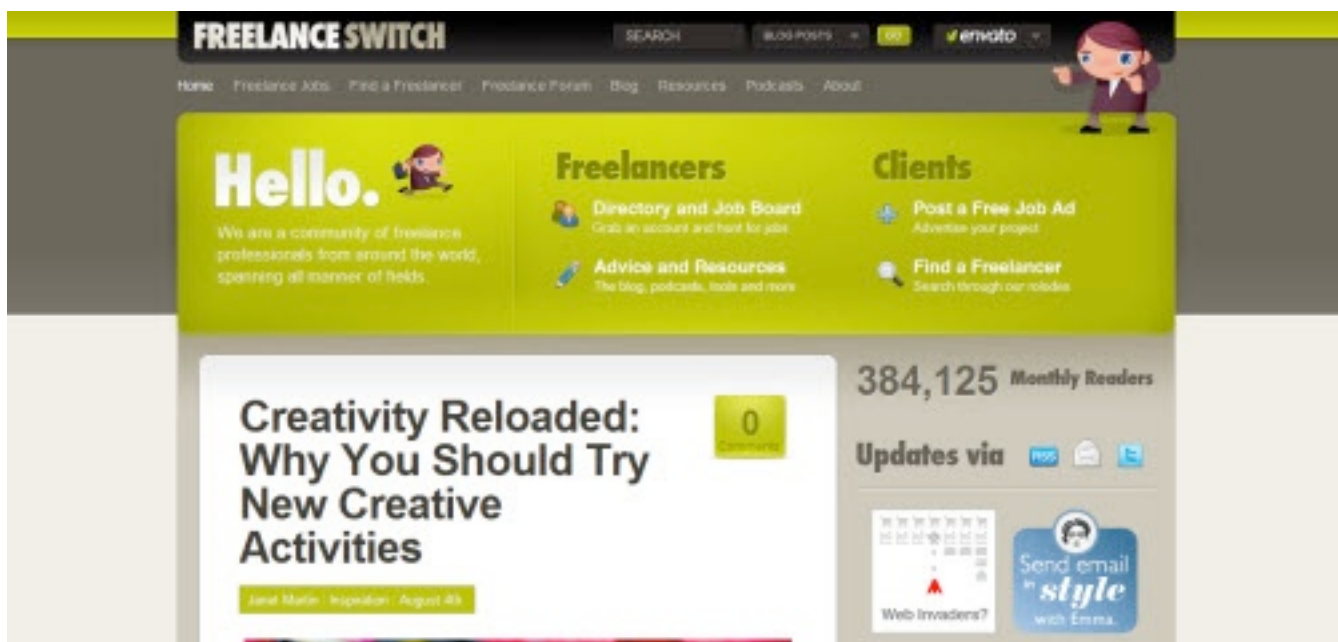
The user experience should actually serve the needs of users.

Here are some skills that professionals bring to the table:

- Knowledge of how to build a website and how it works

- Experience with anticipating user needs

- Delivering services and content that users need

The Audience

Audience members play a different role. Most websites don't offer a broad spectrum of customizable features (for good reason) or feedback options that give insight into how visitors use the website or what they'd like to see improved — specifically, what they'd like to see improved that would make them return on a regular basis. Word of mouth has become a serious force on the Web with Twitter, Facebook and their kin; and while you control the code, they control users' wallets, attention and referrals.
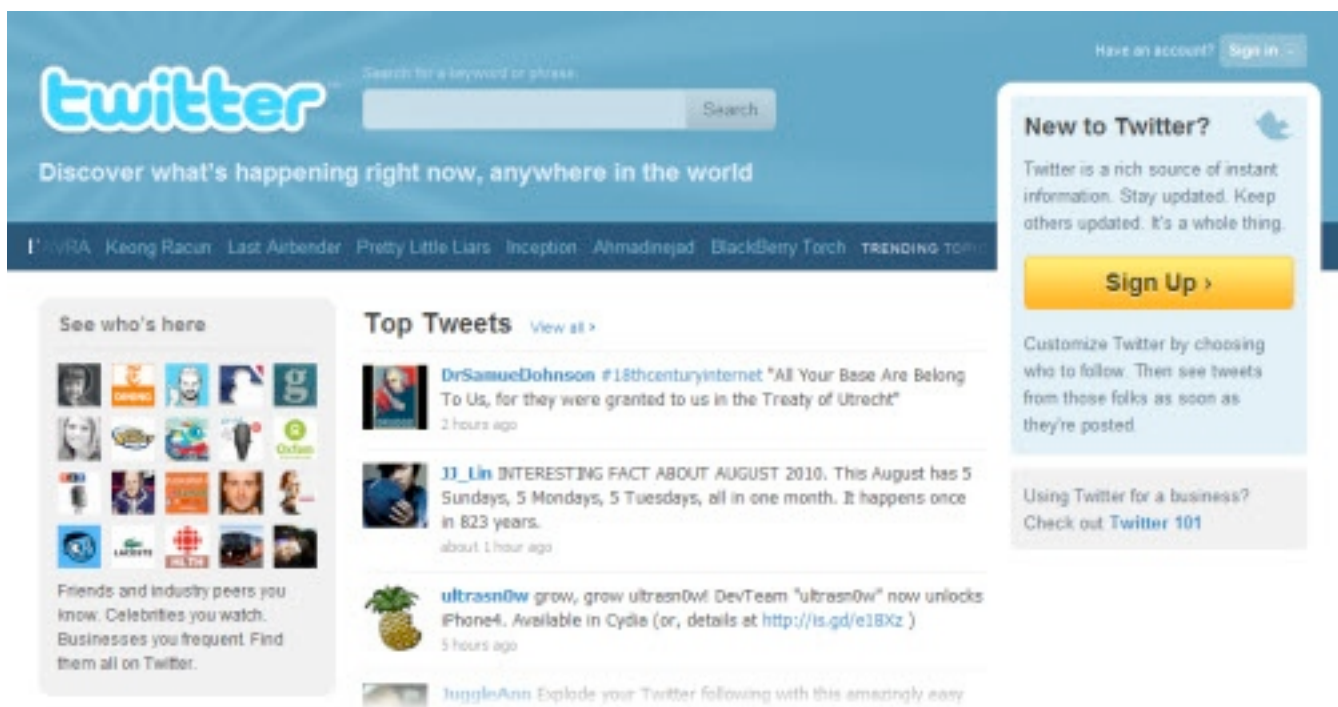
Visitors know what they like, and negative feedback can spread quickly.

Here is what visitors contribute:

- Essential feedback on core service failings

- Word-of-mouth promotion

- Knowledge of what works, not just what's pretty

The roles of professional and user can be reversed on occasions when one expert helps another improve their website, resulting in a mixture of a professional perspective and visitor-oriented goals. Being both visitor and expert gives you empathy for the user and a valuable perspective, especially if you're lucky enough to be in a position of power.

Nature Versus Nurture

Both sides bring goods to the table. Professionals bring the service and content into existence, thereby providing the "location for the party." New visitors share vital details about how you can encourage them to be more involved, and they spread your website to the masses. Maintaining equilibrium between the professional and user is required if your Web presence is going to survive.

The Professionals

As it stands, the professional designer still plays the traditional role, but it should be expanded to encompass the evolving needs of our audience. The ability to nurture and cultivate is a skill ignored by many in our industry. In the past, "If you build it, they will come" described the Web fairly accurately, especially given the limited competition, but easy enterprise and designing for yourself have failed by their own merit.

Here are some problems that face a Web professional:

- Subjective opinion

- Too many tasks to be able to master one

- Fixated on success and reward (depending on the service being offered)

The Audience

Even if your visitor has no understanding of Web design, they will know what works for them, so you should encourage them to share their ideas. You can do this through code, design, experience and interactivity (or functionality). Playing to your audience's strengths makes sense (if only to avoid the MySpace effect: going stale).

Some of the problems facing users are:

- Lacking knowledge of the Web's limitations

- Their diverse backgrounds and characters lead to quirks

- Subjective opinions

Survival of the fittest is the name of the game, and with the increasing pressure to focus on the needs of users and encourage them to connect to your services, it's important that your designs are user-centric: contextual and interactive. The anonymous designer who knows what they want is well equipped for the Internet's future. We can't overlook users as we did in the past.

Elements Of Success

At this point, you're probably wondering why we should place so much trust in users. That's a fair question. Distinguishing between constructive advice and useless drivel can be difficult when many users don't even know what a Web browser is, and determining the relevance of reviews is a challenge when manipulation is commonplace. All interaction with users should be done with respect (and perhaps a grain of salt), but it's still worth encouraging the crowd to take action.

The trick is to use your knowledge (from statistics to semantics), experience, research and trust (of information sources) to your advantage.

By using a mixture of both quantitative data (numbers, statistics, etc.) and qualitative data (opinions, ideas, etc.), you can make informed decisions about what will work best for your clients. Putting the onus entirely on visitors is unfair, but giving them a channel for input will make a difference. If you don't please them, they'll surely take their business elsewhere.
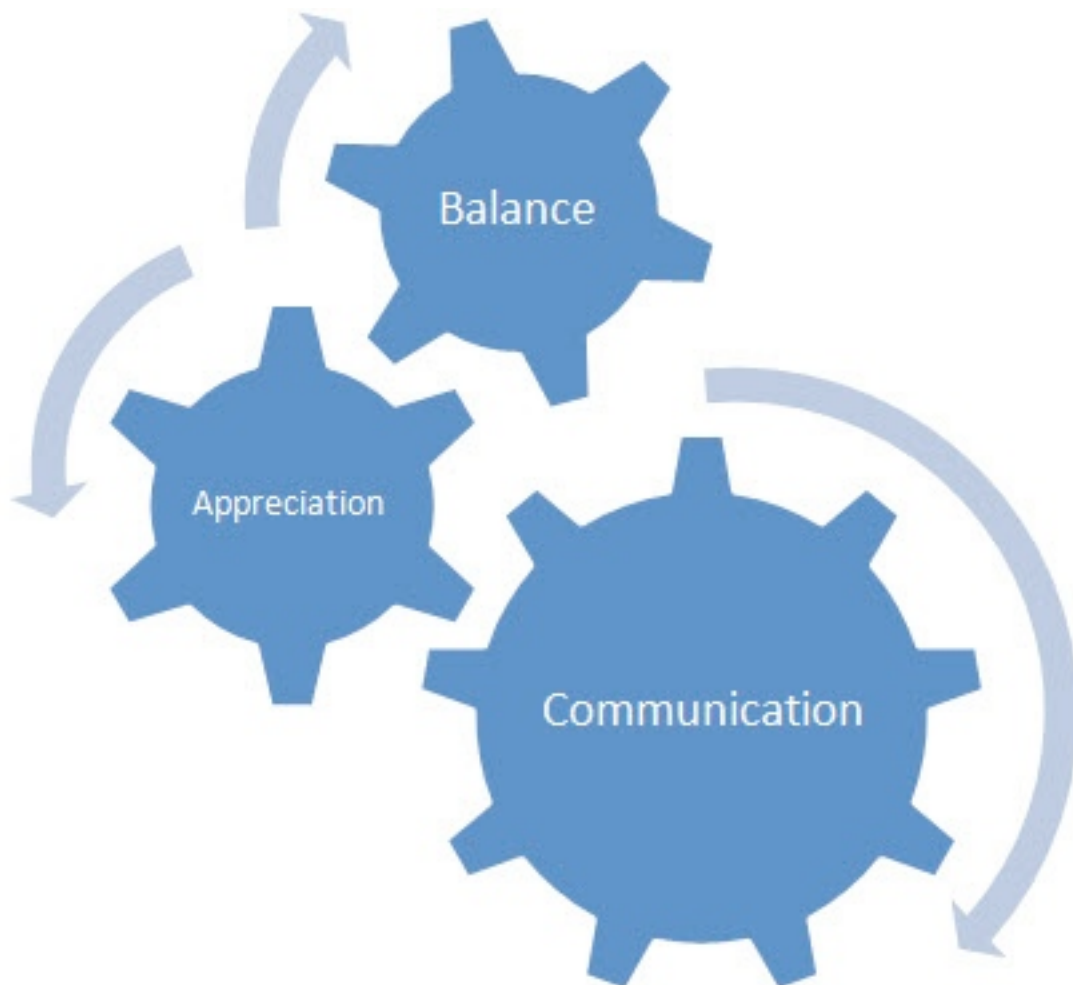
The ABCs of experience: appreciation, balance and communication.

A few of the ideas you get from users are bound to be cringe-worthy, but it's surprising how many good ones you'll get if you implement feedback functionality properly. The stereotypical designer who leads with his gut is a testament to the lack of awareness of user expectations and needs. We must acknowledge this co-dependency, this need for an exchange of perspectives, and the value of the anonymous designer.

Appreciation

Your visitors are so much more than statistics. They deserve to be appreciated. You, the professional, can re-engineer a website until the cows come home, but it's ultimately up to users to decide whether to make use of your services. Acknowledging and understanding their needs, and encouraging diversity, is critical to breaking down barriers.
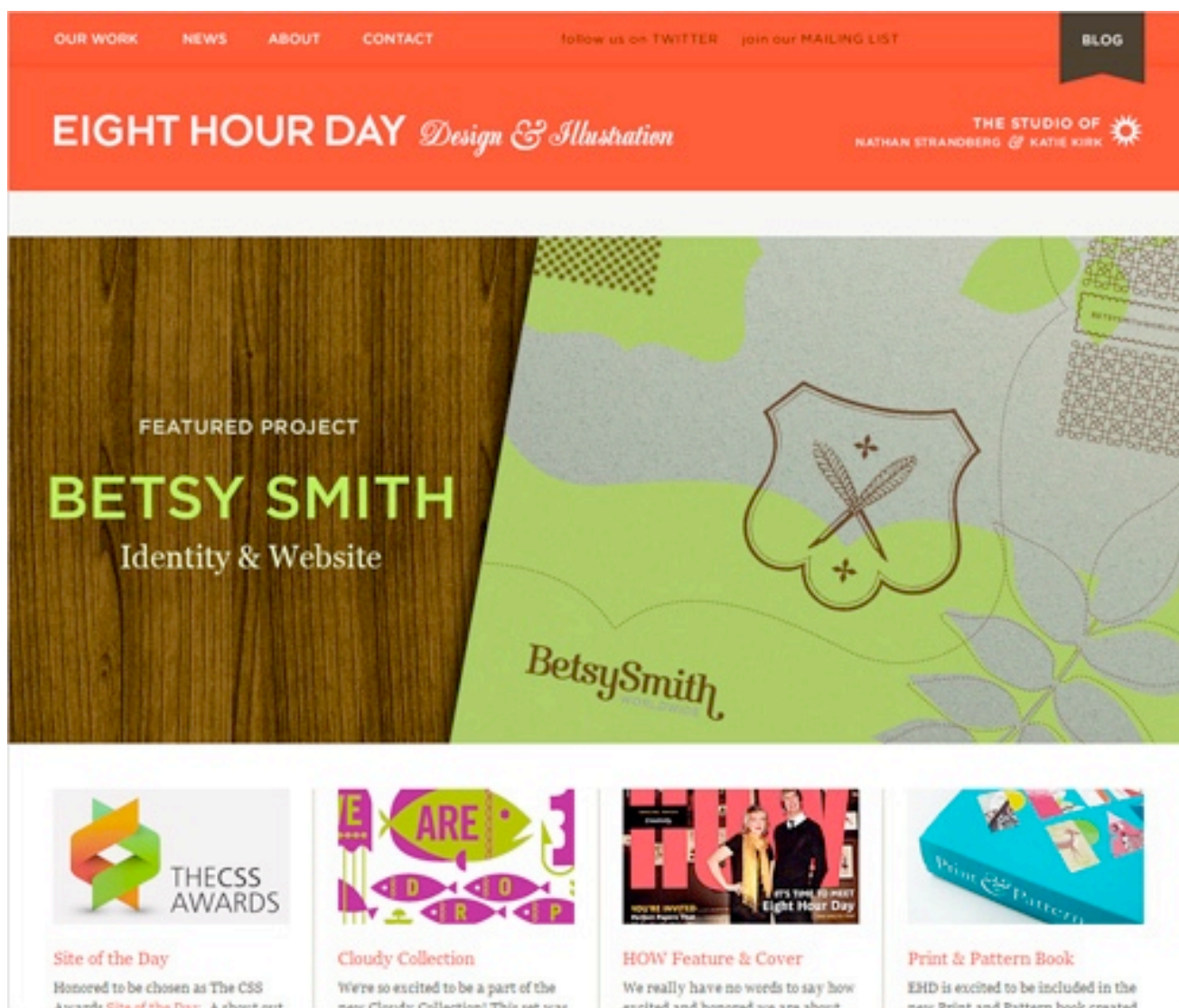
Appreciate your users and the issues they might encounter, and be patient. That also means having a clean, attractive design that communicates. Example: [EightHourDay](#)

Ask your visitors the following questions:

- What, if anything, could my website offer to better meet your needs?

- Do you have any technical difficulties with the way the website functions?

- What more could we offer to deliver additional value?

Balance

Balance is integral to life, and it's probably more critical to design than other Internet-related fields. Designers forge open bonds with visitors that often lead to social relationships that enrich not only our websites but our lives. Visitors hold greater value for our communities than many of us care to admit — think of Wikipedia or Facebook. Maintaining good relationships with them is paramount.

Stay level-headed by maintaining the balance between contribution and ownership.

Ask your visitors the following questions:

- How can we encourage you to participate more often?

- What content or services would you like to see in future?

- How much time do you spend on our website?

Communication

Communication is fundamental to our social and interactive experiences on the Web, and the staggering amount of rich content (not even counting spam!) that visitors contribute could exceed our wildest dreams. The user's job is to come forth and help us help them, but our job is to act as intermediaries between them and the website so that they achieve the ideal experience. We are the Web's "interpreters."
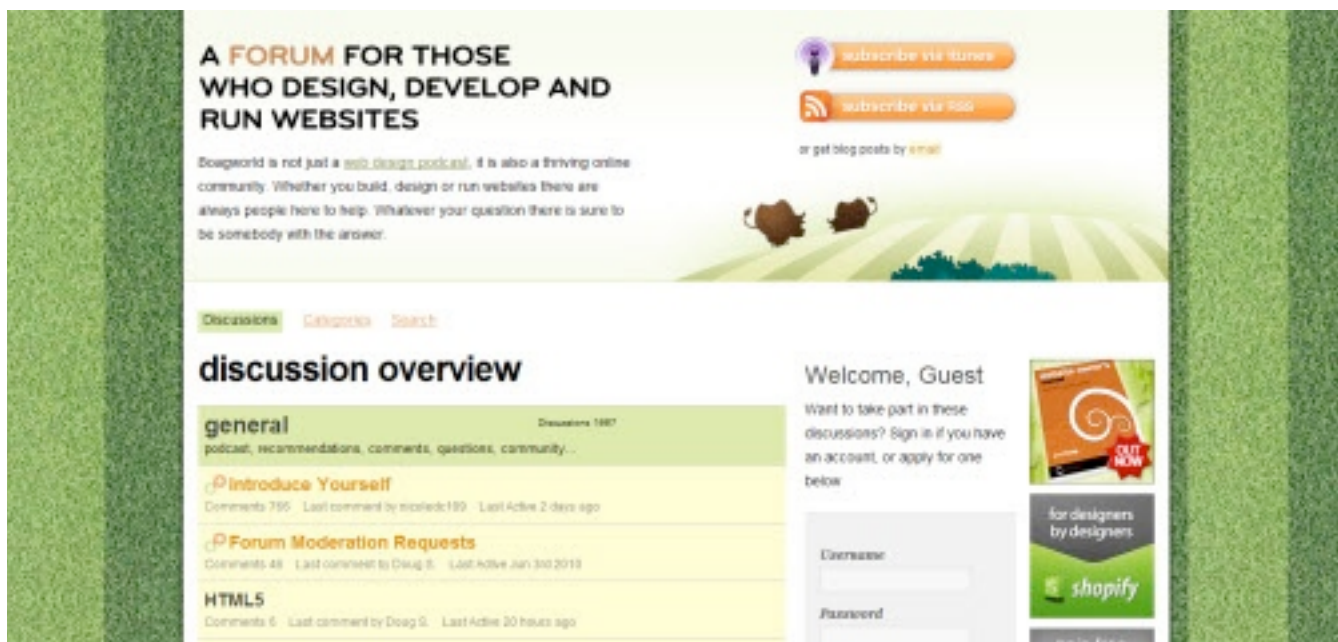
Communication is paramount. Make sure you can be contacted easily.

Ask your visitors the following questions:

- Are we missing an essential feature that would help us help you?

- Do you have any concerns or complaints that we need to address?

- Is there anything you want to ask about our website or services?

The Sociological Switch

Of all that has been addressed in this article, the most important is the process of planning and implementing mechanisms for visitors to interact with your services. So, put time and effort into studying how to make your website better. There are many methods of doing this (audits, surveys, statistics). What's important is that lines of communication with your users stay open, so that you can go about your business successfully.
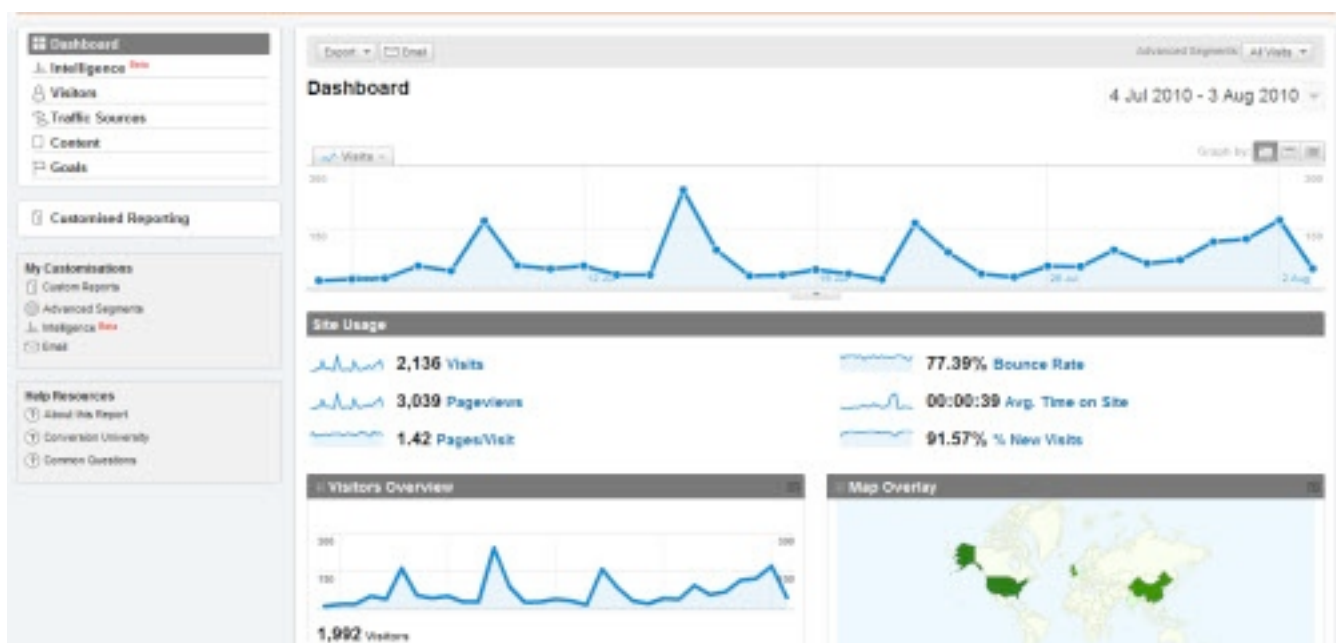
Planning and research make up a huge portion of the creation of the user experience.

The future of the Web will be about being socially connected, and a role reversal that I call the "sociological switch" seems to be taking place. Simply put, we Web professionals are becoming the audience of our users' demands, and users are becoming anonymous designers, deciding where they visit, how the Web should evolve and how media are constructed and consumed.

The Design Matrix: A Powerful Tool for Guiding Client Input

Bridget Fahrland

I used to think the beginning of a website design project was the best part. Hopes are high. People are full of great ideas. Nobody is disappointed yet. But as I gained experience, I found that learning about a client's brand, competitors and customers doesn't always give clear direction about design goals.

Brand discussions can generate goals like "be modern," but they don't necessarily determine how to accomplish those goals. Competitor reviews can devolve into cherry-picking sessions that spawn "frankencomps" rather than provide helpful feedback. And mood boards, which communicate a general feeling, don't help to articulate or prioritize design goals. With a design matrix, you can guide discussions and establish clear direction.

Hey, You Got Math In My Art...

Sometimes the abstract nature of design is enough to make you envy the people over in accounting, with their definite answers and proven formulas. While the beauty of design is that it transcends the world of definite answers, introducing a little math in the form of design matrices can help you create better websites by providing a clear picture of where the website design is today and where it should go tomorrow.

Design matrices don't require any serious math skills because they're based on the coordinate system. Chances are you've seen a competitor matrix

that ranks brands according to two key attributes on X and Y axes (for example, value could be plotted against profit margin). A design matrix is essentially like a competitor matrix but ranks the client's website against competitor websites, and it uses design attributes ("clean" and "warm," for example) instead of other points of competitive comparison.
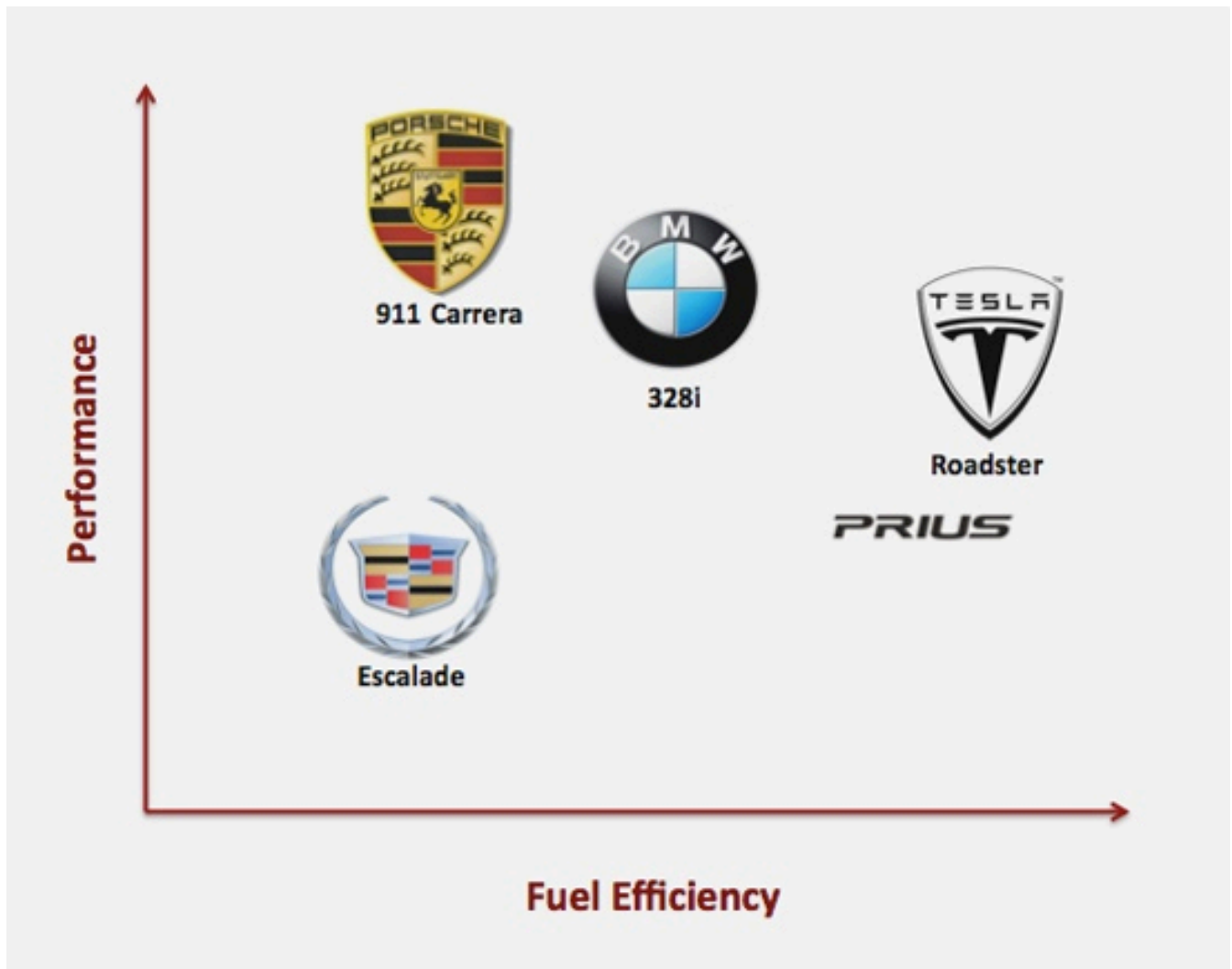
A typical competitor matrix ranks brands according to rational factors. (This example, which compares a few car models, was created for illustrative purposes only.)
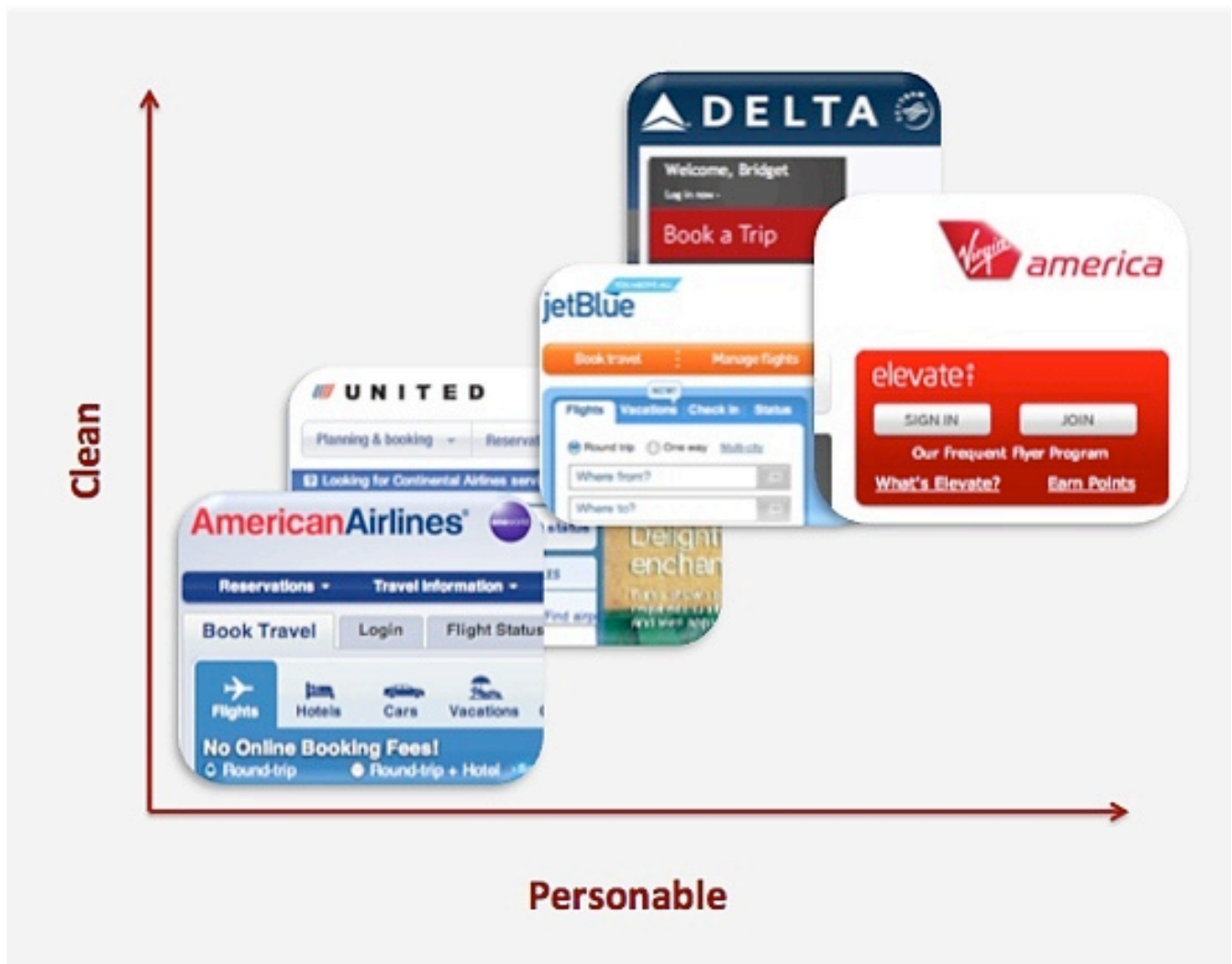
A design matrix ranks website designs according to design attributes. (This example, which compares airline website designs, was created for illustrative purposes only).

Design matrices are powerful tools for determining the path of the website design process, because: they force you to determine two design attributes to focus on; they build consensus within a team; they guide the clients' perception of competitors; and, most importantly, they lead to differentiated website designs.

The Art (And Math) Of Building A Design Matrix

Step 1: Gather Information

To build a design matrix, you will need to know the client's core brand attributes and main competitors. You should also have a broad understanding of what the redesign aims to accomplish (from a design perspective): "the website is cluttered" or "our website is not engaging." The good news is that information gathering is a normal part of the discovery phase.

A design matrix should not be the only piece of work involved in the discovery phase, but it can replace some other approaches. Creating or documenting a brand's position and defining the key redesign goals are essential. However, a design matrix could potentially replace mood boards. A mood board is a collage or grid of images that capture the "feel" or "tone" of a brand. They are valuable tools for providing direction to new brands, but they provide a less concrete direction than a design matrix. If the brand is in its nascent form and needs broad high-level direction, then mood boards work well; but if you are working with an established brand or a client who prefers a concrete approach, then a design matrix is the best bet.

Athletic
Wide Selection
Value
Convenient
Proactive

Inspiring
Personal
Empowering
Relevant
Accessible

Engaging
Warm
Comfortable
Stylish
Trustworthy

Rational   Emotional
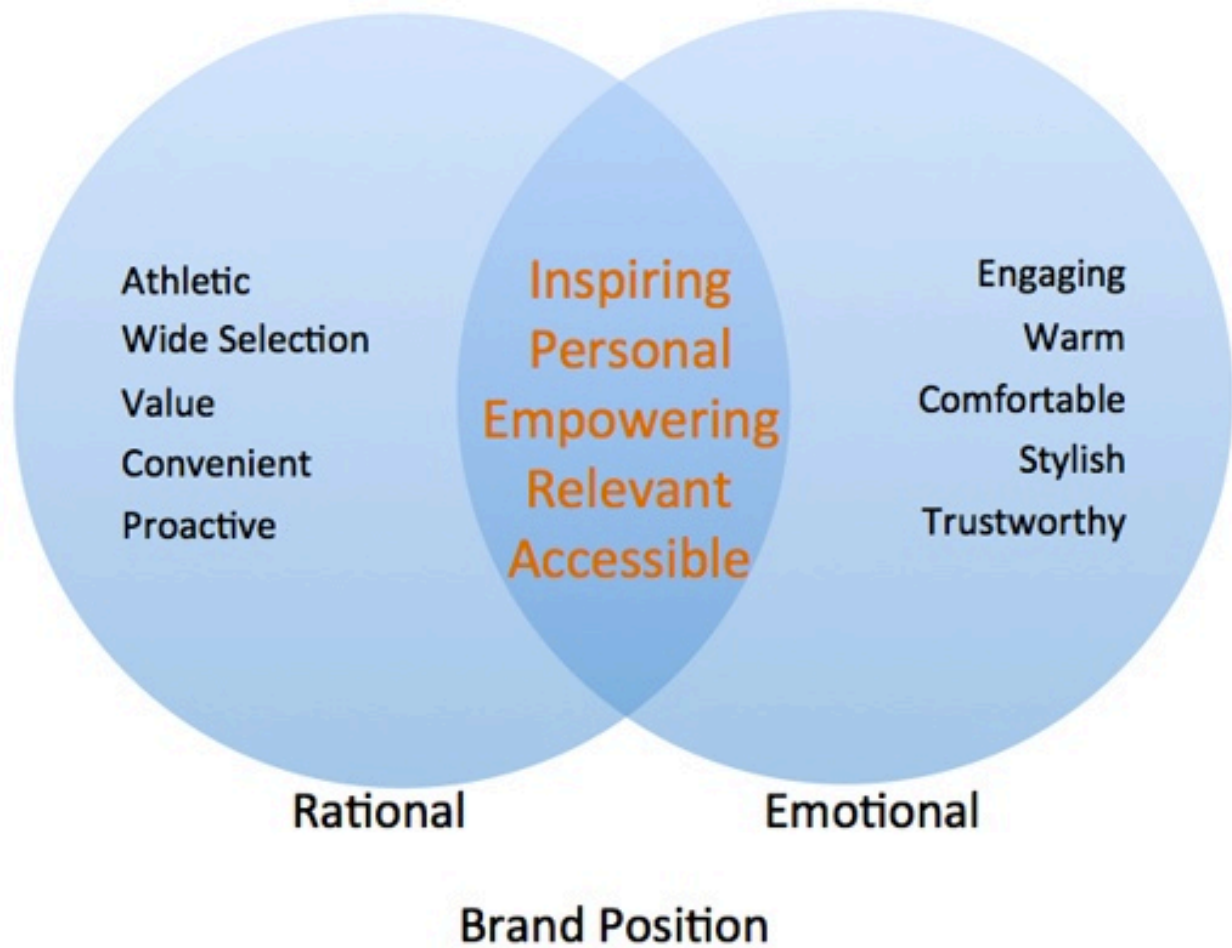
Brand Position

Document the brand's position before creating a design matrix. (For illustrative purposes only.)
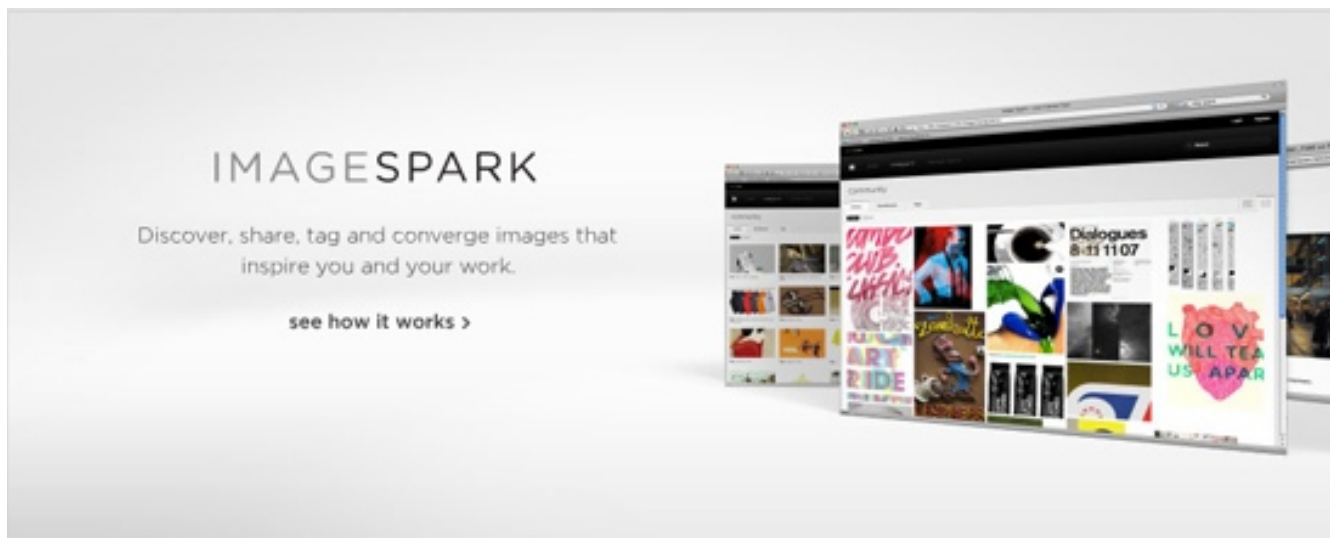
Image Spark is a great resource for creating online mood boards—particularly useful if you are working with a company that requires high-level brand definition.

Another common discovery activity that design matrices can replace is the "competitor website review." Looking at competitors' websites can generate lively discussion, but too often it either shifts the focus to feature sets instead of design direction, or it becomes a cherry-picking session for disparate design elements from a variety of websites that the designer is somehow supposed to mash together into a single coherent website design.

Create a design matrix that shows the current website in relation to competing websites. This way, you are less likely to get distracted by feature sets or be expected to combine all sorts of design elements. That said, if you are looking for an energizing group activity, competitor reviews can generate more brainstorming than a design matrix. Doing both is an option, but if you do that, then do the matrix after the walk-through of competitors.

United Airlines' website.

Delta Airlines' website.

Looking at these individual airline websites, rather than comparing them on a design matrix, can lead to a less design-oriented and more feature-focused conversation.

Step 2: Determine Your X and Y Axes

Narrowing down a design direction to two attributes can be uncomfortable for those of us accustomed to creative briefs that list a litany of brand attributes to guide our design. How often have we heard that a design should be "clean," "inspiring," "warm," "engaging," "approachable" and "trustworthy"? How do we even accomplish just two of these attributes? And if we must choose only two, how do we decide?

Understand that a design matrix is not intended to limit the final design to two attributes. That would be almost impossible. It is intended to illustrate the two most important attributes for taking the website design to the next level and differentiating it from that of competitors.

To determine your X and Y axes, ask yourself the following questions:

- Of all the brand's attributes, what will make this client stand out from the crowd? Which design traits reinforce those brand attributes?

- What are the competitor websites' strengths and weaknesses?

- What does the design need to do better in order to accomplish the website's goals?

The X and Y axes should not be nearly synonymous (for example, "warm" and "engaging"), nor should they be mutually exclusive ("innovative" and "traditional"). There should be a slight tension between the two attributes.

The airline websites, for example, are ranked according to how "clean" and "personable" their designs are. There is a slight, but not negating, tension between these two attributes. Clean websites can come across as cold if they don't have a distinctive voice or warm color palette. Personable websites are often less functionally organized. Achieving a high ranking for both attributes is a worthy challenge, and stepping up to that challenge will definitely create a distinctive website.

You might find that you change the labels of your axes as you place the websites on the matrix (see step 3), but the above process should get you pretty close to determining what the final axes should be.

Step 3: Play a Little

You know the competitors. You have a clear idea of what is important, brand- and design-wise. You have determined your x and y axes. It's time to try some things out.

Place all of the websites on your matrix as you would rank them off the top of your head. As you begin to place them, you will most likely rearrange some as you compare them to others. This is a natural part of the process because the matrix shows relationships as well as individual rankings.
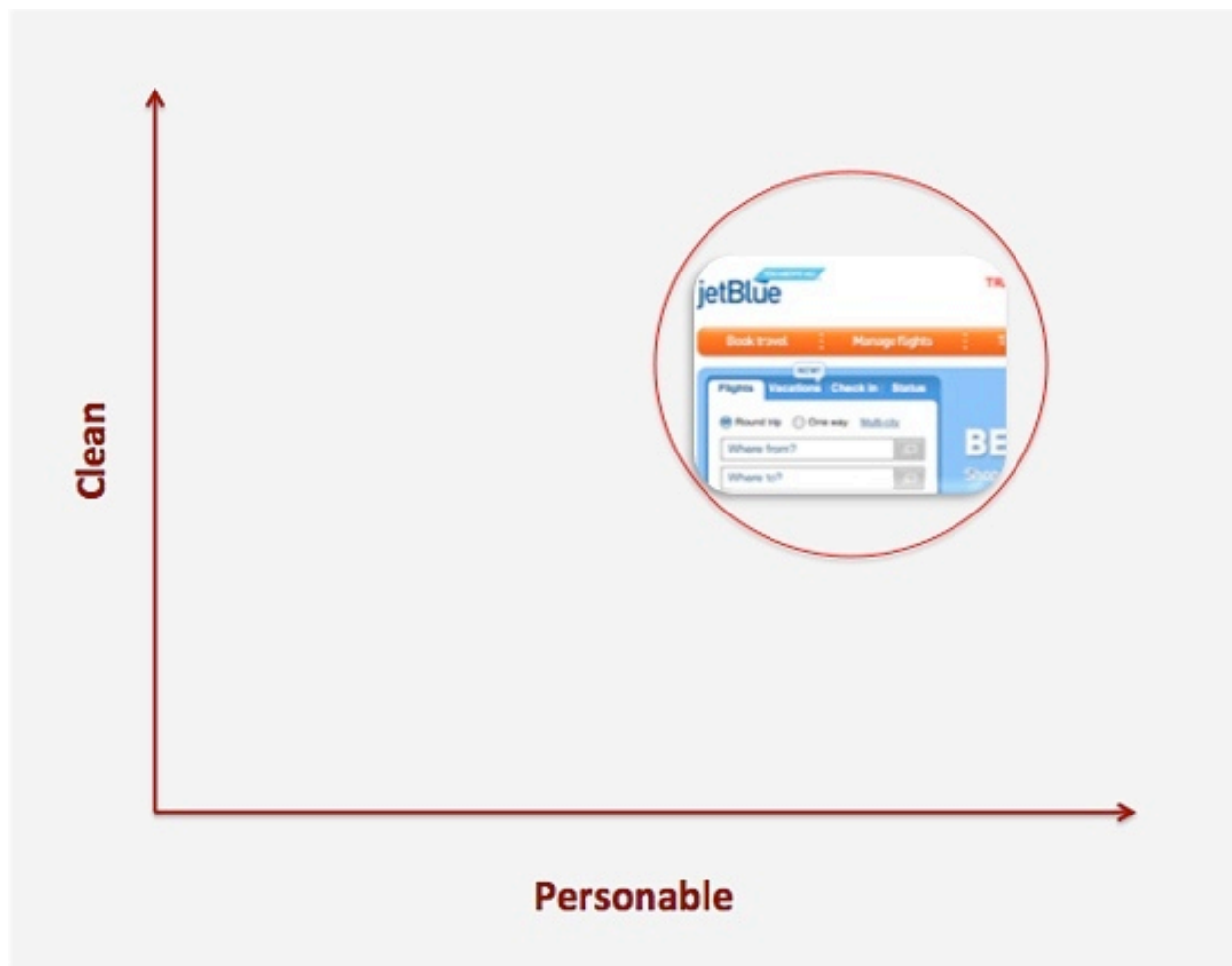
I was influenced by Jet Blue's overall branding and so originally ranked its website's personality fairly high. Later, when I compared it to the Delta and Virgin America websites, I revised the ranking.

Design matrices do not have to be limited to ranking competitors. They can also show a client's website's position among affinity brands (i.e. brands with a similar "feel" and customer base). Mini USA and Apple, for example, might be considered affinity brands because they both exemplify modern design and appeal to similar customer types.

Step 4: Get Serious

Things will take shape fairly quickly, but there is a final step before declaring your design direction matrix done and dusted. Before sharing the matrix with the client team, make sure you can defend it. Show it to others in your agency and see if they agree with your placement decisions. Ask these key questions:

- Do my axes represent the two most important design attributes?

- Can I clearly articulate why I placed each website where I did?

- Will the redesign be able to get the website to the top-right corner? If not, what is holding it back?

If you answered yes, yes and yes (or yes, yes and yes *if* we do a certain thing...), then your website design direction matrix is ready to share with the client.

Creating Buy-In With Design Matrices

Everyone loves talking about design, but with everyone talking, we don't always hear other ideas. If you show a client a design matrix before creating the initial comprehensives, then you will visibly and quantifiably show that you are on the same page; and because of that, you'll likely be successful in the long run.

The design matrix will clearly show which websites you think best capture the desired attributes and where the current website falls into the mix. It is a tangible foundation for a conversation about design.

Invite the client to participate actively in this stage of the design process. Clients usually want to feel like they have had direct input in the design, and designers always prefer that the input comes sooner in a high-level, directional form ("The design feels cold"), rather than later in an overly specific form ("Make that element blue").

Discuss the following questions:

- Does the matrix address the two most important design attributes?

- Do we all agree on the placement of competing and/or affinity brands?

- Do we all agree on the placement of the client's brand?

- If we end up in the top-right corner, are we where we want to be?

Using a design matrix can be risky, mainly for one reason. Some clients have difficulty prioritizing the two most important design elements, and then they dig their heels in and declare that there are in fact four equally important elements. Hopefully, the matrix demonstrates how your choice of attributes distinguishes the website. If you meet with a lot of resistance, just create two matrices or conduct a competitor review (as discussed in step one).

Be prepared to explain your rationale and defend your position — but also be open to suggestions. Maybe there are good reasons to focus on different attributes, or maybe the team feels that the placements of some website are not quite right. Revising a design matrix is much easier than revising a design.

Truly Going The Distance

Creating a design matrix is a great first step, and getting client feedback is an awesome second step, but the most important step is to use the matrix as a resource as you design and when you present your designs to the client.

Ultimately, the purpose of a design matrix is to move a website design in the right direction. Specifically, move it to that space in the upper-right corner that represents the best of both worlds. As you design, continually refer to the matrix and see where your new iterations might fall on it.

Think about these questions as you design, and take notes for upcoming presentations:

- Do the new iterations embody the key attributes?

- Are they better than the competing and affinity brands?

- How do they accomplish the design goals?

When presenting designs to a client, review key findings and recommendations made during the discovery phase. Before presenting your designs, review the matrix with the client, and revisit the matrix at the end of the presentation to show that progress has been made.

Training Wheels: A Step-By-Step Overview Of A Design Matrix For Cannondale

The following walk-through illustrates the design matrix process in its entirety and addresses the kinds of decisions that need to be made when creating a matrix. The exercise below is entirely theoretical. I do not work,

nor have I ever worked, for any major bicycle manufacturer, including Cannondale. Thoughts about what design attributes Cannondale might strive for are purely my opinion. Thoughts about competing website design attributes are informed by looking at their websites and general industry expertise — just as yours will be.

How would you create a design matrix for Cannondale? (This example is purely illustrative.)

Step 1: Understand

For the purposes of this exercise, let's assume that Cannondale has chosen you to redesign its website. Your first step will be to understand its brand, its competition and the desired attributes of its new website. Let's also assume that you left the initial discovery meeting with this information:

- Cannondale's key competitors are Trek, Giant, Diamondback and Fuji.

- Its brand is about performance, innovation and a superior craftsmanship that inspires riders.

When you ask about the desired design attributes, Cannondale's representatives say the website should capture the sense of elation that comes with a successful bike ride. They also want the website to showcase technical innovation, dedication to quality and devotion to the individual rider. Your notes read, "inspiring, innovative, technical, individual, quality."

Step 2: Determine Your Axes

The X and Y axes reflect the client's most important and desired design attributes, but do look at competing websites before naming the axes; they will inform your direction and give you ideas about what would be distinctive.

Upon viewing the competitor websites, I found both Trek's and Fuji's to be "immersive" and "powerful," with clean, bold imagery. Fuji's was slightly colder and more "technical." Diamondback has an inspiring home page, but the website loses steam and doesn't showcase the individual bikes distinctively. Giant has a strong focus on teams and individual riders and helpful bike selection tools, but the design is flat.

So, how does all of this play into naming the axes and creating the matrix? Going back to your note about desired design attributes, we see that Cannondale wants to showcase technical innovation, which Trek and Fuji do well on their websites; Cannondale wants to inspire, which Trek and Fuji do through immersive imagery; unlike Giant, though, Cannondale doesn't want to focus on racing.

At first, it may seem that "inspiring" and "innovative" would be good axes names, but those attributes don't have quite enough tension. They are not synonyms, but there is no balance either. "Inspiring" and "quality" may come to mind, but "quality" is not a design attribute; it's something the client wants to showcase (it's an attribute of the product, not the design).

I chose "inspiring" and "technical" for the desired design attributes. "Inspiring" works because the client wants to inspire riders. "Technical" is a good second attribute because it captures innovation and product quality while striking a balance with "inspiring." There is a healthy tension between the two words. Capturing both emotion and technical detail is difficult. Accomplish that balance and you'll leave the competition in the dust.

Step 3: Place Websites on the Matrix

I always start by plotting all of the websites roughly where I think they fall on the matrix, and then I move them around as I consider the relationships between the websites.

I originally placed Cannondale's website in the lower-left corner but, as I compared it to the other websites, I realized that it's actually more technically focused than others, including Diamondback and Giant. That said, the Trek and Fuji websites are still more technical, with their bold product showcases and detailed imagery.

Inspiration-wise, the current Cannondale website seems to be on par with Giant's: there is imagery of bikers, but it feels flat and diminutive. Diamondback's immersive home page raises it a bit above the others. Trek's warmth and voice put it in the lead for inspiration. I originally had Trek in the upper-right, but I ultimately decided that Fuji's website has a more technical feel to it.

The final matrix (below) is informative on many levels. It shows where Cannondale currently is and where the websites are that it needs to surpass in order to get to where we determined it needs to go. Naturally, the final design will have a unique flavor, but looking at the competing designs will partly uncover how to get there.

A design matrix informs the path of the design process.

Step 4: Consensus

This is a purely illustrative example, so I did not show this to a team (or the client) for feedback. Typically, feedback focuses on the desired design attributes ("Is this where we want to go?") and the placement of all of the websites on the matrix. The most important thing is to agree on direction, of course, and then to determine the goal. The hard part is to design a website that gets there.

Step 5 (the Big One): Using It

The last step is not so much a step as a big stride. Once you've created the matrix, the important part comes: using it to create a better website. Make a copy for everyone involved in the project (including those in other disciplines) and have them put it up somewhere to serve as a daily reminder and motivator.

Refer to the matrix as you design. Are the decisions you are making moving you toward the upper-right? For example, if you were choosing images for the Cannondale website, ask yourself relevant questions:

- Is this image inspirational?

- Does it convey the technical expertise of Cannondale?

- How can the design be more inspiring?

- How can I better convey the technical passion of the brand?

As mentioned, revisiting the matrix when showing comps to the client will help justify your approach, but the real reason to create a matrix isn't to sell comps or do a fun exercise during discovery; rather, it is to remind us of the

path we are on. We could take so many directions, and going down a road that looks good but doesn't take you where you want to be is all too easy.

Think of your design matrix as a compass. It's not as precise (or cold) as a GPS; it's an old pocket compass that wobbles a bit as you walk but still gets you to the summit.

Do's And Don'ts

The beauty of design matrices is that they provide a new way to look at competitors and a tangible foundation on which to begin discussions with clients. They also enable you to play a little as you tweak the axes' names and the websites' placement to get them just right. There is wiggle room in the methodology and application, so have fun with it. That said, there are a few set guidelines worth adhering to for success:

- **Don't be afraid to experiment.**

- **Do get your ducks in a row.** Verify the desired design attributes and the competitor and affinity brands with a client before proceeding, so that the matrix is relevant.

- **Don't base your insights on home pages alone.** A website's design is more than the home page. Your matrix might use a home page screenshot, but include it only if it represents the overall design of the website.

- **Do share your toys.** Get team input about the placement of websites on the matrix. It's not an altogether scientific approach, but be as objective as possible.

- **Don't carve it in stone.** Be open to recommendations from clients. Changing a matrix is easier than changing a comp.

- **Do use it to sell your work.** Present the matrix as part of your comp presentation in order to explain your rationale and sell your comps.

- **Don't matrix and run.** Don't abandon the matrix after the discovery process. Refer to it regularly.

Enjoy the process of creating a matrix and of seeing opportunities to design a distinctive website for your client.

A Design Is Only As Deep As It Is Usable

Louis Lazaris

There are well-known proverbs that imply (or state outright) that beauty is superficial and limited in what it can accomplish. "It's what's inside that counts" and "Beauty is only skin deep" are a few simple examples. Because the Web design industry is now flooded with a lot of raw talent, and because virtually anyone can create a "beautiful" website, recognizing a truly beautiful website experience is becoming increasingly difficult. What appears beautiful to the eye might in fact be more of a hindrance.

In this article, I hope to provide a clear demarcation between what is perceived by most to be beautiful in Web design and what is truly beautiful, along with some guiding principles to help designers today create websites whose beauty is not superficial, but rather improves and enhances the user experience.

Gradients, Drop-Shadows, Reflections, Oh My!

A lot of things could fall in the category of "beautiful" or "attractive" in the context of Web design. But a number of factors would make such beauty shallow. Is a website more attractive if it has tastefully placed drop-shadows, gradients or reflections? What if it has an eye-pleasing color scheme? What about big over-designed buttons? Could these be standards by which a design would be deemed beautiful?

If you've been keeping tabs on the Web design industry in the last five years, you've probably at some point visited one of the many CSS galleries. Visiting those inspirational showcases is great, and I'm sure we've all done

it, but we need to be careful not to fall into the copycat syndrome, whereby we prettify our websites for no other reason than to make them CSS gallery-worthy.
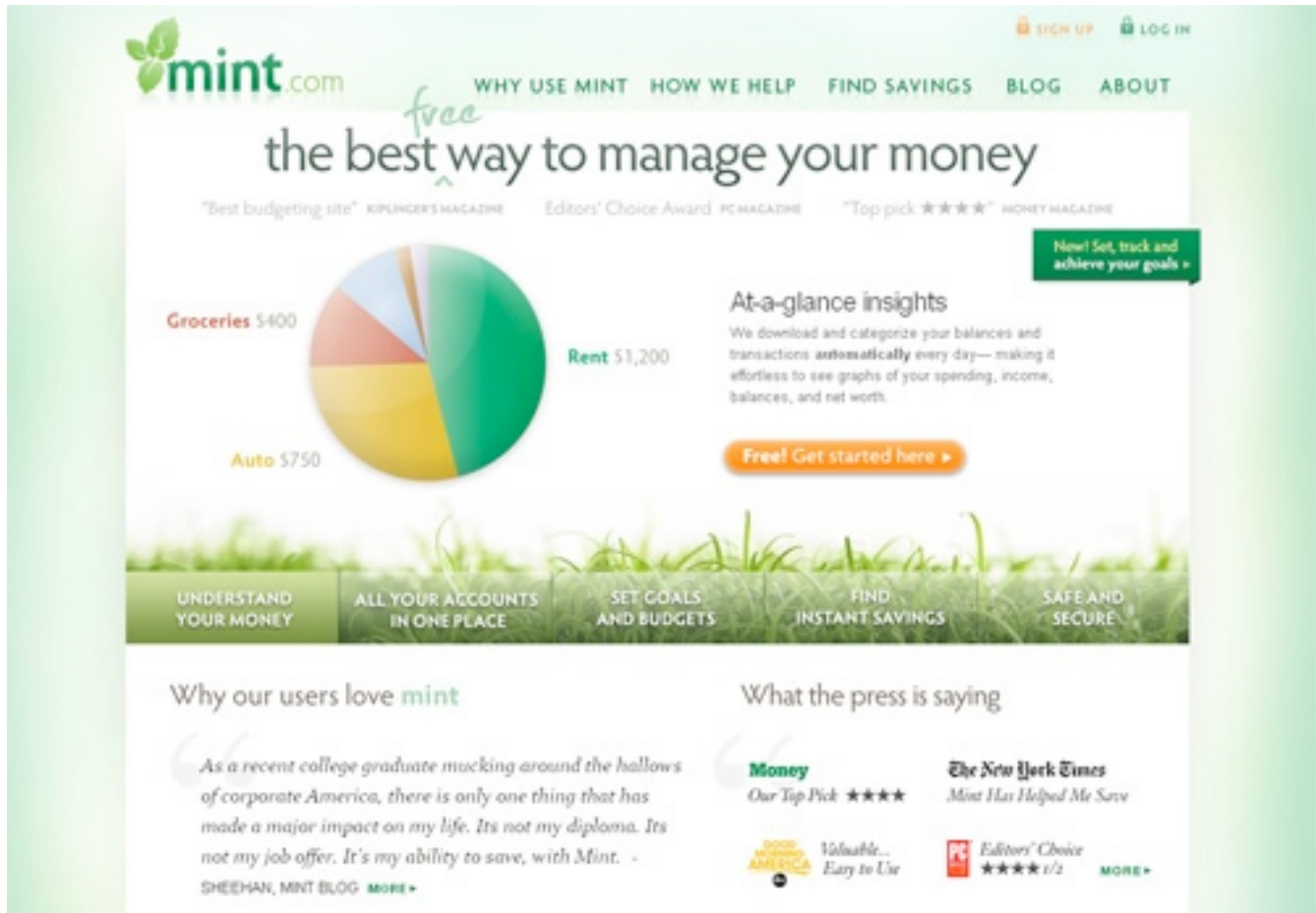
Mint.com has everything a client could ask for in a "Web 2.0 design". Does that mean it's beautiful?

The designers, developers and content strategists who planned and executed many of the websites in those galleries did what they did because they felt it would truly benefit the user experience and their clients' bottom line. The truly beautiful websites and apps in those showcases are not just visually beautiful; they're usable, accessible and optimized to benefit both the user and website owner.

The Dribbble Syndrome

With the recent popularity of *Dribbble*, the copycat syndrome might be gaining momentum. On Dribbble, a designer reveals a sample of something they're working on, and then the style of that small snippet starts spreading. The context and strategy underlying it are unknown, yet the style is still viewed as beautiful in and of itself. The designer may have taken hours, days or weeks to arrive at the decisions that informed the design, but now that it's out in the wild, the snippet becomes nothing more than eye candy.
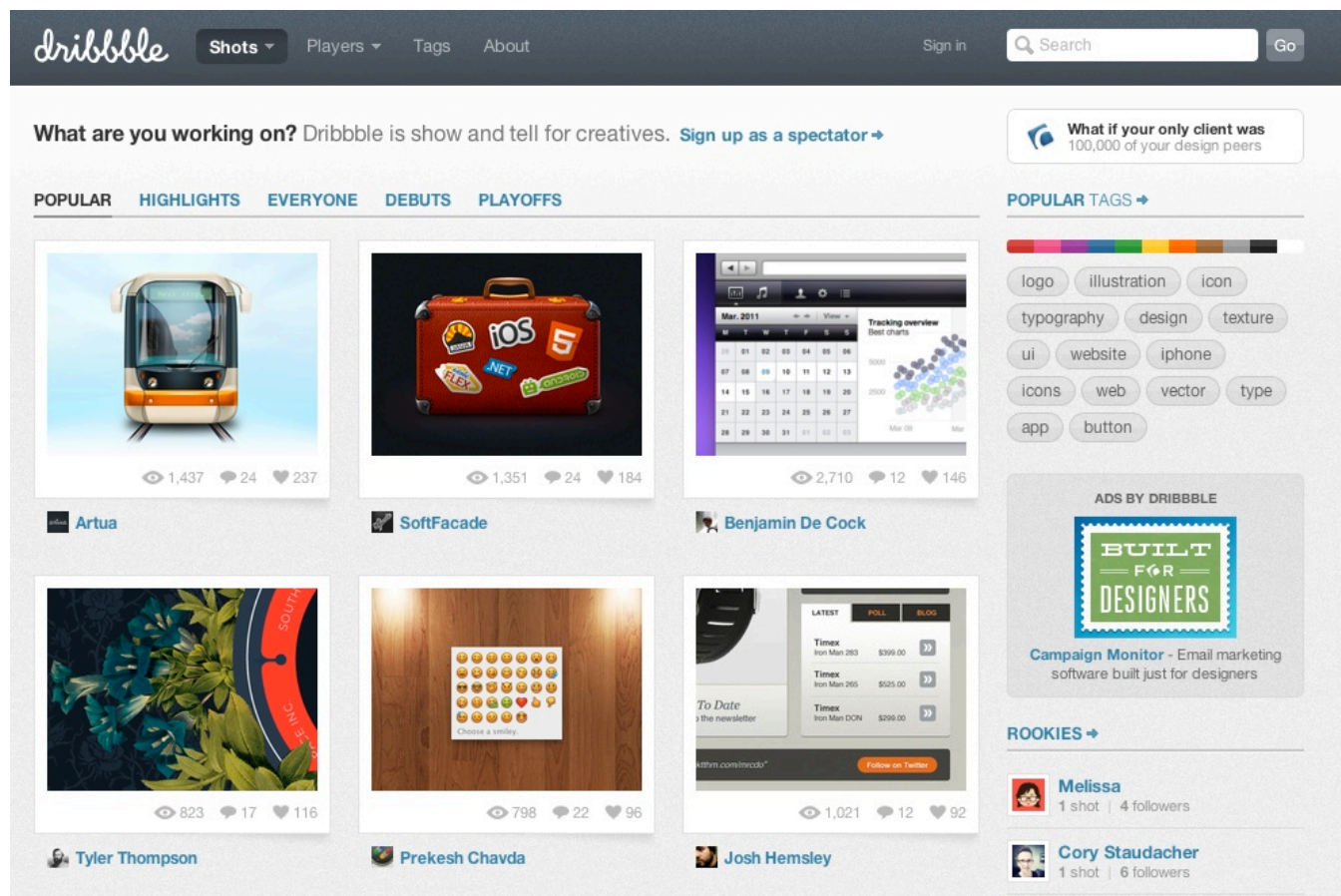
Dribbble shows out-of-context design shots. Is this a bad thing?

Of course, the intent of this article is not to blame those who share their designs on Dribbble, nor to blame those who review these designs and offer feedback. But we mustn't lose sight of the fact that every design decision should have significant reasoning behind it.

The Style-Less Comparison

How do we measure beauty? If a website is difficult to use, then isn't its beauty without purpose? Look at the comparison in the image shown below.
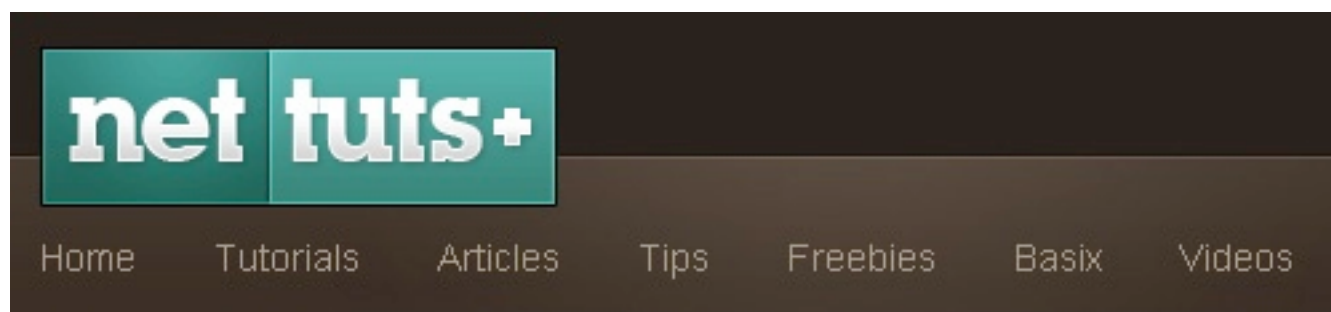
The Nettuts+ logo and navigation bar.

I think Nettuts+ is a very nicely designed website. But is the fancy navigation and logo section shown on top more usable than the plain blue

and white version below it? Taken at face value, some might argue that the plain version is more usable (if only slightly) than the "beautiful" one.

The [Facebook](#) home page.

While the Facebook home page shown on top might not appear the most beautiful design to many of us, it still contains attractive aesthetic elements (colors, gradient background, styled buttons, etc.). But when most of these minor elements are made plain, does it really affect the usability (of course, after you increase the color contrast for the form labels in the right upper corner)?

If prettiness is really as important as we think, then the current Facebook home page should perform much better than the plain alternative. How do we know, though, that the plain version wouldn't outperform the adorned version?

What Makes A Design Usable?

I'm not about to make a case for bringing back blue links on a white background on every website. In fact, as I'll explain, both Nettuts+ and Facebook may very well qualify as truly beautiful websites. The examples above were more illustrative, and not meant to criticize the designers who worked on them.

Rather, I'm encouraging designers to consider two things when adding "beautiful" enhancements to their designs.

- Responsive and intuitive page elements,

- Branding and consistency of theme.

Focusing on these two things will give every pixel in a design a purpose and will contribute to the website's overall usability. Let's consider both of these, with a few simple examples to illustrate their effectiveness.

Responsive and Intuitive Page Elements Make a Design Usable

If a design element makes a website feel more friendly or gives subtle hints as to what's happening, then this adds to its usability. Look at the simple example below from Design Informer:
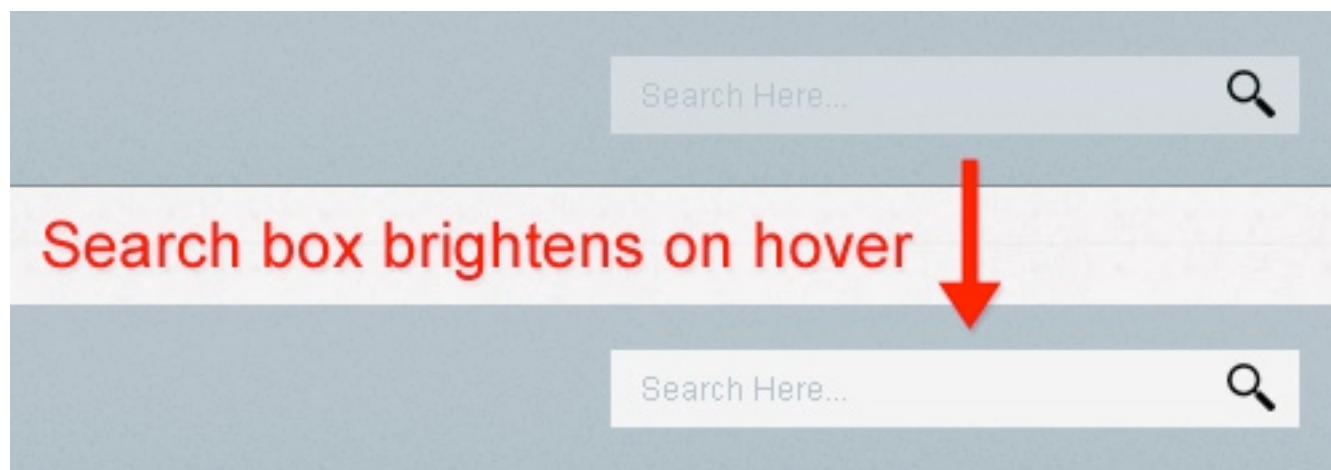
On the Design Informer website, hover over the search box in the top right, and you'll notice it brightens up. This is not intrusive in any way, and it looks especially elegant in WebKit browsers, because the brightening animates with CSS3. The default look of the search box could be a bit brighter to improve the general usability of the site, but in this specific case the idea counts more than the execution.

This very simple effect conveys to the user that this is a usable element, and it makes the search box more inviting. It's a ridiculously simple technique but has a very powerful effect.

But just because you can use an animated effect does not mean you should. If, as in the case of Design Informer, the effect makes the UI more intuitive and responsive, then it is justified. This statement by Stuart Thursby sums it up well:

"If designers think that using HTML5 and CSS3 makes them a better designer just because they use them, then they're sorely misguided."

Include an element only if it accomplishes some usability-related purpose. If the design is not made more usable by a particular technique (whether via CSS3, JavaScript or something else), then the designer should reconsider whether the extra code is worth the effort. Decoration only goes so far and often has an effect opposite to the one intended, so consider yours carefully before including it in your design.

Another example of an animation that enhances usability is found on Soh Tanaka's new website. Look at the screenshot below from this post on his blog:
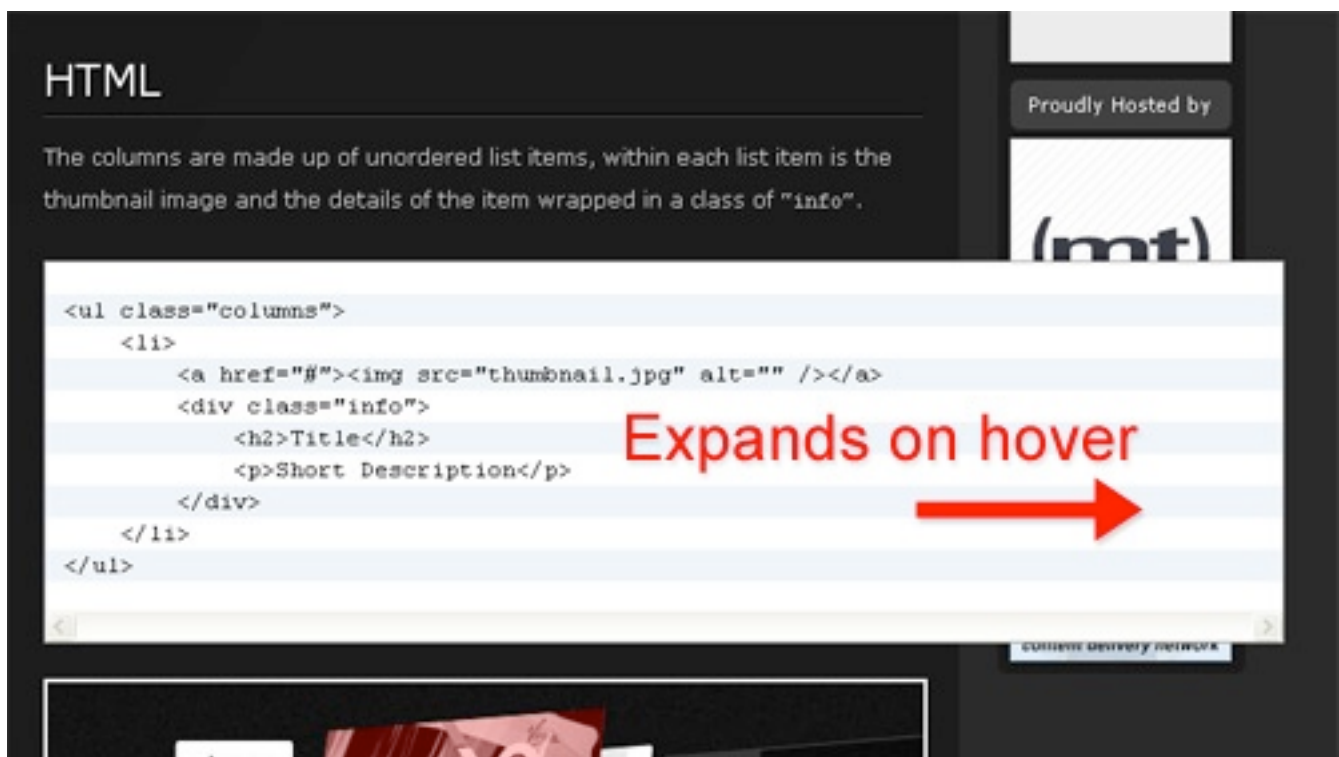
When you hover over any presentation of code on his website, you'll notice that the block expands to the right (probably via jQuery, so it would work in every browser).

Again, a simple effect, but not just eye candy; it has a purpose. In tutorials, HTML code is often too long to fit in the highlighter, so the code either wraps or creates ugly scroll bars. Tanaka's solution makes the code more inviting and readable, and it decreases the likelihood of wrapping or scroll bars.

So whether we're talking about text links that change color on hover, buttons that move when clicked, AJAX that creates subtle yet intuitive effects, we can take a design beyond mere decoration in many ways and truly enhance its usability.

Branding Makes a Design Usable

If an element contributes to a website's overall branding, image or reputation, then it's safe to say that it contributes to its usability. Properly planned and executed branding is not superficial or decorative. Carefully chosen colors and graphic elements create an inviting atmosphere that leads the user to make easy decisions and helps them interact with elements smoothly and intuitively.

Look at the screenshot below from 10k Apart:

The laurel wreath in the background and the distinctive illustration immediately distinguish this website as belonging to A List Apart. Consistency in branding contributes to the usability of this ALA microsite and makes it feel inviting and familiar.

And then we have the beautiful and intuitive design for Launchlist:
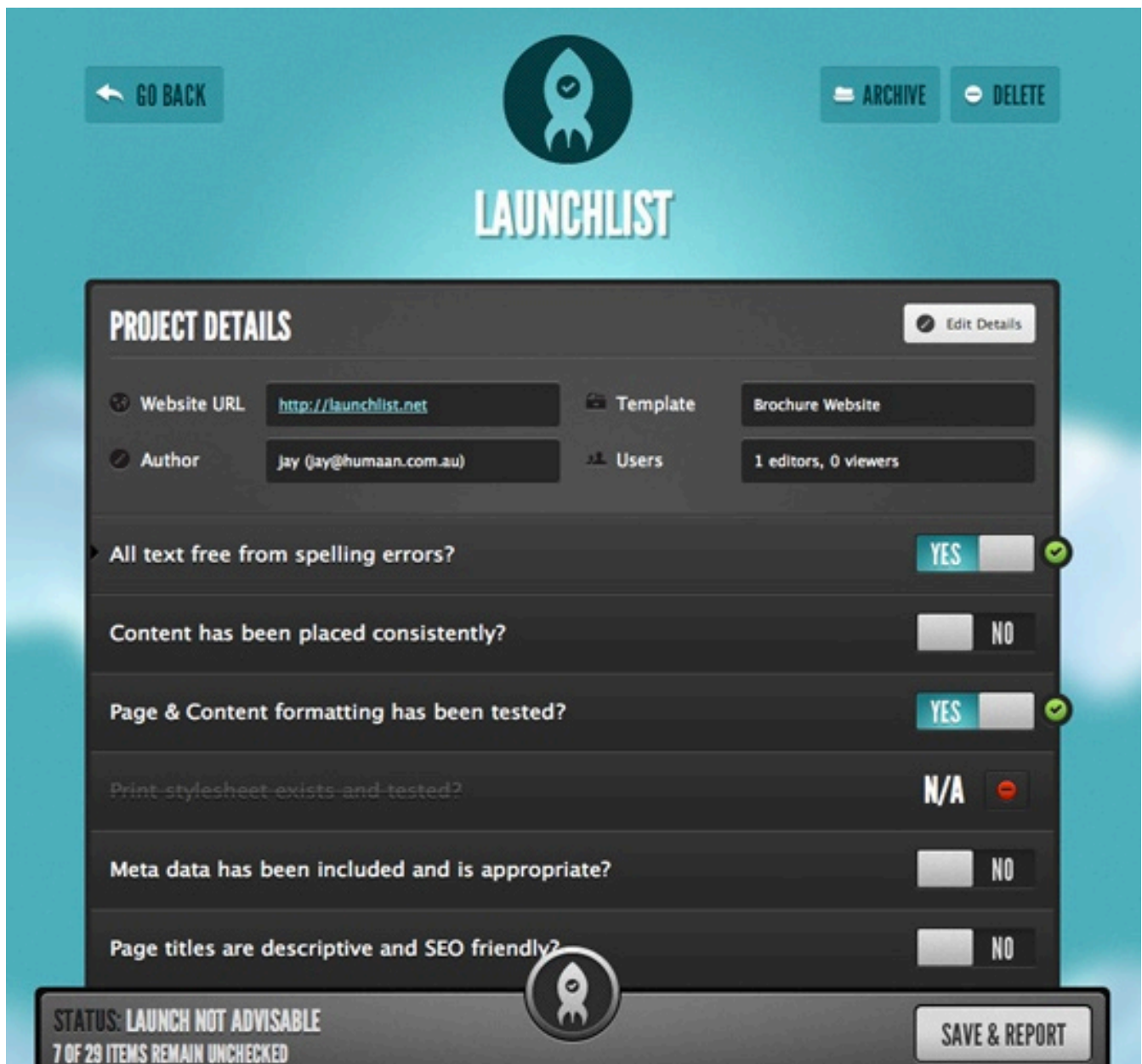
This screenshot doesn't do justice to the website's look and feel; you'll have to poke around to really experience it for yourself. The design might appear decorative and superficial at first glance, but it's not. The elements work together to create a consistent and inviting atmosphere, extending the "launch" theme throughout with subtle animations.

Usable Doesn't Have To Mean Ugly

My purpose here was not to tell designers to forget about slickness, sexiness and beauty. This should be obvious from the beautiful examples shown, which certainly qualify as both usable and attractive. No one expects owners of beautiful websites to suddenly drop their enhancements in favor of the Craigslist look just to make them more usable.

Rather, this article is just a reminder that eye candy is important, but it isn't everything, and that for a design to be truly beautiful, it has to be functional, have purpose and contribute in some way to the website's intuitiveness, usefulness and branding. All of these things contribute to the overall effect of a design.

Web Designers, Don't Do It Alone

Paul Boag

Whether freelancers, small agency founders or website owners, too many of us work alone. The downside of the digital revolution is isolation. The Web allows us to do alone what previously would have required a team of people. It also frees us from the constraints of geography, allowing us to work from home. But while these are benefits, they also leave us isolated.

The Dangers Of Isolation

Over time, working in isolation (even if you function as part of a team) can prove harmful to your mental health, business and website. In fact, even if other people are working on a project of yours, if they are junior to you, you can still feel isolated.

If you don't find a peer with whom you can share ideas and discuss your business or project, you face a number of dangers:

- **Dry up creatively**
  Creativity is born of interaction. Being consistently creative on your own is hard. The best ideas come from people brainstorming together and from one great idea leading to another. Without someone to bounce ideas around with, your business or project will lack a creative spark.

- **Lose confidence**

  Over time we can lose confidence in our abilities or our business. This is especially true when we make mistakes and things go wrong. Without someone to encourage and reassure us, we can begin to second-guess our decisions.

- **Become over-confident**

  While some suffer from a lack of confidence, others are over-confident and need to be challenged and questioned. This is a trait I suffer from; I would happily dive headlong into disaster if my fellow directors did not constantly question my ideas. Without people like this, moving your business in entirely the wrong direction would be too easy.

- **Reach the limit of your knowledge**

  We can't all be experts at everything, and yet running a website and a business requires a broad range of skills. When working in isolation and tackling problems beyond your comfort zone, you can easily reach the limit of your expertise and flounder.

- **Have a blinkered perspective**

  Another problem with working alone is that you have only a single perspective on your work. By adding another set of eyes to your problems, you gain a broader vision and can approach your challenges from a different angle.

- **Feel overwhelmed**

  Running a business or a business-critical website can feel like a burden. You are often required to make big decisions, particularly with hiring and expenditures. Making these decisions alone is a big responsibility and can be really scary. Having someone to share that with would make a big difference.

So, can you identify with any of these traps? If not, then I suggest you read the one about over-confidence again! I don't believe a single website owner or entrepreneur couldn't benefit from an outside perspective.

The question, then, is how do you find someone?

Getting An Outside Perspective

The most obvious solution is to partner with somebody at the outset. Whether you work with someone on a website or form a business with an associate, partnerships can be very beneficial. This is what I did with our company, and I haven't regretted it for a minute. I would be lost without my two co-founders, Chris and Marcus.

That said, I know that not everyone's experiences with partners have been rosy. Also, by the time you read this, the opportunity for this kind of partnership may have already passed.

What can you do then? What other options are available to those seeking an outside perspective and someone to bounce ideas around with?

Here are some options:

- **Sleeping partner**
  This is the approach we took. We have a non-executive director named Brian who works with a number of companies and keeps us on our toes. He has a radically different view of business and constantly challenges us. In return, he has a small stake in the business. He is worth every penny.

- **Paid consultant**

  If you don't fancy having someone so entrenched in your business, why not consider an external consultant with whom you could speak on an ongoing basis? Admittedly, this kind of consultant can be pricey, but they do bring an outside perspective to the table.

- **Mentor**

  Another option is to approach a Web designer or website owner you admire and ask them to mentor you. Obviously, these people are probably busy with their own work, but if you are willing to pay for their time, you might get some valuable advice. You'll usually need only an hour per month to stay on the right track.

- **Buddy**

  A buddy would be a cheaper option, someone in a situation similar to yours. The two of you could agree to chat regularly and share the challenges you face as business or website owners, discussing different approaches and ideas.

- **Community**

  Yet another option would be to look not for a consultant, mentor or buddy, but for a supportive online community. Loads are around, but make sure the one you join is not too big. You want people to remember you and your circumstances.

Whatever you decide is entirely up to you. The point is, if you want to realize the potential of your website or business, you need the help and encouragement of others. Humans by nature work best in groups, and you are no exception. We are not meant to do it alone!

Why Design-By-Committee Should Die

Speider Schneider

No matter where you go in the known universe, there is design-by-committee. It has become a pecking order of disaster for the society that used to pride itself on being a mover and shaker and that allowed its mavericks and dreamers to innovate their way to success. In a business climate fueled by fear and the *"Peter Principle,"* as it is today, a decision not made is a tragedy averted. So, decision by committee provides a safe and often anonymous process for finger-pointing down the line… inevitably leading to the creative, of course.

Why It Happens

[Wikipedia describes it](#) thus: The Peter Principle is the principle that "in a hierarchy every employee tends to rise to his level of incompetence." It was formulated by Dr. Laurence J. Peter and Raymond Hull in their 1969 book The Peter Principle, a humorous treatise which also introduced the "salutary science of Hierarchiology", "inadvertently founded" by Peter. It holds that in a hierarchy, members are promoted so long as they work competently.

Sooner or later they are promoted to a position at which they are no longer competent (their "level of incompetence"), and there they remain, being unable to earn further promotions. This principle can be modeled and has theoretical validity. Peter's Corollary states that "in time, every post tends to be occupied by an employee who is incompetent to carry out his duties" and adds that "work is accomplished by those employees who have not yet reached their level of incompetence.

Whether on staff or freelance, we all walk into meetings prepared for our work to be torn to shreds. And it always is. The client sits there trying to explain to you how a logo the size of a small melon should sit on a 9×12-inch ad.

Our core competency is in creating something that is the perfect communication vehicle for the given message. But then subjectivity walks in the door, and the creative is left standing there, looking like an incompetent who needs a committee to complete their work.

Others Have Noticed Its Effects

Michael Arrington, founder and co-editor of TechCrunch, a blog covering Silicon Valley technology, and a widely respected and influential person on the Web, recently wrote:

> There's a saying I love: "a camel is a horse designed by committee." A variation is "a Volvo is a Porsche designed by committee." Some of the best product advice I've ever heard goes something like "damn what the users want, charge towards your dream." All of these statements are, of course, saying the same thing. When there are too many cooks in the kitchen all you get is a mess. And when too many people have product input, you've got lots of features but no soul.

Through it all, I've heard some wondrous and magical statements come from the mouths of non-creatives as they "join in on the fun" of designing in these dreaded committee meetings.

My favorite exchange to date happened in a meeting that a secretary sat in to take notes but who eventually took over the conversation. I looked at her and then the art director, who sat sheepishly quiet (from too many

emotional beatings, no doubt), and asked why a secretary would be allowed to give design feedback. She pulled herself up in her chair and said, "Well, you do want this to be the best product it can be?"

"The best it can be." She was somehow convinced that her opinion overshadowed all others, including those of the art staff. In her mind, she was actually saving the design. Stories like this abound.

You're Not The Only One

Wanting to feel I was not alone, I posed the question to the art directors among my umpteen connections on LinkedIn. The responses were varied, passionate and maddening at times. One of my favorite Los Angeles art directors gave me a list of her favorite sayings overheard in committee meetings:

My wife wants more circles.

My husband says it doesn't hit him in the gut.

My kids say there are too many words.

My dog didn't wag its tail.

The waiter said he's seen something just like that in France.

I need more oopmh in it.

I'll know it when I see it. So go back and make more.

I love what _____ did. Can you do the same, but with carrots?

What are you doing after work?

The next respondent to my question asked, "Did you forget to take your meds today?" Another chimed in, "I don't want to give you any stories because I don't want to cry!"

One creative director added these: "Why isn't my logo bigger?", "Why can't we use all of this empty space over here?" and "It's too promotional". He adds: "Anything from anyone who's ever said, 'I'm not creative, but...' or 'It needs more... something.' And anything from anyone who 'knows what they don't want but has to actually see what they do want because they can't describe/direct/vocalize it'."

Plenty of responses advised us to let go and just take the fee and do whatever the client or committee wants. This is a "service industry" after all. One graphic designer wrote:

> *One thing I try to do is understand why certain decisions have been made, and I do this by questioning the person doing the direction (this could be a colleague, sales person, client, etc.). If that person has legitimate reasons for asking for specific things, and they can back up that it will work, I'd like to know.*

Another voice added, "He who pays calls the tune, even if they're wrong, and even if they have poor taste. That is important to keep in mind."

As much as I agree, there is still that voice inside me that screams bloody murder at the idiocracy of group decisions. Feeling the same way, an art director in Texas wrote, "The client may pay for the work, but who takes the blame when the client campaign fails miserably because the client did not listen to the advice of the designer?"

Who Should Ultimately Decide?

For better or worse, I agree with another passage in Mr. Arrington's article:

> *Product should be a dictatorship, not consensus-driven. There are casualties, hurt feelings, angry users. But all of those things are necessary if you're going to create something unique. The iPhone is clearly a vision of a single core team, or maybe even one man. It happened to be a good dream, and that device now dominates mobile culture. But it's extremely unlikely Apple would have ever built it if they conducted lots of focus groups and customer outreach first. No keyboard? Please.*

He also illustrates his point brutally with this hard fact:

> *Digg is sort of on the opposite end of the spectrum. The company has been standing still now for years as Facebook, Twitter and others have run laps around it. But the company is famous for listening to its hard core fanatical users.*

My point is best made through the brilliant, funny, intelligent Better Off Ted. In one adventure, the corporation empowers everyone to make decisions about products in committee. See what happens to the simple product. The always classic "Process (aka Designing the Stop Sign)" is another frightening example soaked in truth.

Marketing aims to create consumer interest in goods and services based on the assumption that the target consumer is buying a lifestyle or habit, with some income, location and loyalty considerations thrown in. It draws from information about the target demographic; however, personal preferences about color, type size, logos and so on do not represent those of the target demographic. One person on a committee *might* be a target consumer, but

certainly not the committee as a whole. Should people from disparate demographics second-guess the visual approach taken by the designer to the target consumer?

Mr. Arrington believes that the plan trumps all voices. His article ends with a very assertive video about winners and losers. Most creatives choose to let it wash over them and collect their pay check. I suppose I don't agree because I haven't seen many pay checks made out to "Dance, monkey, dance!"

What's The Solution?

From all the responses and stories, it seems there are few ways to live with the design-by-committee lifestyle. Suggesting what a marketing plan or piece of copy is missing or implying that the secretary is unable to spell will only get you pegged as "difficult" and make you appear as though you "overstep boundaries." Asking a non-creative who gives you excruciating input why they think you're incapable of doing your job will brand you as "defensive" and "combative." Give in, and you'll earn descriptions like "flexible" and "easy to direct."

The sensible answer is to listen, absorb, discuss, be able to defend any design decision with clarity and reason, know when to pick your battles and know when to let go.

A photographer I know once said, "I'll give the model a big mole on her face, and the committee focuses on that and are usually satisfied with the momentous change of removing it and leave everything else as is."

Whether you're on staff or freelance, the political dance of correctness and cooperation brings a new story and new experience every day. And isn't

that one of the great things about this business… even if it goes around and around sometimes? You can just blame someone using the new buzzword, "Commidiot," which is a committee member who has no idea what is going on in front of them but feels they have to say something of importance to justify their presence in the room.

Design Better and Faster with Rapid Prototyping

Lyndon Cerejo

The old adage, "a picture speaks a thousand words" captures what user interface prototyping is all about: using visuals to describe thousands of words' worth of design and development specifications that detail how a system should behave and look. In an iterative approach to user interface design, rapid prototyping is the process of quickly mocking up the future state of a system, be it a website or application, and validating it with a broader team of users, stakeholders, developers and designers. Doing this rapidly and iteratively generates feedback early and often in the process, improving the final design and reducing the need for changes during development.

Prototypes range from rough paper sketches to interactive simulations that look and function like the final product. The keys to successful rapid prototyping are revising quickly based on feedback and using the appropriate prototyping approach. Rapid prototyping helps teams experiment with multiple approaches and ideas, it facilitates discussion through visuals instead of words, it ensures that everyone shares a common understanding, and it reduces risk and avoids missed requirements, leading to a better design faster.

The Rapid Prototyping Process

Rapid prototyping involves multiple iterations of a three-step process:

1. **Prototype**
    Convert the users' description of the solution into mock-ups, factoring in user experience standards and best practices.

2. **Review**
    Share the prototype with users and evaluate whether it meets their needs and expectations.

3. **Refine**
    Based on feedback, identify areas that need to be refined or further defined and clarified.
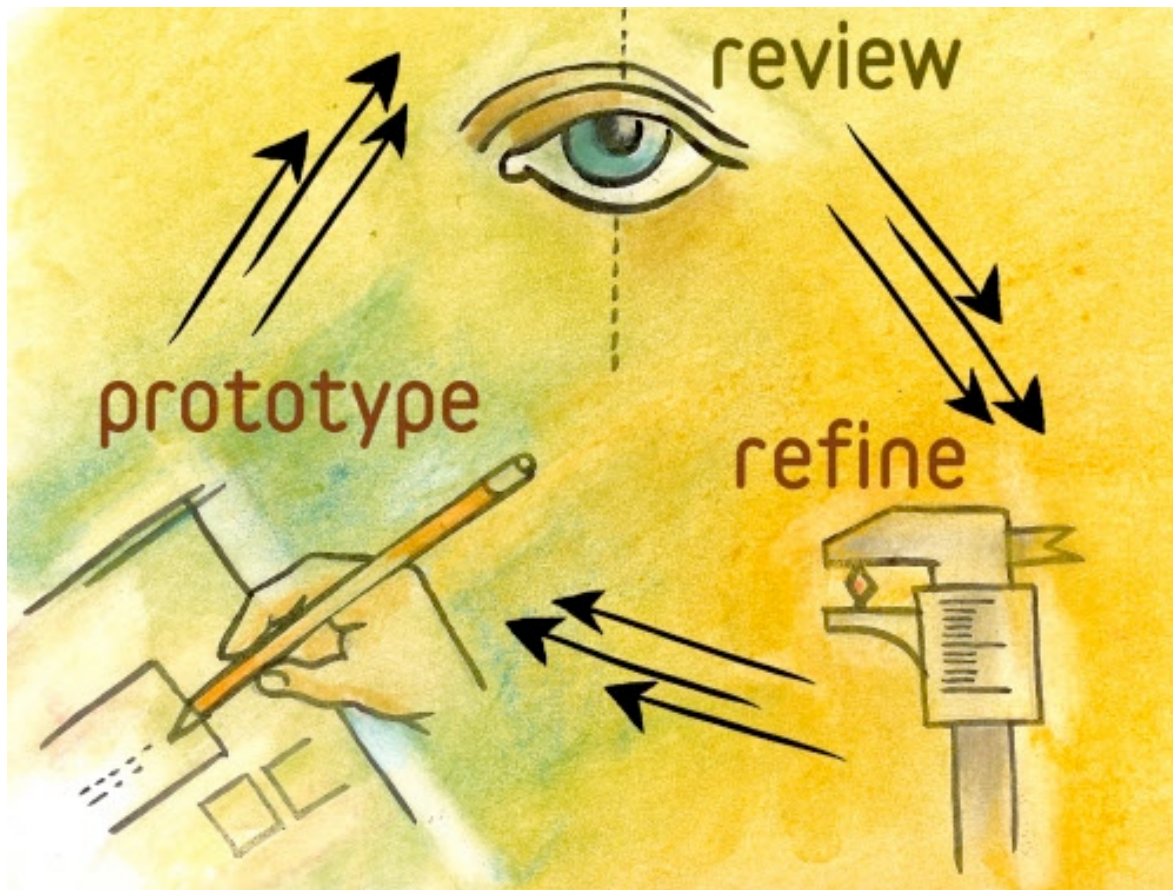
The prototype usually starts small, with a few key areas mocked up, and grows in breadth and depth over multiple iterations as required areas are built out, until the prototype is finalized and handed off for development of the final product. The rapidness of the process is most evident in the iterations, which range from real-time changes to iteration cycles of a few days, depending on the scope of the prototype.

Scoping A Prototype

The word prototype often conjures images of a coded, fully functioning version of an application or interface. Rapid prototypes are not intended to evolve into fully functional solutions, but are meant to help users visualize and craft the user experience of the final product. With that in mind, when scoping a prototype, decide on a few key issues before beginning any prototyping work.

What Needs to Be Prototyped?

Good candidates for prototyping include complex interactions, new functionality and changes in workflow, technology or design. For example, prototyping search results is useful when you want to depart significantly from the standard search experience; say, to introduce faceted search or the ability to preview a document without leaving the search results.

How Much Should Be Prototyped?

A good rule of thumb is to focus on the 20% of the functionality that will be used 80% of the time; i.e. key functionality that will be used most often. Remember, the point of rapid prototyping is to showcase how something

will work or, in later stages, what the design will look like, without prototyping the entire product.

Find the Story

After identifying the areas to be prototyped, weave them together into one or more scenarios: identify the coherent paths through the user experience that the prototype simulates. For a website that sells shoes, one scenario could be "Boring Joe" buying the exact same Nike running shoes that he bought six months ago, while another scenario could be "Exploring Sam" browsing through size 10s to find a pair of Oxfords and pair of loafers that interest him.

Plan Your Iterations

The entire prototype is usually not built in a single iteration but rather piece by piece. A good approach is to start prototyping broadly and widely and then dive deep into selected areas of the solution. For a website, this would mean building out the home page and landing pages for the main sections in the first iteration (sometimes referred to as a horizontal prototype) and then reviewing and revising that framework. Subsequent iterations could drill down into one or more sections of the website (a vertical prototype); for a media download website, this could be the steps a user would take to find a video and to download it, or how they would manage the media in their online library.

Choose the Appropriate Fidelity
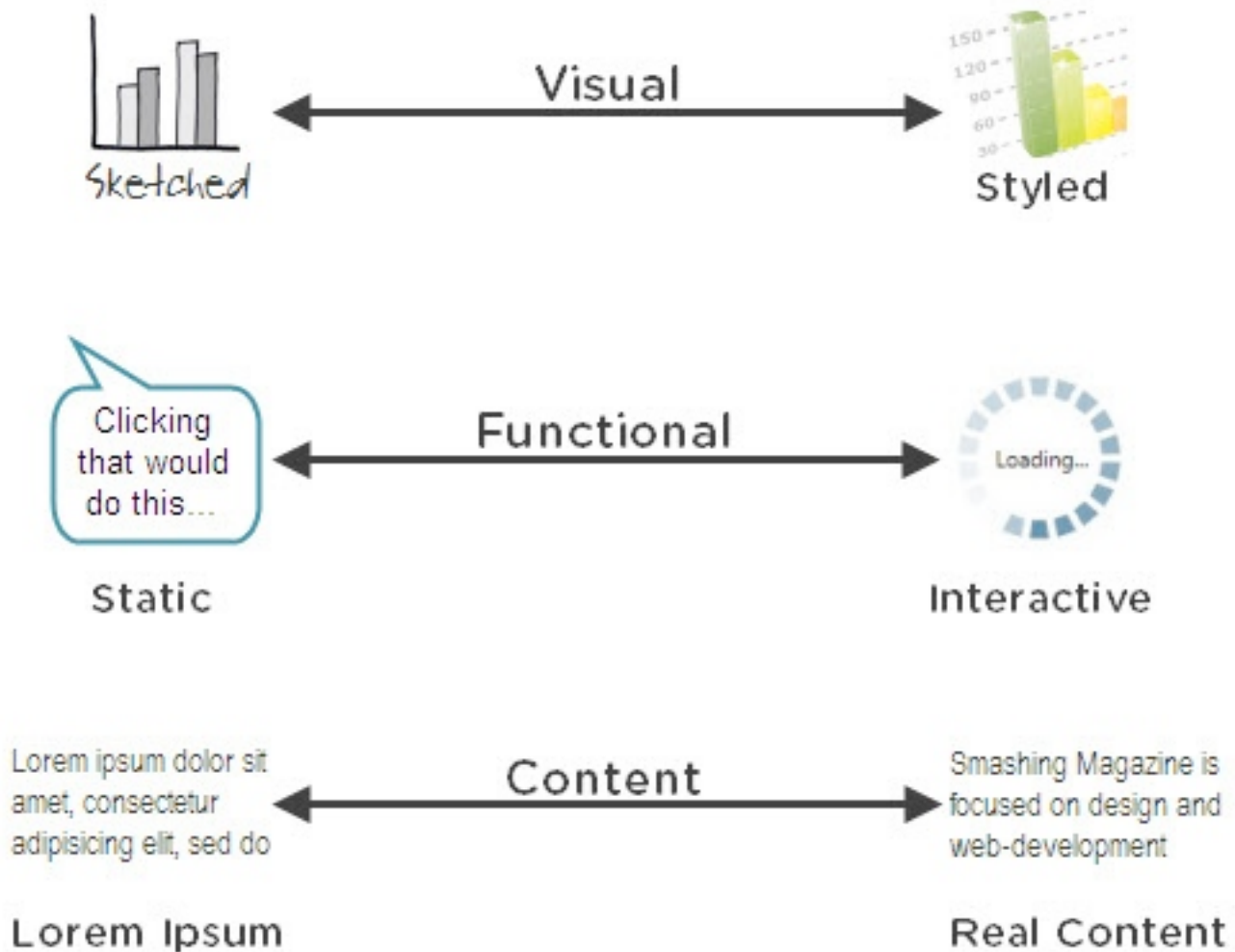
Fidelity refers to how closely a prototype resembles the final solution. There are multiple dimensions of fidelity, and prototypes can lie anywhere on the spectrum for each of these dimensions. Depending on the stage of the design process and the goals of the prototype, select the appropriate fidelity for each of the following:

- **Visual fidelity (sketched ↔ styled)**

  Look and feel are the most noticeable dimension of a prototype's fidelity and, if not properly selected, can sidetrack prototype reviews. Go hi-fi too soon and users will focus on visual design, which is not appropriate in early stages. From a visual standpoint, prototypes do not have to be pixel perfect but should be proportional; for example, if the left navigation area has to occupy one-fifth of a 1024-pixel screen, it does not need to be exactly 204 pixels wide, as long as it is proportionally depicted in the prototype. As prototyping progresses through the design cycle, increase visual fidelity as needed by introducing elements of style, color, branding and graphics.

- **Functional fidelity (static ↔ interactive)**

  Does the prototype reveal how the solution will work (static) or does it appear to be fully functional and respond to user input (interactive)? This dimension is less of a distraction to users, but adding interactivity in subsequent iterations increases functional fidelity and allows the prototype to be used for usability testing and training and communications.

- **Content fidelity (lorem ipsum ↔ real content)**

  Another dimension that often distracts users is the content that is displayed in the prototype. Squiggly lines and dummy text like lorem ipsum are useful to avoid in early stages of prototyping. But as the prototype is refined, evaluate the need to replace dummy text with real content to get a feel for how it affects the overall design.

The Prototyping Spectrum

Low Fidelity

The quickest way to start prototyping is also the easiest: putting pen(cil) to paper. Sketching on paper is a low-fidelity approach that anyone can do; no special tools or experience required. Most often used during the early stages of a design cycle, sketching is a quick way to create rough mock-ups of design approaches and concepts and to get feedback from users. Paper prototyping is ideal during brainstorming and conceptualization and can be done alone in a cubicle with a sketchbook or in a group with a flip chart (or whiteboard) and markers.

Lying at the low-fidelity end of the prototyping spectrum, paper prototypes are static and usually have low visual and content fidelity. This forces users to focus on how they will *use* the system instead of what it will look like, and it makes designers more open to changes based on user feedback.

Low-fidelity prototyping lends itself to rapid prototyping. It has no learning curve but lets you make changes easily and quickly.

Medium Fidelity

As we start using computer-based tools such as Visio and Omnigraffle to prototype, the fidelity increases on most fronts, yielding medium-fidelity prototypes. Wireframes, task flows and scenarios that are created with these tools take more time and effort but look more formal and refined. While visual elements of branding, colors and style can be introduced, prototypers often stay away from them, focusing instead on demonstrating the behavior of the application. Interactivity can be simulated by linking pages or screens, but functional fidelity here is medium at best. These prototypes are best suited to determining whether user needs are met and whether the user experience is optimal.
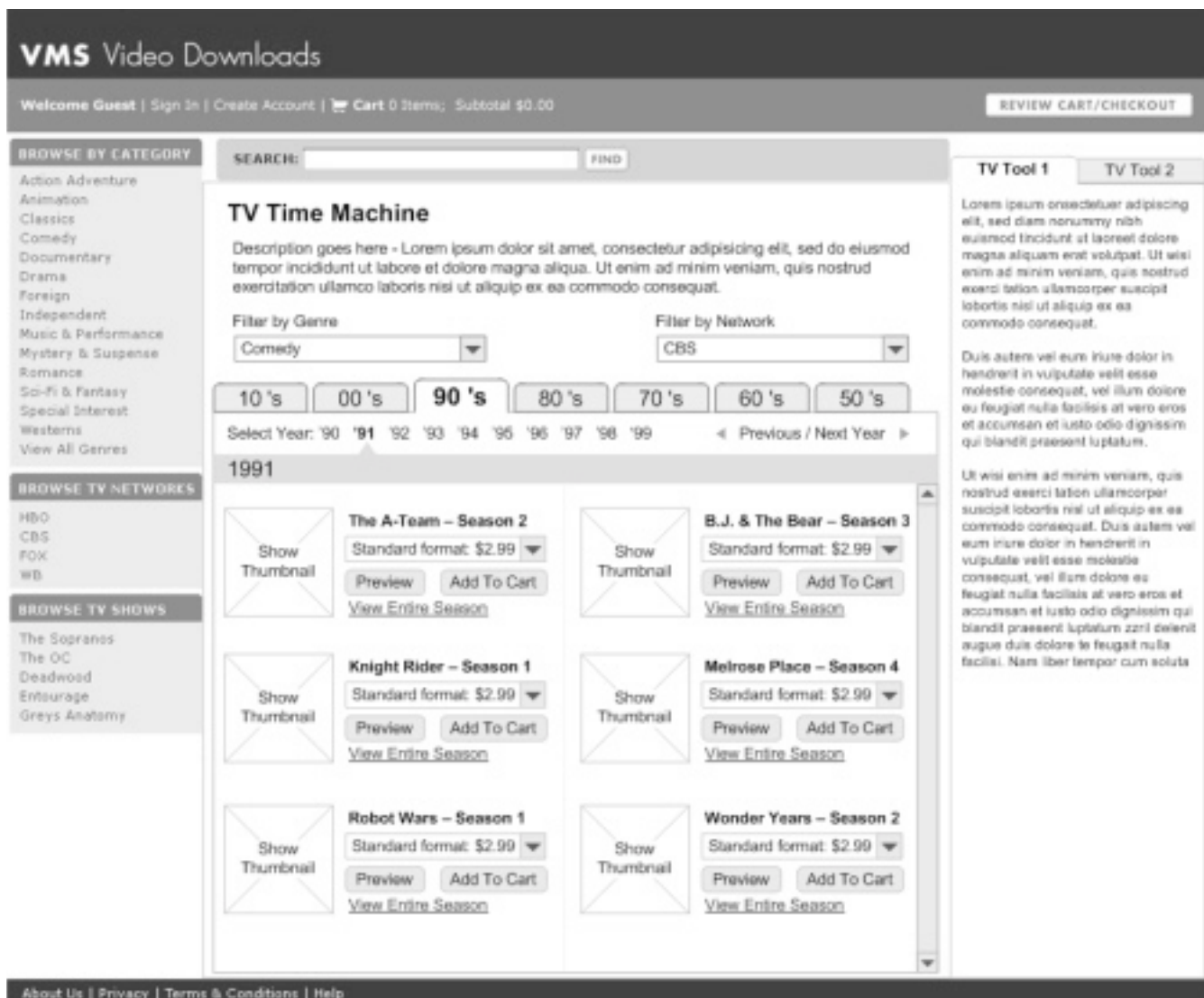
There are two reasons why one might intentionally make a medium-fidelity prototype *not* look like a medium-fidelity prototype:

- The first is that, by using Balsamiq or sketchy Visio stencils to make the prototype look low fidelity, you force users to view it as a draft or work in progress, rather than a polished and finished product.

- The second is that, by giving the prototype a high visual fidelity (for instance, in a comprehensive layout done in Photoshop), you get the user to focus on the visual design and look and feel, including color, fonts, layout, logo and images.

The speed of medium-fidelity prototyping is achieved with templates, stencils and reusable widgets and elements. It gets faster as you become more proficient with your tools of choice.

High Fidelity

High-fidelity prototypes are the most realistic and are often mistaken for the final product, but they are usually time-intensive. A few years ago, the only way to create high-fidelity prototypes was to actually code using a programming language, which often required the designer and developer to work together. These days, however, application-simulation tools allow non-technical users to drag and drop UI widgets to create high-fidelity prototypes that simulate the functionality of the final product, even for business logic and database interactions. Axure and iRise are some examples of application-simulation tools that can be used to create high-fidelity prototypes.

These prototypes are appropriate when high visual and functional fidelity is required; for example, when introducing a new technology (say, when moving from a mainframe application—yes, they still exist!—to a Web-based solution. Most of these prototypes cannot be converted to usable code, but they serve as an excellent reference for developers. These are also useful for conducting usability testing and training users.

High-fidelity prototyping is relatively rapid, considering the level of interactivity and fidelity involved, and it can be accelerated by using drag-and-drop simulation tools. In addition, some of these tools facilitate the

gathering of user feedback and documenting of requirements, further speeding up the design process. Even though you do not need to learn a new programming language, these tools do have a learning curve.

Selecting a Fidelity Level

In choosing the prototype fidelity, there is no one correct approach. Most designs of new products are best started with sketches, then moving to either medium- or high-fidelity prototypes, depending on the complexity of the system and the requirements of the dimensions of fidelity.

In working with one particular client in the pharmaceutical industry, we went from whiteboards to interactive prototypes that had high functional and content fidelity but low visual fidelity. They cared less about the look and feel than about adhering to corporate guidelines.

For another client, this one in retail, our interactive prototype had to have high visual and functional fidelity. The content fidelity did not matter because they would be reusing content and were already familiar with it. To them, the look and feel and interactive experience mattered more because this was their first implementation of SharePoint, and they wanted to make the portal look "non-SharePointy"!

Selecting Tools

Depending on your approach, you have a wide variety of tools to choose from. Dan Harrelson has compiled a list of popular prototyping tools on the Adaptive Path blog.

Each tool has its own feature set and strengths. Based on your needs and the requirements of the projects you work on, evaluate which tool would be

most appropriate. Here are some questions to consider when evaluating tools:

- How easy is it to learn and use the tool?

- Is it flexible to support prototypes for Web, packaged and custom software applications, as well as desktop and mobile applications?

- Is there a repository of reusable stencils, templates or widgets available?

- How easy is it to share the prototype with others for review? Can their feedback be captured using the tool?

- How easy is it to make changes on the fly or to incorporate feedback?

- Does it have any collaboration features, such as allowing multiple people to work on it at the same time?

- What are the licensing terms and costs?

Dos And Don'ts

As you get started, here are a few points about effective rapid prototyping to keep in mind:

Do...

- Work collaboratively with users, business and IT stakeholders while rapid prototyping. Apart from giving valuable feedback, they also gain a sense of ownership of the final product.

- Avoid "prototype creep" by setting expectations for the process, including ones affecting the purpose, fidelity, scope and duration.

Remind everyone, including yourself, that rapid prototyping is a means to an end, not an end in itself.

- When creating interactive high-fidelity prototypes and simulations, build in realistic delays (for instance, for screen refreshing or moving through steps of a transaction), so that users do not expect instant response times from the final product.

- Reuse, reuse, reuse. For computer-based prototyping, this means saving reusable templates, stencils, patterns and widgets for future projects.

- Most importantly, begin every prototype review session with the disclaimer that this is just a prototype, a mock-up, not the actual solution. This reminds users that this is a work in progress, it encourages feedback, and in the case of high-fidelity prototypes, it prevents users from mistaking it for a working solution.

Don't...

- Don't prototype features or functionality that cannot be implemented — often an issue with software package implementations. When in doubt, confirm with developers before starting.

- Don't take every change or request that comes out of a prototype review as a new requirement. Rapid prototyping helps capture missed requirements, but these new requirements should be evaluated carefully. Some may be implemented, while others are pushed to a future release.

- Don't begin prototype review sessions without clear guidelines for feedback. Be very specific about the type of feedback you are looking for. (Are the steps logically arranged? Is the navigation clear and intuitive?) If not, be prepared for, "I don't like the blue in the header," or "Can't we use this font instead?" or "Can you make this bigger, bolder, in red and flashing?"

- Don't be a perfectionist. In most cases, rapid prototyping does not have to be 100% perfect, just good enough to give everyone a common understanding.

- Don't prototype everything. Most of the time, you shouldn't have to.

When a Thousand Words Is Worth a Picture

Felicity Evans

Good design speaks for itself, right? Unfortunately, that is rarely the case. Most of us don't have the privilege of designing for ourselves; we design for clients, clients who have their own taste and ideas, clients who ultimately need to be persuaded on why we've made certain decisions. Good design doesn't speak for itself; it needs an advocate.

This article examines both why design requires justification and how you can go about providing it in a way that is clear and understandable. While we'll focus on visual design, the principles described here are applicable to any creative process or endeavor. Indeed, we learned most of these lessons while presenting Web interfaces and prototypes to clients, which took place after the visual designs had been agreed on.

Beyond The Talk

There is a saying oft quoted by professionals:

> *"Those who can, do; those who can't, teach."*

> *— George Bernard Shaw, Man and Superman*

I disagree with the premise of this statement, but I do sympathize with its tone. It is borne of the frustration that comes from the separation between the practice of a discipline and its theory.

Web design is to some degree immune from this criticism, being an extremely practical discipline by nature, and also a very new one. But we don't have to look far to find university professors teaching students how to design layouts using tables!

At the other end of the spectrum, we have the salespeople. These people may not be great at precise explanation, but they can certainly put forward a compelling argument. Ever sat in on a sales pitch and barely recognized the product being promoted? By the time the salesperson was through talking, you were convinced that the CMS could take you to the moon. I once heard this described as "truth massage."

[The WaSP InterACT](#) (featured) and Opera Web standards curriculum were created to address the problem of institutions getting out of step with the Web industry.

While the full-time teacher (and even full-time salesperson) certainly have a place in the industry, explaining our designs is too important a task to leave to professional "talkers."

We need to reconnect the practice of design with the theory of design. To do this, we need to learn the words and arguments for making a strong case for our designs.

The Importance Of Communicating A Message

"Marketing without design is lifeless, and design without marketing is mute."

— Von R. Glitschka

Despite what many designers think, "marketing" is not a dirty word. We can learn much from the advertising and marketing industries about the importance of communication. Some advertising is as subtle as a sledgehammer, practically screaming at the user "Buy now!"; nevertheless, a great deal of thought went into how each message would be perceived and what would convince buyers most effectively.

When it comes to Web design, communication tends to be a more subtle affair: a delicate balance between satisfying users needs and delivering on business objectives. Here, design can play a key role by simplifying processes and persuading users.

Consider the registration form. Usually, business objectives demand as much information from the user as possible, so that the company can streamline and direct its marketing efforts. However, users are reluctant to supply information about themselves and are put off by long forms.

In this instance, trends dictate that the user experience wins out; registration forms now commonly consist simply of an email address and password. But business objectives are not entirely dismissed either: the reduction in effort to sign up translates into more registered users for the website, and people are generally becoming more willing to submit their email address. Plus, a short form fits neatly onto most pages, grabbing the attention of users when they are most receptive — for example, after having just read an interesting article.

The micro-blogging platform Tumblr epitomizes the simple sign-up form.

Design Doesn't Just Happen

Design is itself a process of deduction. It involves a number of decisions, both conscious and unconscious. During this process, the designer dismisses some ideas as unworkable and pursues others in order to arrive at a solution.

But this process is completely opaque to the client. The client likely views the design not as the outcome of an in-depth process, but as a response to the brief, merely a visual representation of the constraints and considerations set before the designer. As the designer, part of your job is to educate the client and reveal the design process to them.

If you fail to do this, you risk reliving the scenario parodied The Oatmeal's post on "How a Web Design Went Straight to Hell." In short, what happens is that client (rightfully) wants some input into the process, so they make suggestions or improvements. But they don't fully understand the reasons behind your design choices, and so their "minor tweaks" take the website further and further away from its purpose. The design suffers, the designer gets exasperated, the client feels unappreciated and the relationship breaks down. Sound familiar?
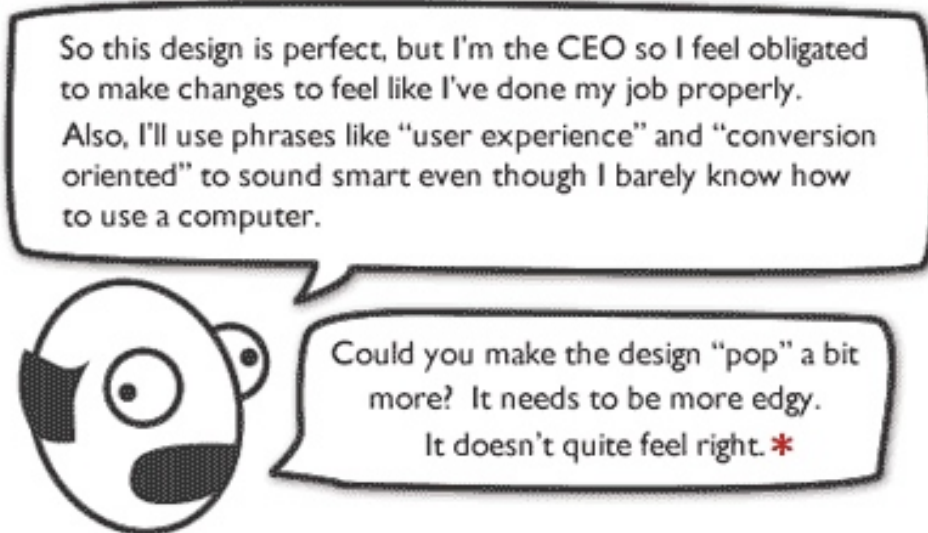
[The Oatmeal comic](#) describes how poor communication leads to Web design hell.

As designers and Web professionals, we needn't be afraid of criticism. Good design stands up to criticism, because it is more than a matter of taste. View criticism as an opportunity to explain the reasons behind your decisions, to invite the client into the design world so that they feel like a part of the process.

If you treat criticism this way, then both you and the client will benefit. The client begins to articulate "problems" with the design in language that you can respond to. Instead of just saying, "Make it green," they might now say, "The red feels too dominant, and we want to push the environmental benefits of our product, so I feel a more natural color is needed. Perhaps green?" Similarly, you will be less precious about your designs,

understanding that they will be open to change unless the arguments behind your design decisions are convincing.

Learning The Vocabulary

If we are to explain our designs to clients, then we need to know the language. Translating a visual medium into words is not straightforward. Designers suffer from being so well versed in the language of design that they struggle to articulate it to others. To them, it just "looks right."

The second difficulty faced by designers is assumed knowledge. They forget that much of their understanding of design comes from past experience, and they can become impatient with clients who are resistant to their suggestions.

In these situations, going back to the basics — i.e. the elements and principles of design — can be helpful. These basics are the "fundamental ideas about the practice of good visual design that are assumed to be the basis of all intentional visual design strategies" (from Wikipedia). A few are listed below:

Elements of design	Principles of design
Line	Balance
Shape	Gradation
Size	Repetition
Texture	Contrast
Colour	Harmony
Colour	Dominance
Tone	Proximity

The elements and principles of design.

You might wish to familiarize yourself with these elements and principles, although you very likely use them liberally in your designs already. For example, white space is really an expression of balance, because it affords each element enough space to have the right amount of focus. Similarly, we draw on proximity when designing forms to make clear the relationship between labels and inputs.

Understanding the principles and elements of design can also help in the design process. When assessing your options, you are able to fall back on more than "It just looks right."

Making A Convincing Argument

Ultimately, if we are going to persuade clients of the value of our designs, then we need to make convincing arguments. Clients are people, and

people have biases; so, the best way to convince them is to frame your argument in the right context.

Clients usually have various pressures put on them, and have to deal with various expectations of the website. Understanding these expectations from the outset will help you at each stage of the process, including the design stage. Some expectations will be formalized as objectives, which should be documented and serve as the basis of the design brief; other expectations might be politically motivated and therefore more difficult to identify.

While I don't advocate playing the political game, spending some time understanding your client's motivations will pay dividends as the project proceeds.

Broadly speaking, motivations fall into five categories:

1. Usability and accessibility,

2. Return on investment (ROI),

3. Return on engagement (ROE),

4. Deadline-driven,

5. Financial.

Depending on the dominant factors in a project, the client will be sympathetic to certain solutions and hostile to others. Let's look at one factor in detail.

Usability and Accessibility

These tend to be a strong motivator with government, not-for-profit and service-oriented clients, usually stemming from a combination of legislative

pressures and a desire to be inclusive to all audience segments. Of course, usability and accessibility should be core considerations in all projects, but they are rarely client-driven.

In such a case, Flash and other highly interactive technologies could be dismissed out of hand for not being accessible. This will also have a bearing on decisions related to color, font and clarity, which in turn will have a huge impact on the overall look and feel of the design.

Understanding this from the outset will help you present the design in the right framework and thus make the design appeal more to the client.

Presenting Your Case

Always present a design to the client in person (or by remote conferencing when that is not possible). Never just send it as an attachment to an email, otherwise you will:

1. Look incredibly unprofessional,

2. Have no control over the environment (or mindset) in which your design is viewed.

Once you have set aside a time to present your design, how should you proceed? The first thing to do is set clear expectations for the meeting.

Whether you like it or not, the way you present a design sends a message. For example, if you show up at a kick-off meeting with professionally printed boards of design ideas, you will be sending two conflicting messages, because no matter how much you talk about "sketching out" ideas and "exploring" concepts, your boards will be communicating

something completely different. The danger here is that the client gets hung up on what they see, which can derail the project in later stages.

As a designer, you have many tools in your arsenal: wireframes, mood boards, sketches, scamps, flat designs, HTML templates and even the backs of napkins. Pick the method that conveys the right amount of professionalism, while offering enough flexibility.

And remember, whatever expectations you have for the meeting, make sure to communicate (and agree on) them before a single design is made.

Dan M. Brown's book Communicating Design focuses on those deliverables (wireframes, concept models, site maps) that serve as the primary communication tools between designers and clients.

The second thing I find really helpful, closely related to setting expectations, is giving the client a task. Believe it or not, clients actually want to help. But like a wayward child in the kitchen, this help can sometimes be more damaging than constructive.

The best way to do this is to bring the client on board with your own goals. Talk them through the design brief and how you have answered it, or reiterate your overall concept to them. Invite the client to look at the design from this standpoint and to use it to shape their comments. They should consider themselves the website's quality control officer, pointing out where you might have veered from your own principles. If you do this, then you are far more likely to hear the sort of helpful criticism we spoke of earlier.

To Recap

- Remember, your design doesn't speak for itself. If it is to stand up to scrutiny, then it needs an advocate.

- In explaining your design, resist the temptation to use meaningless catchphrases. Instead, use plain, easily understood language.

- If possible, include your client in the design process. Walk them through the process you took to arrive at your decisions.

- Don't be afraid of criticism, because good design is more than a matter of taste.

- Don't be precious with your design. Be prepared to change elements that you cannot make a good argument for.

- Translating a visual medium into words can be difficult. But the principles and elements of design will help you articulate this visual language.

- To make your argument convincing, frame it in the right context. Understand the pain points, and justify your design in these terms.

- Always present your design in person, and ensure that any expectations are communicated (and agreed on) before a single decision is made.

- Finally, be proud of your work. When you've finished speaking, give it plenty of space to say a few words of its own!

The Authors

Alexander Dawson

Alexander Dawson is a freelance Web designer, author and recreational software developer specializing in Web standards, accessibility and UX design. As well as running a business ([HiTechy](#)) and writing, he spends time in Twitter, SitePoint's forums and other places, helping those in need.

Bridget Fahrland

[Bridget Fahrland](#) has worked in Web design since the days of blink tags. As a creative lead with a copywriting background, she works with clients to ensure their sites have a distinct voice and design. She is currently the Director of Creative Strategy at [Fry, Inc.](#)

David Travis

David Travis is a user experience consultant and trainer at [Userfocus](#). He has a BSc and a PhD in psychology and he is a Chartered Psychologist.

Felicity Evans

Interface developer and user experience specialist at [Reading Room](#) with a strong desire to marry robust, semantic code together with beautiful and usable websites. Felicity has been producing websites for almost 9 years and is still learning how to do it better. Projects she has worked on include Skoda, Hill & Knowlton, Wellcome Collection and NHS Smokefree.

Kayla Knight

[Kayla Knight](#) is a full-time freelance Web designer and developer with several years of experience. She specializes in Wordpress development, creating PHP/MySQL custom applications, and effective user interface design. In her spare time she writes for a number of top related design and development blogs as well.

Louis Lazaris

Louis Lazaris is a freelance Web developer based in Toronto, Canada. He blogs about front-end code on [Impressive Webs](#) and runs [Interviews by Design](#), a website that publishes brief interviews with talented designers.

Lyndon Cerejo

Lyndon Cerejo is a certified user experience & usability [strategist](#) with a successful track record with clients including Allstate, American Express, Coca-Cola, General Motors, Merrill Lynch, and Walmart. His key areas of expertise are user experience analysis, information architecture and rapid prototyping usability testing, online strategy & marketing. He is the co-author of [marketing.com](#) - a book about marketing adaptations on the Internet.

Paul Boag

Paul Boag is the founder of UK Web design agency Headscape, author of the [Website Owners Manual](#) and host of award-winning Web design podcast [Boagworld](#).

Speider Schneider

Speider Schneider is a former member of The Usual Gang of Idiots at MAD Magazine, "among other professional embarrassments and failures." He currently writes for local newspapers, blogs and other Web content and has designed products for Disney/Pixar, Warner Bros., Harley-Davidson, ESPN, Mattel, DC and Marvel Comics, Cartoon Network and Nickelodeon among other notable companies. Speider is a former member of the board for the Graphic Artists Guild, co-chair of the GAG Professional Practices Committee and a former board member of the Society of Illustrators. He also continues to speak at art schools across the United States on business and professional practices.

Vitaly Friedman

Vitaly Friedman is editor-in-chief of Smashing Magazine, an online magazine dedicated to designers and developers.

Bruce Lawson, Remy Sharp

Remy and Bruce are two developers who have been playing with HTML5 since Christmas 2008. Bruce evangelizes Open Web Standards for Opera. Remy is a developer, speaker, blogger, and contributing author for jQuery Cookbook (O'Reilly). He runs his own Brighton based development company called Left Logic, coding and writing about JavaScript, jQuery, HTML5, CSS, PHP, Perl and anything else he can get his hands on. Together, they are the authors of Introducing HTML5, the first full-length book on HTML5 (New Riders, July 2010).