

User Experience

Practical Techniques

Volume 1

A large, stylized, dark red letter 'U' is positioned in the bottom right corner. A small, light red number '1' is placed inside the lower curve of the 'U', serving as a visual element for the volume number.

U
1

Imprint

Copyright 2012 Smashing Media GmbH, Freiburg, Germany

Version 1: February 2012

ISBN: 978-3-943075-25-0

Cover Design: Ricardo Gimenes

PR & Press: Stephan Poppe

eBook Strategy: Thomas Burkert

Technical Editing: Andrew Rogerson

Idea & Concept: Smashing Media GmbH

ABOUT SMASHING MAGAZINE

[Smashing Magazine](#) is an online magazine dedicated to Web designers and developers worldwide. Its rigorous quality control and thorough editorial work has gathered a devoted community exceeding half a million subscribers, followers and fans. Each and every published article is carefully prepared, edited, reviewed and curated according to the high quality standards set in Smashing Magazine's own publishing policy. Smashing Magazine publishes articles on a daily basis with topics ranging from business, visual design, typography, front-end as well as back-end development, all the way to usability and user experience design. The magazine is — and always has been — a professional and independent online publication neither controlled nor influenced by any third parties, delivering content in the best interest of its readers. These guidelines are continually revised and updated to assure that the quality of the published content is never compromised.

ABOUT SMASHING MEDIA GMBH

[Smashing Media GmbH](#) is one of the world's leading online publishing companies in the field of Web design. Founded in 2009 by Sven Lennartz and Vitaly Friedman, the company's headquarters is situated in southern Germany, in the sunny city of Freiburg im Breisgau. Smashing Media's lead publication, Smashing Magazine, has gained worldwide attention since its emergence back in 2006, and is supported by the vast, global Smashing community and readership. Smashing Magazine had proven to be a trustworthy online source containing high quality articles on progressive design and coding techniques as well as recent developments in the Web design industry.

About this eBook

Are you familiar with the term User Experience (UX) and the principles that determine a user-friendly website? If yes, then all you need to do is to learn some more useful techniques that will make your user interface designs even better! **User Experience, Practical Techniques, Volume 1** features seven selected articles on usability principles which will help you design user-centered websites by showing you helpful examples of the best practices and which common mistakes to avoid in your next project.

Table of Contents

[10 Principles Of Effective Web Design](#)

[Five More Principles Of Effective Web Design](#)

[10 Useful Techniques To Improve Your User Interface Designs](#)

[10 Useful Web Application Interface Techniques](#)

[12 Useful Techniques For Good User Interface Design](#)

[An Extensive Guide To Web Form Usability](#)

[10 Usability Nightmares You Should Be Aware Of](#)

[About The Authors](#)

10 Principles Of Effective Web Design

Vitaly Friedman

Usability and the utility, not the visual design, determine the success or failure of a web-site. Since the visitor of the page is the only person who clicks the mouse and therefore decides everything, user-centric design has become a standard approach for successful and profit-oriented web design. After all, if users can't use a feature, it might as well not exist.

We aren't going to discuss the implementation details (e.g. where the search box should be placed) as it has already been done in a number of articles; instead we focus on the main principles, heuristics and approaches for effective web design — approaches which, used properly, can lead to more sophisticated design decisions and simplify the process of perceiving presented information.

Please notice that

- you might be interested in the usability-related articles about [10 Usability Nightmares](#) and [30 Usability Issues](#) we've published before,
- we'll cover more principles of effective design in our following posts. Therefore you might want to [subscribe to our RSS-feed](#).

Principles Of Effective Web Design

In order to use the principles properly we first need to understand how users interact with web-sites, how they think and what are the basic patterns of users' behavior.

How do users think?

Basically, users' habits on the Web aren't that different from customers' habits in a store. Visitors glance at each new page, scan some of the text, and click on the first link that catches their interest or vaguely resembles the thing they're looking for. In fact, there are large parts of the page they don't even look at.

Most users search for something interesting (or useful) and clickable; as soon as some promising candidates are found, users click. If the new page doesn't meet users' expectations, the Back button is clicked and the search process is continued.

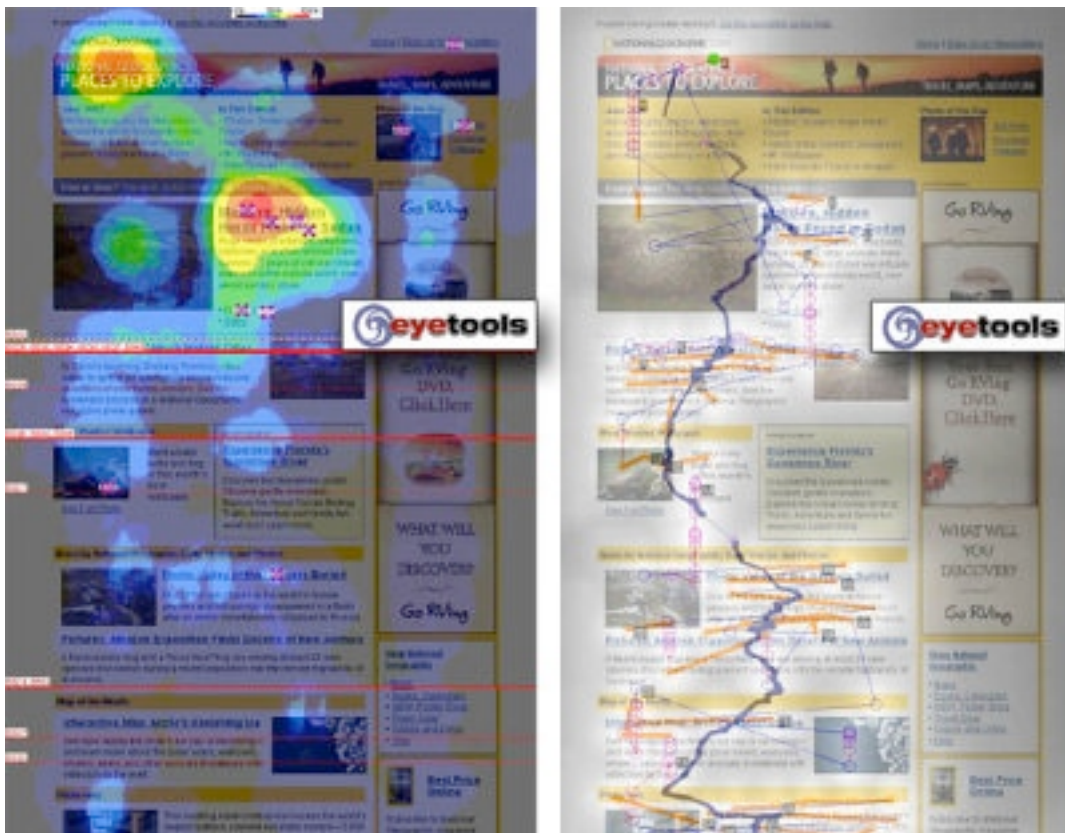
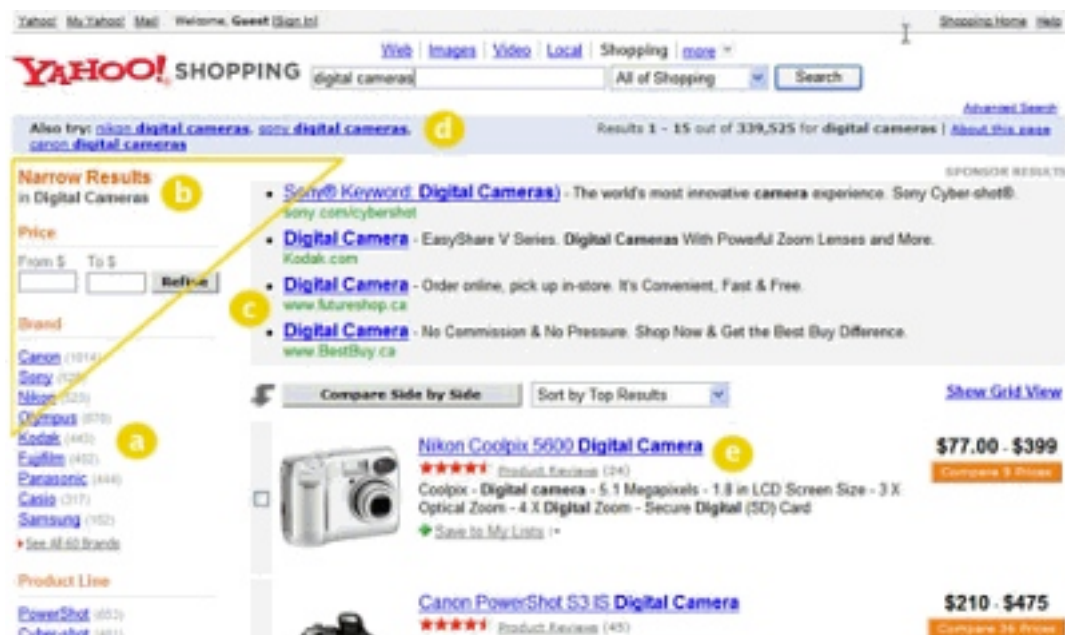
- **Users appreciate quality and credibility.** If a page provides users with high-quality content, they are willing to compromise the content with advertisements and the design of the site. This is the reason why not-that-well-designed web-sites with high-quality content gain a lot of traffic over years. Content is more important than the design which supports it.
- **Users don't read, they scan.** Analyzing a web-page, users search for some fixed points or anchors which would guide them through the content of the page.



Users don't read, they scan. Notice how "hot" areas abrupt in the middle of sentences. This is typical for the scanning process.

- **Web users are impatient and insist on instant gratification.** Very simple principle: If a web-site isn't able to meet users' expectations, then designer failed to get his job done properly and the company loses money. The higher is the cognitive load and the less intuitive is the navigation, the more willing are users to leave the web-site and search for alternatives. [JN / DWU]

- **Users don't make optimal choices.** Users don't search for the quickest way to find the information they're looking for. Neither do they scan web-page in a linear fashion, going sequentially from one site section to another one. Instead users satisfice; they choose the first reasonable option. As soon as they find a link that seems like it might lead to the goal, there is a very good chance that it will be immediately clicked. Optimizing is hard, and it takes a long time. Satisficing is more efficient.
[\[video\]](#)



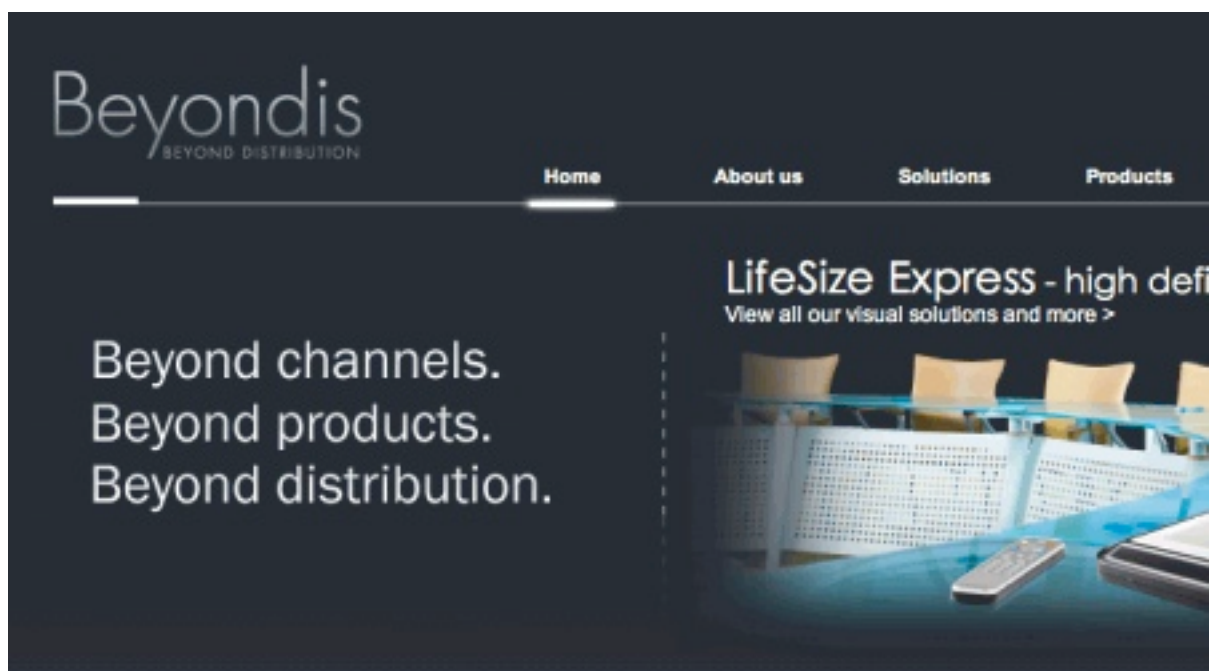
Both pictures show: sequential reading flow doesn't work in the Web. Right screenshot on the image at the bottom describes the scan path of a given page.

- **Users follow their intuition.** In most cases users muddle through instead of reading the information a designer has provided. According to Steve Krug, the basic reason for that is that users don't care. "If we find something that works, we stick to it. It doesn't matter to us if we understand how things work, as long as we can use them. If your audience is going to act like you're designing billboard, then design great billboards."
- **Users want to have control.** Users want to be able to control their browser and rely on the consistent data presentation throughout the site. E.g. they don't want new windows popping up unexpectedly and they want to be able to get back with a "Back"-button to the site they've been before: therefore it's a good practice to *never open links in new browser windows*.

1. Don't make users think

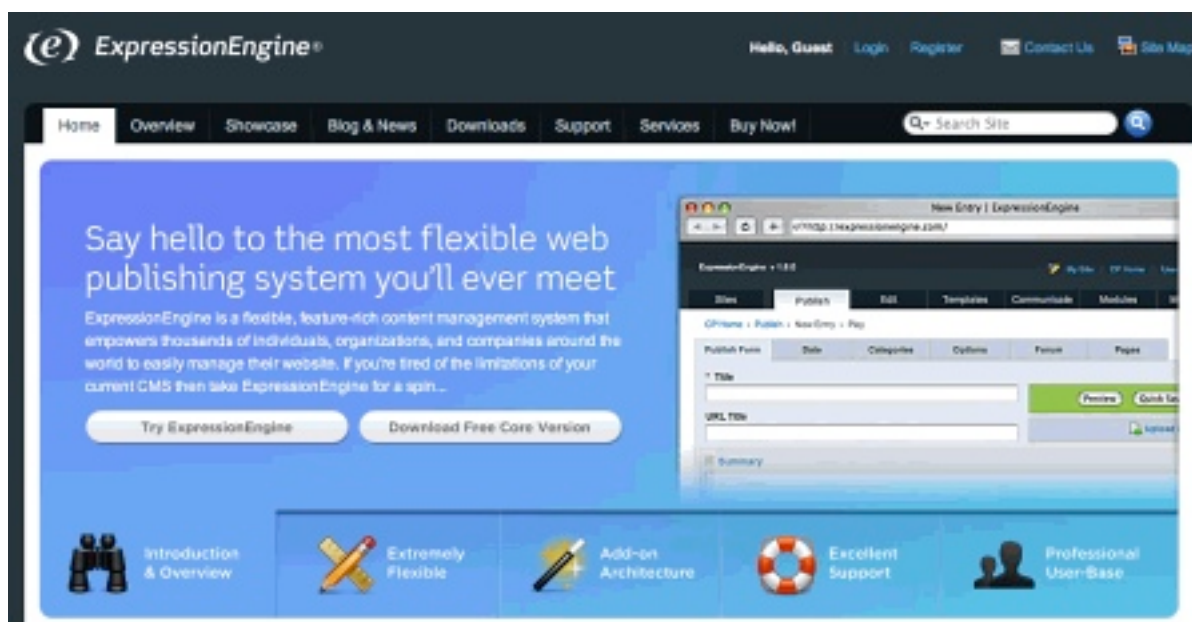
According to Krug's first law of usability, the web-page should be obvious and self-explanatory. When you're creating a site, your job is to get rid of the *question marks* — the decisions users need to make consciously, considering pros, cons and alternatives.

If the navigation and site architecture aren't intuitive, the number of question marks grows and makes it harder for users to comprehend how the system works and how to get from point A to point B. A clear structure, moderate visual clues and easily recognizable links can help users to find their path to their aim.



Let's take a look at an example. Beyondis.co.uk claims to be “beyond channels, beyond products, beyond distribution”. What does it *mean*? Since users tend to explore web-sites according to the [“F”-pattern](#), these three statements would be the first elements users will see on the page once it is loaded.

Although the design itself is simple and intuitive, to understand what the page is about the user needs to search for the answer. This is what an *unnecessary question mark* is. It's designer's task to make sure that the number of question marks is close to 0. The visual explanation is placed on the right hand side. Just exchanging both blocks would increase usability.



[ExpressionEngine](#) uses the very same structure like Beyondis, but avoids unnecessary question marks. Furthermore, the slogan becomes functional as users are provided with options to try the service and download the free version.

By reducing cognitive load you make it easier for visitors to grasp the idea behind the system. Once you've achieved this, you can communicate why the system is useful and how users can benefit from it. People won't use your web site if they can't find their way around it.

2. Don't squander users' patience

In every project when you are going to offer your visitors some service or tool, try to keep your user requirements minimal. The less action is required from users to test a service, the more likely a random visitor is to actually try it out. First-time visitors are willing to play with the service, not filling long web forms for an account they might never use in the future. Let users explore the site and discover your services without forcing them into sharing

private data. It's not reasonable to force users to enter an email address to test the feature.

As Ryan Singer — the developer of the 37Signals team — [states](#), users would probably be eager to provide an email address if they were asked for it after they'd seen the feature work, so they had some idea of what they were going to get in return.

A screenshot of the Stikkit website's sign-up form. The header is blue with the 'stikkit' logo (a yellow sticky note with an arrow) and navigation links: 'contact', 'forums', 'api', and 'sign in'. The main content area is yellow and titled 'sign up for stikkit'. It contains four input fields: 'email address' (with 'editor@smashingmagazine.com'), 'nickname' (with 'smashing'), 'password' (masked with dots), and 'password again' (masked with dots). Below the fields is a checkbox with a blue checkmark and the text 'I agree to the Stikkit Terms of Use and Privacy Policy. (We also provide a summary of your rights and obligations.)'. At the bottom right is a blue 'sign up' button with a hand cursor icon pointing at it.

stikkit

contact forums api sign in

sign up for stikkit

email address

editor@smashingmagazine.com

nickname

smashing

password

.....

password again

.....

☒ I agree to the Stikkit [Terms of Use](#) and [Privacy Policy](#). (We also provide a [summary of your rights and obligations](#).)

sign up

[Stikkitt](#) is a perfect example for a user-friendly service which requires almost nothing from the visitor which is unobtrusive and comforting. And that's what you want your users to feel on your web site.

Einen neuen Account erstellen.

1. Firmenname & Login-Kürzel.
Das Login-Kürzel darf nur Buchstaben und Ziffern enthalten.
Firmenname:
Login-Kürzel:
appmte.de/login/smashing_mag

2. Dein erster Benutzer.
Die E-Mail-Adresse dient als Benutzername.
E-Mail:
Vollständiger Name:

3. Wähle ein Passwort für den ersten Benutzer.
Das Passwort muss zwischen 6 und 30 Zeichen haben.
Passwort:
Passwort wiederholen:

☒ Ich bin mit den [AGB](#) einverstanden.
☒ Passwort in der Aktivierungs-Mail mitsenden.

Account erstellen »

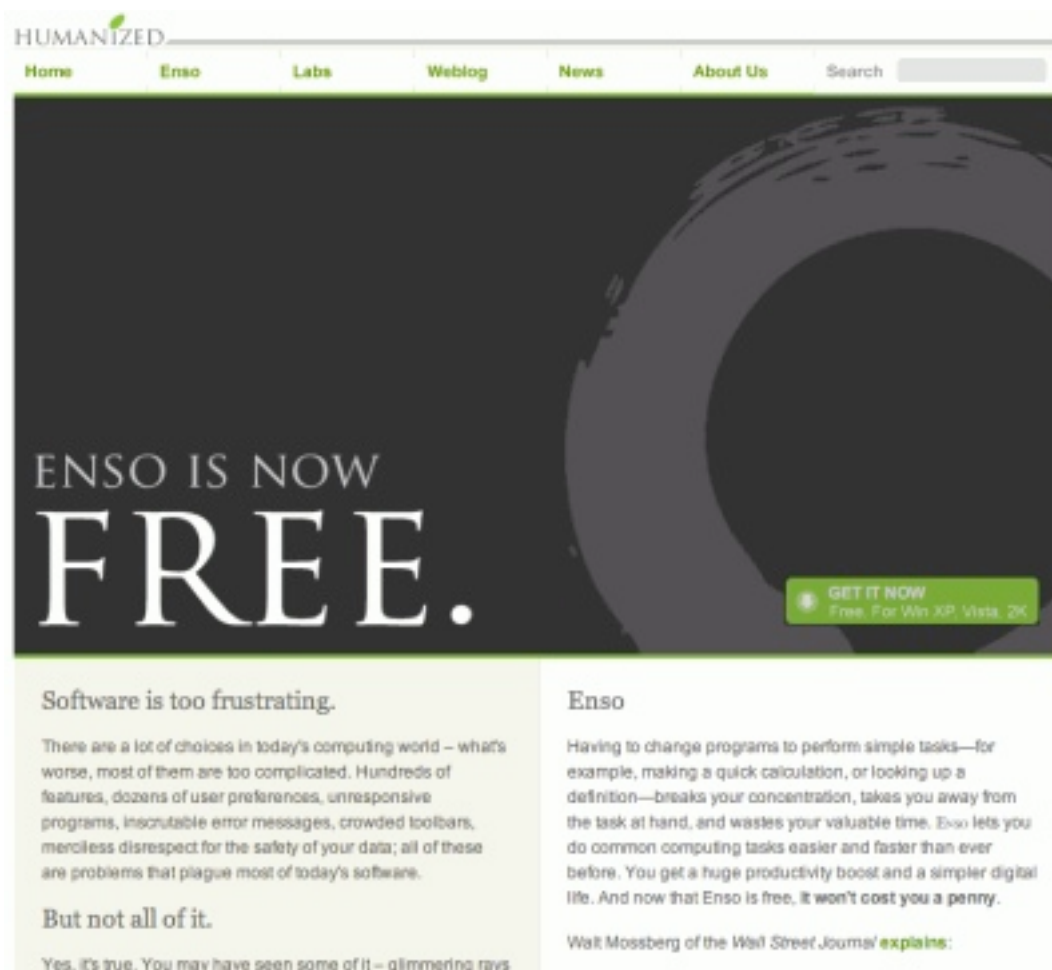
Apparently, Mite requires more. However the registration can be done in less than 30 seconds — as the form has horizontal orientation, the user doesn't even need to scroll the page.

Ideally remove all barriers, don't require subscriptions or registrations first. A user registration alone is enough of an impediment to user navigation to cut down on incoming traffic.

3. Manage to focus users' attention

As web-sites provide both static and dynamic content, some aspects of the user interface attract attention more than others do. Obviously, images are more eye-catching than the text — just as the sentences marked as bold are more attractive than plain text.

The human eye is a highly non-linear device, and web-users can instantly recognize edges, patterns and motions. This is why video-based advertisements are extremely annoying and distracting, but from the marketing perspective they perfectly do the job of capturing users' attention.



Humanized.com perfectly uses the principle of focus. The only element which is directly visible to the users is the word “free” which works attractive and appealing, but still calm and purely informative. Subtle hints provide users with enough information of how to find more about the “free” product.

Focusing users’ attention to specific areas of the site with a moderate use of visual elements can help your visitors to get from point A to point B without thinking of how it actually is supposed to be done. The less question marks visitors have, the better sense of orientation they have and the more trust they can develop towards the company the site represents. In other words: the less thinking needs to happen behind the scenes, the better is the user experience which is the aim of usability in the first place.

4. Strive for feature exposure

Modern web designs are usually criticized due to their approach of guiding users with visually appealing 1-2-3-done-steps, large buttons with visual effects etc. But from the design perspective these elements actually aren’t a bad thing. On the contrary, such guidelines are extremely effective as they lead the visitors through the site content in a very simple and user-friendly way.



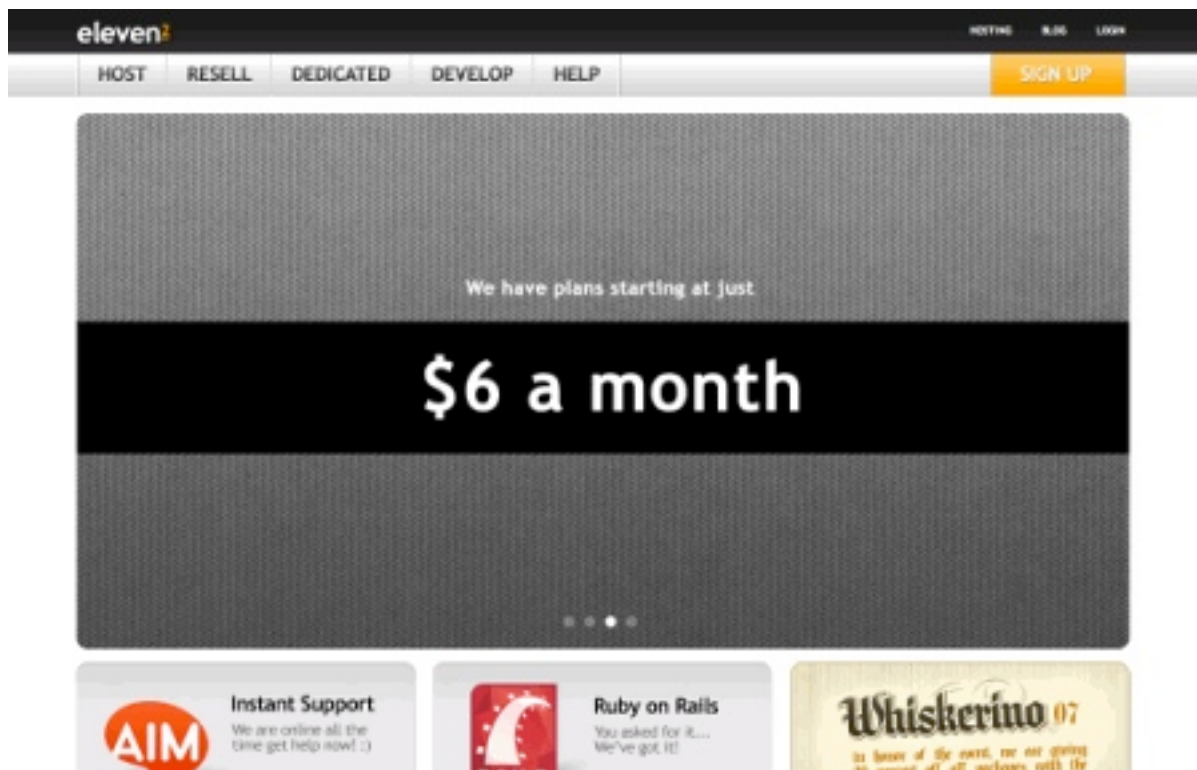
Dibusoft.com combines visual appeal with clear site structure. The site has 9 main navigation options which are visible at the first glance. The choice of colors might be too light, though.

Letting the user see clearly what functions are available is a fundamental principle of successful user interface design. It doesn't really matter how this is achieved. What matters is that the content is well-understood and visitors feel comfortable with the way they interact with the system.

5. Make use of effective writing

As the Web is different from print, it's necessary to adjust the writing style to users' preferences and browsing habits. Promotional writing won't be read. Long text blocks without images and keywords marked in bold or *italics* will be skipped. Exaggerated language will be ignored.

Talk business. Avoid cute or clever names, marketing-induced names, company-specific names, and unfamiliar technical names. For instance, if you describe a service and want users to create an account, “sign up” is better than “start now!” which is again better than “explore our services”.



Eleven2.com gets directly to the point. No cute words, no exaggerated statements. Instead a price: just what visitors are looking for.

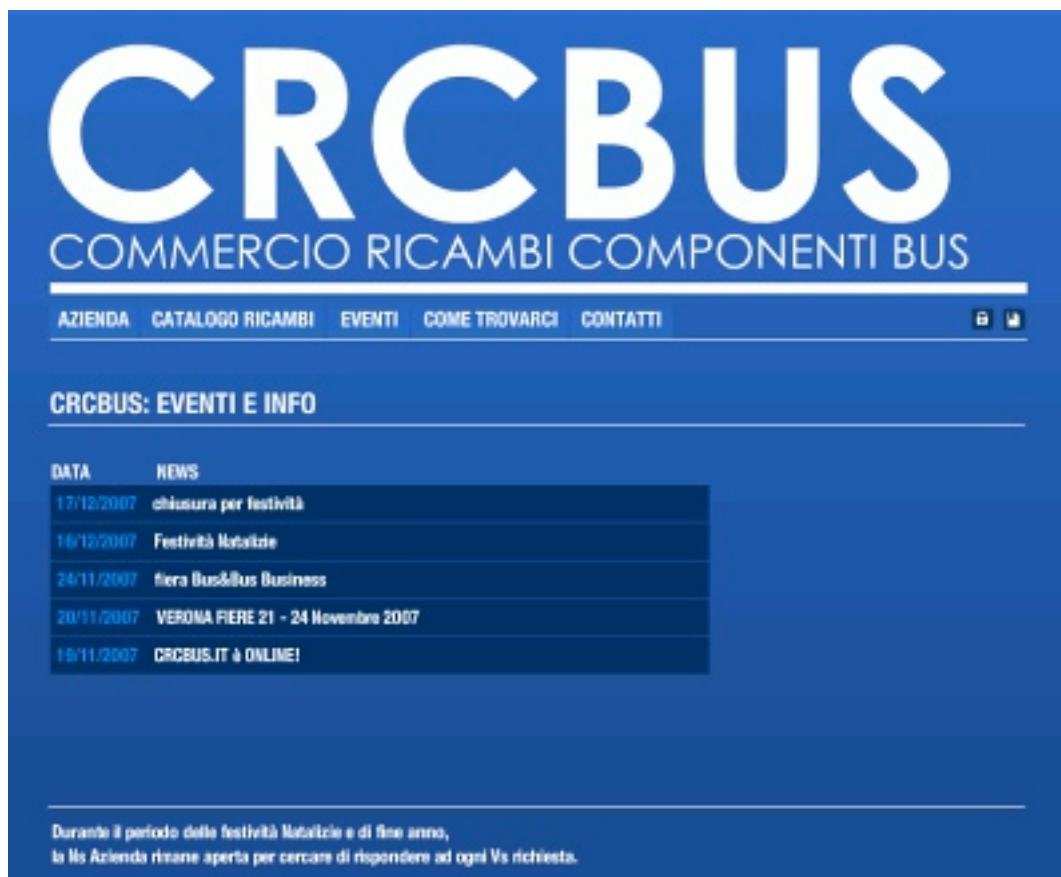
An optimal solution for effective writing is to

- use short and concise phrases (come to the point as quickly as possible),
- use scannable layout (categorize the content, use multiple heading levels, use visual elements and bulleted lists which break the flow of uniform text blocks),

- use plain and objective language (a promotion doesn't need to sound like advertisement; give your users some reasonable and objective reason why they should use your service or stay on your web-site)

6. Strive for simplicity

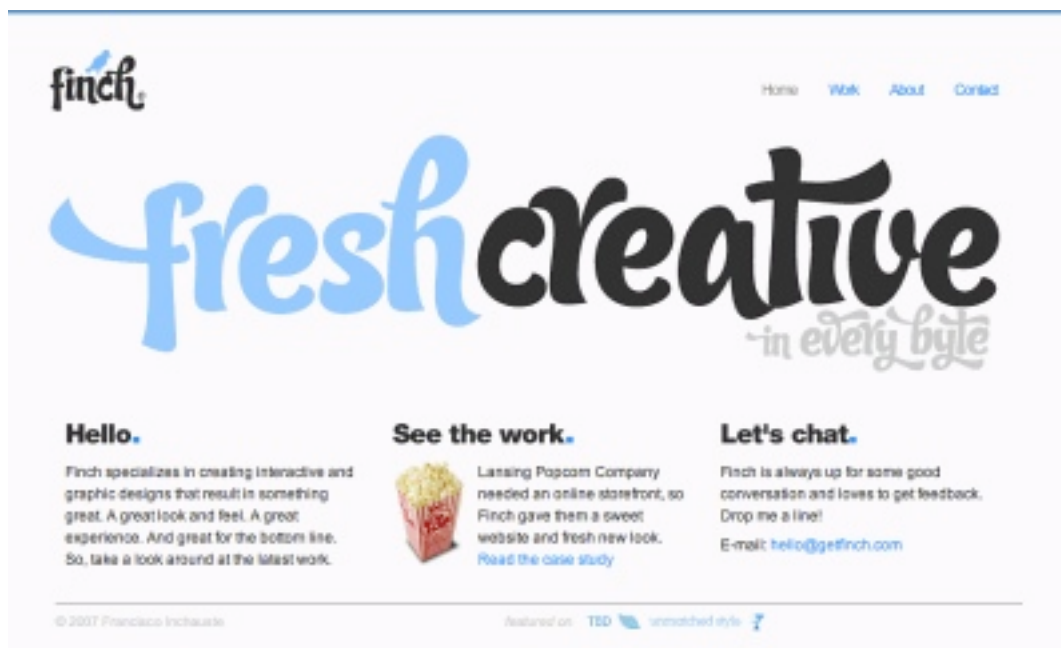
The “keep it simple”-principle (KIS) should be the primary goal of site design. Users are rarely on a site to enjoy the design; furthermore, in most cases they are looking for the information *despite* the design. Strive for simplicity instead of complexity.



[Crcbus](#) provides visitors with a clean and simple design. You may have no idea what the site is about as it is in Italian, however you can directly recognize the navigation, header, content area and the footer. Notice how

even icons manage to communicate the information clearly. Once the icons are hovered, additional information is provided.

From the visitors' point of view, the best site design is a pure text, without any advertisements or further content blocks matching exactly the query visitors used or the content they've been looking for. This is one of the reasons why a user-friendly print-version of web pages is essential for good user experience.



[Finch](#) clearly presents the information about the site and gives visitors a choice of options without overcrowding them with unnecessary content.

7. Don't be afraid of the white space

Actually it's really hard to overestimate the importance of white space. Not only does it help to reduce the cognitive load for the visitors, but it makes it possible to perceive the information presented on the screen. When a new

visitor approaches a design layout, the first thing he/she tries to do is to scan the page and divide the content area into digestible pieces of information.

Complex structures are harder to read, scan, analyze and work with. If you have the choice between separating two design segments by a visible line or by some whitespace, it's usually better to use the whitespace solution. Hierarchical structures reduce complexity (Simon's Law): the better you manage to provide users with a sense of visual hierarchy, the easier your content will be to perceive.



White space is good. Cameron.io uses white space as a primary design element. The result is a well-scannable layout which gives the content a dominating position it deserves.

8. Communicate effectively with a “visible language”

In his papers on effective visual communication, Aaron Marcus states [three fundamental principles](#) involved in the use of the so-called “*visible language*” — the content users see on a screen.

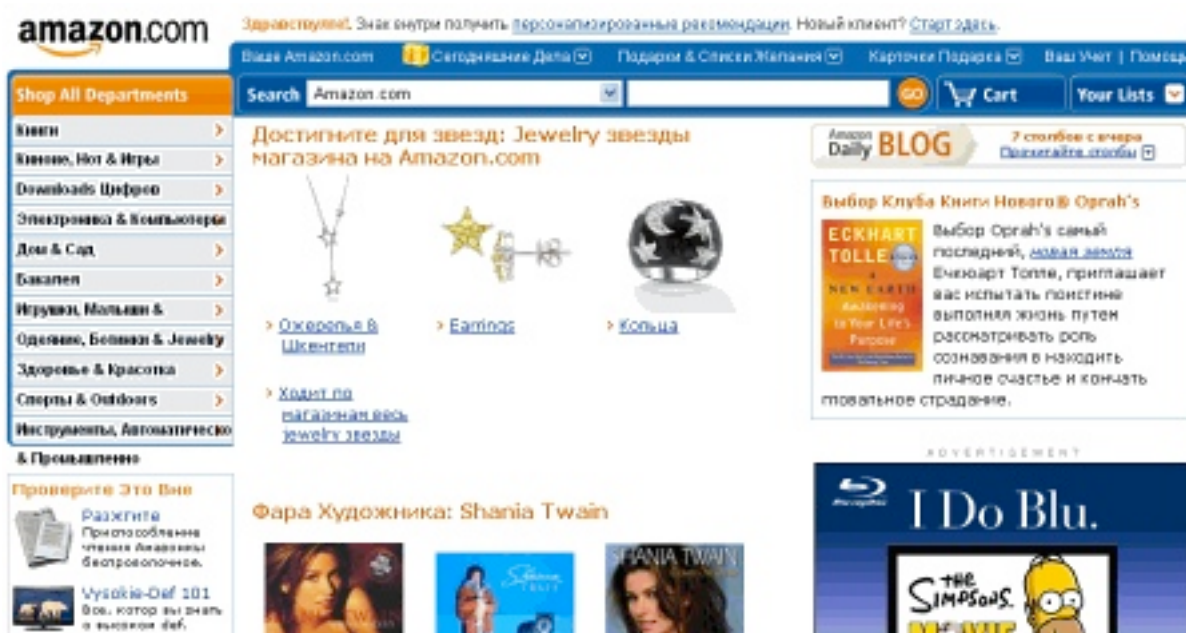
- **Organize:** provide the user with a clear and consistent conceptual structure. Consistency, screen layout, relationships and navigability are important concepts of organization. The same conventions and rules should be applied to all elements.
- **Economize:** do the most with the least amount of cues and visual elements. Four major points to be considered: simplicity, clarity, distinctiveness, and emphasis. *Simplicity* includes only the elements that are most important for communication. *Clarity*: all components should be designed so their meaning is not ambiguous. *Distinctiveness*: the important properties of the necessary elements should be distinguishable. *Emphasis*: the most important elements should be easily perceived.
- **Communicate:** match the presentation to the capabilities of the user. The user interface must keep in balance legibility, readability, typography, symbolism, multiple views, and color or texture in order to communicate successfully. Use max. 3 typefaces in a maximum of 3 point sizes — a maximum of 18 words or 50-80 characters per line of text.

9. Conventions are our friends

Conventional design of site elements doesn’t result in a boring web site. In fact, **conventions are very useful** as they reduce the learning curve, the need to figure out how things work. For instance, it would be a usability

nightmare if all web-sites had different visual presentation of RSS-feeds. That's not that different from our regular life where we tend to get used to basic principles of how we organize data (folders) or do shopping (placement of products).

With conventions you can gain users' confidence, trust, reliability and prove your credibility. Follow users' expectations — understand what they're expecting from a site navigation, text structure, search placement etc. (see Nielsen's [Usability Alertbox](#) for more information)



BabelFish in use: Amazon.com in Russian.

A typical example from usability sessions is to translate the page in Japanese (assuming your web users don't know Japanese, e.g. with [Babelfish](#)) and provide your usability testers with a task to find something in the page of different language. If conventions are well-applied, users will be able to achieve a not-too-specific objective, even if they can't understand a word of it.

Steve Krug suggests that it's better to innovate only when you know you really have a better idea, but take advantages of conventions when you don't.

10. Test early, test often

This so-called TETO-principle should be applied to every web design project as usability tests often provide crucial insights into significant problems and issues related to a given layout.

Test not too late, not too little and not for the wrong reasons. In the latter case it's necessary to understand that most design decisions are local; that means that you can't universally answer whether some layout is better than the other one as you need to analyze it from a very specific point of view (considering requirements, stakeholders, budget etc.).

Some important points to keep in mind:

- according to Steve Krug, testing one user is 100% better than testing none and testing one user early in the project is better than testing 50 near the end. According to Boehm's first law, errors are most frequent during requirements and design activities and are the more expensive the later they are removed.
- testing is an iterative process. That means that you design something, test it, fix it and then test it again. There might be problems which haven't been found during the first round as users were practically blocked by other problems.
- usability tests always produce useful results. Either you'll be pointed to the problems you have or you'll be pointed to the absence of major design flaws which is in both cases a useful insight for your project.

- according to Weinberg's law, a developer is unsuited to test his or her code. This holds for designers as well. After you've worked on a site for few weeks, you can't observe it from a fresh perspective anymore. You know how it is built and therefore you know exactly how it works — you have the wisdom independent testers and visitors of your site wouldn't have.

Bottom line: if you want a great site, you've got to test.

Five More Principles Of Effective Web Design

Vitaly Friedman

Web design has significantly improved over the last years. It's more user-friendly and more appealing today — and there is a good reason behind it: over the years we've found out that design with focus on usability and user experience is just more effective. Modern cut-edge design isn't filled with loud happy talk and blinking advertisements. We've learnt to initiate the dialogue with visitors, involve them into discussions and gain their trust by addressing their needs and speaking with them honestly and directly.

We've presented 10 Principles Of Effective Web Design — a comprehensive article about effective Web design and provided you with insights about how users actually think as well as with some examples of how effective designs can be achieved.

This article highlights 5 further principles, heuristics and approaches for effective Web design — approaches which, used properly, can lead to more sophisticated design decisions and simplify the process of perceiving presented information.

1. Use An Effective Marketing Principle

In order to sell a product or a service you need to be able to effectively initiate the dialogue with random visitors of your site. Since your visitors have actually come to your site, they are willing to hear to you and learn

what you have to offer. So how do you approach this potential clientele to maximize your earning at the end of the month?

Suggested by Strong in 1925, **AIDA** is the effective marketing model which describes a common list of events that are very often undergone when a person is selling a product or service:

- **A** – Attention (Awareness): attract the attention of the customer.
- **I** - Interest: raise customer interest by demonstrating features, advantages, and benefits.
- **D** – Desire: convince customers that they want and desire the product or service and that it will satisfy their needs.
- **A** – Action: lead customers towards taking action and/or purchasing.
- Nowadays some have added another letter to form AIDA(S): **S** – Satisfaction – satisfy the customer so they become a repeat customer and give referrals to a product.

In this context customers should notice that AIDA is usually the way how potential buyers are tricked into buying products or services they actually don't need. To gain users' trust designers need to make sure that the site provides genuine information and there is no hidden context in which the content can be understood in a different way.

It's also more effective to offer visitors concrete arguments, situations and ideas of how a product or a service can be used instead of bombarding them with loud and empty ad-slogans.

traffik

Tour Features Pricing Designer FAQ Blog Support Contact

FREE 30 DAY TRIAL

A Content Management System Made For Business

Meet the first content management system born to run online businesses. Your website and a central customer database. Email marketing and eCommerce. Web analytics and customer profiling. It's one, central console with everything you need to run your online business.

a

Next



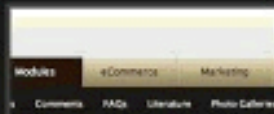
Intro Run Your Website Capture Leads Know Your Customers Easy eCommerce Email Marketing Executive Insight

What's Next

i

How Can Traffik Help You?

i



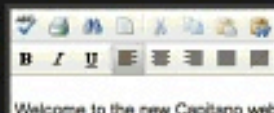
One Stop For Online Success

Content management, email marketing and online shopping. Integrated contact management and in depth analytics.



Built For Business

Traffik focuses on what matters to a business owner, so you get a faster return on your investment.



Easily Manageable For You

You can edit your site, shop and email campaigns yourself.



Who Says Support Isn't A Feature?

We're committed to our top notch phone and email support as well as our online training videos.

Traffik is an online, hosted content management system that allows you to:

- Build and manage your website
- Write a blog and build a forum
- Run an online shop and accept payments
- Create email marketing campaigns
- Build a customer database
- Plus, a whole lot more!

d

Why Use Traffik For Your Website?

Everything is designed to work together easily and intuitively while leaving serious designers the power to create some really cool stuff. Heck, we built this site with Traffik!

TAKE THE TOUR

FREE 30 DAY TRIAL

a

level9
DESIGN

Traffik is a product of Level9 Design
© 2008 Level9 Design. All Rights Reserved.

Consider the example presented above. [traffik](#) follows the AIDA principle. If the visitors expect some information about a Content Management System their expectation is immediately confirmed with a brief introduction at the top of the site. Thus, potential customers are attracted and remain on the site (*Attraction*, a). The interest is then raised by showcasing features and benefits of the product (*Interest*, i).

To communicate how the service will satisfy user's needs the brown info-block at the right hand at the bottom of the page is used (*Desire*, d). Finally the users are guided to "Take the tour" and "Free 30 day trial" buttons which attempt to lead customers towards purchasing (*Action*, a).

A lot of other models are known in order to sell, e.g. the BOSCH-Formula (developed by Peter Hubert):

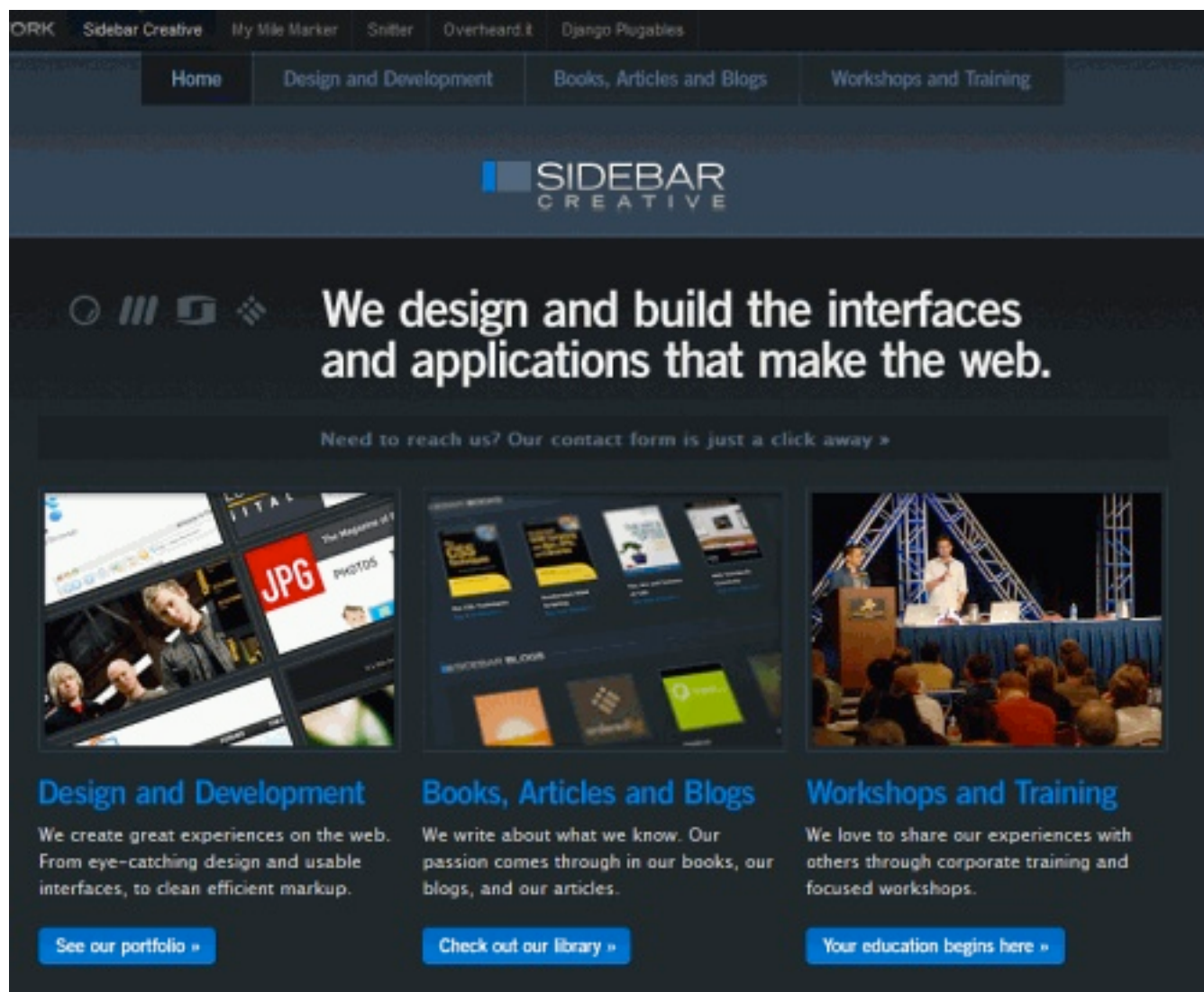
- Be inquisitive, ask open questions
- Offer solutions, talk about the end result benefits for the customer
- Stimulate the senses, let the customer test your product
- Cross your sales, think of all the necessary accessories
- Hit the closing point, sell when the customer is ready to buy.

2. Experiment With Few Colors

Dark body copy on white background and light body copy on black background are passive. Vibrant colors can help to highlight some specific elements of the site which you'd like to point users' attention to.

However, you don't need to use a vibrant palette to create an effective Web design. Sometimes it's useful to pick a moderate number of colors and use them efficiently. Thus your visitors will immediately see what's more

important. It will also be easier for you to focus their attention on the most important areas of the site.



Sidebarecreative.com uses only one dominant vibrant color — blue color stands for links, buttons and clickable elements. White stands for important information which summarizes the content of the current page. The rest is the content of the site with some light blueish hover effects. Excellent, modest and expressive use of colors. That's effective.

Posts

[More posts / Feed](#)

Indistinguishable from magic

I love video games. I'm terrible at most of them. But I'm a sucker for a game with a good story.

[Read / 13 Comments](#)

Airport express

Recently I learned two things about interaction design and user experience from waiting in lines at the airport.

[Read / 14 Comments](#)

Shouts and echoes

There have been some situations lately that have got me thinking a lot about the Internet as a megaphone for personal communication.

[Read / 7 Comments](#)

Mantra

I learned too little from my grandfather, but what I did learn I will remember, and repeat, forever.

[Read](#)

Photos

[More photos](#)

Dry bones

Links

[More links / Feed](#)

The Audacity of Data Fascinating look at some of Obama's key advisors and how they think.

A Camera That Frees Your Face Sony's new dual-sensor DSLR (so you can actually preview images on the LCD screen as you shoot) looks really nice.

I Don't Multitask This is one of the best descriptions I've seen of how I work, and I imagine a lot of people in my generation (I'm 27) would identify with it on a similar level.

Laura Brunow Miner is JPG's New Editor Title says it all. I couldn't be more proud and happy for her — for being amazing at something she really loves.

Erik Svedang Developer's blog for a beautiful-looking independent game called Blueberry Garden.

Hallelujah Great chronicle of the history of a sequence of covers of a great Leonard Cohen song.

Steve Jobs speaks out 15 nuggets of wisdom from the man himself in a Fortune Q&A.

The trouble with Steve Pretty standard-fare mythology-ming article about Steve Jobs, but noted for my favorite quote in the whole piece, from Jean Louis Gasse: "Democracies don't make great products. You need a competent tyrant."

Don't laugh, sugar pills are the future Placebo is powerful medicine. The psychological effects of medicine might be as important as the physiological effects, but since we can't test them, we discount them.

2008: A Type Odyssey Beautiful recreations of the PALL terminal screens from 2001.

Ephemera



Sky Blue Sky / Wilco

I've been listening to this album a lot lately and it's settling in—in a really good way. It's nowhere near the ambition or energy of *Ghost is Born*, but it feels like a natural progression, and it's just a really good album. And I still call Wilco America's greatest living rock band.

Infinite Jest / David Foster Wallace

I had to be recommended this book quite a few times before I actually picked it up, and it had to sit on my shelf for a while before I actually read it. It just seems really dense (which it is) but it's also incredibly entertaining and satisfying to read. Rollicking, even.

The Assassination of Jesse James by the Coward Robert Ford / Andrew Dominik

I couldn't wait to see this movie, and I wasn't disappointed. Such a perfect balance between all the elements that make a great movie: amazing cinematography, impeccable pacing, fascinating performances, pitch-perfect writing.

About the author

[Contact](#)

Wilson Miner is a designer and web developer in San Francisco.



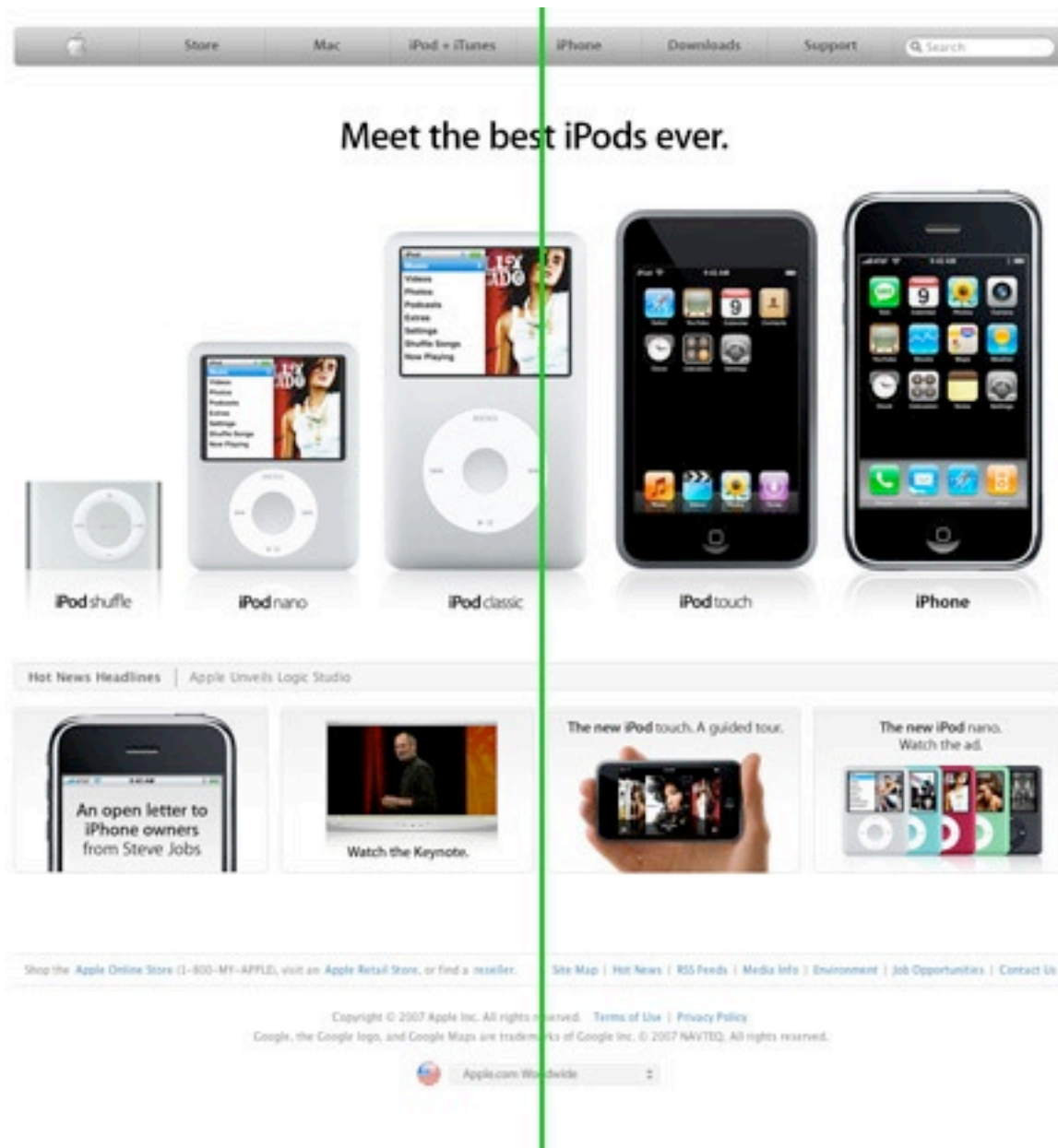
This site's design and contents are copyright © 2002-2008 Wilson Miner. Words and opinions here are mine and nobody else's.

Powered by Django. Hosted by Webfaction.

[Wilson Miner](#) uses green, blue and pink. The dominating green color looks fresh and comforting. Blue stands for the links and pink for the hover effect. Notice the structure of the site. Although only few colors are actually used, one can immediately recognize clickable areas and detect the different sections of the site.

3. Strive For Balance

In Web design balance is important because it provides users with some sense of closure and gives the feeling of permanence and stability. Balanced designs are easier to digest and to understand, because they create a visual hierarchy which clearly displays what is more important on the web-site and what is less important.



Source: [The Elements of Design Applied to the Web](#)

[Apple](#)'s design is probably one of the best examples of the symmetrical (also called *formal*) balance. Single design elements have either very distinctive (above) or very similar (below) weight. They are placed on both sides of an imaginary vertical line on the page which goes through the middle of the user's screen. The design is rather static and restful which

makes it easier for the visitors to comprehend the information which is presented through it. Result: balance creates a simple visual harmony.

Why is Apple's design so efficient? Because it's user-centric and product-centric. The design of Apple's site resembles a theatre stage where visitors are the audience and the product is the entertainer. Notice that the site presents only the product and *nothing else*. Apple manages to initiate the dialogue offering only most important options.

Agentur für Gestaltung & Webentwicklung

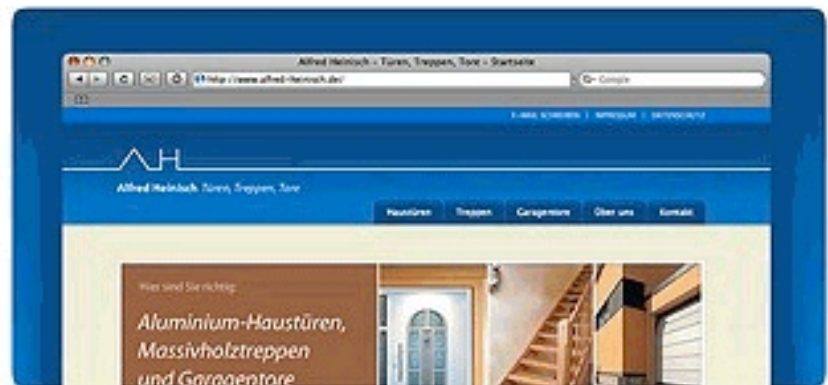
signalfeuer. Das sind die Webentwickler und Grafikdesigner Leif Langsdorf und Dennis Nicolai, die sich nach knapp zehn Jahren gemeinsamer und erfolgreicher Zusammenarbeit im Team, nun auf eigene Beine gestellt haben. Seit April 2008 führen sie eine Internetagentur in Butzbach / Hessen. Neben den klassischen Leistungen einer Internetagentur, bieten sie unter anderem auch grafische Leistungen, wie Logoerstellung und die Entwicklung von Corporate Designs, an.

Aktuelles Projekt

Relaunch: www.alfred-heinisch.de

Türen, Treppen und Tore vom Profi. Für den Fachmann für Türen, Treppen und Tore aus dem Rhein-Main-Gebiet wurde ein kompletter Relaunch der Unternehmenspräsentation vorgenommen.

Neben der vollständigen Neugestaltung und der technischen Umsetzung wurde außerdem ein Suchmodul für die Produktpalette der Haustüren integriert.

[Weiter zu Referenzen](#)


signalfeuer. Agentur für Gestaltung & Webentwicklung

Langgasse 9

35510 Butzbach

[Der Weg zu uns »](#)[Kontaktformular »](#)

Leif Langsdorf Geschäftsführer

Telefon 06033 928 54 51

Telefax 06033 928 54 53

Mobil 0151 149 649 75

Skype llangsdorf

E-Mail leif@signalfeuer.info

Dennis Nicolai Geschäftsführer

Telefon 06033 928 54 52

Telefax 06033 928 54 53

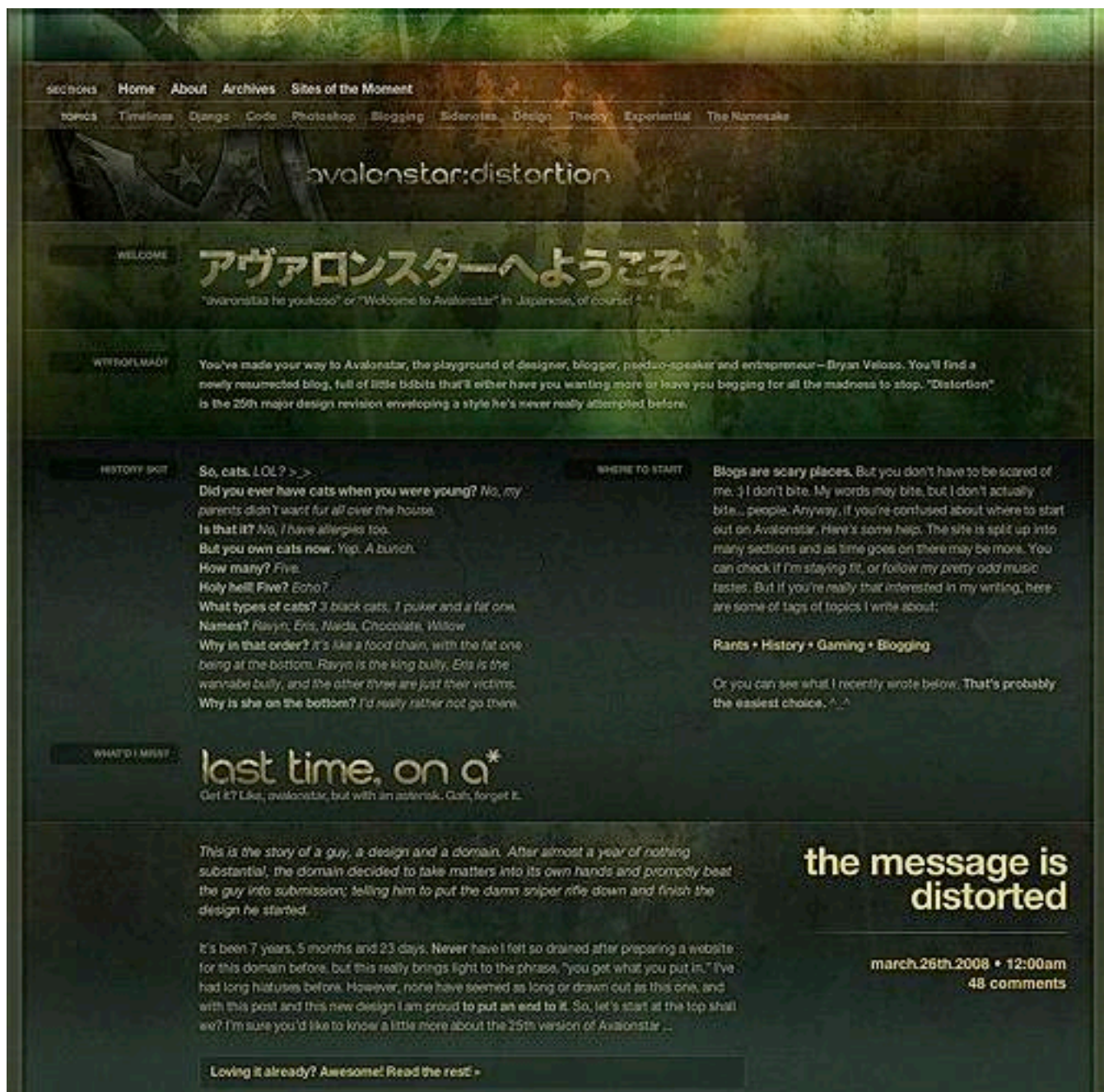
Mobil 0151 149 649 73

Skype dennis.nicolai

E-Mail dennis@signalfeuer.info

Signalfeuer.info uses an asymmetrical (*informal*) balance which allows for the more dynamic use of white space (and this is probably the main advantage of asymmetrical balance). The left block is smaller than the right one, however the site offers a very strong balance by using an impressive and balanced grid. Also the choice of colors is quite impressive.

According to the article [Principles of Design](#), “asymmetrical balance may be unequal in position and intensity. To create asymmetrical balance, there must be an increase in intensity to compensate for the change in position. Intensity can be increased by changing size, shape, or tone. For a particular job, the designer might choose to position the elements to one side of the picture plane. The white space opposing must then act as a counter-balancing force.”



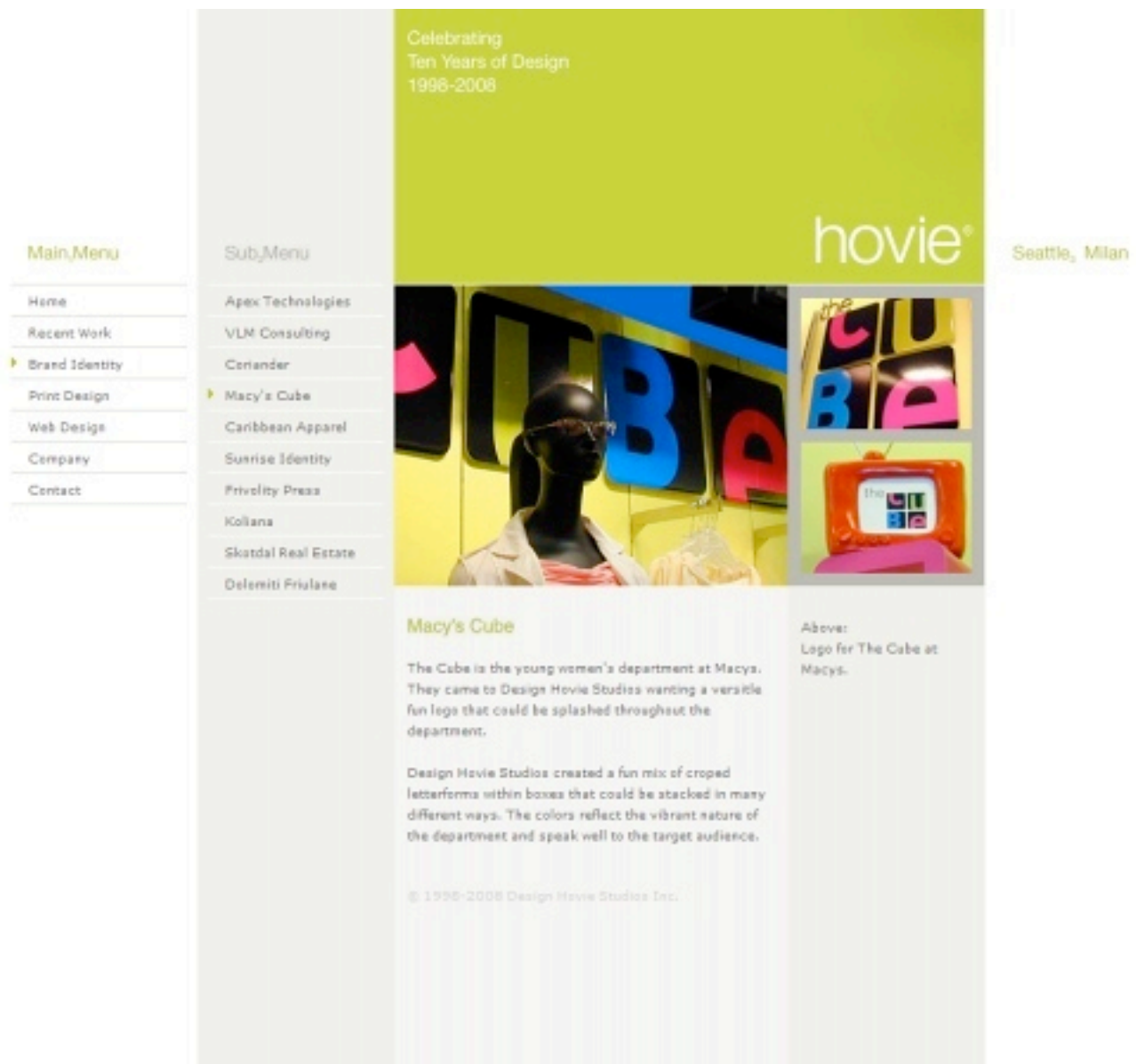
This is exactly what has been done on Avalonstar.com. Notice the different shapes of blocks and “hanging” heading of the blocks. White space functions as a counter-balancing force to the content presented on the right.

[Grid-based approach](#) can be helpful or even necessary for achieving balanced design layouts.

4. Strive For Clarity

Independently of the design approach, the clarity of both layout and presented information should be given one of the highest priorities in the design process. If there is any chance of misunderstandings and ambiguous titles make sure to get rid of them or specify explicitly what is meant. Being more specific is usually better than being less specific.

To achieve clarity you don't need to make use of "standard" design layouts or basic templates. Risk design experiments if you want to, but keep the clear visual hierarchy and structure of the site's content in mind. The more organized your categories are, the easier it will be for your visitors to find their path through your site.



Hovie.com displays an unusual yet very clear design solution. The site consists of five blocks; the first one stands for the main menu which appears when the site is loaded. Sub-menu appears when some section in the main menu is selected.

The current position is displayed by arrows next to the selected option in the navigation menu. In the content area a brief description and the caption of the images displayed above is presented. The design is consistent

throughout all pages. Is there any possibility to somehow misunderstand the design? Not really. That's simple, clear and effective.



Another example: Astheria.com focuses on white space and has a balanced grid-based design solution. Although the design is rather subtle and uses only one appealing color, you can see immediately where you are now and what options are available. Power of typography at its best. Unfortunately, the site doesn't display which links have already been visited (*Update: Kyle Meyer has added :visited-state now*).

5. Address Your Users' Needs

As web-developer you have the primary task to comfort your visitors, addressing their needs and providing them with sufficient clues for seamless and intuitive navigation. However, to achieve this you need more than just considering an average profile of your users and stick to the decision he/she will be happy with.

To maximize your exposure you have to consider various types and profiles of your visitors. You need to identify major groups and offer each group the functionality its members would be comfortable with. That doesn't mean that you need to create multiple versions of your web-site; it rather means that you can integrate multiple levels of user interaction in one single design layout.

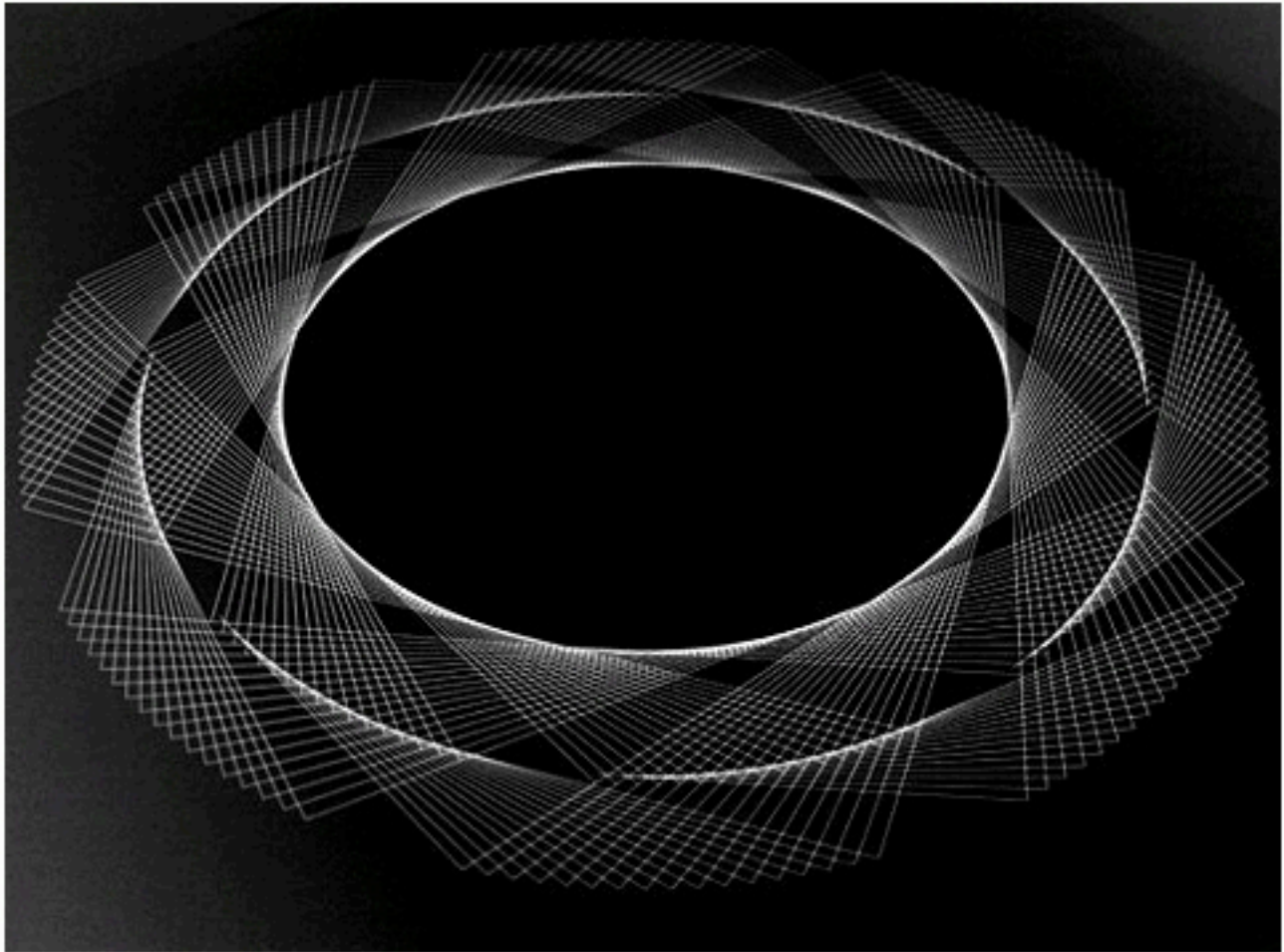
For instance, according to Shneiderman's principles of User Interface Design, it's important to enable frequent users to use shortcuts — to increase the pace of interaction use abbreviations, special keys, hidden commands etc.

[tile\(v\)](#) | [top\(h\)](#) | [prev\(k\)](#) | [next\(j\)](#)

Quoted from: [Edward Hann Portfolio, On Rails.](#)

www.edwardhann.co.uk/images/rails/spiral1.jpg

2 days 12 hours ago [saved by 8 people](#)



[Ffffound.com](#) offers both usual navigation and a Javascript-based keyboard navigation. This functionality is not directly seen to newbies who can browse through the site in a usual manner. Advanced users, however, are able to navigate more efficiently. Of course, you need to make it explicit and clear how to use any advanced functionality and what benefits it offers.

Another possibilities would be bookmarklets, toolbars or some specific tools such as e.g. Flickr uploader. Offering these tools you'll offer your users easier ways to use your service and they'll have no need to search for alternatives.

Conclusion

Effective Web design doesn't have to be colorful and pretty — it needs to be **clear and intuitive**. Make sure you help your visitors to understand the benefits of your web-site and offer them an easy way to explore and use your site.

Symmetrical and asymmetrical balance can be helpful in achieving a solid and clear design solution and thus provide a foundation for rich user interaction. To maximize your exposure, consider different types of users and offer them multiple levels of user interaction which would bring them back to your site.

10 Useful Techniques To Improve Your User Interface Designs

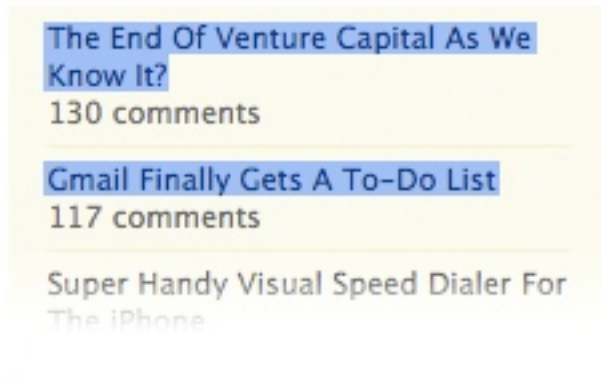
Dmitry Fadeyev

Web design consists, for the most part, of interface design. There are many techniques involved in crafting beautiful and functional interfaces. Here's my collection of 10 that I think you'll find useful in your work. They're not related to any particular theme, but are rather a collection of techniques I use in my own projects. Without further ado, let's get started.

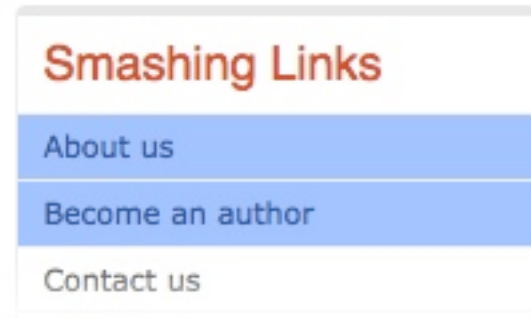
1. Padded block links

Links (or anchors) are inline elements by default, which means that their clickable area spans only the height and width of the text. This clickable area, or the space where you can click to go to that link's destination, can be increased for greater usability. We can do this by adding padding and, in some cases, also converting the link into a block element. Here's an example of inline and padded links, with their clickable areas highlighted to show the difference:

Inline links on TechCrunch:



Padded links on Smashing:



Click areas highlighted blue:



Obviously, the larger the clickable area is, the easier it is to click on the link because there is less of a chance of missing it. Converting links into block elements makes the text area span the whole width of the container, unless the width is specified otherwise. This makes it ideal for links located in sidebars. We can do it with the following code:

```
a {  
  display: block;  
  padding: 6px;  
}
```

Make sure to also add a healthy dose of padding to the links, because converting a link into a block only affects its behavior and width; adding padding ensures that the link is high enough and has some room to breathe.

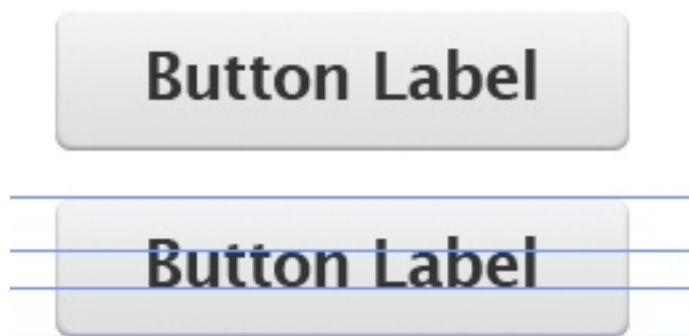
2. Typesetting buttons

Attention to every detail is what separates a great product from a mediocre one. Interface elements such as buttons and tabs are clicked on many times

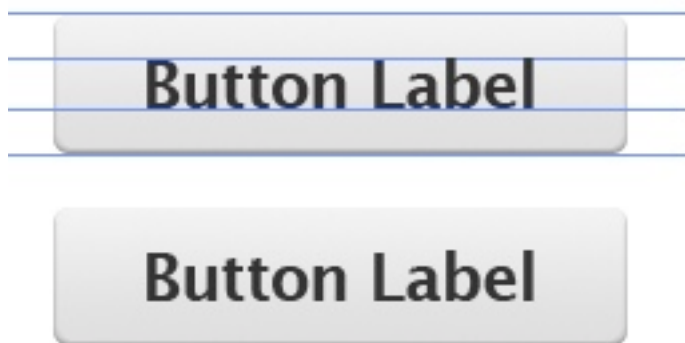
a day by your users, so it pays to typeset them properly; and by typesetting I mean positioning the label. Here's a couple of examples of the kind of misplaced labels I sometimes notice:



At first glance they look okay, but notice that the text is placed too high because the lowercase letters have been used as a guide to align the text vertically in the center, like so:



However, if we use uppercase letters as well as lowercase letters with ascenders (“t,” “d,” “f,” “h,” “k” and “l”), the balance shifts upwards, making the label appear too high on the button. In such cases, we should set the type using the uppercase height as a guide — or set it a little bit higher if most of the letters are lowercase. Here's what I mean:



This gives the whole button a more balanced look and feel. Little touches like this go a long way towards making your interface more polished and satisfying to use.

3. Using contrast to manage focus

Similarly, you can also manage the focus of your visitors' attention with contrast between elements. Here's an example of a post headline and some meta information underneath regarding who posted the article and its date:

Lorem Ipsum Dolor Sit Amet
1st Jan, 50 BC - by *Cicero*

All the text is set in black. Let's decrease the contrast between the meta information (the date and author's name) and the background by putting the text in a light shade of gray:

Lorem Ipsum Dolor Sit Amet

1st Jan, 50 BC - by *Cicero*

The highest contrast element here is the headline, so it literally pops out at us. The other elements fade into the background. Here, I've chosen the author as the second-most important element, and the date as the least. The font also differs in size and style, but the contrast level can be very powerful. Let's reverse the order of importance to date, author and headline:

Lorem Ipsum Dolor Sit Amet

1st Jan, 50 BC - by *Cicero*

You can see how effective it is in shifting focus: the date now pops out at you, while the headline fades away. This technique comes in very handy for information-heavy websites, such as blogs, forums and social networks, in which you want to make a lot of information easily scannable while still showing a lot of additional things, like dates. Fading the extras allows visitors to easily focus their attention on the most important pieces of text.

4. Using color to manage attention

Color can also be used to effectively focus your visitors' attention on important or actionable elements. For example, during the US presidential election, pretty much all of the candidates' websites had the donation button colored red. Red is a very bright and powerful color so it attracts attention,

and it stands out even more when the rest of the website is blue or another colder color.

Warmer tones like red, yellow and orange are naturally bright and so tend to attract the eye. They also “expand” when set against colder colors like blue and green. This means that an orange button on a blue background looks like it’s flowing outwards and taking the front seat. Conversely, a blue button on an orange background contracts inward, wishing to stay in the background. Here’s a picture to illustrate:



Here’s a couple of examples of websites that use color effectively to direct users’ attention to the important elements:



Function features a “We’re Hiring” link on its jobs page. To ensure the link is not missed, the designers set it against a red background that pops out from the dark background header, effectively grabbing attention.



Causecast use color effectively. Four bright pink elements pop out at you: the logo, the feedback link, the donate link and the website description message.

Want the “About” blurb on your website to grab the visitor’s focus? Make the background yellow. Want to make the “Join” button stand out? Color it orange. Make sure not to highlight too many elements, though; if you do, they may get lost in each other’s company.

5. White space indicates relationships

One of the most crucial elements in an interface is the white space between elements. If you’re not familiar with the term white space, it means just that: space between one interface element and another, be it a button, a navigation bar, article text, a headline and so on. By manipulating white space, we can indicate relationships between certain elements or groups of elements.

So, for example, by putting the headline near the article text we indicate that it is related to that text. The text is then placed farther away from other elements to separate it and make it more readable. Here's an example in which white space could be improved:

Usability Inspection

Usability inspection is a review of a system based on a set of guidelines. The review covers the concepts of usability in design. The experts focus on a list of areas in design that

Pluralistic Inspection

Pluralistic Inspections are meetings where users, developers, and human factors experts review a design. As more people inspect the scenario for problems, the higher the probability to find issues are resolved.

Consistency Inspection

In consistency inspection, expert designers review products or projects to ensure consistency with their own designs.

The text looks all right and is certainly readable, but because the spaces above and below each heading are equal, they don't separate each piece of text clearly. We can improve this by increasing the white space between each section and also by slightly tightening the line height of the paragraphs:

Usability Inspection

Usability inspection is a review of a system based on a set of guidelines. The review focuses on the concepts of usability in design. The experts focus on a list of areas in design that are likely to cause usability problems.

Pluralistic Inspection

Pluralistic Inspections are meetings where users, developers, and human factors experts review a design. As more people inspect the scenario for problems, the higher the probability to find and resolve issues are resolved.

Consistency Inspection

In consistency inspection, expert designers review products or projects to ensure they are consistent with their own designs.

This results in more clearly defined blocks; we can easily tell which headings go with which pieces of text and can see the separate sections clearly. Good designers often squint or glance at their work from a distance, which lets them see the blocks of elements separated by white space as they merge together. If you cannot see these groups clearly then you may need to tweak your white space.

6. Letter spacing

Web design is pretty limiting for typographers. But while there are only a few safe Web fonts and not a great many things you can do to style them, it's worth remembering that we do still have some level of control. "Tracking" is a term used in the field of typography to describe the adjustment of spacing between letters in words. We've got the ability to do this with CSS using the *"letter-spacing"* property.

If used with restraint and taste, this property can be effective in improving the look of your headlines. I wouldn't recommend using letter spacing on

the body text because the default spacing generally provides the best readability for smaller font sizes.

Here's an example of letter spacing in use:

h1. Lorem ipsum dolor

h2. Lorem ipsum dolor

H3. LOREM IPSUM DOLOR

And here's the CSS code used for the above examples:

```
h1 {  
  font-family: Helvetica;  
  font-size: 27px;  
}  
  
h2 {  
  font-family: Helvetica;  
  font-size: 27px;  
  letter-spacing: -1px;  
}  
  
h3 {  
  font-family: Georgia;  
  font-size: 24px;  
  letter-spacing: 3px;  
  font-variant: small-caps;  
  font-weight: normal;  
}
```

The effect can be useful when you want to craft a more aesthetically pleasing or more original heading. Here, I've used only a couple pixels for letter spacing, but already it makes a big difference to the style of the font.

7. Auto-focus on input

Many Web applications and websites feature forms. These may be search forms or input forms inviting you to submit something. If this form is the **core feature** of your application or website, you may want to consider **automatically focusing the user's cursor on the input field** when the website loads. This will speed things up because users can start typing right away without having to click on it. A good example of this is Google and Wikipedia's websites.



Upon arriving at [Wikipedia.org](https://www.wikipedia.org), the search box is already highlighted, ready to accept text.

To automatically focus on input fields, you'll need a little bit of JavaScript. There are various solutions, and the one you should use depends on the functionality you want to achieve. The simplest way to do it would be to add the following to your body tag:

```
<body onLoad="document.forms.form_name.form_field.focus()">
```

Your form code should look something like:

```
<form method="get" name="form_name" action="#">
<input type="text" name="form_field" size="20" />
<input type="submit" value="Go" />
</form>
```

Now, every time the page loads, the text field called “form_field” will be automatically selected, ready for input.

The only problem with this is that if your users want to return to the previous page using the Backspace key, they will be out of luck because they’ll just be deleting characters in the input field. Thankfully, Harmen Janssen has another simple JavaScript solution you can find [here](#). Harmen’s script allows the Backspace key to go to the previous page when there are no characters left in the input field to delete.

8. Custom input focus

While the default look of form elements suffices for most functions, sometimes we want something a little prettier or a little more standardized across various browsers and systems. We can style input fields by simply targeting it with an “id,” “class” or plain old “input,” like so:

```
input {
  border: 2px solid #888;
  padding: 4px;
  font-size: 1em;
  background-color: #F8F8F8;
}
```

Default input text field (OS X, Firefox):

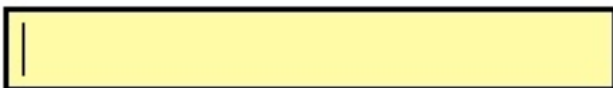
A simple, empty rectangular text input field with a thin grey border, representing the default style in OS X and Firefox.

Styled input text field:

A rectangular text input field with a light grey background and a thin grey border. A vertical cursor is visible on the left side, indicating it is in focus.

What's more interesting is also being able to style the input field when it's in focus; that is, the state it's in when it has been clicked. To do this, we need to attach a “**:focus**” after the “input” property:

```
input:focus {  
  border-color: #000;  
  background-color: #FFFE9D;  
}
```

A rectangular text input field with a yellow background and a thick black border, representing the focus state defined by the CSS code above.

If you're using custom backgrounds to style your input field, they may clash with some browsers and operating systems' default focus styles. For example, here's a screenshot of a custom-styled form clashing with the default blue OS X glow effect:



In such cases, you could also use the “input:focus” property to remove the default styling. The default blue glow in the screenshot above can be removed by disabling the “outline” property:

```
input:focus {  
  outline: none;  
}
```

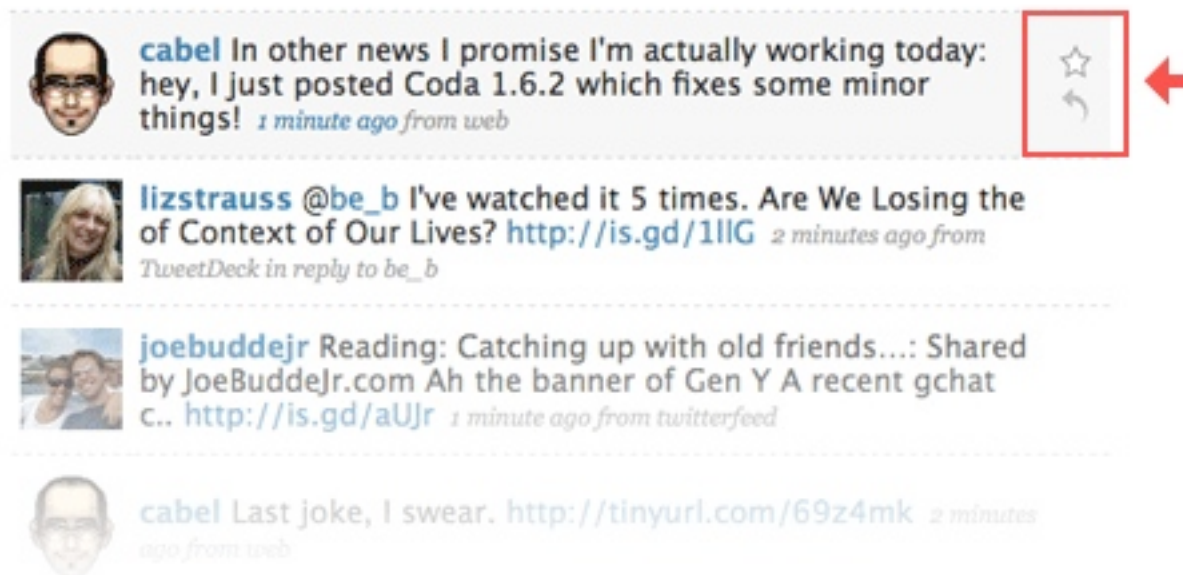
The blue glow effect will now be gone:



Obviously you would only want to remove the outline if you’re replacing it with your own styling, so that you don’t negatively affect the accessibility and usability of your forms.

9. Hover controls

Some Web applications have extra utility controls, such as edit and delete buttons, that don’t necessarily have to be shown beside every item at all times. They can be hidden to simplify the interface and focus visitors’ attention on the main controls and content. For example, these hover controls are used in Twitter when you hover over messages:



These hover controls can be achieved with some simple CSS code, without any JavaScript. Simply style the <div> with the controls when its parent <div> is in a hover state. Here's the code to hide and show the controls (using a <div> with the class "controls" inside a <div> with the class "message"):

```
.message .controls { display: none; }  
.message:hover .controls { display: block; }
```

When you hover over the "message" <div>, the "controls" <div> inside it will appear, along with all of its content, giving you the same functionality as shown in the Twitter screenshot above.

There may be an issue with accessibility because screen readers may not be able to read the hidden <div>. There are plenty of other ways to hide the inner <div>, such as offsetting it with a negative margin that takes it off the page (e.g. "left-margin: -9999px"), coloring its text the same color as the background or simply placing another <div> on top of it.

This technique should of course be used with restraint because you don't want to hide your important controls; but if used correctly, it can be useful for cleaning up your interface by removing those extra utility links that you don't want to show up at all times.

Note that this doesn't work in IE6, so you'll need to override the hiding property in your IE6-specific style sheet or, if you don't have one, simply use the following IE6-specific code inside the <head> section of your code:

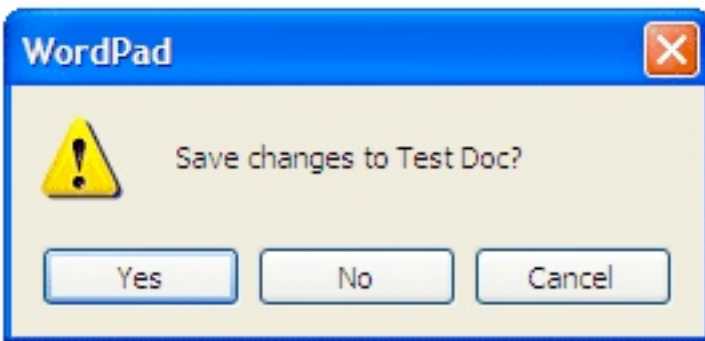
```
<!--[if lt IE 7]>
  <style type="text/css" media="screen">
    .message .controls { display: block; }
  </style>
<![endif]-->
```

10. Verbs in labels

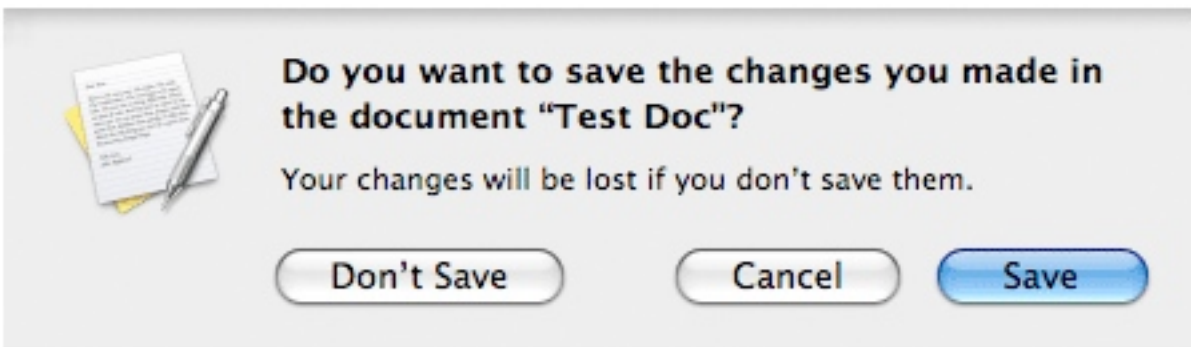
You can make options dialogs much more usable by thinking through the labels you use on buttons and links. If an error or message pops up and the options are “Yes,” “No” and “Cancel,” you have to read the whole message to be able to answer. Seems normal, right?

But we can actually speed things up by using verbs in the labels. So, if instead of “Yes,” “No” and “Cancel,” we have “Save,” “Don't Save” and “Cancel” buttons, you wouldn't even need to read the message to understand what the options are and which action to perform. All the information is contained in the button labels.

Windows XP WordPad save dialog:



OS X TextEdit save dialog:



Using verbs in labels on buttons and links makes the options dialogs more usable because the labels contain all of the information the user needs to be able to make a decision.

10 Useful Web Application Interface Techniques

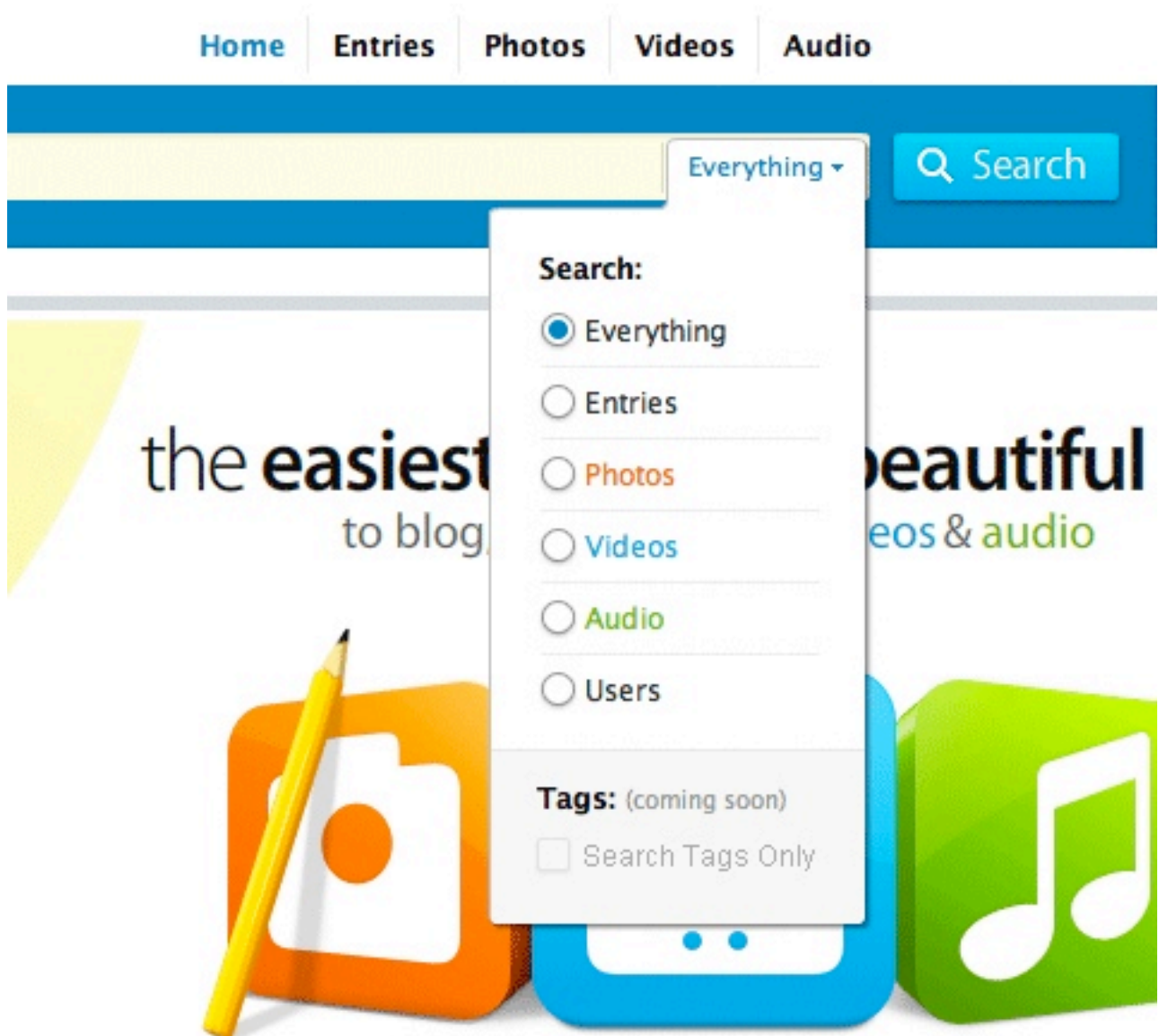
Dmitry Fadeyev

More and more applications these days are migrating to the Web. Without platform constraints or installation requirements, the software-as-a-service model looks very attractive. Web application interface design is, at its core, Web design; however, its focus is mainly on function. To compete with desktop applications, Web apps must offer simple, intuitive and responsive user interfaces that let their users get things done with less effort and time.

In the past we didn't cover web applications the way we should and now it's time to take a closer look at some useful techniques and design solutions that make web-applications more user-friendly and more beautiful. This article presents the first part of our extensive research on design patterns and useful design solutions in modern web applications. Below you'll find a collection of 10 useful interface design techniques and best practices used in many successful web-applications.

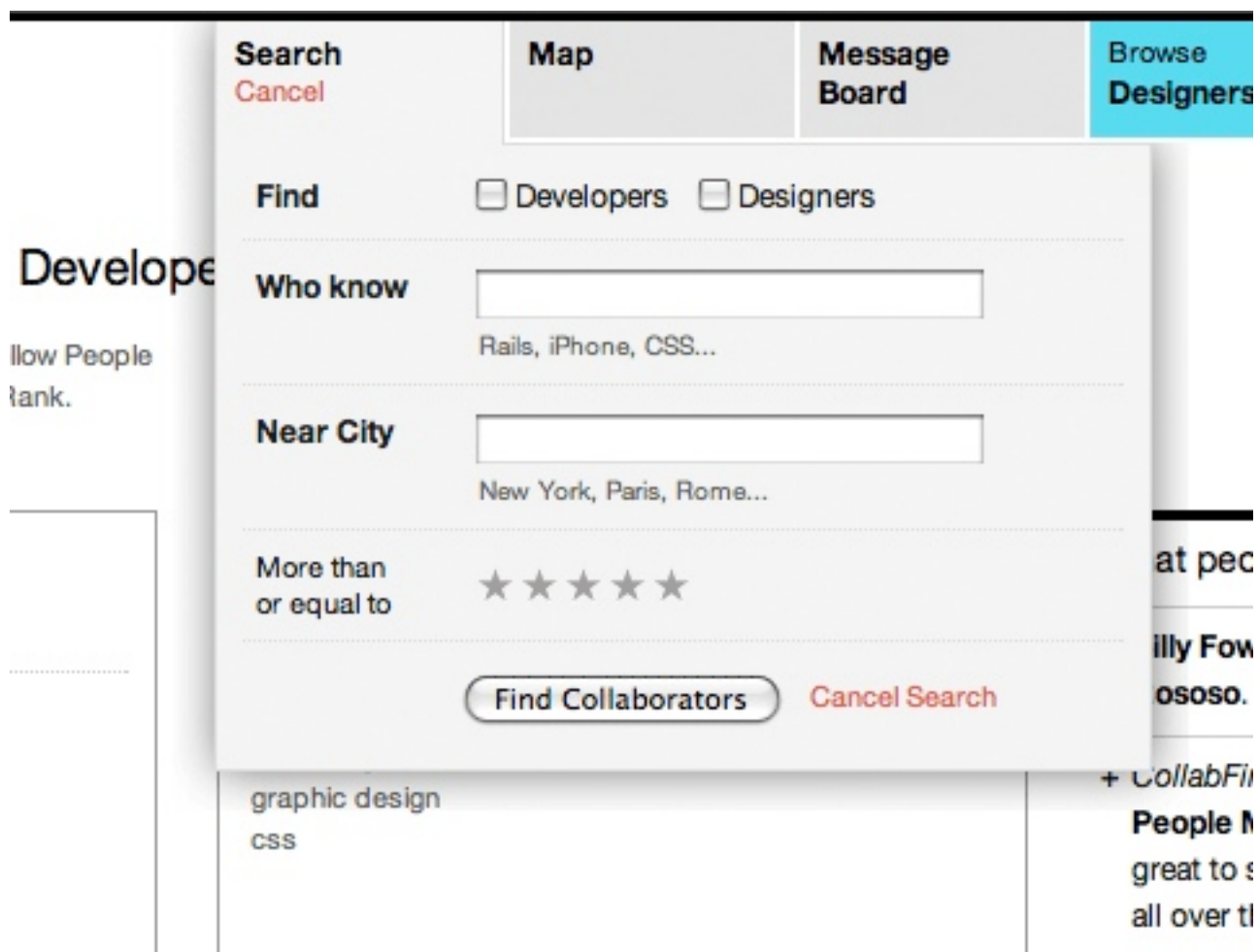
1. Interface elements on demand

Simplicity is important in user interface design. The more controls you display on the screen at any time, the more time your users will have to spend figuring out how to use your interface. When there is less choice, the available functions become more apparent and are easier to scan. Simplifying an interface isn't easy though, especially if you don't want to limit the app's functionality.



When you click on the search link in [Kontain's](#) search box, a similar drop-down menu appears. So, if you need to narrow your search, you can use the menu to select the sort of content you're looking for. Tucking these options away simplifies the search box.

One way of making things simpler is to hide or conceal advanced functionality. Find out the most commonly used functions of your interface and tuck away the rest. You can do this with pop-up menus and controls, which are very common on desktop software. For example, if your search bar has advanced filters, put them away in a special drop-down menu at the end. If users need those filters, they can enable them with just a couple of clicks. Deciding what to keep and what to conceal isn't a simple task, though, and will depend on how important and how frequently used each of the controls is.



When you click on the search link in [CollabFinder](#), you aren't taken to a different page. Instead, the search box controls drop down, allowing you to begin your search straight away.

2. Specialized controls

It's important to select the right interface controls for the situation. Different situations can be handled in different ways, and certain controls are better at their intended task than others.

The image shows a 'Set a reminder' dialog box. At the top, there's a title bar 'Set a reminder'. Below it is a text input field. To the right of the input field is a 'When?' label and a dropdown menu currently showing 'At a specific time...'. To the right of the dropdown is a 'Set reminder' button. Below the dropdown is a compact calendar for January 2009. The calendar shows days of the week (S, M, T, W, T, F, S) and dates. The date '8' is highlighted in black, indicating it is the selected date. Below the calendar, it says 'Due: January 8, 2009'. Underneath that is a time picker with 'at' followed by two spinners for hours (1) and minutes (20), and a dropdown for 'am'. At the bottom, there's a question 'Repeat this reminder?' followed by a dropdown menu currently showing 'no'.

Set a reminder

When? At a specific time... Set reminder

<< Jan 2009 >>

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Due: January 8, 2009

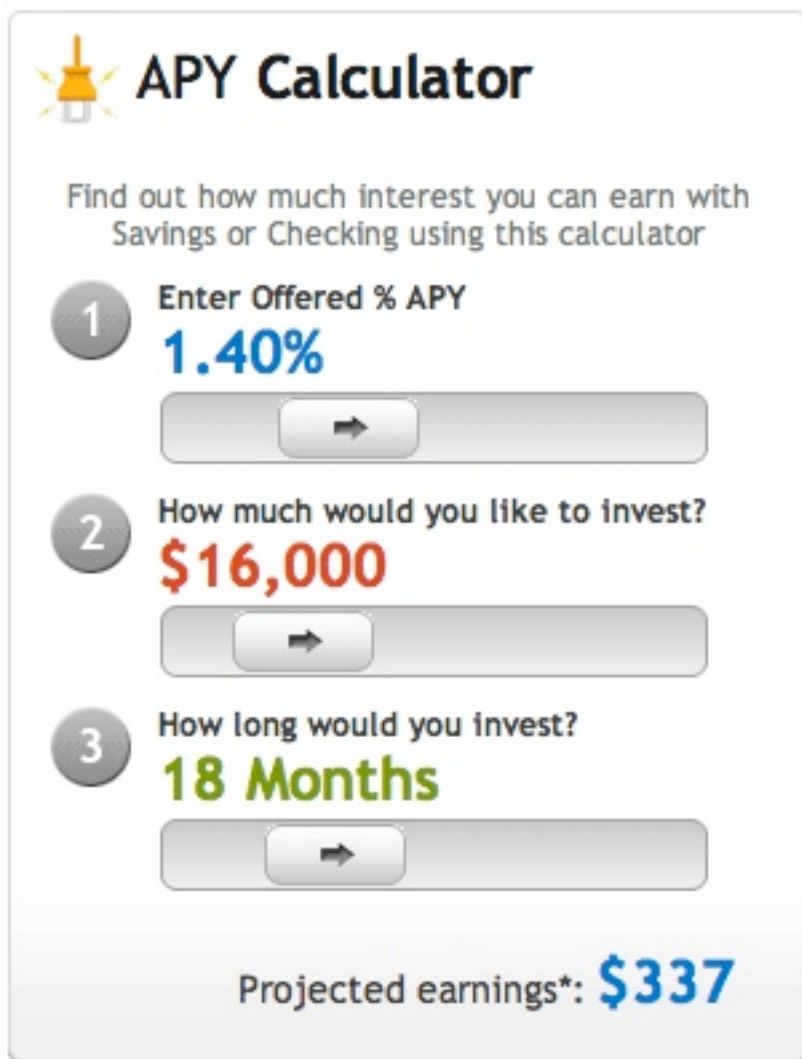
at 1 : 20 am


Repeat this reminder?

no

[Backpack](#) has a compact calendar date and time picker for selecting a reminder date.

For example, you can select a date by using drop-down lists for day, month and year. Drop-downs aren't very efficient, however, when compared to a calendar picker, where you can click directly on a day you want. Calendar pickers also help you see the days, weeks and months (and especially workdays and weekends) more easily and so allow you to make a more informed decision more quickly than you would with a simple drop-down list.



 **APY Calculator**

Find out how much interest you can earn with Savings or Checking using this calculator

- 1** Enter Offered % APY
1.40%
- 2** How much would you like to invest?
\$16,000
- 3** How long would you invest?
18 Months

Projected earnings*: **\$337**

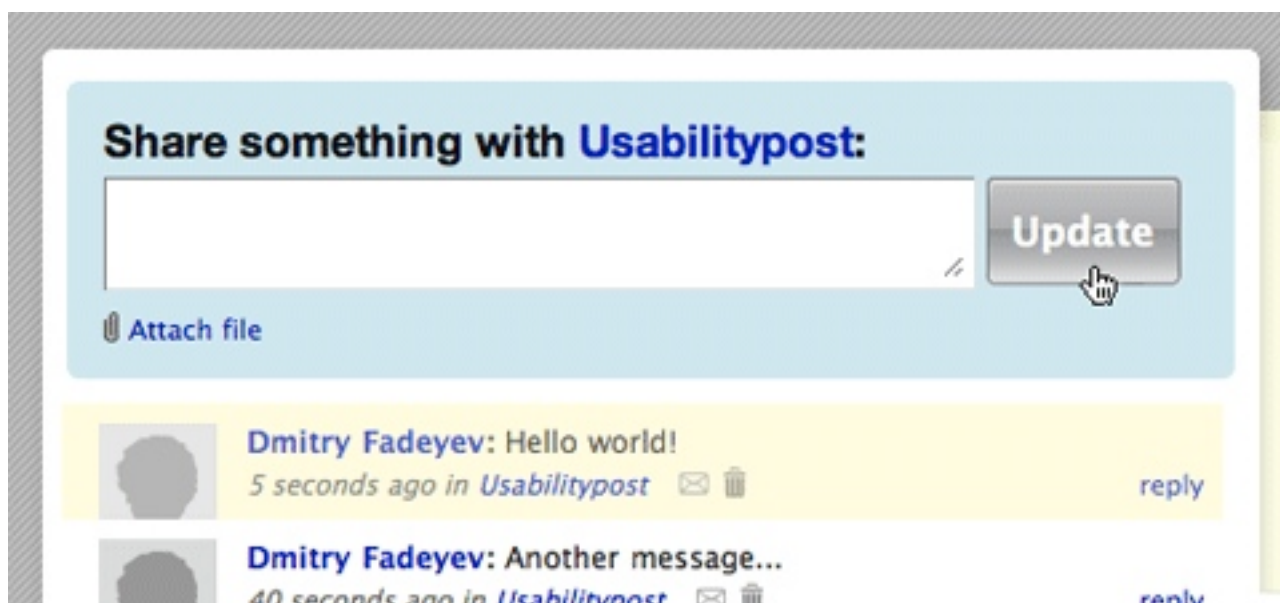
[*MyBankTracker's*](#) APY calculator features easy-to-use slider controls for quickly trying out different projections.

Another good example of this are sliders. Yes, you can always input a number manually, but for certain situations, slider controls do a much better job. Not only are they easy to use — just click and drag — but you can also see how your selection fits between the minimum and maximum of an available range.

3. Disable pressed buttons

One of the problems Web applications encounters with forms is the submission process. With very simple forms, if you click the “Submit” button twice or more very quickly, the form will be submitted two or more times. This is obviously problematic because it will create duplicates of the same item. Preventing duplicate submissions isn’t very hard, and it is essential to do this for most Web apps.

There are two tiers to this safeguard: client-side and server-side. We won’t go through the server-side safeguard here because this will vary depending on the programming language you use and your back-end architecture. What you should essentially do is put in a check to ensure during the processing stage that whatever is being submitted is not a duplicate, and if it is to block it.



[Yammer](#) disables the “Update” button while your new message is being submitted.

The client-side stage is much simpler. All you have to do is disable the “Submit” button the very moment it is clicked. The easiest way to do this is to add a piece of JavaScript to the “Submit” button like this:

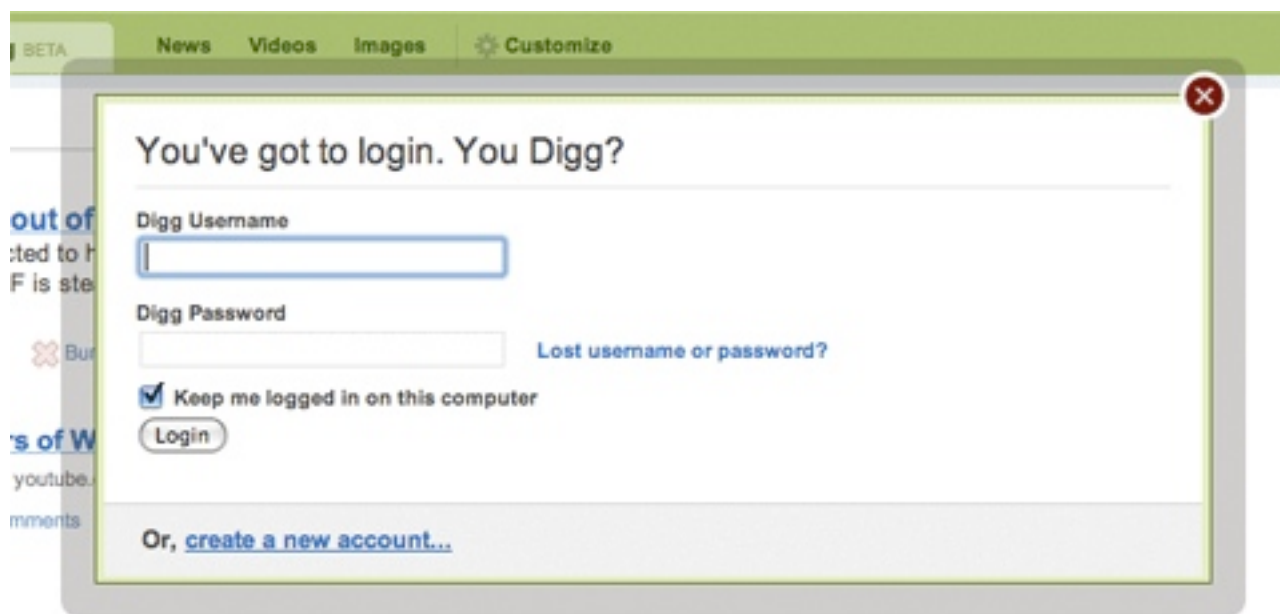
```
<input type="submit" value="Submit"
onclick="this.disabled=true" />
```

Of course, we would advise you to also implement server-side checks to be sure that duplicates don’t get through.

4. Shadows around modal windows

Drop shadows around pop-up menus and windows aren’t just eye candy. They help the menu or window stand out from the background by reinforcing its dimensions. They also block out the noise of the content beneath the window by darkening the area around it with a shadow.

This technique has its roots in traditional desktop applications and helps the user to focus his/her attention on the appearing window. Since most modal windows aren't as easy to distinguish from the main content as in desktop applications, shadows help them to appear closer to readers, because the window appears to be three-dimensional and lay above the rest of the page.



[Digg's](#) log-in window has a thick shadow around it to block out the noise of the page beneath.

To achieve this effect, designers often create a container with a transparent PNG-image as background and place the content inside the container – with equidistant padding on all sides of the box. Another option is to use a background image with transparent borders and position the content box within this box using absolute positioning. This is exactly what Digg does — [this](#) is the image they are using (`dialog.png`). And this is the markup and CSS-style they are using:

(X)HTML:

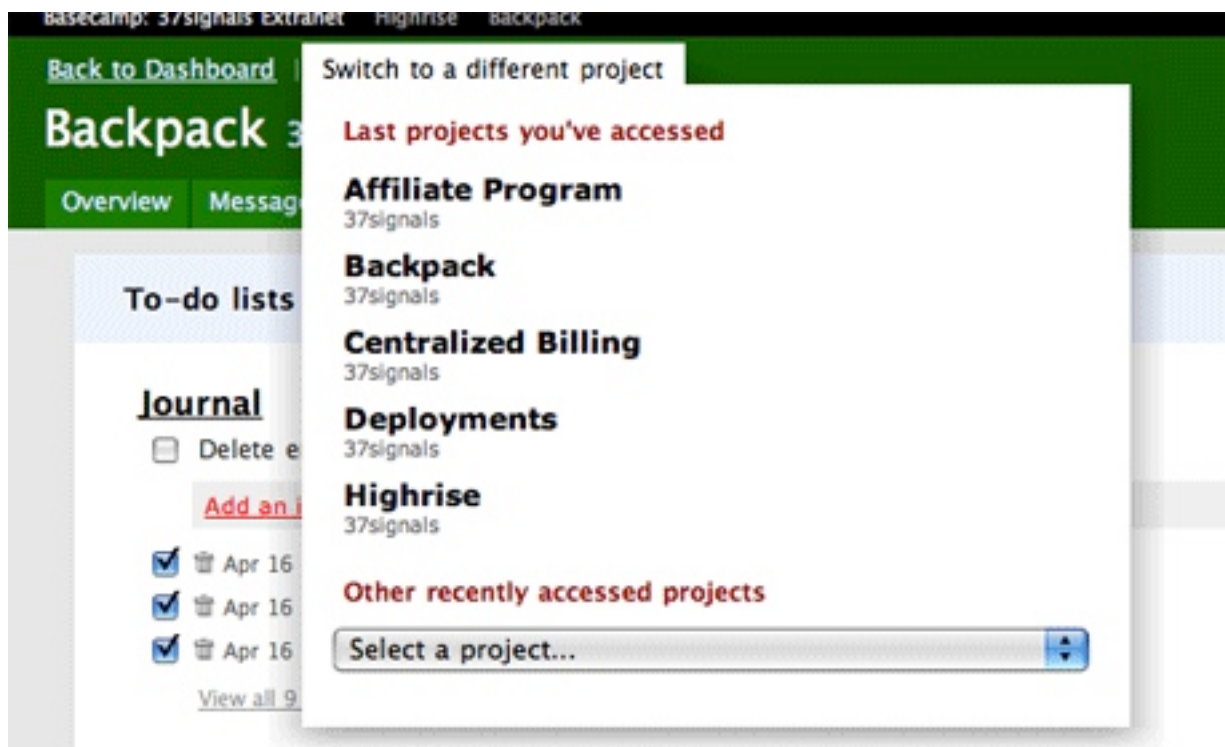
```
<div id="container">
  <div style="display: block; top: 236px; opacity: 1;"
class="dialog">
    <div class="body">
        <div class="content">
            ...
        </div>
    </div>
</div>
</div>
```

CSS:

```
.dialog {
    position: absolute;
    left: 50%;
    margin-left: -315px;
    width: 630px;
    z-index: 100001;
}

.dialog .body {
    background: url(/img/dialog.png) 0 0; /* semi-transparent .png image */
    padding: 40px 13px 10px 40px;
}
```

Alternatively, you can also use JavaScript-based light boxes or drop shadows using [CSS3-attributes we've described earlier](#), but you need to be aware that Internet Explorer won't support them.



[*Basecamp's*](#) project switcher window has a large soft drop shadow that helps the menu area stand out.

5. Empty states that tell you what to do

When you're designing a Web application, it's important not only to test it with sample data, but to ensure that it looks good and is helpful when there is nothing there yet. You should design the empty states.

When there is no information for a page or query yet, a helpful message telling the user how to start could go in that empty space. For example, a project management application's home page may list the user's projects, but if there are no projects yet, you could provide a link to the project creation page. Even if there is already a button to do that on the page, an extra bit of help doesn't hurt.

Welcome to Campaign Monitor

You're currently in the Manage Clients section of your account. To get you started, we've added Vitaly as the first client you'll be sending campaigns for. You can add as many clients as you like.

Once you start creating and sending campaigns, this page will list all the activity across your account.

 **Create your first campaign**

[Campaign Monitor](#) points you in the right direction when you start building an email campaign.

This technique encourages users to actually try out the service and proceed directly with using the service after registration. Guiding the user through single steps of the application may help him or her to understand what advantages the application offers and if it's useful or not. It is also important to present most important options to the users and only them — it doesn't make sense to flood them with numerous options. Keep in mind that users usually want to get a more or less concrete idea of what is offered to them, but they don't want to jump into details — they have neither time nor interest in it.

Using empty states to motivate users and animate actions, you can significantly reduce the amount of “drop-outs” and help your potential clients to gain a better understanding of how the system works.



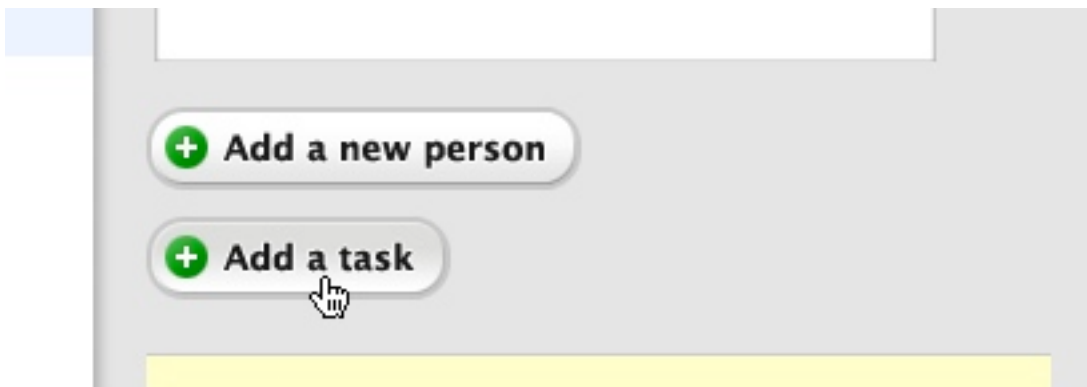
[Wufoo's](#) forms page has a large, friendly message inviting you to create a new form if none yet exist.

6. Pressed button states

Many Web applications have custom-styled buttons. These are anchors or input buttons that have custom images assigned as their backgrounds. The default input buttons may not be suitable in some cases, and the text links are sometimes too subtle. The challenge is, when you make your links look like buttons, they should act like buttons — and this includes having a “pressed” look when the user clicks on them.

This isn't a purely visual tweak. Giving instant feedback to the user will make the application feel more responsive and bring the experience closer to what the user experiences on desktop applications.

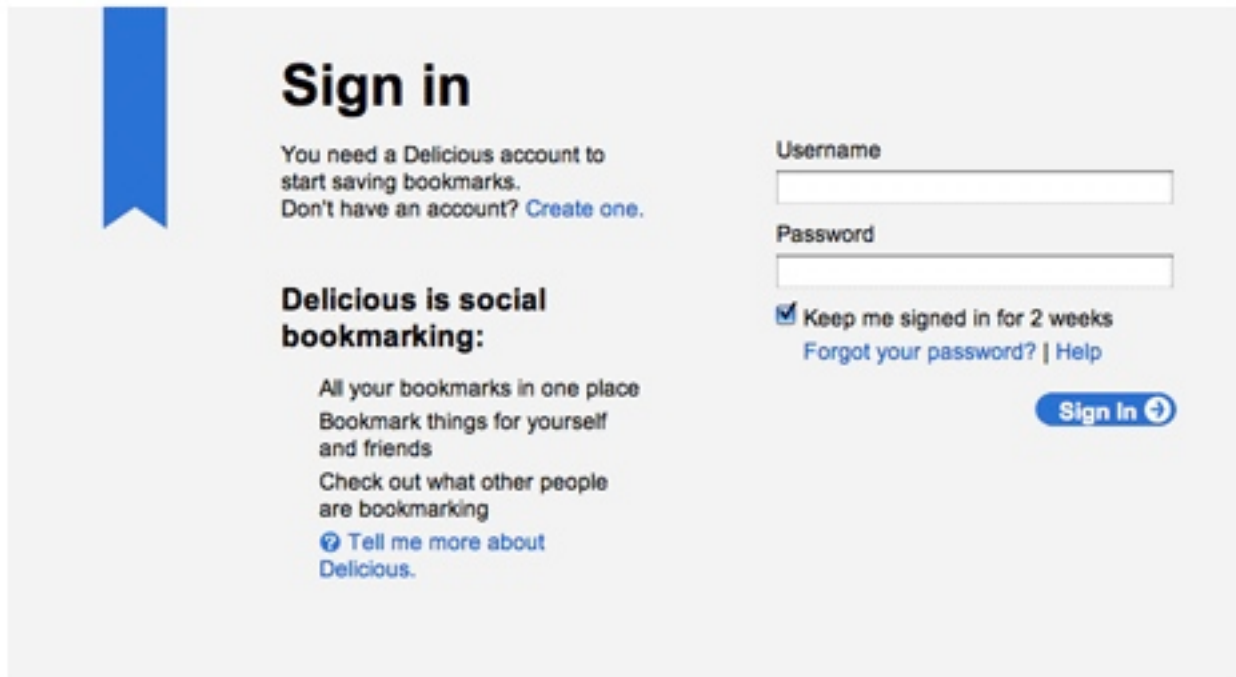
You can add a pressed button state with CSS by styling the `active` pseudo-class of the link in question. So for example, if your anchor has the class `add_task_button`, you can style its active class by targeting `add_task_button:active`.



Buttons in [Highrise](#) actually show a pressed state when you click on them, providing the user with a satisfying responsive feel.

7. Link to the sign-up page from the log-in page

Some people who haven't yet signed up to your application will inevitably end up on the log-in page. They likely want to try out your application but can't find the registration page in a hurry. Perhaps they've tried accessing a feature that's only available to registered users.

The image shows a 'Sign in' page for Delicious. On the left, there is a blue bookmark icon. The main heading is 'Sign in'. Below it, a message states: 'You need a Delicious account to start saving bookmarks. Don't have an account? [Create one.](#)'. To the right of this message is a section titled 'Delicious is social bookmarking:' followed by three lines of text: 'All your bookmarks in one place', 'Bookmark things for yourself and friends', and 'Check out what other people are bookmarking'. Below this text is a link: 'Tell me more about Delicious.' To the right of the main heading, there are two input fields labeled 'Username' and 'Password'. Below these fields are two checkboxes: 'Keep me signed in for 2 weeks' (checked) and 'Forgot your password? | Help'. At the bottom right of the form area is a blue 'Sign In' button with a right-pointing arrow icon.

Sign in

You need a Delicious account to start saving bookmarks.
Don't have an account? [Create one.](#)

Delicious is social bookmarking:

All your bookmarks in one place
Bookmark things for yourself and friends
Check out what other people are bookmarking
[Tell me more about Delicious.](#)

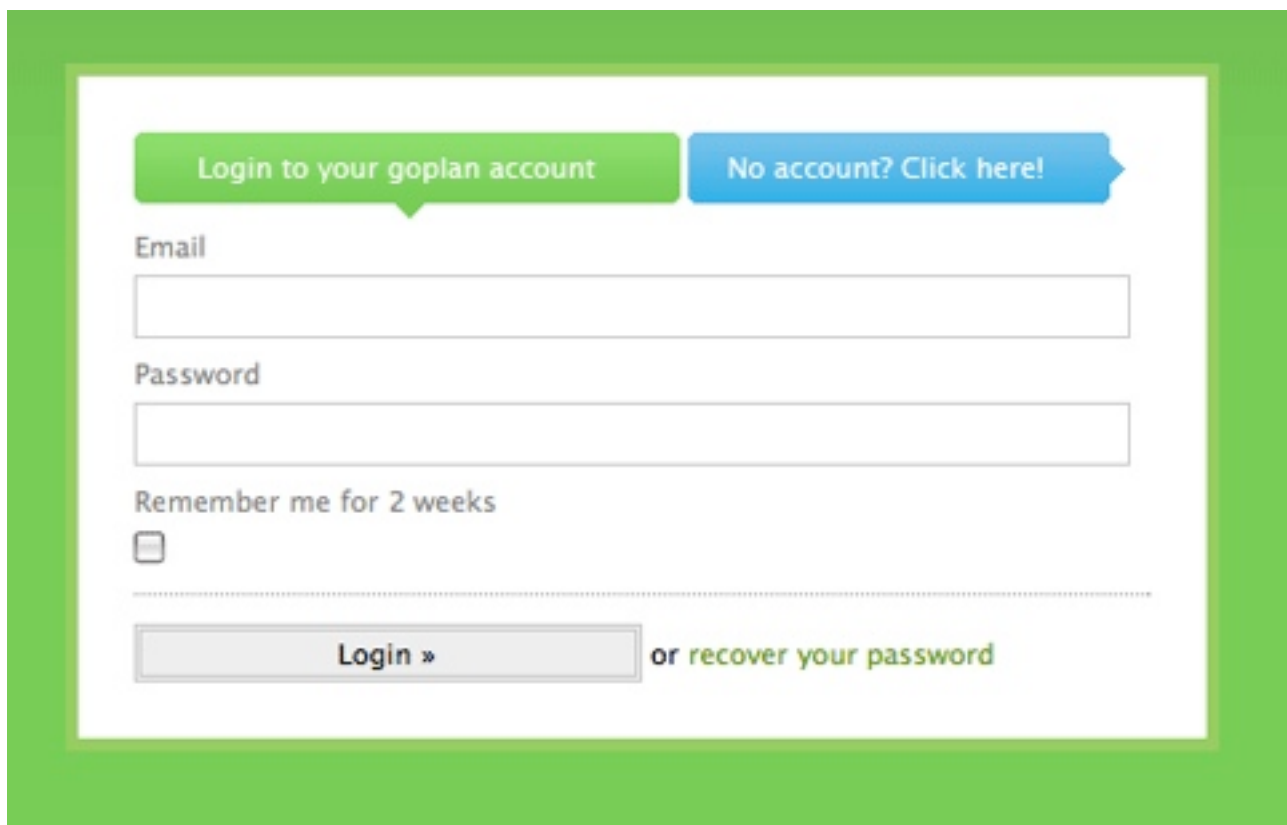
Username

Password

☒ Keep me signed in for 2 weeks
[Forgot your password?](#) | [Help](#)

[Sign In](#)

Don't have a [Delicious](#) account? No problem; a sign-up link is provided on the Delicious log-in page.



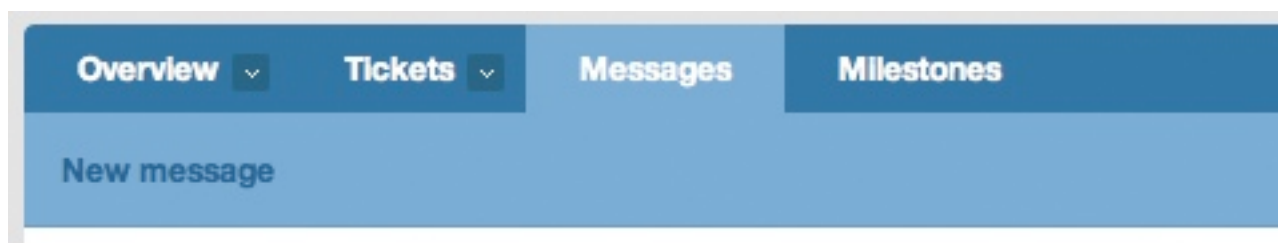
[Goplan](#) has a nice colored button on the log-in page pointing to the sign-up page.

Make things easy for these folks by placing a registration link on your log-in pages. If they haven't got an account yet, they shouldn't have to look for a registration page. Our studies [confirm](#): 18% have a sign-in form or a link to the sign-in form placed next to it (e.g. YouTube, Reddit, Digg, Lulu, Metacafe).

8. Context-sensitive navigation

It's important to think about what the user expects to see and what they need in every given context. You don't need to display the same navigation controls everywhere because users simply may not need them in every situation.

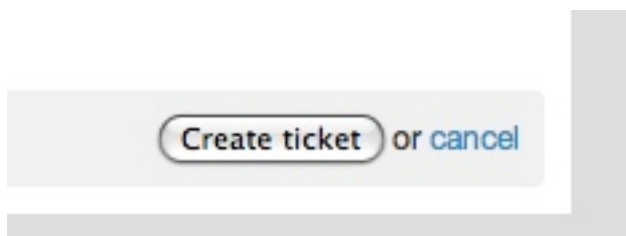
One of the best examples of context-sensitive controls is the recent change in the Microsoft Office 2007 interface, in which the default set of toolbars was replaced by ribbon controls. Each tab on the ribbon holds different controls relating to a particular activity, be it editing graphs, proofreading or simply writing. Web applications can also benefit from such context-sensitive controls because these controls help unclutter interfaces by showing only what the user needs, not everything that's available.



[*Lighthouse*](#) features a familiar tabbed navigation menu; however, it also has a second level of menus right under the set of tabs. This level displays only the items associated with the active section of the website.

9. More emphasis on key functions

Not all controls hold the same importance. For example, on a screen for creating a new item, you may have two buttons: “Create” and “Cancel.” The “Create” link is more important because that’s what the user will be doing most of the time. Only rarely will they need to cancel the screen. So if these controls are located side by side, you may not want to give both the same emphasis.



The “Create ticket” button in [Lighthouse](#). You can see the “cancel” link next to it, in plain text. The button not only commands more importance but also has a larger clickable area and is easier to spot because of its frame.

To shift emphasis to the “Create” link, we can simply use different styles or types of controls. Some applications use the form input button for the create action, and have the cancel action as a text anchor. This not only gives the create button more clickable area, it also helps to grab the users gaze better when they’re looking for it.

10. Embedded video

While pictures and text are a great way to communicate and teach your users about your app’s features, video can be an even better alternative if you have the resources to produce it. Video has been gaining popularity on the Web in recent years. For Web apps, videos are generally used on the marketing website as a kind of screencast to show off a product’s features; however, this isn’t the only way to use video.

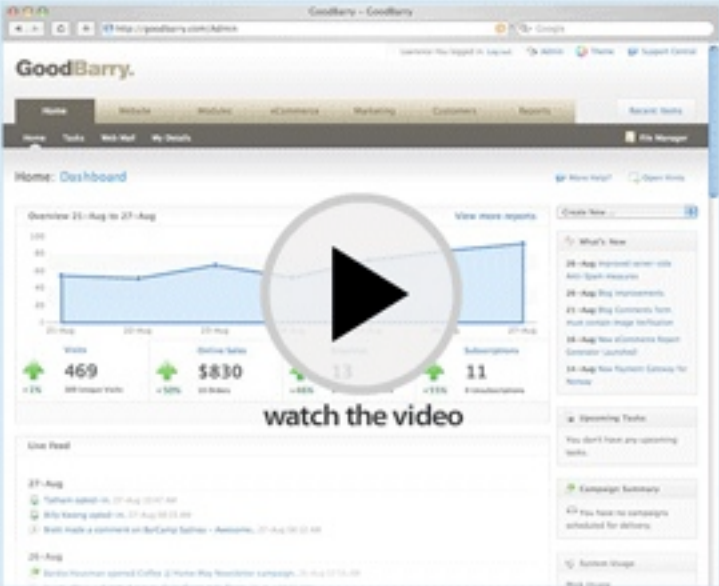
GoodBarry. We're hiring! We've made some significant shipping improvements during the holidays. We've received

Home Overview Case Studies Pricing & Plans Resellers **Free Trial** Support Blog About Contact Jobs

Hi I'm Barry.

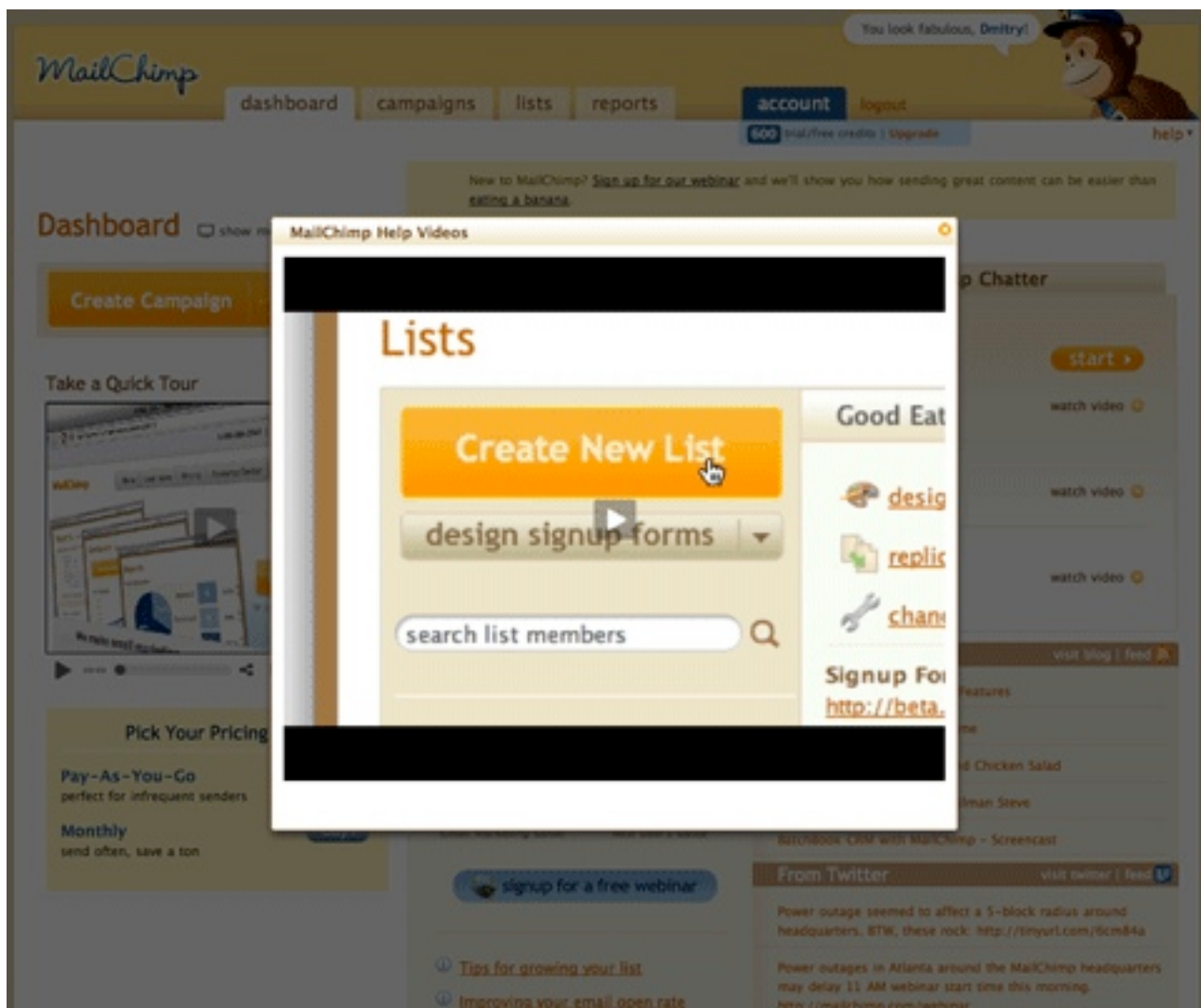
I'm helping 1000s of business owners save time and grow their online business. How?

They're using GoodBarry - my integrated system that runs their website, email marketing and online shop while automatically growing their customer database. My customers have one central console with everything they need to run their business. At only \$39/month for the entire package and no contracts, shouldn't you join them?



The screenshot shows a web browser displaying the GoodBarry dashboard. The dashboard includes a line chart for 'Overview 25-Aug to 27-Aug' with data points for 25-Aug, 26-Aug, and 27-Aug. Below the chart are four key metrics: Visits (469, +2%), Online Sales (\$830, +50%), Subscriptions (11, +55%), and a fourth metric (11, +55%). The right sidebar contains a 'What's New' section with a list of updates and a 'Campaign Summary' section.

[GoodBarry](#) features a video screencast on its front page showing off the product. It also uses screencasts inside the app to teach people on how to get started.



[MailChimp](#) includes tutorial videos right on the admin panel to help out new users.

Some Web apps use video inside the application itself to teach users how to use certain features. Video is a fantastic way to quickly demonstrate how your product can be used, because it is easier to consume than a page of text, and it is also much clearer because the viewer can see exactly what to do.

12 Useful Techniques For Good User Interface Design

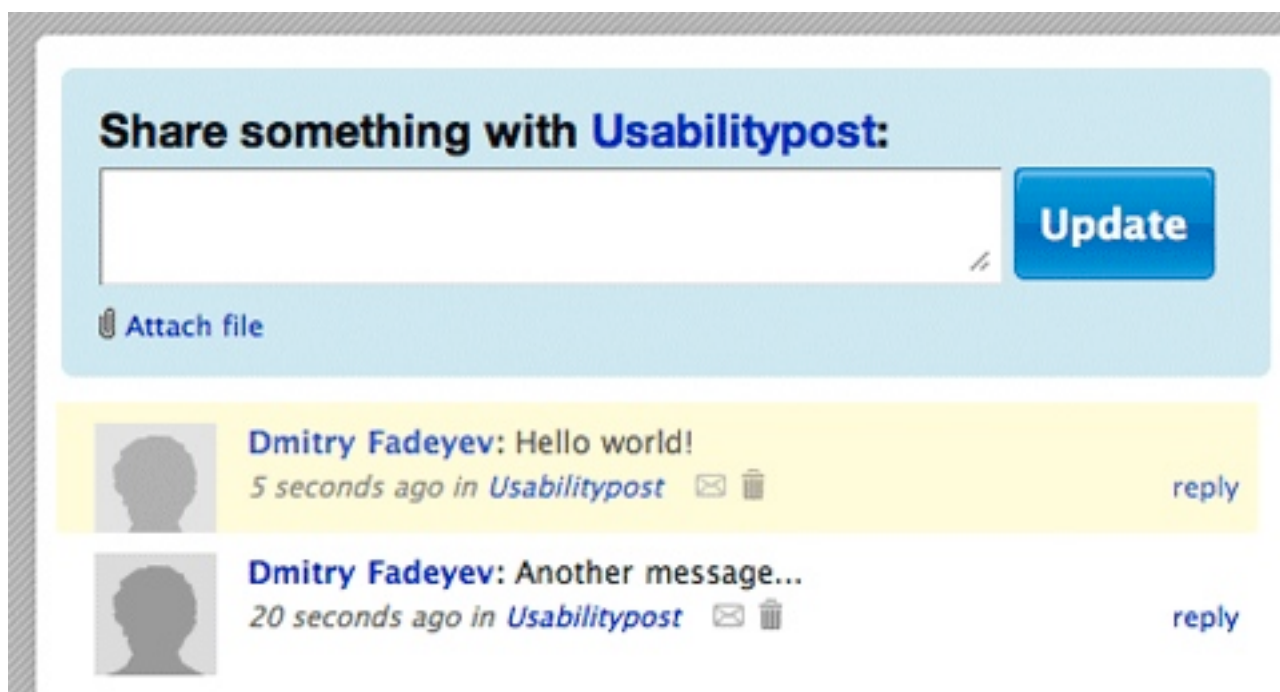
Dmitry Fadeyev

We presented 10 Useful Web Application Interface Techniques, the first part of our review of useful design trends in modern Web applications. Among other things, we highlighted embedded video blocks, specialized controls and context-sensitive navigation. We also encouraged designers to disable pressed buttons, use shadows around modal windows and link to the sign-up page from the log-in page.

This post presents the second part of our review: 12 useful techniques for good user interface design in Web apps. We also discuss how to implement these techniques so that they are properly used. Please feel free to suggest further ideas, approaches and coding solutions in the comments below.

1. Highlight important changes

One of the most significant elements of a good user interface is visibility of the system's status. Users must notice immediately what's going on behind the scenes and whether their actions have actually led to the expected results. To achieve a more sophisticated level of system visibility, Web applications these days use AJAX (of course), which allows users to update portions of a Web page at any time without having to refresh the whole page. AJAX brings the level of responsiveness and interactivity of Web apps much closer to desktop-grade applications.



[Yammer](#) applies not one but three effects on all new messages in a feed: fade in, slide down and highlight.

However, this dynamic nature means that when you click on a button, the page doesn't refresh but something *does* happen. The majority of websites still don't use AJAX extensively, so some users may not be sure whether anything has happened at all or whether the button was properly clicked. To fix this, you need to provide some visual feedback for each of the user's interactions.

A list

☐ Lorem

☐ ipsum

☐ dolor

Add this item or [Close](#)

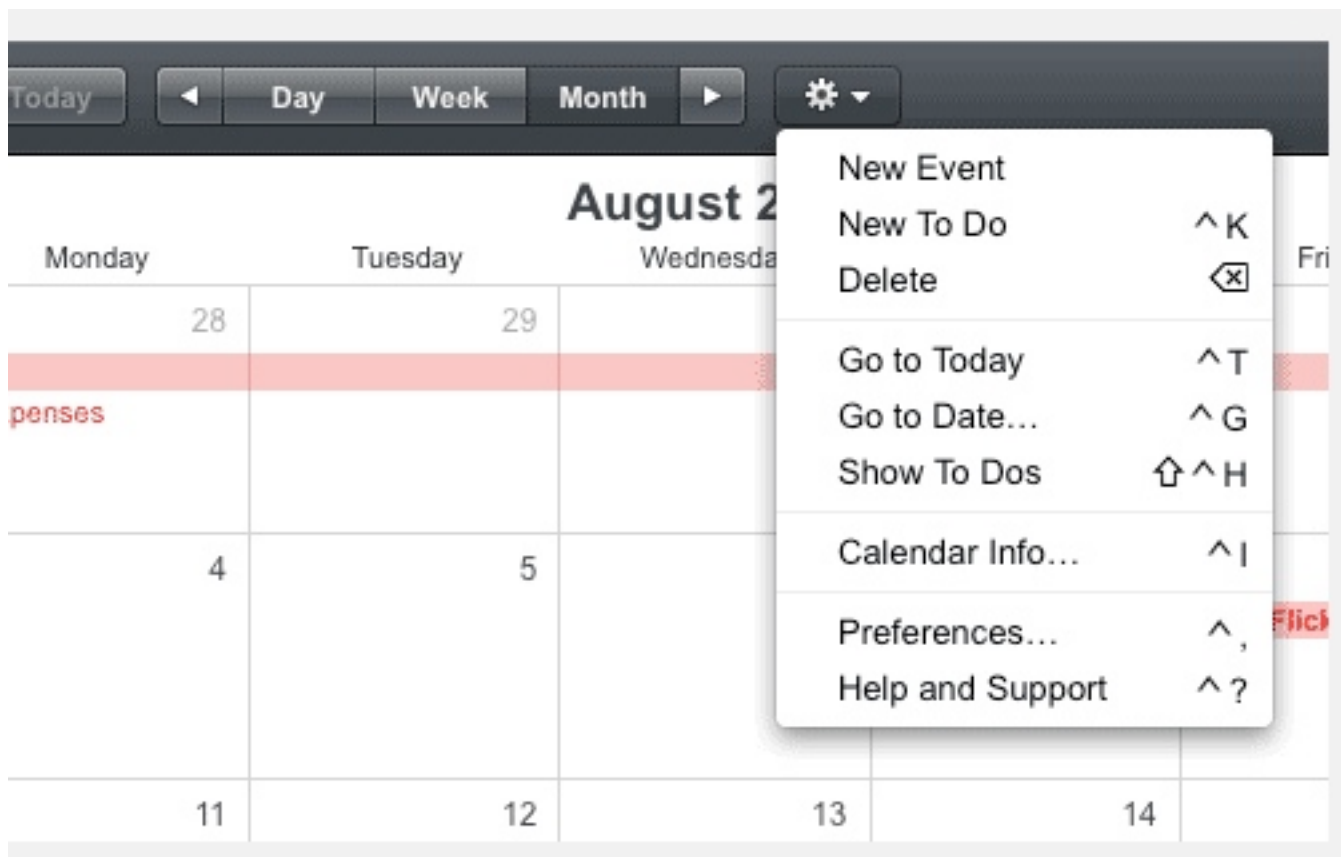
[Backpack](#) applies a highlight effect to all new items in a task list, which lasts for a second before fading out.

One great way to do this is with animation. The human eye can notice movement fairly well, especially if the rest of the page is static. Playing a highlight animation when users add items to their shopping carts, for instance, will attract their eyes to those items. They'll see that their action has worked. Animations can be implemented with JavaScript and are a nice way to provide visual feedback. Just be sure to not overdo it; adding too many animations could cause interface friction because the speed with which the user performs each action will be slowed down by the duration of the animation.

2. Enable keyboard shortcuts in your Web application

As the advanced features of modern Web applications (such as dragging and dropping, modal windows, etc.) steadily gain on those of desktop apps, developers of these applications are trying to offer users more responsive and interactive user interfaces. One of the techniques used to achieve this is the integration of keyboard shortcuts or navigation. Just as with classic

applications, this little feature can significantly improve the workflow of your users and make it easier for them to get their tasks done.



In fact, various Web applications already use shortcuts (for example, [Google Spreadsheets](#), [MobileMe](#), etc.), and even regular websites, such as [Ffffound](#) or [Boston.com](#), allow visitors to use them for basic navigation. Ffffound enables users to use shortcuts to switch from the thumbnail view to list view and vice versa (using “v”), go back to the top (“h”) and navigate to the previous (“k”) and next (“j”) image. Boston.com also uses “k” and “j” for the same functions.

It’s worth mentioning that your shortcuts should be intuitive and self-explanatory. For instance, it wouldn’t make sense to make the shortcut letters for “Previous” and “Next” navigation too far apart from each other on the keyboard; rather, pick ones that are close together. The reason is, if a

user makes a mistake and jumps to a page she doesn't want to visit, she can immediately return to her page without looking at the keyboard. The "j/k" configuration is one option. It would actually make perfect sense to have some common conventions for keyboard shortcuts used throughout various websites, but we haven't been able to detect such conventions thus far.

How do you implement this? Essentially, you just have to use the `onKeyPress-DOM` event and manipulate the appearance of the document using the `window.scrollTo` function in JavaScript.

Ffffound.com uses an `onMouseDown` effect in its markup, which is not a good solution because it doesn't adhere to the [main guideline of unobtrusive JavaScript coding](#): Thou shalt separate JavaScript functionality from CSS style sheet and (X)HTML markup.

```
<a id="float-navi-prev" href="javascript:void(0);"
  onmousedown="try { move_asset_auto(-1) ;} catch (e) { }"
  onclick="return false;">prev(<span class="shortcut">k</
span>)
</a>
```

Here it would make more sense to use a JavaScript library instead (like jQuery) and call the element via its identifier or class. That's (almost) what [Boston.com](#) does. All of the images in its gallery are labelled with the class `bpImage` in the markup, with the JavaScript pointers to them added to the array. The `onKeyPress` event triggers the scrolling window in the background, and the window's browser is manipulated using the `window.scrollTo()` function.

Here is the (X)HTML:

```
...


<div class="bpCaption">...</div>


<div class="bpCaption">...</div>


<div class="bpCaption">...</div>

...
```

And the JavaScript:

```
function bpload(){

    // put pointers to all images with the class "bpImage" in
    an array
    imgArr = getElementsByClassName(document.body, "bpImage")
    isLoading = 1;
}

document.onkeypress = function(e) {
    if (!e) e = window.event;

    // Pick up what key was pressed
    key = e.keyCode ? e.keyCode : e.which;

    // 107 is the ASCII code for 'k'
    if(( key == 107 ) && ( isLoading ) ) {

        // if there are images...
        if ( currImg > 0 ) {
```

```

        // decrease the counter for the current image
        currImg--;

        // offsetTop returns the vertical coordinate of
the
        // upper-left corner of the image
        // (we are not sure why the script adds 174px.
Any idea?)
        window.scrollTo(0,imgArr[currImg].offsetTop+174)

    }
    else {
        if (currImg==0) {
            currImg--;
            window.scrollTo(0,325)
        }
        else
        {
            if (currImg<0) {
                window.scrollTo(0,325)
            }
        }
    }

}

if ( ( key == 106 ) && ( isLoaded ) ) {
    // a similar code snippet for 'j'
    ...
}

}

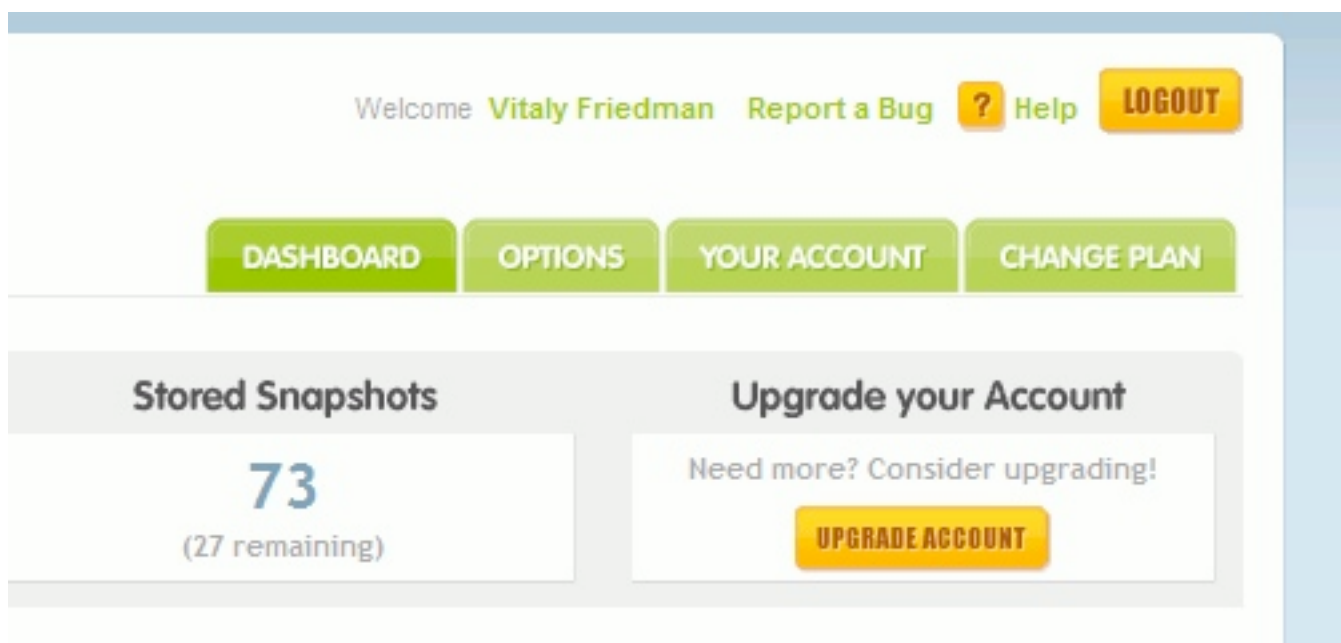
```

But you need to make sure that you clearly communicate that 1) keyboard shortcuts are available, and 2) they can be used to perform certain tasks more efficiently. If a user can easily manage his tasks with your application,

he is less likely to switch to another application, if the feature set is more or less similar.

3. Upgrade options from the account page

If your application features several subscription plans, make sure to remove any interface friction for customers deciding to upgrade. Most users like to try the basic version of an application first to get a better sense of what it offers and how it works. If they are convinced the application meets their expectations, they will consider upgrading to a more advanced plan. It's the designer's task to make sure this transition is as simple and intuitive as possible.



[CrazyEgg](#) integrates the option “Change plan” in its main navigation.

In fact, a lot of Web applications put upgrade options right on the user's account page, making them easily accessible. This design choice has the simple advantage of providing users with an overview of available options and supported functionalities right away.

Plan

Limits	Gold	Platinum	Diamond
Product limit	5	25	100
Images per product	1	3	5
Custom pages	1	3	10
Features			
Custom CSS	•	•	•
Custom HTML		•	•
Inventory tracking		•	•
Use your own domain		•	•
Discount codes		•	•
More navigation & search		•	•
Statistics			
Product views	•	•	•
Use 3rd-party stats		•	•
Visitor tracking		•	•
Top referrers			•
Top search terms			•
Prices	Gold	Platinum	Diamond
Price	Free	\$9.99 monthly	\$19.99 monthly
	Current	Upgrade	Upgrade

[Bigcartel](#)'s upgrade plans are available in the app itself.

Note, though, the importance not only of featuring the available upgrade plans, but also of identifying the plan that the user is *currently* using and the features that are *currently* available to her. It is vital to provide users with precise information about what advantages they gain by upgrading their account. Take a look at our article [Pricing Tables: Examples and Best Practices](#) as well.

4. Advertise features of the application

Even though you've created a detailed marketing page, outlining your application's every feature, and crafted a thorough help section on your website, your users are unlikely to have read it all. They're probably not familiar with all the features of your product and would benefit from little tips inside the application itself.

Advertise new features in your application. These would usually go in the sidebar, out of the way of the main functions. If a user is nearing the maximum capacity of a certain feature for her chosen subscription plan, you should point this out and give her an option to quickly upgrade.

People



[Dmitry Fadeyev](#) that's you! ★owner admin
dmitry@usabilitypost.com

edit person

■ ■ ■ this month

You're out of user accounts. Upgrade instantly to the **Basic** plan for 4 more user

The [Freckle](#) time-tracking app tells you when you've run out of people on your current plan. The message also links to actions you can take if you need to upgrade.



NEW! Try one of our CSS Themes from the [Wufoo Form Gallery!](#)

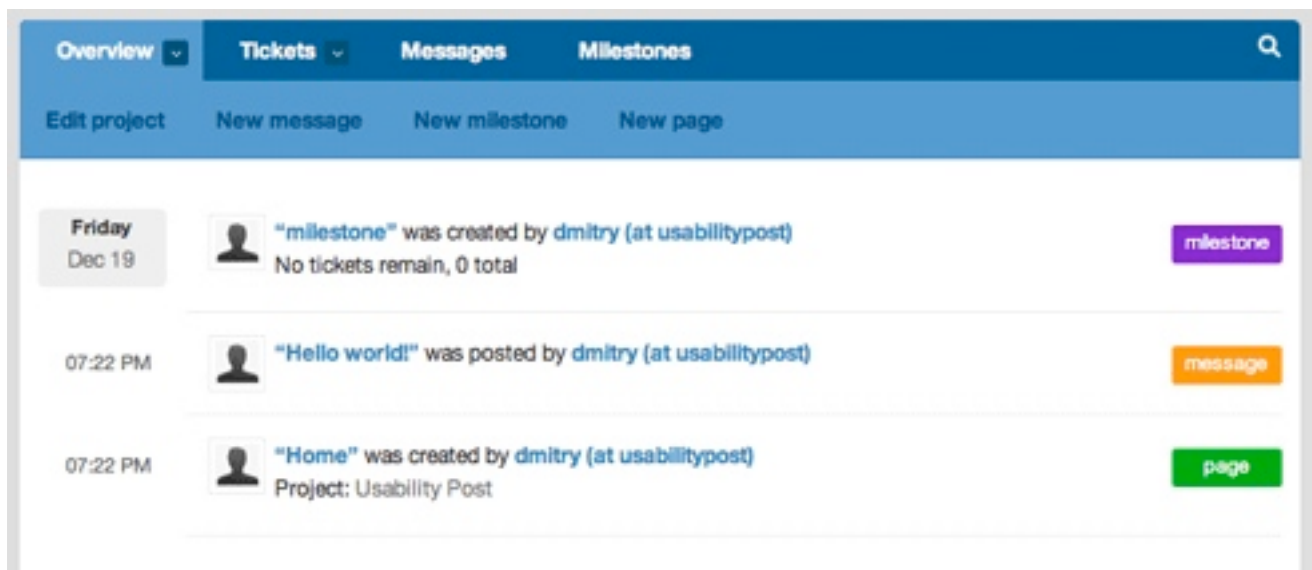
Wufoo highlights its new form gallery feature at the bottom of the form creation page, making sure customers get the most value out of the app.

5. Use color-coded lists

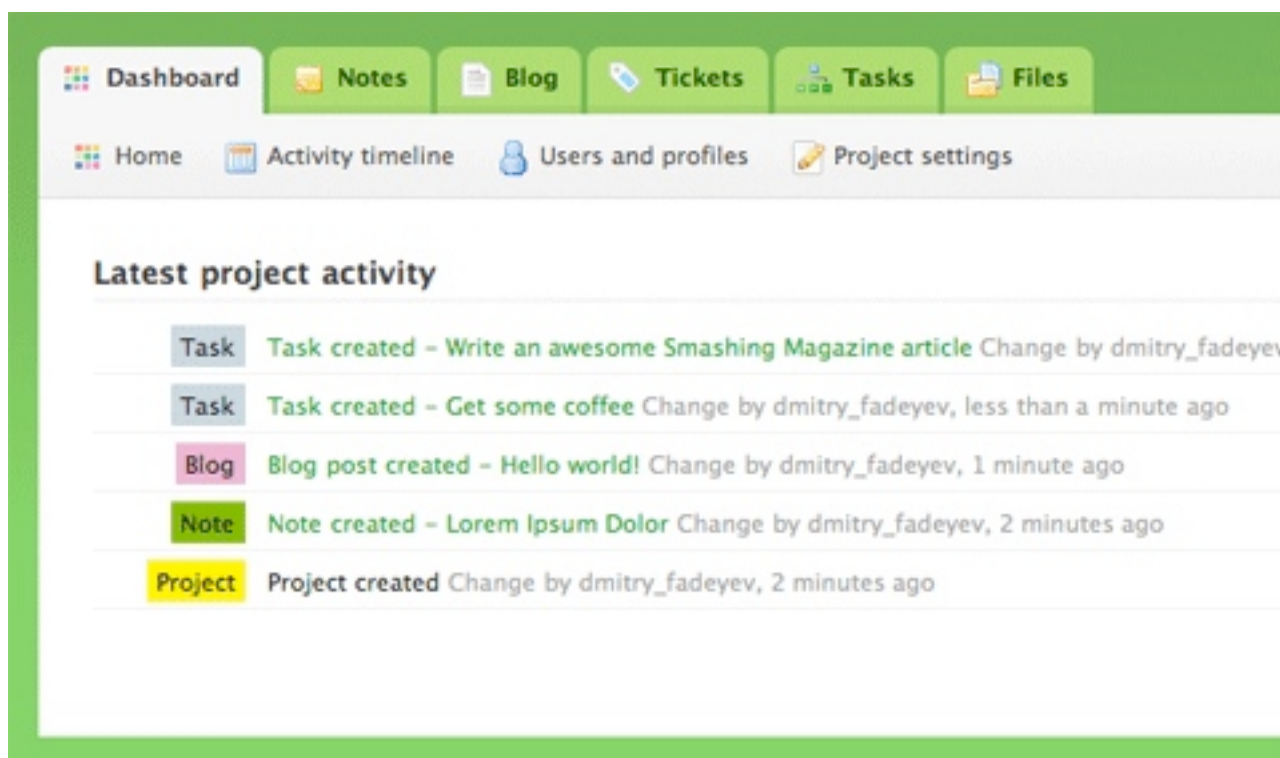
Some applications feature feeds that aggregate various types of content. For example, you may have a project management application that shows you all the latest messages, tasks and files on the home page. If these items all appear together in one list, it may be difficult to tell what's what. Many applications use color coding to help visually distinguish between different

types of entries. A simple way to do this is to place a text label inside a colored box. This way, the list becomes easily scannable.

It's important not to use various colors for the same task or similar colors for completely different tasks. The color scheme should not be random but should implicitly indicate the function each item serves.



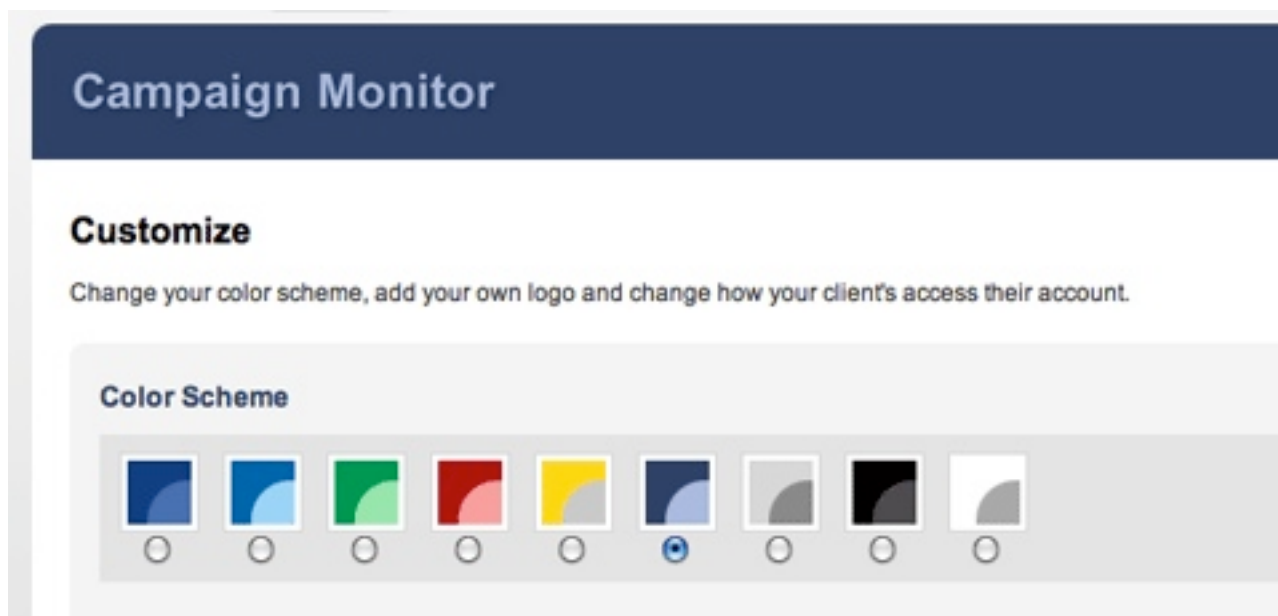
The [Lighthouse](#) issue-tracking app has color-coded labels on the right-hand side of each item on the overview page, which helps you quickly scan the list.



The Goplan dashboard uses similar color-coded labels to differentiate various items, like tasks, notes and files, so you can quickly find what you need.

6. Offer personalization options

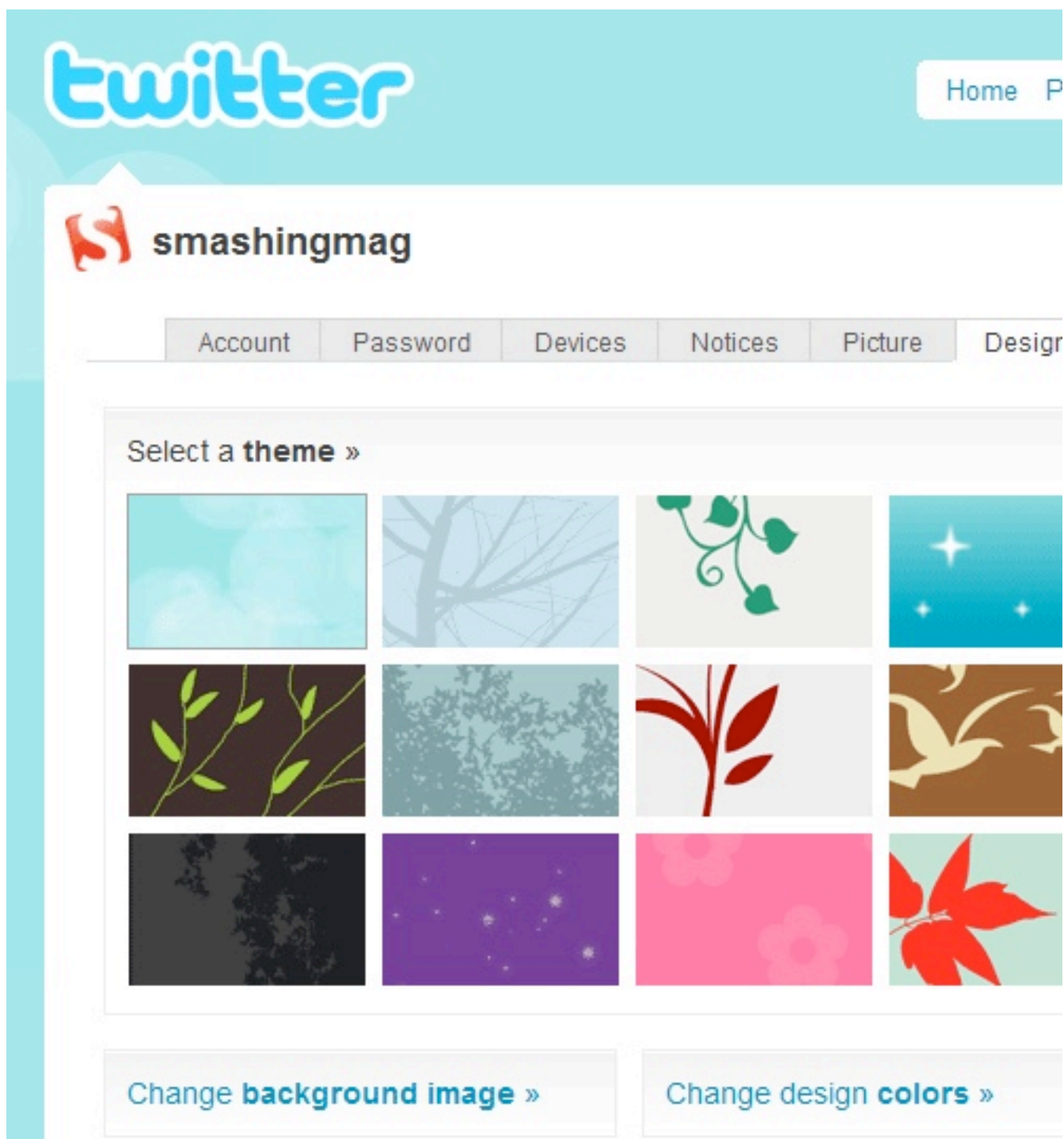
Many applications provide custom workspaces for people and businesses. Personalization can help make your users feel more at home. This can be done by giving users options to customize the look and feel of the application interface. Let them select the color theme, the link colors, the background and so on. Even a small amount of customization will allow your users to make their pages their own.



Campaign Monitor lets you choose a color theme for your account and upload your business logo. This helps businesses infuse the colors of their brands into the Web apps they use.

Personalization is certainly one of the simplest and most effective methods of binding your customers to your service, but it's important to understand that the various personalization options should never come at the expense of the core application's functionality. The system should always be capable of performing its functions and thus meet users' expectations, despite how exactly users have personalized the application.

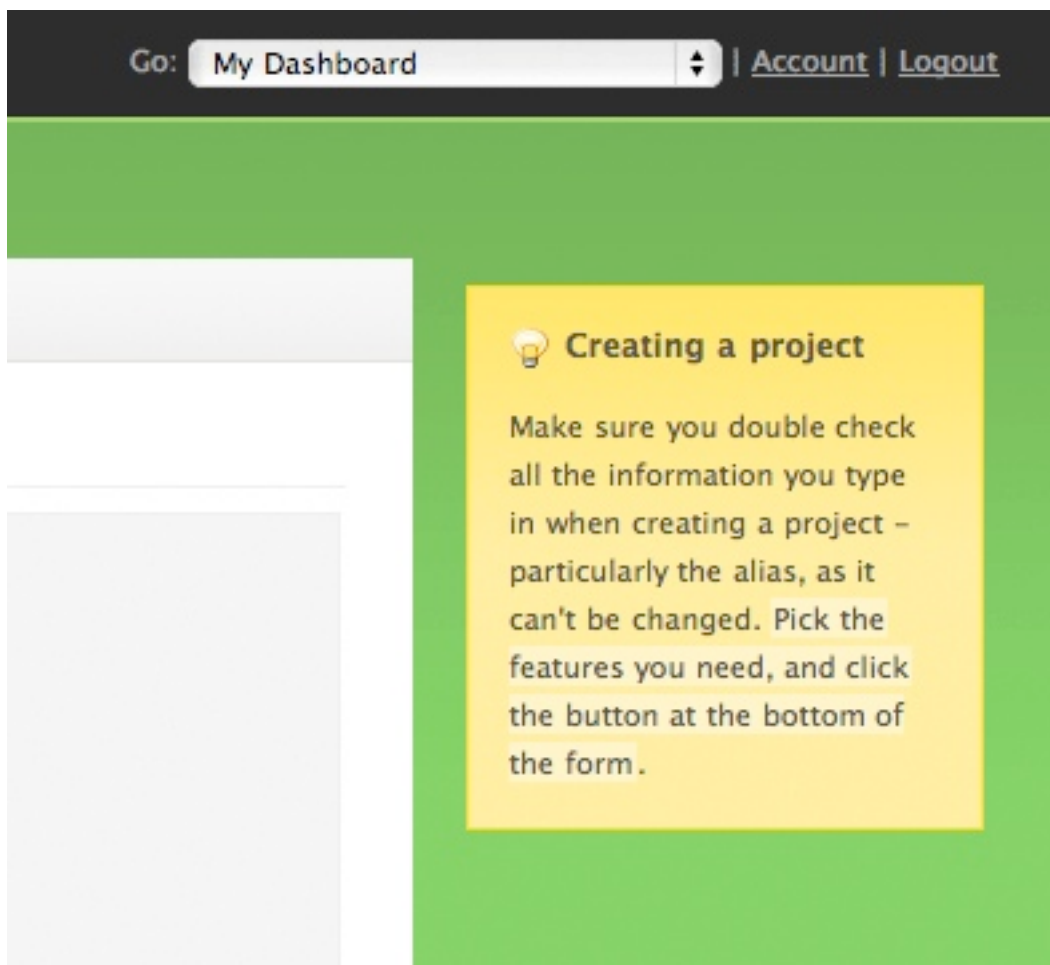
One useful approach to finding a compromise between a website's core functionality and the user interface is to introduce various levels of personalization, depending on whether the user is a novice or an advanced user. It is also a good idea to allow the user to revert his account to the default settings or restore the settings that he had saved in his previous session.



[Twitter](#) lets the user customize his profile page background and colors, allowing him to craft a unique spot on the popular micro-blogging network.

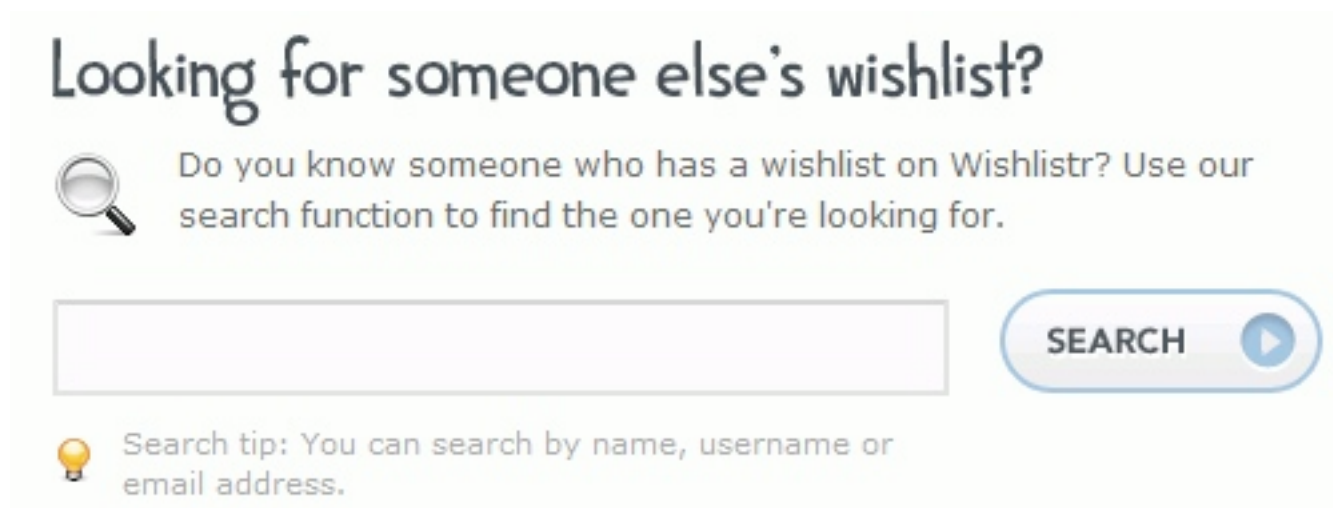
7. Display help messages that attract the eye

Every Web application is different and has its own way of doing things. If the function of a particular element isn't immediately apparent, you can provide short help messages to get people started. One important thing to note is that if you want to help people who aren't sure what they're doing yet, you need to attract their attention to this message. One way to attract attention is with color — putting a yellow “sticky” message in the sidebar, for example, is sure to stand out.



Goplan puts help messages in bright yellow boxes resembling paper stickies. The bright color ensures that users don't miss them.

Alternatively, if you are looking for a subtler solution that doesn't require much space to display and isn't obtrusive for regular users of your application, you can consider displaying vibrant visual graphics (for example, small icons) next to the design element needing explanation. For instance, pointing users to new, updated or useful features of an application's search engine, as [Wishlitr.com](https://www.wishlitr.com) does, makes sense.



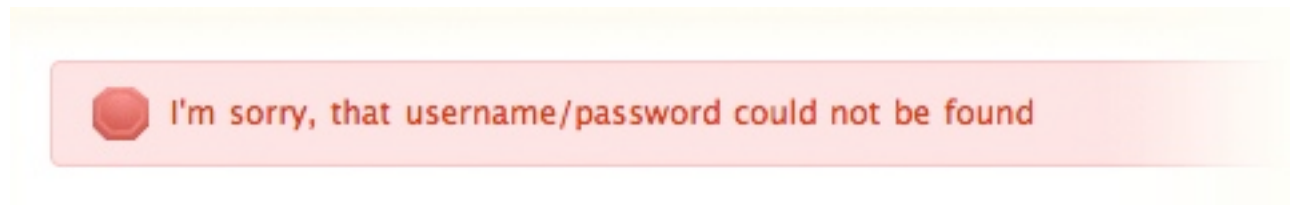
Classic: Wishlitr uses a light bulb to focus the user's attention on the available search functionality of its system.

8. Design feedback messages carefully

Pretty much every application has some form of feedback messages. These are little messages that pop out when there is an error or warning or perhaps when an action is completed successfully. Designing these messages correctly is important because you don't want to confuse or startle your users when there's nothing to worry about.

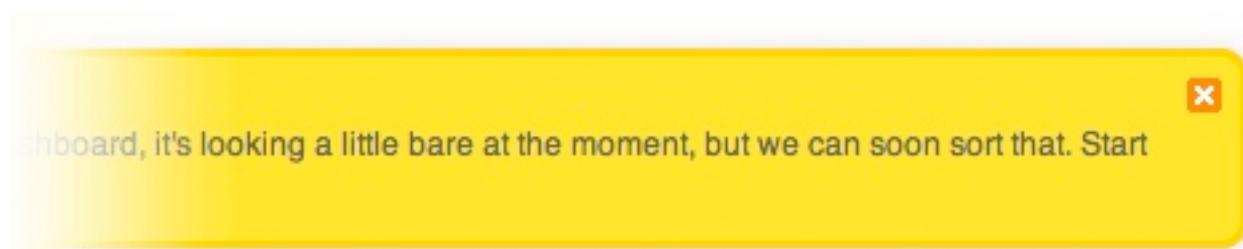
A good practice here is to do a couple of things. First, color code the different types of messages. Messages that notify users of successful actions are usually colored green. These employ the traffic-light analogy of green meaning "Go." Warning and error messages are colored yellow. Same

traffic-light analogy here: yellow means slow down and wait. You can also distinguish between warning messages and error messages by coloring errors red and warnings yellow.



Mailchimp uses color effectively for its error messages.

The second thing to do is add a unique icon for each message type. Icons can convey meaning instantly, without the user having to read the message. For example, a tick icon can symbolize completion of a successful action. An exclamation mark in a triangle is a warning sign. People will instantly recognize that this message warns them about something and will pay attention.

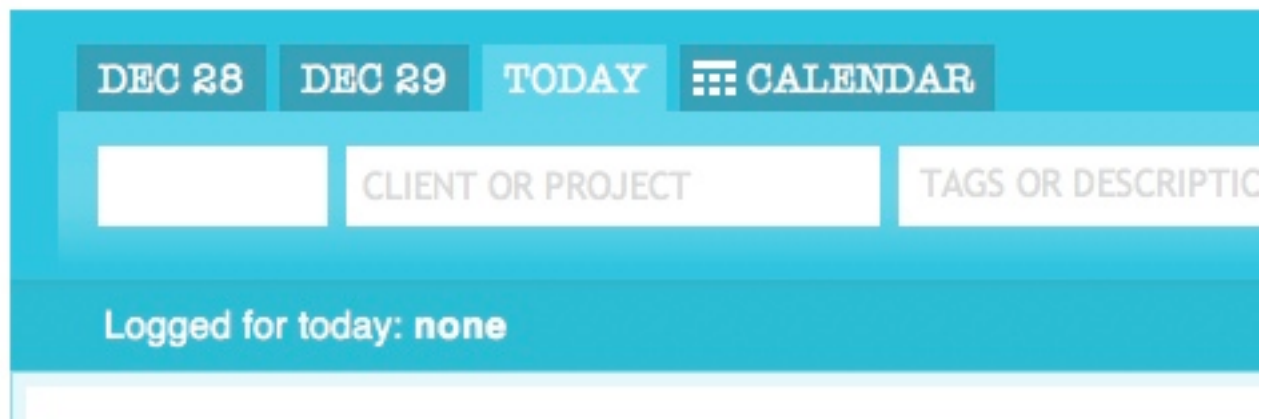


GetSignOff allows you to close notifications by using the little button in the top-right corner of the message box.

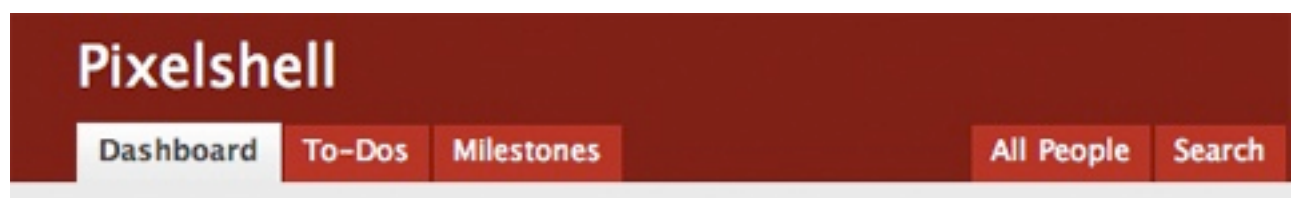
Lastly, you should provide a way for users to close the notification if they are likely to remain on the page for a while.

9. Use tabbed navigation

Many Web applications have adopted the tabbed navigation approach for their main navigation menu. Tabbed navigation is a menu that looks like each item is a tab on a file folder, with the active tab connected to the body of the page. Tabbed navigation isn't just eye candy; it provides a usability benefit.



Freckle uses tabbed navigation in a sub-menu relating to the time input menu.

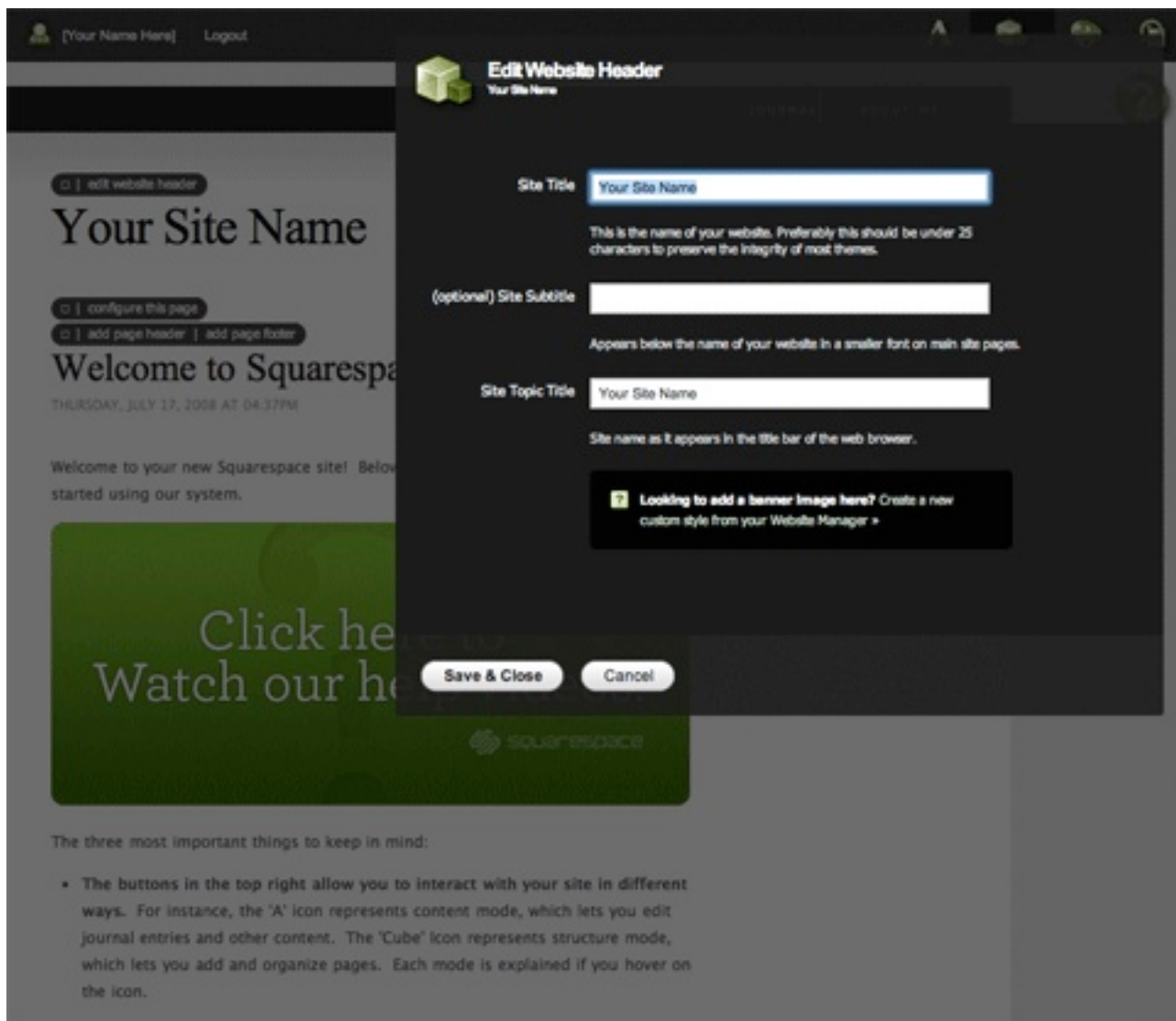


Basecamp features the standard tabbed menu for the main navigation of its app.

If you make the menu look like tabs on folders, almost everyone will be able to figure out what it is and how it works. This is because the visual metaphor is strong and clear. The current page or section also becomes easy to see. Knowing where they are puts users at ease because they gain a greater sense of control.

10. Darken background under modal windows

In some applications, you may want to display a bit of information or quick input form that doesn't really deserve a full page of its own. Some developers put that message or form in a modal window. Modal windows are little windows that pop up on top of the current page and that users need to interact with to proceed.

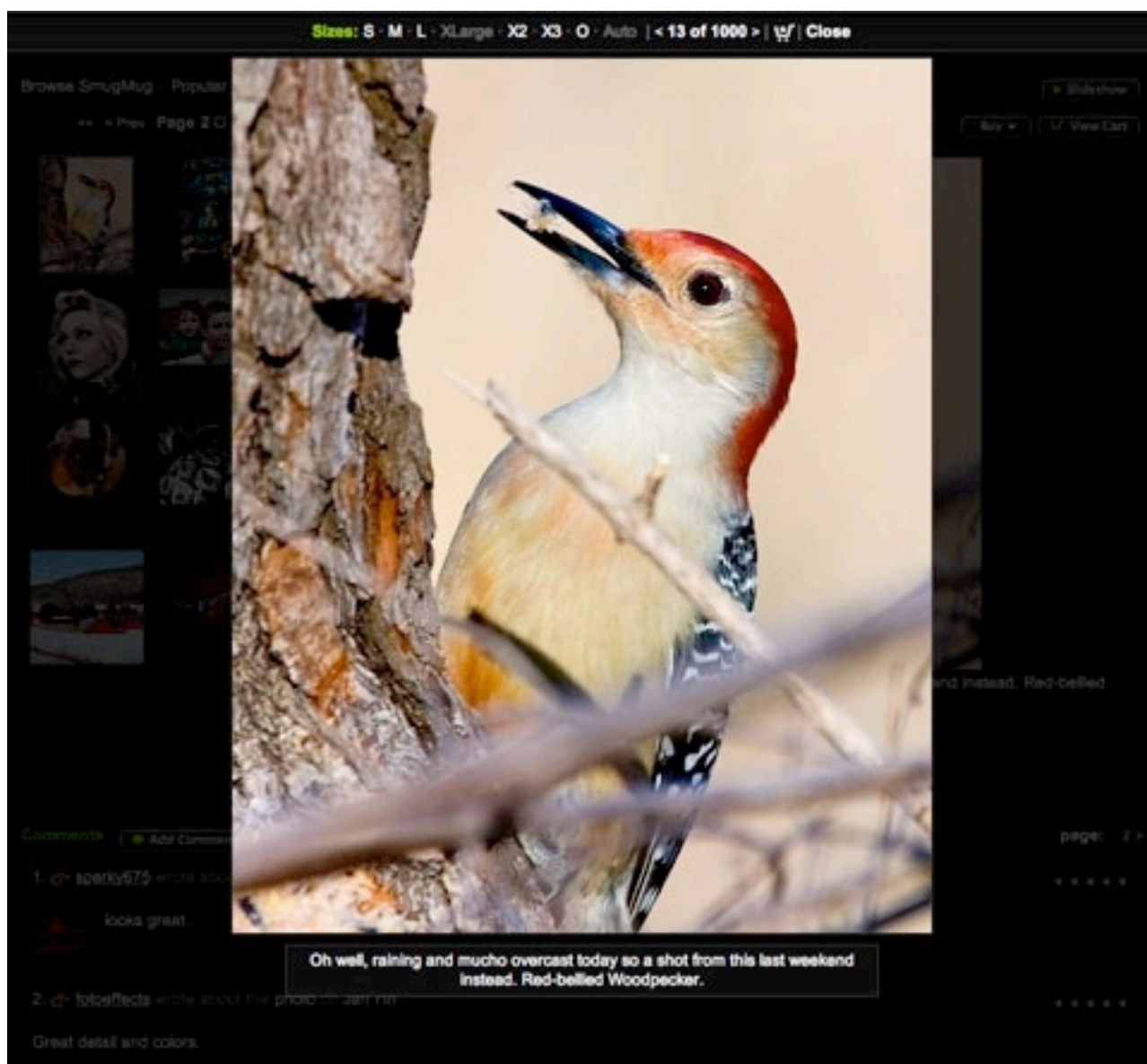


When editing things in the [Squarespace](#) website creation app, the background darkens to shift focus to the edit window.

To make this window stand out better, you can darken the content below it. The darker background will block out all the noise of the content behind the box and make the modal window the center of attention. This is very similar to using shadows around the window but is even more powerful in directing focus. The darker background also indicates that interaction with the content beneath is disabled and that the user should instead interact with the modal window.

11. Lightboxes and Slideshows

Some applications include a lot of images that users may want to browse. Displaying every image on its own page may not be the most efficient way to do it — both for your visitors and your server. Your visitors will need to navigate back and forth, and your server will incur extra hits that can be avoided.



[SmugMug](#) uses lightboxes to enlarge photos. (Photo by Kurt Preston.)

Enter lightboxes and slideshows. Lightboxes and slideshows are used to display photos without having to load a new page. For example, the lightbox method will enlarge an image and place it as a modal window above the rest of the page, allowing the user to focus on the image itself while the background is darkened. This means less noise interfering with the viewing experience.

12. Short sign-up forms

The sign-up form is potentially one of the biggest barriers between you and potential customers. The longer the form, the more effort your visitors will have to make before becoming members of your website or, perhaps, paying customers. To minimize the barrier, we've got to speed up the process. This means removing all optional elements from the form and leaving only the core essentials. The optional stuff can be filled in later.

Register for Evernote

Full name	<input type="text"/>
Email address*	<input type="text"/>
Username*	<input type="text"/>
Password*	<input type="password"/>
Confirm password*	<input type="password"/>

☐ You accept our [Terms of Service](#) and confirm that you are at least 13 years old.

Lieut

RACING

Type the two words:

↺

🔊

?

reCAPTCHA™

stop spam.
read books.

Register

Opening an [Evernote](#) account is easy, with only a handful of fields to fill in, all of which are grouped together in a compact box.

Conclusion

Making your application beautiful may lead to a satisfying user experience, but it will not guarantee a *usable product*. For example, an ugly website like Craigslist performs its function fairly well. Its poor aesthetic did not stop it from becoming hugely successful. Similarly, the minimalist interface of the

Google search engine manages to fully accomplish its objectives without getting in your way. The interface disappears, letting you focus on getting things done.

Steve Jobs once said, “design is not just what it looks like and feels like. Design is how it works.” In fact, the usability and overall usefulness of a Web app is governed by how well it performs its functions and how easily those functions are accessed. Design with a goal in mind — a goal that the interface helps your users achieve. Not every technique will work in every situation or for every application. Only implement interface elements if they make sense in your particular context.

An Extensive Guide To Web Form Usability

Justin Mifsud

Contrary to what you may read, peppering your form with nice buttons, color and typography and plenty of jQuery plugins will not make it usable. Indeed, in doing so, you would be addressing (in an unstructured way) only one third of what constitutes form usability.

In this article, we'll provide practical guidelines that you can easily follow. These guidelines have been crafted from usability testing, field testing, website tracking, eye tracking, Web analytics and actual complaints made to customer support personnel by disgruntled users.

Why Web Form Usability Is Important

The ISO 9241 standard defines website usability as the “effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.” When using a website, users have a particular goal. If designed well, the website will meet that goal and align it with the goals of the organization behind the website. Standing between the user's goal and the organization's goals is very often a form, because, despite the advances in human-computer interaction, forms remain the predominant form of interaction for users on the Web. In fact, forms are often considered to be the last and most important stage of the journey to the completion of goals.

Let's clarify this last point by discussing the three main uses of forms. As Luke Wroblewski states in his book [Web Form Design: Filling in the Blanks](#), every form exists for one of three main reasons: commerce, community or productivity. The following table translates each of these reasons into the user and business objectives that lie behind them:

	Commerce	Community	Productivity
User Objective	Obtain Info / Buy	Join Community	Get Things Done
Business Objective	Maximize Sales	Grow & Increase Engagement in the Community	Increase Content & Time Spent on Site
Example of Type of Site	E-Commerce	Social Networks	Online Banking
Example of a Form	Checkout Form	Registration Forms	Transfer of Funds

Uses of forms, based on Luke Wroblewski's [Web Form Design: Filling in the Blanks](#).

Thus, the relationship between forms and usability have two aspects:

1. **Forms can make a website usable** or unusable, because they stand in the way of the user achieving their goal;
2. **Forms need to be usable** in order to help the user achieve that goal.

This post will focus on the second point, because a usable form will naturally contribute to the overall usability of the website, hence the first aspect.

The Six Components Of Web Forms

Web forms are a necessity and often a pain point for both designers and users. Over time, users have formed expectations of how a form should look and behave. They typically expect Web forms to have the following six components:

1. **Labels** These tell users what the corresponding input fields mean.
2. **Input Fields** Input fields enable users to provide feedback. They include text fields, password fields, check boxes, radio buttons, sliders and more.
3. **Actions** These are links or buttons that, when pressed by the user, perform an action, such as submitting the form.
4. **Help** This provides assistance on how to fill out the form.
5. **Messages** Messages give feedback to the user based on their input. They can be positive (such as indicating that the form was submitted successfully) or negative (“The user name you have selected is already taken”).

6. **Validation** These measures ensure that the data submitted by the user conforms to acceptable parameters.

5. Message → Please review information that you have entered.

1. Label → First name* Last name*

2. Input Field → Justin Mifsud

Your email address* Repeat email*

badEmail

6. Validation → Email address not valid

Note: no-one can see your email address.

☒ By email

saved *chocolate*

Can't read the text in the box?

Refresh Listen Help

4. Help

Type the text above here*

Yes, I have read and I accept the [Skype Terms of Use](#) and the [Skype Privacy Statement](#)

I agree - Continue

3. Action

[Skype](#)'s registration form contains all six components.

Tackling Usability Via Three Aspects Of Forms

Despite differences in layout, functionality and purpose, all forms have three main aspects, as noted by Caroline Jarrett and Gerry Gaffney in their book [*Forms That Work: Designing Web Forms for Usability*](#):

1. **Relationship** Forms establish a relationship between the user and the organization.
2. **Conversation** They establish a dialogue between the user and the organization.
3. **Appearance** By the way they look, they establish a relationship and a conversation.

For a form to be usable, all three aspects need to be tackled. Let's look at each aspect in turn to figure out how to make a form truly usable, along with practical guidelines that you can easily follow.

Aspect 1: The Relationship

“No man is an island,” the 17th-century English poet, satirist, lawyer and priest [John Donne](#) once said. Indeed, human beings thrive on relationships, be they amorous, friendly, professional or business. A form is a means to establish or enhance a relationship between the user and the organization. When done badly, it can pre-empt or terminate such a relationship.

With this in mind, a number of principles emerge:

- Relationships are based on trust, so establishing trust in your form is critical. This can be achieved through the logo, imagery, color, typography and wording. The user will feel at ease knowing that the form comes from a sincere organization.

- Every relationship has a goal, be it love and happiness in a romantic relationship or financial gain in a business relationship. Ask yourself, what is the goal of your form?
- Base the name of the form on its purpose. That name will inform users what the form is about and why they should fill it in.
- Just as in a relationship, getting to know the other person is essential. Get to know your users and always consider whether the questions you're asking are appropriate and, if so, whether they are timely. This will instill a natural flow to your form.
- Knowing your users will also help you choose appropriate language and remove superfluous text. And it will help you craft an interface that balances your needs and the user's.
- Do not ask questions beyond the scope of the form. In a relationship, you would become distrustful of someone who asked questions that were out of place. The same thing happens online. Consult with relevant stakeholders to see what information really is required.
- Sudden changes in behavior or appearance will make users edgy. Likewise, never introduce sudden changes between forms or between steps in a form.

my details

LOG-IN / REGISTER

You don't have to be registered to shop with Debenhams, but if you do you'll find that it makes shopping saved securely in My Details for you to amend at any time.

If you are already registered simply enter your *Log-in ID email address and password in the boxes below. Note that your password is case sensitive.

*Your Log-in ID email is the email address that you registered with.

Log-in (if you're already registered)

Email address:

*

Password:

*

[Forgotten your password?](#)

LOG-IN

Register (for new customers)

Email address:

*

Confirm your email address:

*

Password (minimum 6 characters):

*

Confirm your password:

*

REGISTER ME

Know your users. Make it easy for registered users to log in and for new users to register. [Debenhams](#) makes this distinction barely noticeable.

Sign In

What is your e-mail address?

My e-mail address is:

Do you have an Amazon.com password?

☐ No, I am a new customer.

☒ Yes, I have a password:

[Forgot your password?](#)

Sign in using our secure server



Sign In Help

Forgot your password? [Get password help.](#)

Has your e-mail address changed? [Update it here.](#)

[Amazon](#), on the other hand, simplifies the process for registered and new customers.

Aspect 2: The Conversation

A form is a conversation. And like a conversation, it represents two-way communication between two parties, in this case, the user and the organization. In fact, the user has filled out the form in order to initiate communication with the organization.

For instance, with a social network, a user would fill out a registration form to inform the organization that they would like to join. In inviting their request (whether automatically or manually), the organization would ask the user a number of questions (in the form of labels), such as their first name, last name, email address and so forth. Upon acceptance (or denial), the company would inform the user of the outcome, thus completing the communication process.

Viewing forms from this perspective yields some useful guidelines:

- As mentioned, **a form is a conversation, not an interrogation.** Aggressive wording in labels will make users feel edgy, and (if they do not leave) they will most likely give you the answers that you want to hear, rather than the truth.
- **Order the labels logically**, reflecting the natural flow of a conversation. For example, wouldn't it be weird to ask someone their name only after having asked a number of other questions? More involved questions should come towards the end of the form.
- **Group related information**, such as personal details. The flow from one set of questions to the next will better resemble a conversation.

Select an ID and password

Yahoo! ID and Email

@

yahoo.com

⌵

Check

Password

Password Strength

☐

☐

☐

☐

Re-type Password

In case you forget your ID or password...

Alternate Email (optional)

Secret Question 1

- Select One -

⌵

Your Answer

Secret Question 2

- Select One -

⌵

Your Answer

[Yahoo's](#) registration form effectively groups related content through purple headings and fine lines.

The image shows a user registration form with two distinct sections separated by a horizontal line. The first section, titled "First, which service do you want to try?", contains a sub-header "Why we suggest! trying one product at a time." and three radio button options: "Email Marketing", "Online Survey", and "Event Marketing". Each option is followed by a link "What's included? | Pricing". The second section, titled "Tell us about yourself.", contains several input fields: "First Name:", "Last Name:", "Organization Name:", "Country:" (a dropdown menu showing "United States"), "State:" (a dropdown menu showing "Choose State"), "Phone Number:" (three separate input boxes), "Email:", and "Website:". Below the "Phone Number:" field is the text "So we can get in touch about our free coaching." Below the "Email:" field is the text "We'll send you an email to confirm your trial and [verify](#) your address."

First, which service do you want to try?

Why we suggest! trying one product at a time.

I want to try: ☐ Email Marketing [What's included?](#) | [Pricing](#)

☐ Online Survey [What's included?](#) | [Pricing](#)

☐ Event Marketing [What's included?](#) | [Pricing](#)

Tell us about yourself.

First Name:

Last Name:

Organization Name:

Country:

State:

Phone Number:

So we can get in touch about our free coaching.

Email:

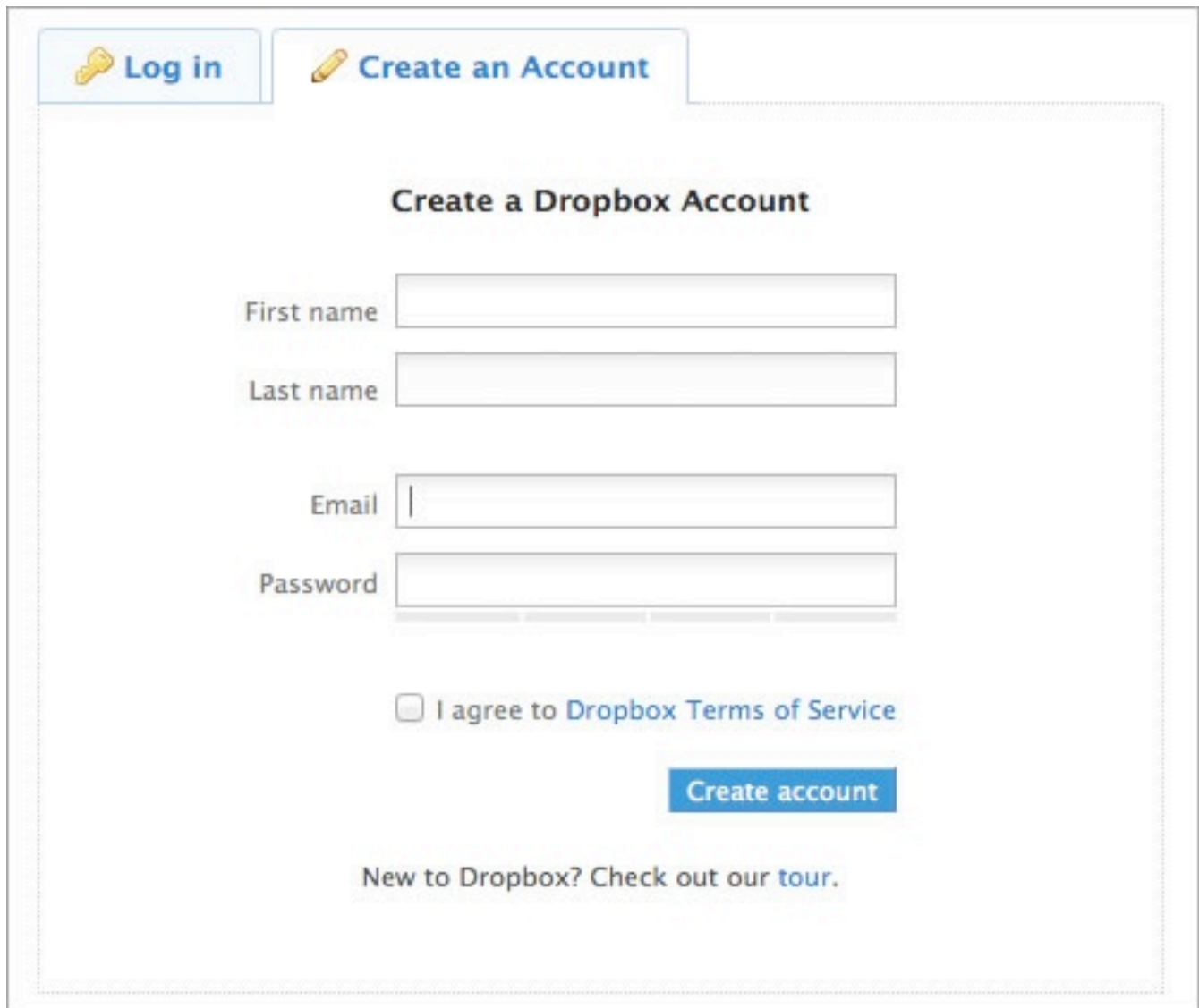
We'll send you an email to confirm your trial and [verify](#) your address.

Website:

While [Constant Contact](#) groups related content, it separates the groups too much, which could confuse the user.

- As in a real conversation, each label should address one topic at a time, helping the user to respond in the corresponding input field.
- The natural pauses in a conversation will indicate where to introduce white space, how to group labels and whether to break the form up over multiple pages.

- In any conversation, people get distracted by background noise. So, remove clutter such as banners and unnecessary navigation that might distract users from filling out the form.

A screenshot of the Dropbox account creation form. At the top, there are two buttons: 'Log in' with a key icon and 'Create an Account' with a pencil icon. Below these is the title 'Create a Dropbox Account'. The form contains four input fields: 'First name', 'Last name', 'Email', and 'Password'. Below the 'Email' field is a checkbox labeled 'I agree to Dropbox Terms of Service'. At the bottom right is a blue 'Create account' button. At the bottom center is a link: 'New to Dropbox? Check out our tour.' The form is clean and uncluttered, with a white background and a thin border.

[Dropbox](#) provides a fine example of what a registration form should look like. The white space is effective, and the page uncluttered.

Aspect 3: The Appearance

The appearance or user interface (UI) is central to the usability of a Web form, and there are several guidelines for this. To simplify the discussion, let's group them into the six components presented earlier.

1. LABELS

- **Individual words vs. sentences** If the purpose of a label is simple to understand, such as to ask for a name or telephone number, then a word or two should suffice. But a phrase or sentence might be necessary to eliminate ambiguity.

Registration

New to Amazon.com? Register Below.

My name is:

My e-mail address is:

Type it again:

My mobile phone number is: (Optional)

[Learn more](#)

Protect your information with a password

This will be your only Amazon.com password.

Enter a new password:

Type it again:

Create account

[Conditions of Use](#) [Privacy Notice](#) © 1996-2011, Amazon.com, Inc. or its affiliates

[Amazon's](#) registration form contains full sentences, whereas individual words would have sufficed.

- **Sentence case vs. title case** Should it be “Name and Surname” or “Name and surname”? Sentence case is slightly easier—and thus faster—to follow grammatically than title case. One thing is for sure: never use all caps, or else the form would look unprofessional and be difficult to scan.

Sign In or Create an Account

To use this feature, you must sign in or create an account.

If you already have an account, please [sign in](#).

EMAIL ADDRESS: *

RE-ENTER EMAIL ADDRESS: *

FIRST NAME: *

LAST NAME: *

PASSWORD: *

CONFIRM PASSWORD: *

(Use 6-15 letters, numbers, or numeric symbols. CaSe SeNsItIvE. No spaces.)






SECURITY QUESTION *

Select



See how difficult it is to quickly scan the labels in [Barnes & Noble's](#) registration form?

- **Colons at the end of labels** UI guidelines for some desktop applications and operating systems such as Windows recommend adding colons at the end of form labels. Some designers of online forms adhere to this, primarily because old screen readers mostly rely on the colon symbol to indicate a label. Modern screen readers rely on mark-up (specifically, the `label` tag). Otherwise, the colon is a matter of preference and neither enhances nor detracts from the form's usability, as long as the style is consistent.
- **Alignment of labels: top vs. left vs. right** Contrary to common advice, above the input field is not always the most usable location for a label. It's ideal if you want users to fill in the form as fast as possible. But there are times when you'll want to deliberately slow them down, so that they notice and read the labels attentively. Also, keeping a long form to a single column and making users scroll down the page is better than breaking it up into columns in an attempt to keep everything "above the fold." Each style of alignment has its advantages and disadvantages:

	Top	Right	Left
Speed of Completion	Fastest		Slowest
Horizontal Space Needed	Least		Most
Vertical Space Needed	Most		Least
Space Available For Label Text	Most		Least
Proximity To Input	Closest		Least Close
User Eye Movement	Down	Down & Right	Down & Right
Time To Move From Label To Input (*)	50ms	240ms	500ms
Ideal For	Simple Forms	Less Simple Forms	Complex Forms

*Times retrieved from “[Label Placement in Forms](#)” by Matteo Penzo.

BRUSHES EYES LIPS FACE PALETTES KITS NEW ITEMS CLEARANCE

Shipping & Policies Login My Account

Create an Account

PERSONAL INFORMATION

FIRST NAME *

LAST NAME *

EMAIL ADDRESS *

LOGIN INFORMATION

PASSWORD *

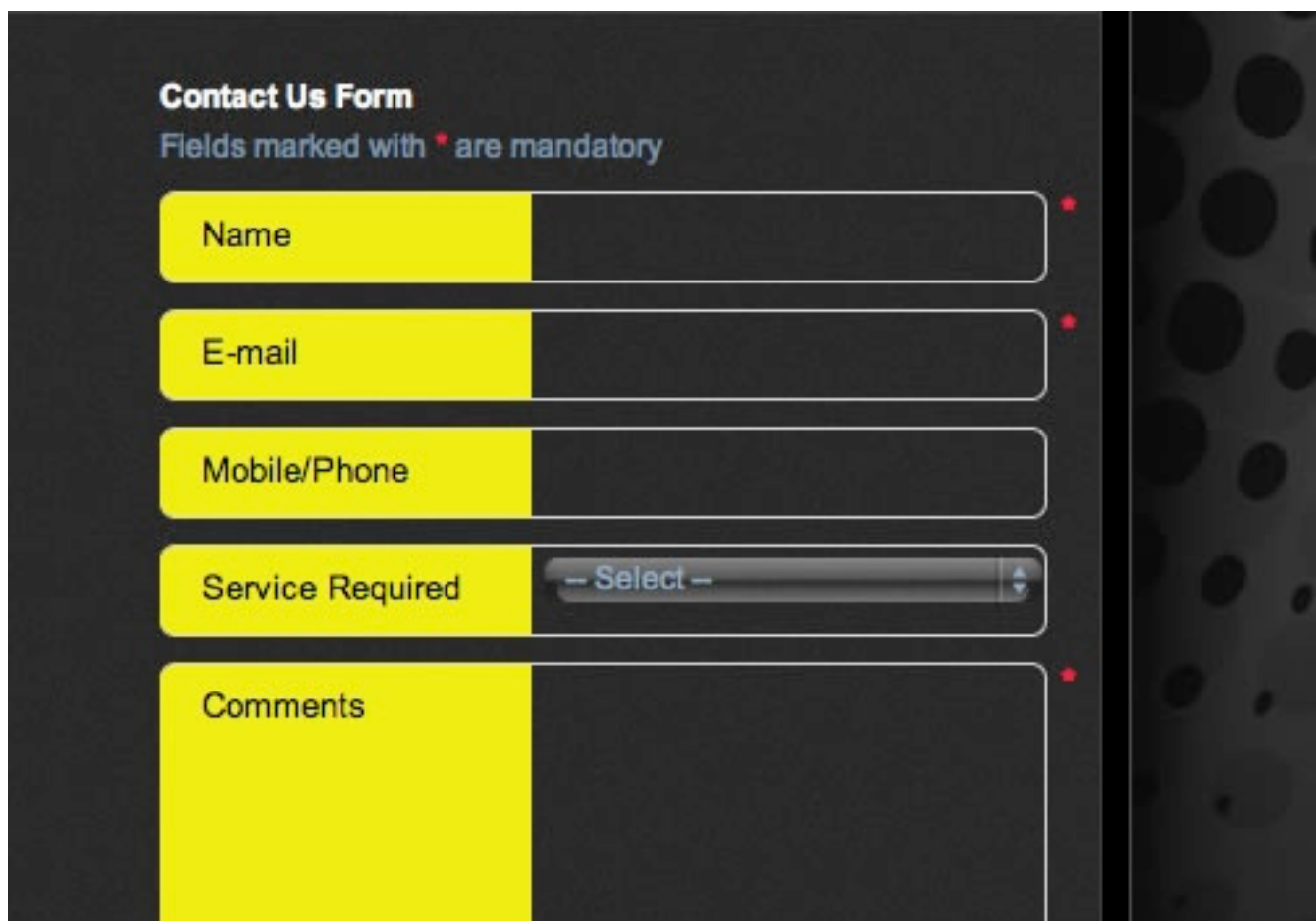
CONFIRM PASSWORD *

SUBMIT

Forms should never consist of more than one column. Notice how easy it is to ignore the column on the right here on [Makeup Geek](#) (not to mention the note about “Required fields” at the bottom).

2. INPUT FIELDS

- **Type of input field** Provide the appropriate type of input field based on what is being requested. Each type of input field has its own characteristics, which users are accustomed to. For instance, use radio buttons if only one option of several is permitted, and check boxes if multiple choices are allowed.
- **Customizing input fields** Do not invent new types of input fields. This was common in the early days of Flash websites, and it seems to be making a comeback; I have seen some odd input fields implemented with jQuery. Simple is often the most useful. Keep input fields as close to their unaltered HTML rendering as possible.



Contact Us Form
Fields marked with * are mandatory

Name *

E-mail *

Mobile/Phone *

Service Required -- Select --

Comments *

Altering the interface of input fields will confuse users.

- **Restricting the format of input fields** If you need to restrict the format of data inputted by users, then at least do so in a way that won't irritate users. For example, instead of displaying MM/DD/YYYY next to a text field for a date, consider using three drop-down fields or, better yet, a calendar control.
- **Mandatory vs. optional fields** Clearly distinguish which input fields cannot be left blank by the user. The convention is to use an asterisk (*). Any symbol will do, as long as a legend is visible to indicate what it means (even if it's an asterisk).

3. ACTIONS

- **Primary vs. secondary actions** Primary actions are links and buttons in a form that perform essential “final” functionality, such as “Save” and “Submit.” Secondary actions, such as “Back” and “Cancel,” enable users to retract data that they have entered. If clicked by mistake, secondary actions typically have undesired consequences, so use only primary actions where possible. If you must include secondary actions, give them less visual weight than primary actions.

Before you submit this form, be certain that you have entered all of the required information. If you fail to submit a required field you will receive an error message. If you receive an error, please use the back button on your browser to return to this form and try again.

The following fields are required before you may successfully submit this form:

- Student number
- Last name
- First name
- Email Address
- Address
- City, State and Zip
- Home Phone
- Course Registration
- Total Cost
- Payment information
- Additional information

Submit Form Reset form

Not clearly distinguishing between primary and secondary actions can easily lead to failure. The above action buttons are found at the end of a lengthy form for enrolling in [St. Louis Community College](#). Just imagine pressing the “Reset Form” button by accident.

- **Naming conventions** Avoid generic words such as “Submit” for actions, because they give the impression that the form itself is generic. Descriptive words and phrases, such as “Join LinkedIn,” are preferred.

4. HELP

- **Text to accompany forms** You should never have to explain to users how to fill out a form. If it does not look like a form or it’s too complicated to fill out, then redesigning it is your only option. Accompanying text should be used only where needed, such as to explain why credit card data is being requested or how a birth date will be used or to link to the “Terms and conditions.” Such text tends to be ignored, so make it succinct and easy to read. As a rule of thumb, do not exceed 100 words of explanation (combined).
- **User-triggered and dynamic help** Rather than include help text next to each input field, show it only where required. You could show an icon next to an input field that the user can click on when they need help for that field. Even better, show help dynamically when the user clicks into an input field to enter data. Such implementation is becoming more common and is relatively easy to implement with JavaScript libraries such as jQuery.

5. MESSAGES

- **Error message** This notifies the user that an error has occurred, and it usually prevents them from proceeding further in the form. Emphasize error messages through color (typically red), familiar iconography (such as a warning sign), prominence (typically at the top of the form or beside where the error occurred), large font, or a combination of these.

- **Success message** Use this to notify users that they have reached a meaningful milestone in the form. If the form is lengthy, a success message encourages the user to continue filling it out. Like error messages, success messages should be prominent. But they should not hinder the user from continuing.

6. VALIDATION

- **Only where needed** Excessive validation is as bad as its complete absence, because it will frustrate users. Restrict validation to confirming key points (such as the availability of a user name), ensuring realistic answers (such as not allowing ages above 130) and suggesting responses where the range of possible data is finite but too long to include in a drop-down menu (such as a country-code prefix).
- **Smart defaults** Use smart defaults to make the user's completion of the form faster and more accurate. For example, pre-select the user's country based on their IP address. But use these with caution, because users tend to leave pre-selected fields as they are.

The image shows a Twitter registration form with the following elements:

- Name field:** Contains "Justin Mifsud". A green checkmark and the text "✓ Name looks great." are displayed to the right.
- Email field:** Contains "justinmifsud@gmail.com". A red X and the text "✗ This email is already registered. Want to login or recover your password?" are displayed to the right.
- Password field:** Contains a masked password (dots). A green checkmark and the text "✓ Password is perfect!" are displayed to the right.
- Username field:** Contains "justinmifsud". A red X and the text "✗ This username is already taken!" are displayed to the right.
- Suggestions:** Below the username field, suggestions are listed: "MifsudJustin", "MifsudJustin", and "justin_mifsud".
- Keep me logged-in:** A checkbox is checked, with the text "Keep me logged-in on this computer."
- Terms of Service:** A section titled "By clicking the button, you agree to the terms below:" followed by a scrollable area containing the text: "These Terms of Service ('Terms') govern your access to and use of the services and Twitter's websites (the 'Services'), and any information, text, graphics, photos or other materials".
- Printable versions:** Links for "Terms of Service" and "Privacy Policy" are provided.
- Create my account button:** A large yellow button with the text "Create my account".

[Twitter's](#) registration form uses both dynamic validation (for the name, email address, password and user name) and smart defaults ("Keep me logged in").

Conclusion The Beginning

The word "conclusion" is not right here. Let this be your starting point to take what I have written about and apply it to your own forms. The good news is that there is much more to say about all this; you can find an abundance of resources on each point made here. For starters, three books are listed below that inspired me when writing this post. As I stated at the beginning, taking shortcuts by only tweaking the UI will not make your forms more usable. What more can I say? The theory is now with you. Go get your hands dirty.

10 Usability Nightmares You Should Be Aware Of

Vitaly Friedman

Sometimes you just want to get the information you're after, save it and move along. And you can't. Usability nightmares — which are rather the daily routine than an exception — appear every now and again; usually almost every time you type your search keywords in Google. In his article "[Why award-winning websites are so awful](#)" Gerry McGovern points out that "*the shiny surface wins awards, real substance wins customers*" and that is absolutely true.

Nevermind what design you have, and nevermind which functionality you have to offer — if your visitors don't understand how they can get from point A to point B they won't use your site. In almost every professional design (except from special design showcases such as, e.g., portfolios) you need to offer your visitors

- a clear, self-explanatory navigation,
- precise text-presentation,
- search functionality and
- visible and thought-out site structure.

And that means that you simply have to follow the basic rules of usability and common sense. You want to communicate with your visitors, don't drive them away, right?

In this article we take a look at some of the usability nightmares you should avoid designing functional and usable web-sites. At the end of the article you'll also find 8 usability check-points you should probably be aware of.

1. Hidden log-in link.

[Backpack](#), 37signal's one of the most usable organizational and project management tools out there, explains exactly what the tool can be used for, how one can use it and which features it has to offer. However, once you've signed up, you might need few minutes to find out what you should do to actually start using the tool.



Backpack

[Home](#) | [Tour](#) | [Examples](#) | [Buzz](#) | [Customer forum](#)

Get organized!

Keep your to-dos, notes, ideas, and
schedule online with Backpack.

Take a tour or **Sign up for free**

Over **300,000 people** signed up! Got an account? [Log in.](#)

“Tremendous.”

PC WORLD

“Best of the Web. Editor's & Readers Choice.”

BusinessWeek

“A cool organizational tool. How very cool.”

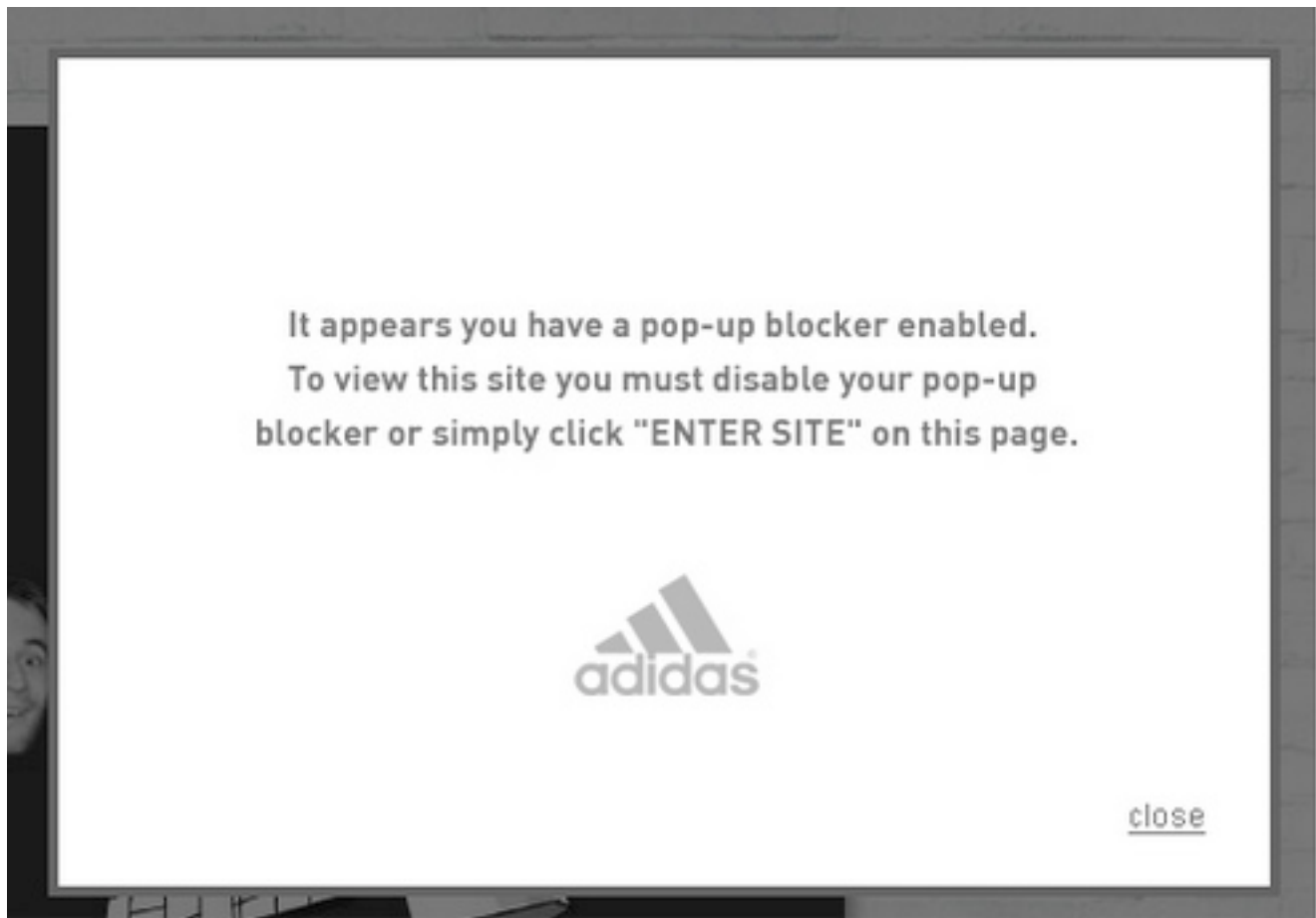
The New York Times



The “Log in”-link (hint: in the yellow box) should have a greater font-size — also icons would do the job. Every registered user has his/her own personalized page, however new registered users will need a while to find out how they can log in. As Lee in our comments said, “someone that's invested their time to sign up for a service will spend more than two seconds finding out how to log in.” Ok, but why should he/she? Why not just place the link somewhere where it's visible? It sounds reasonable to us.

2. Pop-ups for content presentation.

Almost every modern web browser uses a popup-blocker to prevent pop-ups, ad blocks and further site content recognized as advertisement. Firefox, Safari, Opera and Internet Explorer make use of it — therefore the idea to use pop-ups to present the main content isn't probably the most reasonable idea web-designer might come up with.



However, Adidas developers team seem to consider pop-ups as a creative approach for their target audience. It's not clear what ImpossibleStory.com is actually about — most users will never find out; even we were too lazy to deactivate our popup-blocker.

3. Dragging instead of vertical navigation.

Actually, this technique — not a real nightmare, but very unusual — might become a new trend in the future. Dragging, as used from .pdf-documents, can also be adapted to web-sites. [Fichey](#) offers a Flash-based solution. Interesting approach, however you can't bookmark specific parts of the site — a typical problem for Flash-based designs.



An introductory info box on the start page of the site informs its visitors about the new way of navigation. The tool shows sites which are currently popular in social media. By the way, the displayed links don't work — apparently, the tool shows the screenshots which are saved and embedded as images.

4. Invisible links.

Visitors have to know where they are, where they've been and where they can go next. If designers don't present this information in an appropriate way, visitors can have serious problems with site navigation. [Real Player had been using a bunch of invisible links](#) for a while. Take a look at the picture and the labels below (originally created by Trenton Moss) — can you spot the links?



1, 3, 4, 6, 7 and 11 are links, and 2, 5, 8, 9 and 10 aren't.

5. Visual noise.

Often less is more. Visual noise is probably one of the most typical problems large web-sites' designers have to cope with. And it's extremely easy to get it wrong. As [Overstock](#) does.



Jaeger LeCoultre
Reverso Grande
Automatic Watch

Our Price: \$6,249.99

List: \$8,250.00 (24% off)



Invicta Men's Swiss
Quartz Steel Watch

Our Price: \$119.99

List: \$475.00 (75% off)

Review: ★★★★★



Lucien Piccard Men's
Chronograph White
Dial Watch

Our Price: \$129.99

List: \$450.00 (71% off)

Review: ★★★★★



Sturhling Alpine Men's
Automatic Skeleton
Watch

Our Price: \$149.99

List: \$795.00 (81% off)

Review: ★★★★★



Oris Williams F1
Team Chronograph
Men's Watch

Our Price: \$1,749.99

List: \$2,575.00 (32% off)



Invicta Pro Diver
Men's Automatic Steel
Watch

Our Price: \$99.99

List: \$285.00 (65% off)

Review: ★★★★★

Bonus: It's also not quite clear what is a link and what's not. Overstock also uses a number of different link presentations and hover effects. Link or not a link? You might have a lot of fun finding out.

6. Dead end.

You can use different approaches to introduce your new web-service to your visitors. [Scriblink](#) welcomes its visitors with a pop-up and a Java-applet. Visitors have to provide some input to start browsing through the site.



Nothing helps:

- clicking on “cancel” doesn’t help,
- OK doesn’t help,
- full-screen mode doesn’t help,

- clicking somewhere else doesn't help,
- closing the browser window doesn't work,
- clicking on the question mark for help doesn't work.

Unfair, really. We just wanted to see what the tool does...

7. Content blocks layering upon each other.

A typical problem which always appears if large Flash-movies are used on large web-sites. The site navigation is hidden; whatever users might be willing to browse to — they have no chance.

[Download](#) | [Support](#) | [Solution](#) | [Company](#)

Ulead Worldwide ▼

Asia Pacific

Australasia

Canada

China

Italy

Japan

Netherlands

Poland

Russia

Spain

Taiwan

United Kingdom

United States

Ulead® PhotoImpact® 12 is a complete page-editing suite. Choose powerful yet easy-to-use tools and share in your photos, greeting cards, labels, calendars, prints, email and more.

xpressFix Mode

le is perfect for beginners to get started fast.

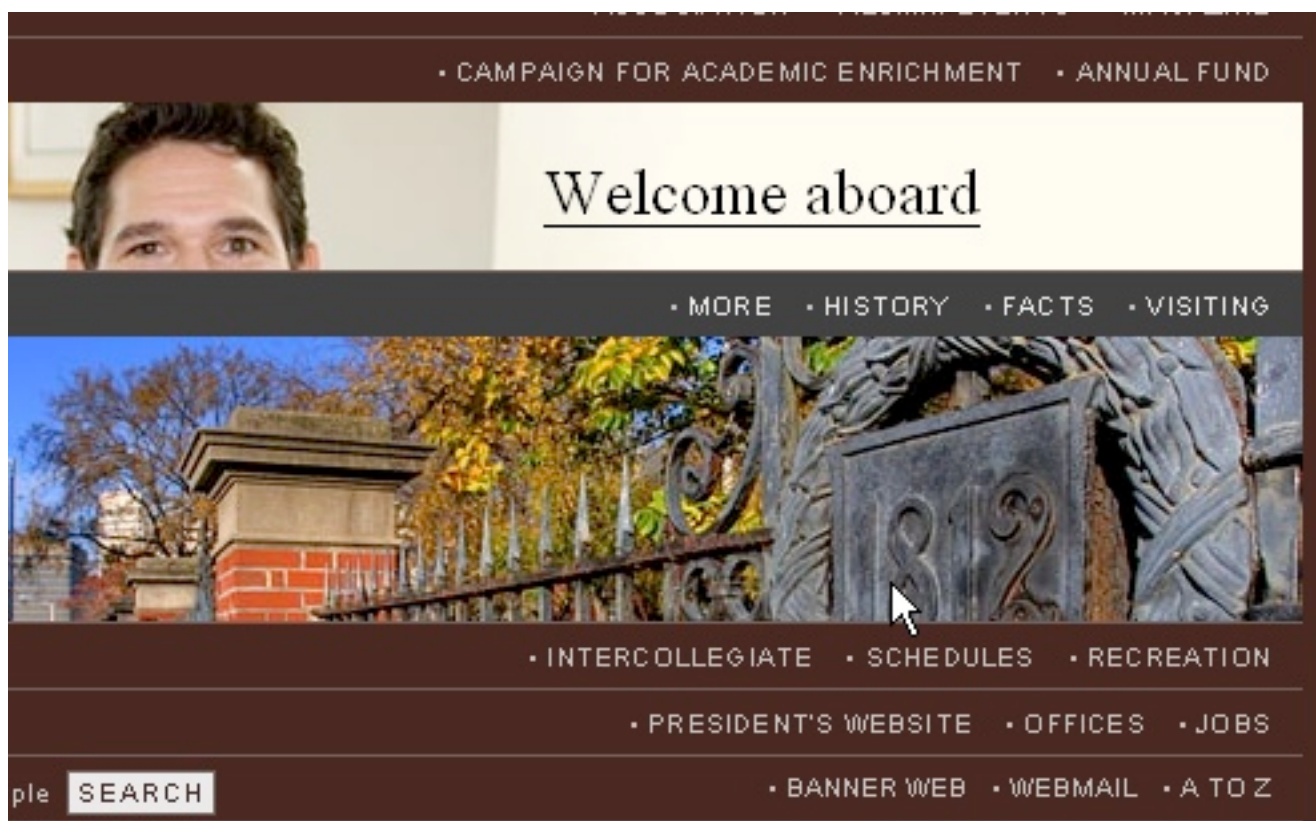


The same problem on [Yahoo.com](#):

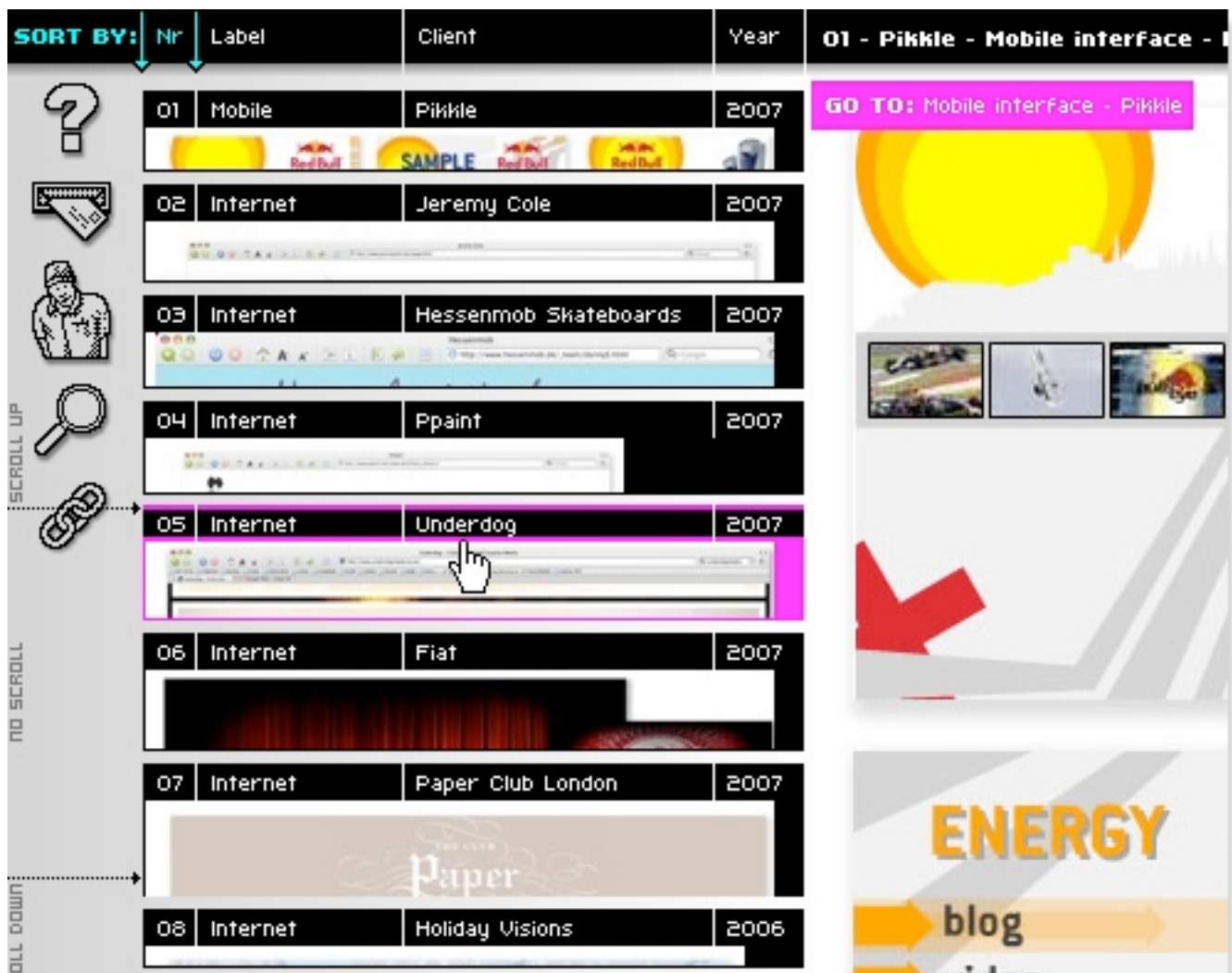


8. Dynamic navigation.

What looks like a usable site navigation, shows itself soon enough from its worst side. Nevermind where you point your mouse pointer to — supporting images slide down and up and change the focus of the link you’ve clicked on. On the image below the link “history” was clicked — it moves away; and no, the sliding images aren’t linked to the page you’d like to browse to.



The effect appears in both left- and right-side navigation menus. Visitors might need some time to find out what is happening. However, once the idea is understood, the navigation isn't that hard to deal with.



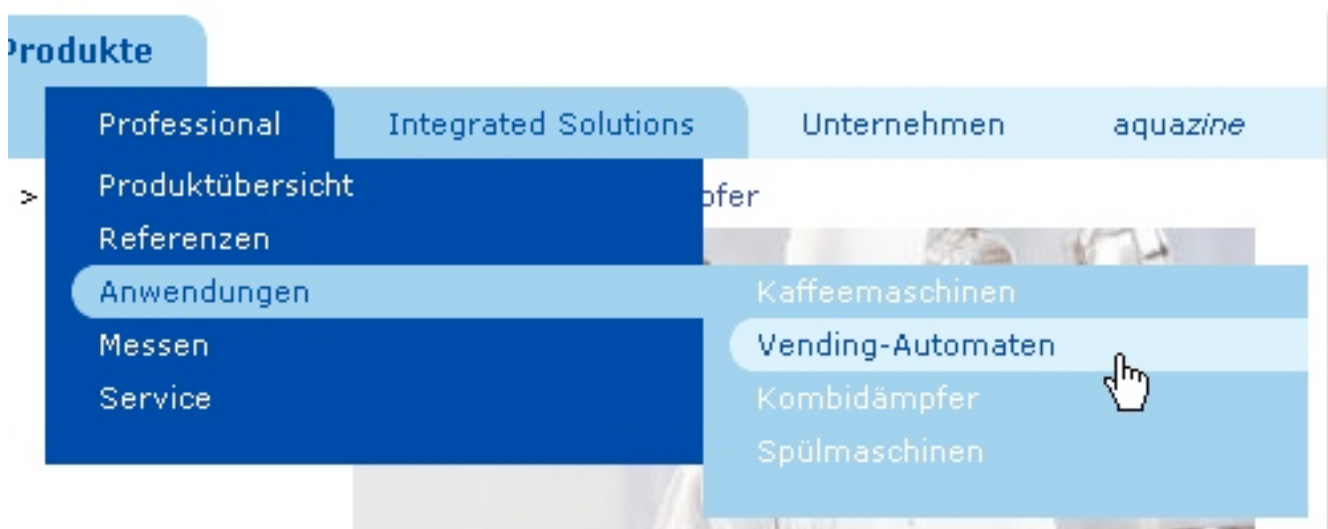
The same idea on [Kabe243](#). Bonus: visual noise at its best.

9. Drop-Down Menus.

Drop-Down menus are useful for web-developers and almost always get on users' nerves. If you — as a designer — hide navigation items in a drop-down menu you can save yourself a large amount of vertical space; however **users have to focus the mouse precisely to get to the section they'd like to visit**. It's not usable.

However, it can be even worse. If the distance between different levels of navigation is too large (for instance because some navigation items have more text) users have to move the mouse horizontally. If the mouse focus changes its vertical position, users have to start from the beginning.

[2Advanced](#) and [Brita.net](#) are excellent examples for these usability nightmares.



Notice: it took us a dozen of attempts to actually make these screenshots.

10. Blinking images.

Sometimes you just want to read the content of a web-site you are visiting. And you can't. To fight against the banner blindness advertisers make use of animated ads — usually animated .gif-images or Flash-movies. In both cases it might become extremely hard to focus on reading if such images are blinking all around the content.

[Digital Web Magazine](#), an excellent online magazine we read on a regular basis, has a disturbing animated ad on the left side of every article.

☰ articles by title

☰ articles by type



Advertise with us

Publishing services provided by
Blue Flavor

☰ Published on September 24, 2007

Digital Web: So what is Sidebar Creative? Are you did this all come about?

Bryan Veloso: We've all worked with each other level in the past, but it was an idea that has floated since SXSW2006. I just remember poking Dan one saying, "We really need to get this thing started. Let get a chat going to see what will happen."

Steve Smith: They asked Jonathan and me if we interested. A few Skype chats later, and there it w. partnership of like-minded individuals that just war great websites.

Jonathan Snook: I often refer to it is a "freelance sorts.

Future Nightmare?

Pointing the mouse instead of clicking.

Dontclick.it, an interesting experiment in interaction design, explores a clickfree environment. It wants to explore how and what changes for the user and the interface once you can't rely on the habit of clicking.



Within this interface you won't find any buttons. Instead you navigate the contents in a different way — by pointing the mouse to the areas of the site you are interested in. What sounds like an interesting approach might soon become a usability nightmare — just imagine web-users trying to open external links in a new browser window; besides it's interesting to know how exactly users will be able to, let's say, open a list of links in new browser tabs (updated to clarify why we've put Dontclick.it in this post).

8 Usability Check-Points You Should Be Aware Of

1. **You don't use pop-ups.**

Pop-ups interrupt the browsing session of the visitors and require an instant feedback. Respect your visitors.

2. **You don't change users' window size.**

The same argument as the one against pop-ups holds. Some browsers, e.g. Internet Explorer, saves the browser dimensions and uses them for further browser sessions. As Ben Bodien commented, “it's just plain inconsiderate to assume that you know better than the user how their software environment should be configured?”

3. **You don't use too small font sizes.**

Long passages are harder to read, and to read brief sentences readers need more time. It holds also for links, buttons, forms, search boxes and other elements. Good news — in Web 2.0 the opposite is the case.

4. **You don't have unclear link text.**

Links have to be precise and lead to the destination they describe. Ambiguous link descriptions should be avoided.

5. **You don't have dead links.**

There are too many of them anyway; why would you want to point your visitors to a dead end?

6. **You have at most one animation per page.**

If blinking images are wide-spread through the site, it's extremely hard to focus on one single site element. Give your visitors an opportunity to perceive your content. Using animated ads, don't place them right along your articles.

7. You make it easy to contact you.

Maybe because you just don't want to be contacted, but If visitors do want to get in touch with you, but can't find any contact information, you lose their interest and trust. Disastrous for online-shopping, a missed opportunity for the rest.

8. Your links open in the same window.

Visitors want to have control over everything what happens in their browser. If they'd like to open a link in a new window they will. If they don't want to, they won't. If your links open in a new window you make the decision which is

About The Authors

Dmitry Fadeyev

Dmitry Fadeyev is the founder of the [Usability Post](#) blog, where you can read his thoughts on good design and usability.

Justin Mifsud

I am a user interface designer and user experience consultant by day and blogger by night. I own and run [Usability Geek](#) where I evangelize about the importance of making the web a usable place and, more importantly, how to do it. Ah yes, I like to [Tweet](#) too!

Vitaly Friedman

Vitaly Friedman loves beautiful content and doesn't like to give in easily. Vitaly is writer, speaker, author and editor-in-chief of [Smashing Magazine](#), an online magazine dedicated to designers and developers.

More Smashing eBooks

You can buy Smashing eBooks from our selected retailers, and for various e-readers. While you can buy Smashing eBook bundles in our Smashing Shop, single eBooks are available from Amazon.com, Apple iBookstore and Google eBooks.

The table below illustrates retailer and device compatibility, with the optimal device for each retailer highlighted in blue.

Retailer	iPad*	Kindle	Android
Apple iBookstore	X		
Amazon.com	X	X	X
Google eBooks	X		X
Smashing Shop	X	X	X

**includes devices such as iPhone and iPod Touch.*