UXPin

Tactical UI Design Patterns

The Handbook to Faster Design

UXPin

Tactical UI Design Patterns

The Handbook to Faster Design

Copyright © 2015 by UXPin Inc.

All rights reserved. No part of this publication text may be uploaded

or posted online without the prior written permission of the publisher.

For permission requests, write to the publisher, addressed “Attention: Permissions Request,” to hello@uxpin.com.

Index

What's a UI Pattern? 7

How Users Interact With Design Patterns 8

Following Universal Design Conventions 11

Applying Empathy to UI Design Patterns 16

Conclusion 20

Why Use UI Patterns? 21

Why Patterns Work 23

Expectations Reinforce Themselves 24

Deadline-Busting Communication 26

Why *not* use patterns? 27

The Importance of Prototyping First 29

Got a Pattern? Plan it Out 30

Thinking Through the Process 32

Patterns Take Guesswork Off of Developers' Plates 34

User Testing: Insights You Can't Ignore 36

Prototyping UI Patterns **40**
Explaining the Gray Box 41
Pattern Libraries Are Prototyping Shortcuts 44
Reusable elements 45
Patterns and Prototypes Work Together 47

Applying UI Design Patterns **48**
Building a Pattern Library 49
Riffing on Design Patterns 52
Tweaking Pattern Styles 54
Going forward 57

Useful UI Pattern Examples **59**
Formatting Data 61
Getting input 67
Navigation 83
Teasers 88

Ben Gremillion is a Content Designer at UXPin. Previously, he was a Design Writer at ZURB. He started in newspaper design, saw a digital future, and learned HTML/CSS in short order. He later earned an Adobe Certification, and learned up regex, PHP, MySQL, and other impressive-sounding acronyms. He also builds and maintains a CMS for webcomic artists, and participates in bi-annual NaNoWriMo challenges.

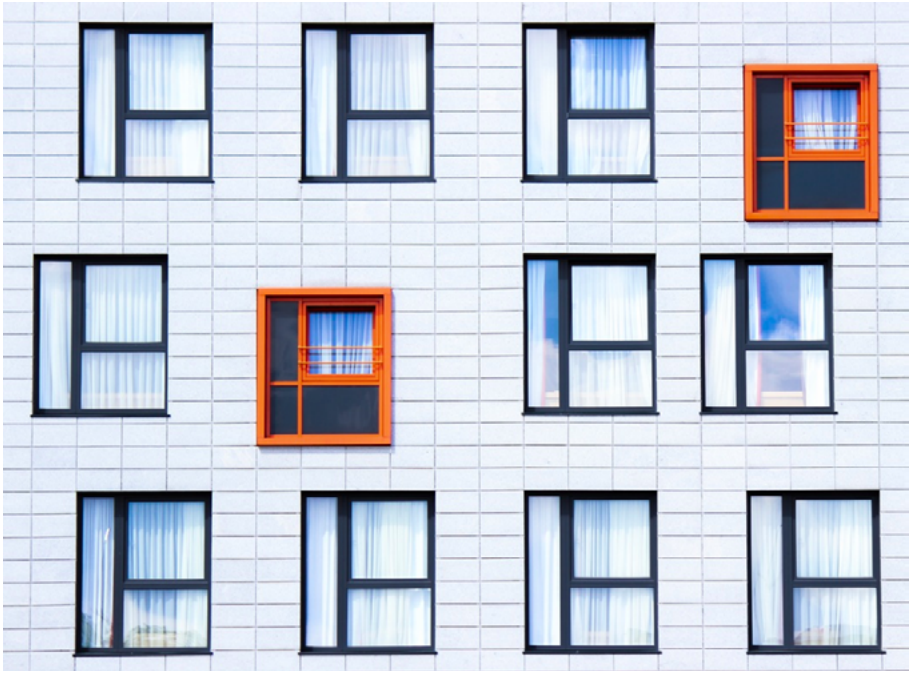
Jerry Cao is a content strategist at UXPin where he gets to put his overly active imagination to paper every day. In a past life, he developed content strategies for clients at Brafton and worked in traditional advertising at DDB San Francisco. In his spare time he enjoys playing electric guitar, watching foreign horror films, and expanding his knowledge of random facts.
Follow me on Twitter

Co-founder and head of product, Kamil previously worked as a UX/UI Designer at Grupa Nokaut. He studied software engineering in university, but design and psychology have always been his greatest passions.
Follow me on Twitter @ziebak

What’s a UI Pattern?

The web’s been around for more than 20 years, giving us ample time to discover user experience problems – in fact, to discover “user experience” at all – and invent solutions.

Some solutions worked so well that designers everywhere began to use them repeatedly. And through familiarity, users came to expect them. This was the beginning of web design patterns: a common visual language that both designers and end users understand.



It wasn't hard to do. People are hard-wired to recognize patterns, even where there are none.

Shapes in clouds, numbers in lottery tickets, predictable stock market fluctuations, stellar constellations, and faces on toast all show how people see things that don't exist. Research shows that people look for relationships between values, and patterns reinforce those relationships. That's a recurring theme in design patterns. They're self-fulfilling and self-reinforcing.

UX designers can easily take advantage of users' pattern-seeking tendencies. How do people navigate many pages? How do we indicate that, when clicked, a video will start playing?

The answers established, reusable solutions to design problems. They're also time-savers. Patterns are the first solutions people go to when they don't have time to invent something new. And like many "it just works" solutions, patterns tend to stick around.

How Users Interact With Design Patterns

First impressions are critical since users know within 10-20 seconds if they'll leave your site. Patterns communicate function and intent as clearly as possible within that window of opportunity.

Here's how it works. Let's say a user wants to accomplish a task, like to book a hotel.

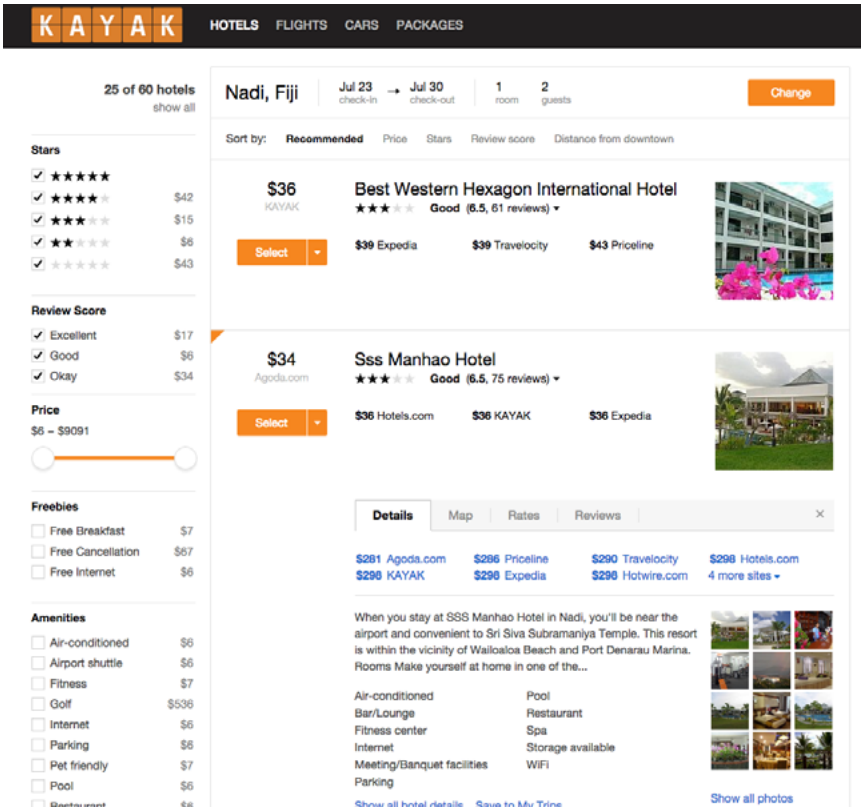


Photo credit: Kayak

Take a look at a possible series of user actions:

1. A travel site presents the user with a list of possible places to stay.

2. The user taps the photo of a charming bungalow. Nothing happens. Lesson learned: Photos do not lead to more information about each property.

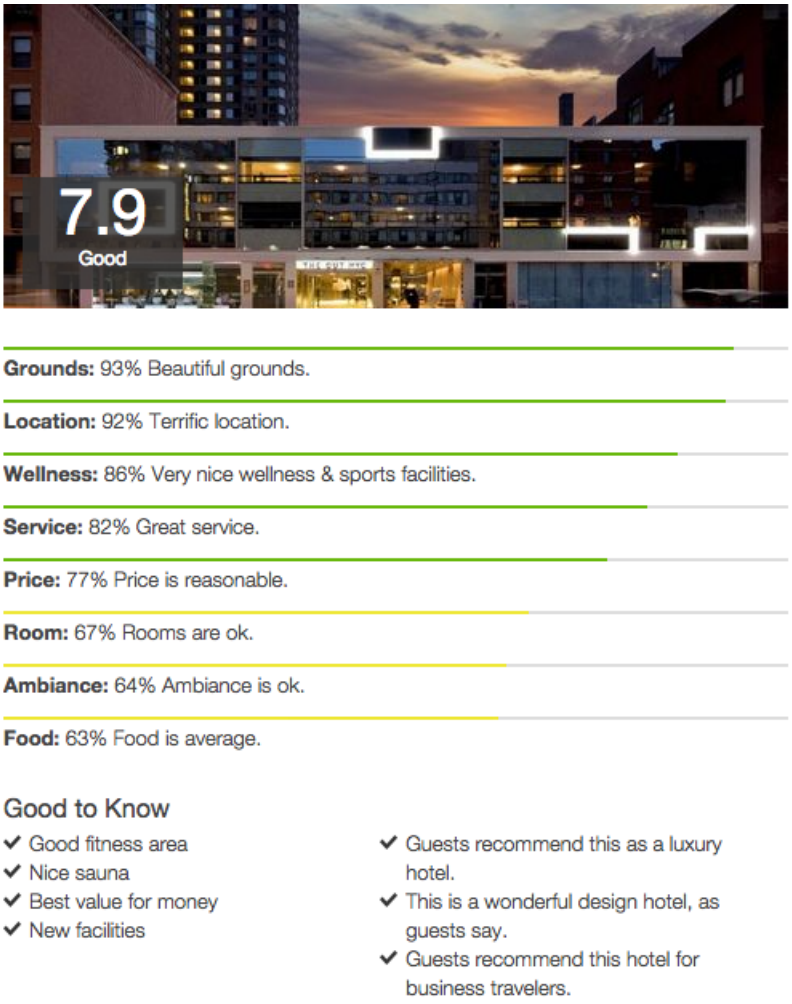
3. The user continues to explore. He or she may look for a "more info" link or tap on the nightly rate. They may try tapping the photo again, just in case. Left on their own, a determined user will continue to tap different elements in search of more information about the bungalow.

4. Finally, they tap the hotel name (perhaps presented as a button or link), and they're redirected to a more detailed page. The user

sees a charming atmosphere but no complimentary breakfast. It doesn’t fit their criteria, so they keep searching.

Now, here's where the pattern kicks in. Having learned that one thumbnail photo isn't tappable, but the hotel’s name is, the user now knows exactly what to do.

This learned behavior is something people do especially well. While many factors influence how people learn, a little experience goes a long way. This is a form of instrumental learning, in which people remember the result of past experiences well enough to alter their future behavior.



Not every pattern is so absolute as the search results example. The *idea* of a pattern supercedes its implementation. For example, maybe bullet points are round dots, little squares, check boxes or funny squiggles. But if an object sits to the side of a line of text, and there’s more than one of them, users understand it functions as a bullet point.

In a similar vein, links don’t always need to change from blue to purple. Like the different bullet formats, users understand that the real pattern is that links change color to reflect their status. You can implement that pattern by making unvisited links green and visited links grey, and users would still understand.

Users remember pattern behavior more than how the pattern itself looks. If you want to play it safe, always stay consistent with the behavior.

Following Universal Design Conventions

Why did our hypothetical hotel-seeker even try to tap a tiny images of hotels? Because small photos, each next to brief descriptive blurbs, is common.

News sites use it. Blogs use it. Amazon makes product images in search results tappable. So does Google and, more to the point, TripAdvisor, Orbitz and Yelp. This is an *established pattern.* It's simply understood without explanation.

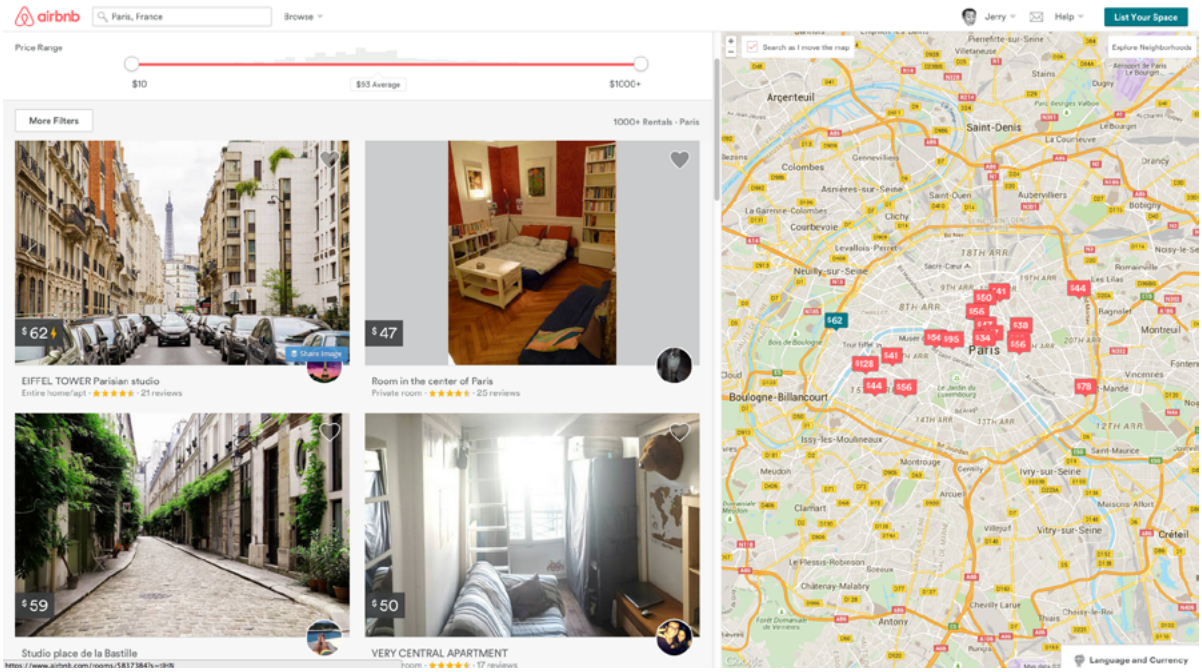


Photo credit: Airbnb

In our hotel example, we broke the established pattern. The user tapped a thumbnail on our site and nothing happened. They got slightly irked. Not enough to drive them away (remember that they kept tapping) but enough to set a bad precedent.

Broken expectations are cumulative. As described in *Consistency in UI Design*, if a design deviates too far from expected conventions (externally inconsistent), designers face resistance to using the site. If a design deviates *from itself (*internally inconsistent), then it just becomes downright frustrating.

It takes a strong curiosity, or exceptional surprise and delight, to educate users about new conventions. But old habits die hard, especially on the web, where conventions reinforce each other with every new hyperlink. We described many visual patterns in *Web Design for the Human Eye*, but let's recap some fundamentals behind them.

1. **Gestalt**

   People draw connections between disparate shapes. In 1910, psychologist Max Wertheimer watched flickering lights of a railroad crossing. The pattern created an illusion that the lights were moving in a circle, even though they were merely alternating with good timing.

   Wertheimer, along with his colleagues, developed theories on sight perception that are foundational to web design. The theories expound on Aristotle's simplistic but classic axiom, "The whole is greater than the sum of its parts," as applied to visuals. The Gestalt principles tend to revolve around a handful of similar concepts. While relating to all sight perception, these ideas are especially applicable to UI design.

Photo credit: UXPin

   The logo above doesn't have a capital letter "H" – but our tendency to see shapes where there are none suggests otherwise.

   Application to web design: Shapes and contours should take precedence over lesser details. It doesn't matter how fascinating a clickable button looks if your users don't know it's a button. These "signifiers" help suggest the function (known as affordances).

2. Proximity

People perceive that closely-placed elements are related, especially if separated from other groups by even more space. This also ties into the time and spatial elements of Hick's Law. This principle uses negative space to push certain elements apart, and others closer together, to create apparent groups that don't need lines or other indicators to spell out "this is a group" like, say, default HTML fieldsets.

Application to web design: Grouping elements into sections is vital to page composition. Sidebars need to stand apart from main content. Sections within content need breathing space to indicate that they're sections. Photos, titles and descriptive text in search results need to form clusters to form discrete units.

Photo credit: UXPin

Like you can see on the right in the above illustration, the search results become discrete items when you add a little space to indicate "this is a group."

3. Color

Bright colors seem to "rise above" cool, dark shades and hues. When adding depth to design, few techniques are more effective than darkening items in the background. To understand why, we break down color theory into three parts:

- **Contrast:** Every shade of color has a set opposite – an "arch-nemesis" whose contrast is far greater than any other color.
- **Complementation:** Colors aren't always at odds with each other. Complementary colors accent each other and bring out their best, the opposite of contrast. These are the colors immediately next to each other on the color wheel, for example, purple's complements are blue and pink.
- **Vibrancy**: Each color evokes a specific moods: the brighter warm colors (red, orange, yellow) tend to energize a user and make them more alert, while darker cool shades (green, blue, purple) tend to be more relaxing and tranquil.

Photo credit: UXPin

Two variations of a logo create two different feelings. On the left, cool shades in the background make the gestalt shapes pop off the

canvas. On the right, warm colors are more inviting – and possibly exciting. But they fall behind the white H shape, which dominates and defines the logo itself.

None of this registers to the average user. All she or he wants is to book a clean room. The fastest way from searching to booking is to use a pattern they're already familiar with. As we mentioned, search results – hotel or otherwise – have an established format.

In the case of a hotel-finding site, we must present users with relevant search results so they can make an informed decision. Sometimes that means foregoing our creative impulses and following tried-and-true-and-boring processes. But that doesn't mean we can't have fun.

Applying Empathy to UI Design Patterns

We can use familiar patterns to our advantage. But to consciously use their unconscious tendencies, we need a little empathy and a lot of planning.

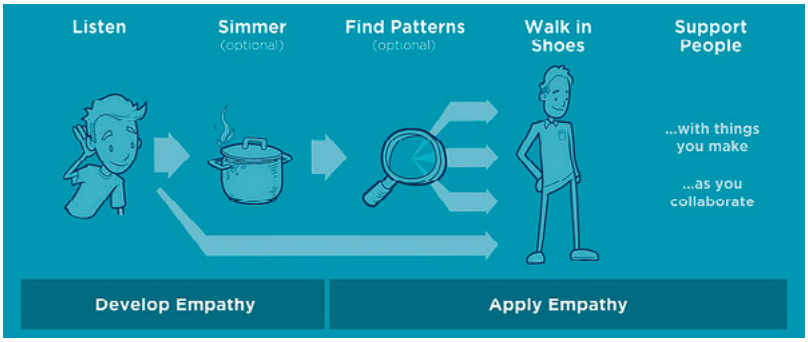


Photo credit: Rosenfeld Media

To start, we must define exactly what our problems are. That's a discipline in itself, and often includes questions like:

- Which elements are more important than others?
- What's clickable, and what isn't?
- How do users know they can scroll for more information?
- Which icon should I use for a given task?
- What tells users that a function is disabled?
- How do users know when one section ends, and another begins?
- How do we tell users "be patient – this is still loading?"

… and other questions about communication that designers ask themselves. But to approach it with empathy, we have to turn things around. See things from the user's point of view.

- Where am I on the site?
- What should I look at first?
- Where can I go from here?
- How do I make this modal window go away?
- Should I tap this, or should I tap that?
- For that matter, what *can* I tap?
- Does this page contain the information I want?
- What can I ignore on this page?

These are the questions we must address as patterns: not what problems in design we want to solve, but how we solve our users' problems with design.

Designers' questions are based on setting up users for success. Users' questions are driven by the need to disregard anything they do not

want, sifting out what they do. Anything that distracts from that goal isn't necessarily bad design… but it sure doesn't help.

When we seek patterns on the web, we're hunting for proven solutions that help users accomplish their goals, whether it's choosing a hotel, browsing articles, reading sports statistics, posting pictures or chatting with friends.

Luckily we're surrounded by patterns. They're so common that it's easy to overlook them. But they exist. Aside from looking for them yourself, sources include:

1. **UI Patterns** – A repository of samples from around the web, organized by subject with helpful descriptive text.

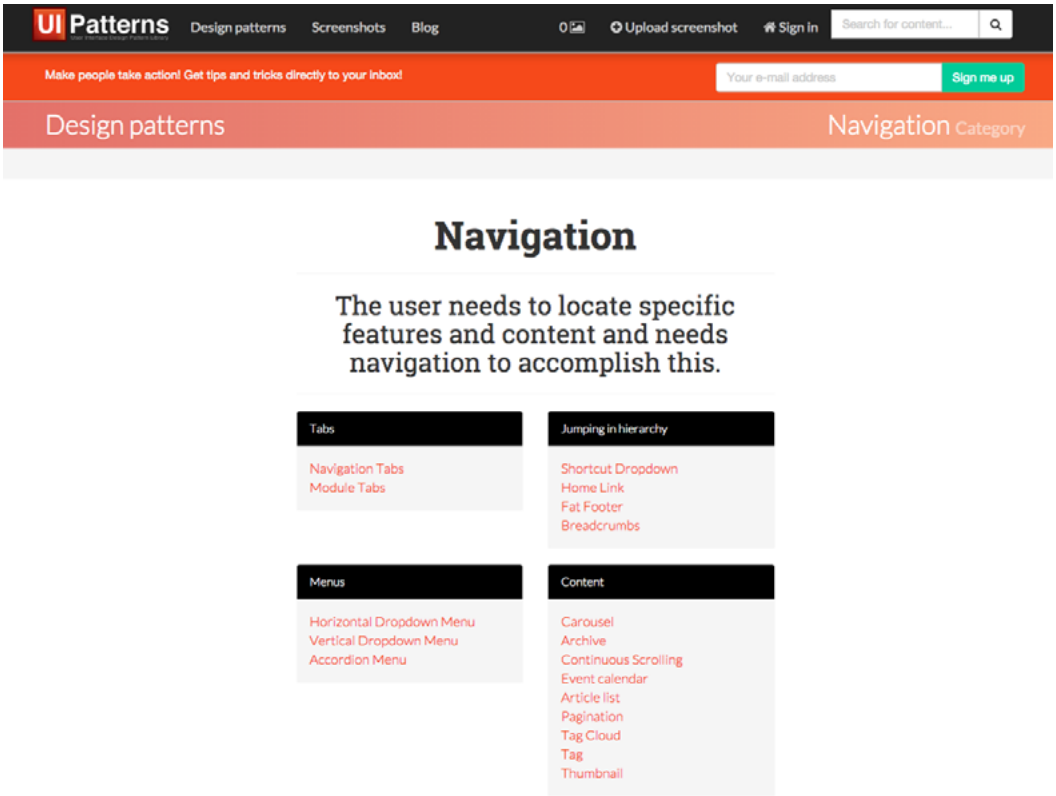
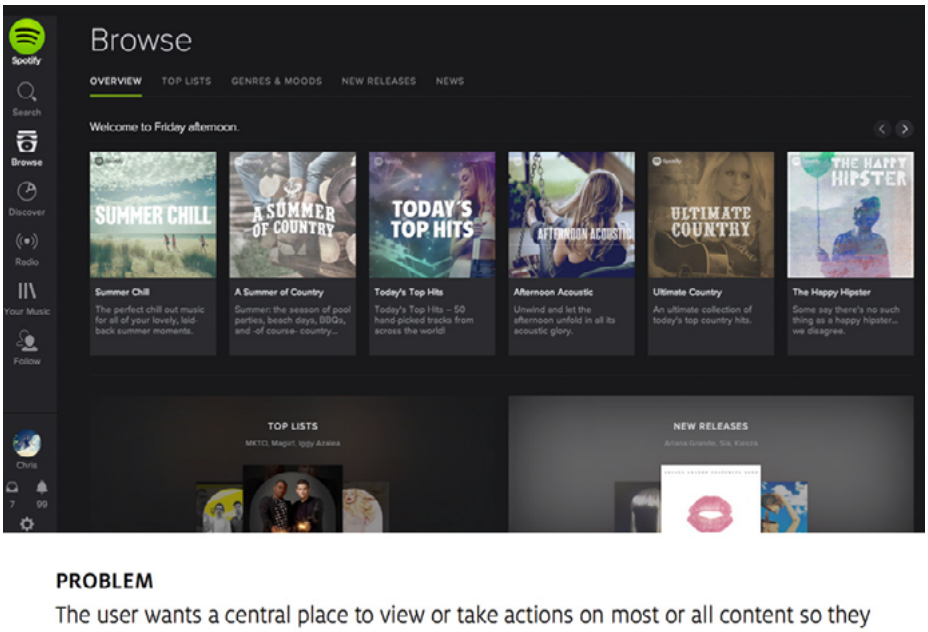


Photo credit: UI Patterns

2. **Web UI Design Patterns** – Packed with information about the how and why of patterns, our free e-book covers current trends in UI design.



PROBLEM

The user wants a central place to view or take actions on most or all content so they

3. **Persuasive triggers** – Search the web long enough and you'll find gems like this Smashing Magazine article about psychological and behavioral patterns.

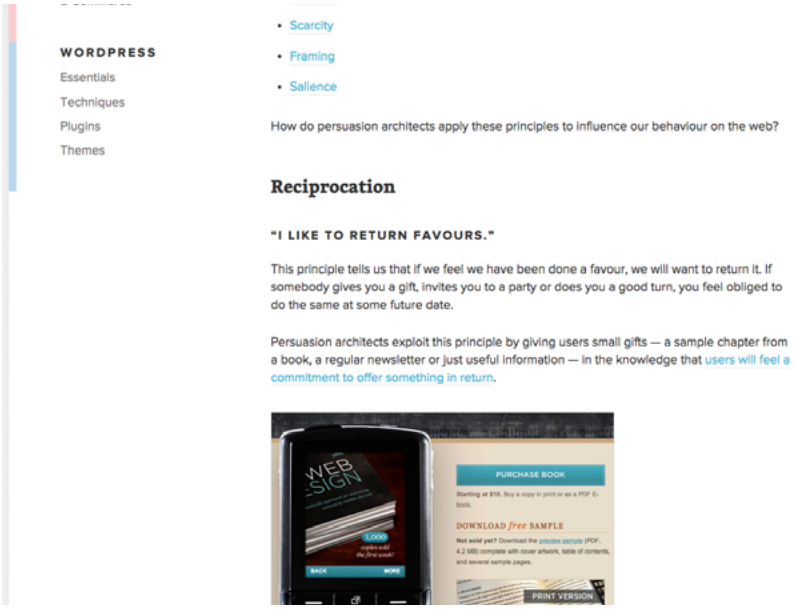


Photo credit: Smashing Magazine

4. **PatternTap** – A vast collection of screenshots, organized with tags, that drive inspiration for design elements.

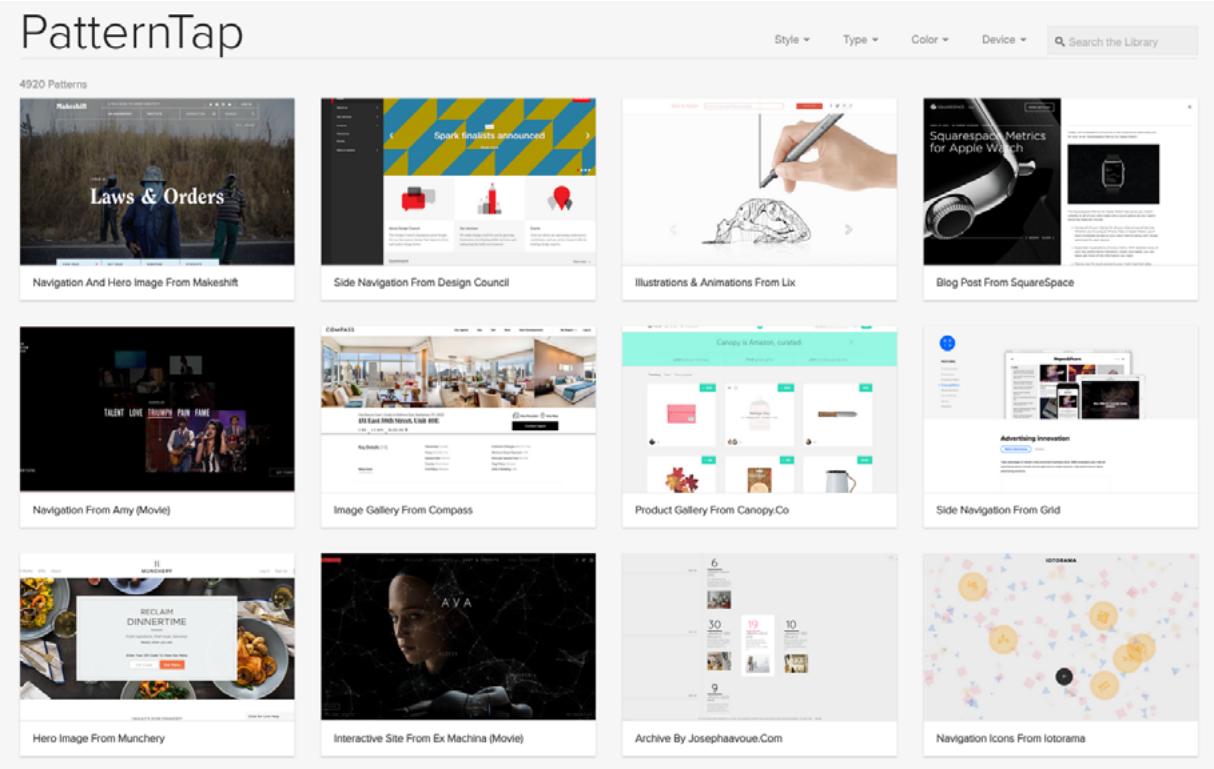


Photo credit: PatternTap

Conclusion

The more designers use patterns, the more users come to understand them, making them more valuable for designers to use. This reciprocal relationship has persisted since the first days of web design. Trends come and go, but the underlying ideas remain.

Why Use UI Patterns?

Few clients ask for boring designs.

Plain designs don't win awards. And no one advances from repeating themselves, let alone copying others. So at first blush a designer should avoid repeating what's come before. Standing out in a world of competitors is paramount, so why do the same thing as everyone else?

Because patterns work for users.

No, strike that: because you can make patterns work for you.

For all their strength, reliability and acceptance, design patterns are starting points. They're frameworks upon which you can hang your unique look. A good pattern is an idea, not a particular look – a guidebook, not a roadmap, to the ultimate destination.

The two are sometimes confused. For example, there are drag-n-drop elements like carousels which come with their own styles. A down-

loadable carousel plugin is not a pattern, though. It's a component built on a pattern that says, "put image in the middle, arrows on either side. Clicking the arrows advances/returns the next/previous image in a set."

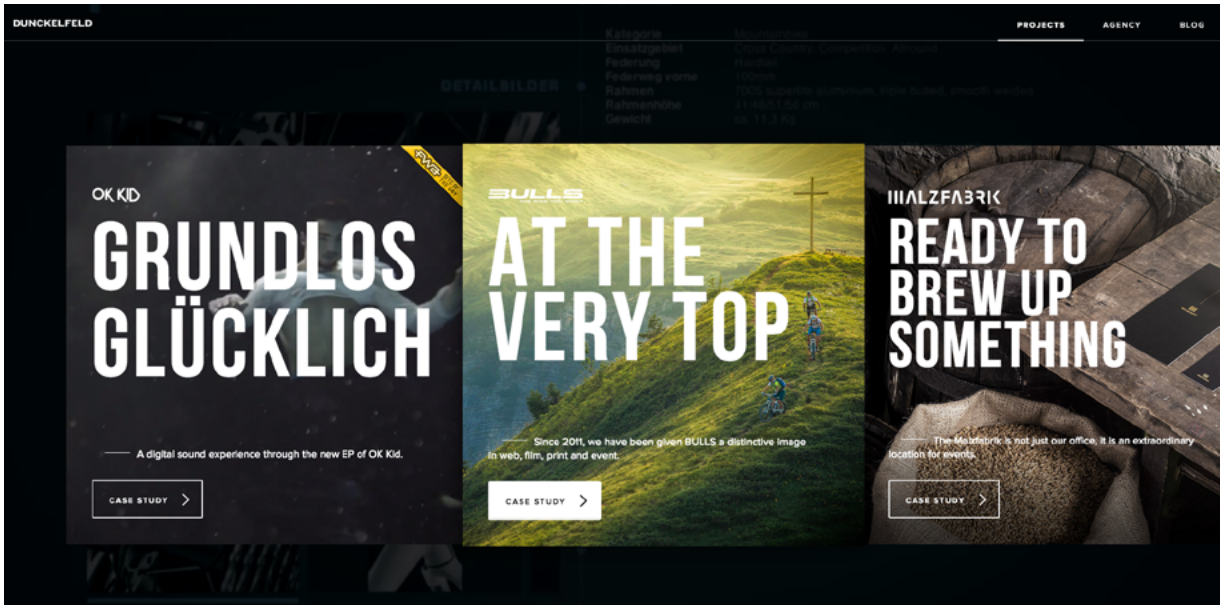


Photo credit: Dunckelfeld

Or take breadcrumbs, the little links that help users traverse a hierarchy. The idea is straightforward: one link represents each level, starting with the highest level, separated by a non-tappable character or glyph. Seems simple enough, and you can download pre-defined sets from many places. Yet there are endless ways to run with that idea. Therein lies the advantage.

Home / Products / Widgets

Home > Products > Widgets

HOME : PRODUCTS : **WIDGETS**

home ▸ products ▸ **Widgets**

UI design patterns only restrict you to common sense. Using patterns gives you a head start in your work – if you understand the reasons behind them.

Why Patterns Work

The first reason to use an established solution is that it has stood the test of time. Patterns are proven, known quantities: technically, visually and practically. The most frequently-used patterns survive because designers have found they accomplish a task the best, e.g. search fields above results and form submit buttons at the bottom of forms.

Take, for example, tappable logos in the upper left corner of a web page.

- **Technically**: HTML for images and hyperlinks is as old as the web. The code is universally accepted across all browsers, and is relatively easy to write and edit.
- **Visually**: Users have come to expect a graphic in a roughly the same area of every page. That prominent corner is high enough to earn notice on first load without getting in the way of the page's main content.
- **Practically**: It's good for responsive web design too, because most compositions base their layout on the upper left corner, making it the most reliable real estate on screens of any size.

If these three points seem like common sense, then you've hit upon the fourth point. We use good design patterns because they're the most likely solution. And the more we use them, the more people come to expect them.

Expectations Reinforce Themselves

When was the last time you had to think about how to use a search form?

Users acclimate to a design pattern through repeated exposure. They more they see it, the more they recognize it. Eventually they come to expect it and, without realizing their savviness, intuitively know how to use it. We use patterns in part because people use them without thinking. They're frictionless.

Take the humble search form. People recognize that long rectangle with a button to the right, especially with a magnifying glass icon, will let them search a site for keywords. Bonus points if it's in the upper right corner of a page, or dead center if the page is about searching. No one has to think about what to do. It's obvious – not because of what it does, but because of what it looks like.

Turn things around for a moment to see it anew.

“It’s broken!” might be the immediate reaction, although such a form is technically valid. But users are conditioned to expect a certain convention. Even though it’s functionally the same, users need to pause for a moment and think about it. Never create that sort of unexpected friction because the more hesitations they encounter, the less likely they are to keep using your site.

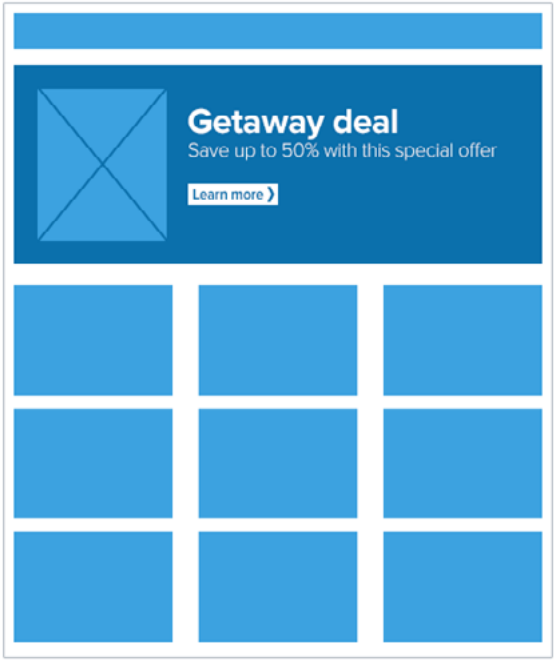
It gets even worse if you start disregarding your own work. Users get confused if you keep changing your visual voice. No matter how proven your patterns are, if you change tack in mid-stream, then users have to spend precious seconds figuring out what just happened. Imagine a search box in top right on some pages, and top middle on others. Someone looking for a hotel doesn’t want to relearn a website from page to page. They just want to find a clean room for the night.

So we use conventions and patterns to keep from confusing users. But reusing patterns has another benefit: They make you look consistent. Purposeful. Like you intend every design decision you make – when all you did was stay true to a proven solution. UI patterns help keep your design consistent within itself. When documented in a style guide, you cut down on the small mistakes that can lead to serious confusion.

Deadline-Busting Communication

If you're on deadline, patterns offer another benefit: speed of development. Whether you're making lo-fi sketches or writing final code, patterns make plugging in design easy. And when crunch time hits, "easy" translates to "fewer headaches."

Most patterns are self-contained entities you can quickly deploy in any context. A hypothetical home page might offer travelers special deals with one large image. The image spans the width of the home page. You can't miss it. It has a few choice words in large text to introduce the deal. A few smaller sentences explain the savings. A friendly button entices them to tap to book a room before it's too late.



All of this follows a familiar "hero element" pattern that most UI designers know. They know it so well that the describing it to developers is almost overkill. "Hero pic goes here" is sufficient without

explaining the details. Patterns make communication easy between designers and developers by providing a common vocabulary. If a designer says “add a modal window here,” chances are developers know what they mean without a lengthy definition of modals.

Best of all, you don’t need to look up many conventions. Just by spending time on the web, your brain has been picking up on subtle cues from the many pages it processes daily. You know patterns without knowing you know them.

Why *not* use patterns?

Patterns are great tools to expedite your work. But they also establish precedents that lull users into certain expectations. When you want to grab attention, break the mold.

- All of your icons are 32 × 32 pixels – except the important one that measures 48 square.
- Top-level headings are serif. All other subheads are sans-serif.
- Paragraphs in a blog post have 3em margins. Blockquotes have none, and so break out of the negative space established by the paragraphs.
- Images have no borders… unless they’re tappable.
- Navigation items have no background – except for the one that’s currently active.

- Your backgrounds are blue. The 404 page is red.
- All form fields have grey borders – except required fields, which are rimmed with orange.

Handy tip: You know it's time to break out of a pattern when you use the words "unless" or "except."

More importantly, you might hit the local maximum of your design if you just iterate *within* your patterns. Don't be afraid to radically break a pattern, as Rian van der Merwe suggests, as long as you thoroughly research your users beforehand and conduct A/B tests after.

Using UI patterns doesn't mean repeating what's come before. Patterns are a starting point, not an end. You can put your unique twist on a UI pattern – once you know what makes it work.

The Importance of Prototyping First

There are many reasons to skip wireframing and lo-fi prototyping. None of them are valid. Well, maybe if you need it tomorrow and plan to revise later: build first, plan second, hope for the best all the while.

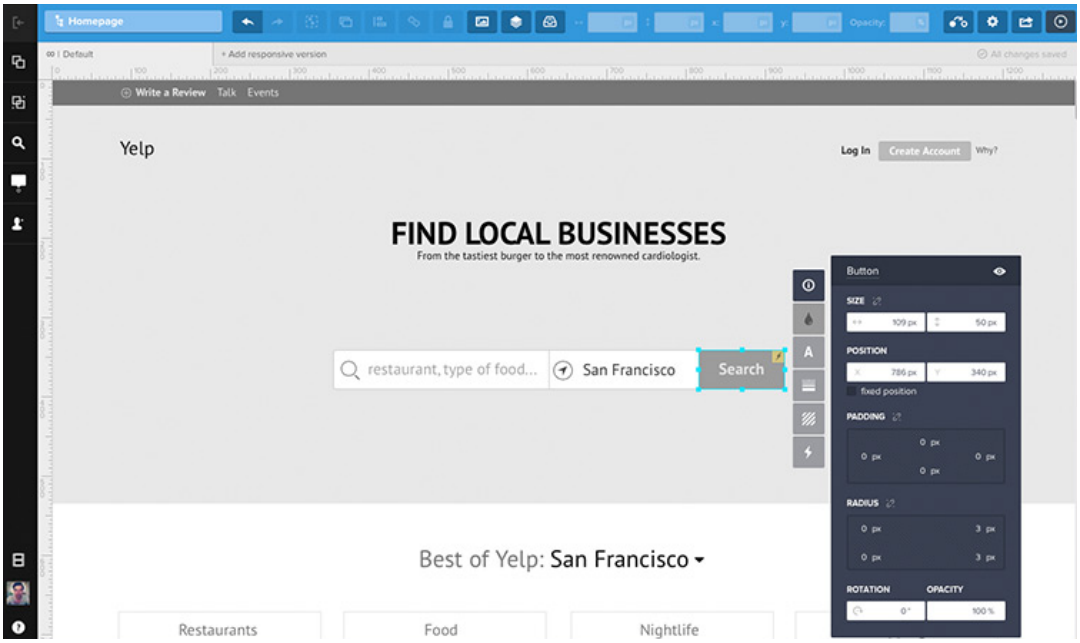


Photo credit: UXPin

But prototyping is a critical step in product design. Final products are complicated, and therefore difficult to change. Prototypes are relatively cheap to build and rearrange. And while they take effort up front, planning and testing a prototype is well worth the effort.

Prototypes create a common design language. With them, designers can seek out problems while content people can see how their work fits into the project. Meanwhile developers understand the interactions and clients can give approval or suggest changes.

Got a Pattern? Plan it Out

Design patterns aren't limited to the visual. There are accepted process patterns as well. Before we choose visual patterns, we need to seek these overall process patterns.

Take, for instance, the find-a-hotel process we've already discussed. At first glance it's not even worth mentioning. The pattern is so common that we take it for granted.

Here, we'll even spell it out:

1. Traveler arrives on the home page.
2. Traveler searches for a hotel.
3. Traveler compares options.
4. Traveler chooses a hotel.

That could even be simplified to "search, compare, choose." A loose design pattern. But it's glossing over a few critical details – details that developers can't take for granted, and details that designers should carefully consider.

A more accurate process might include:

1. Traveler arrives somewhere on the site, depending on how they found the site to begin with.
2. The traveler uses the search tool – either the ever-present menu bar search field, or the home page's more prominent "find it!"
3. The traveler taps through pages of options, somehow comparing promising hotels against each other.
4. They choose a hotel by tapping "book this one" on either its detail page or the search results page.
5. The site presents a form for their name, the number of people and nights, dates and credit card information.
6. The traveler enters their information and clicks "book now."
7. The site sends information to the appropriate hotel.
8. They receive an email confirming their appointment.

All those "ifs" and "dependings" and "somehows" are open for interpretation. Within that, we find many design decisions to make.

Should the home page say "find it" or "search now"? How many results appear per page, and how are they spaced apart? What if a given hotel has only one photo, while another has 20? Should we include steps for when the traveler wants to write a review? By what parameters do they search to begin with? And on and on…

Prototypes communicate to developers what they're building. Removing ambiguity means they have less to decide themselves. In doing so, you also uncover missing critical steps. You won't discover what those steps are, much less find creative ways to stand apart from the competition, unless you put yourself in the user's shoes and try the design yourself.

Thinking Through the Process

A funny thing happens when you prototype : you start to use the design. Unlike static mockups, you'll find yourself working both visually and interactively at the same time. This dual approach leads to all sorts of creative UI pattern decisions, like:

1. **Shopping for Hotels**

   What if we invent a hybrid view between search results and a detail page, on which the traveler can compare select hotels by price, location, amenities, etc?

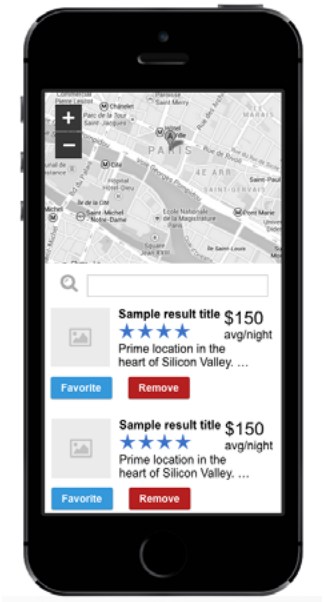
   - Checkboxes and an "add this to your options" button.
   - The ability to rank and sort favorite hotels, maybe by drag-and-drop
   - The ability to say "nope" to certain options
   - The ability to search for similar hotels based on their favorites

Check out the demo.

2. Search by Location

What if our search results page was a map instead of a list?

- Developers would need to tap into a search, most likely Google's, API.
- Choosing multiple hotels for our hypothetical comparison view might become problematic, depending on the API's options.
- Mobile users would need a way to scroll through the page, avoiding the map.



All of these design decisions come from questioning assumptions while walking through a prototype.

Patterns Take Guesswork Off of Developers' Plates

All of the descriptions so far are, obviously, text-based. Give this to a developer and he or she will get a rough idea of what to build.

But all too often, text conceals assumptions. "This page will have a search form" leaves too much to the imagination. Don't do that. It's your job as a designer to make those decisions and come up with as many possible exceptions as you can.

A hotel-finding home page has many possible configurations. It could showcase special deals, a search form, the most popular properties, or whatever else its designer feels will best serve travelers in search of a place to stay. Any of these solutions might work, depending on your priorities and restrictions. But it's not enough to tell developers. You have to show them how the patterns work.

Photo credit: picjumbo

Coders code. Implementation is their job. They don't need to obsess over how a product should behave, scrutinizing user personas, or dealing with clients' requests and design restrictions. Their concern with making a site mobile-friendly relates to media queries more than how many photos fit comfortably on a tablet. Prototyping makes their job easier by spelling out exactly how a product should work, leaving them free to dive into HTML/CSS/PHP/Ruby.

So prototypes also help designers flesh out patterns to their fullest extent. It's easier said than done, as planning a design requires much thought at the beginning of a project. That's where the decision-making should happen, however: early on, before we deliver a prototype to developers.

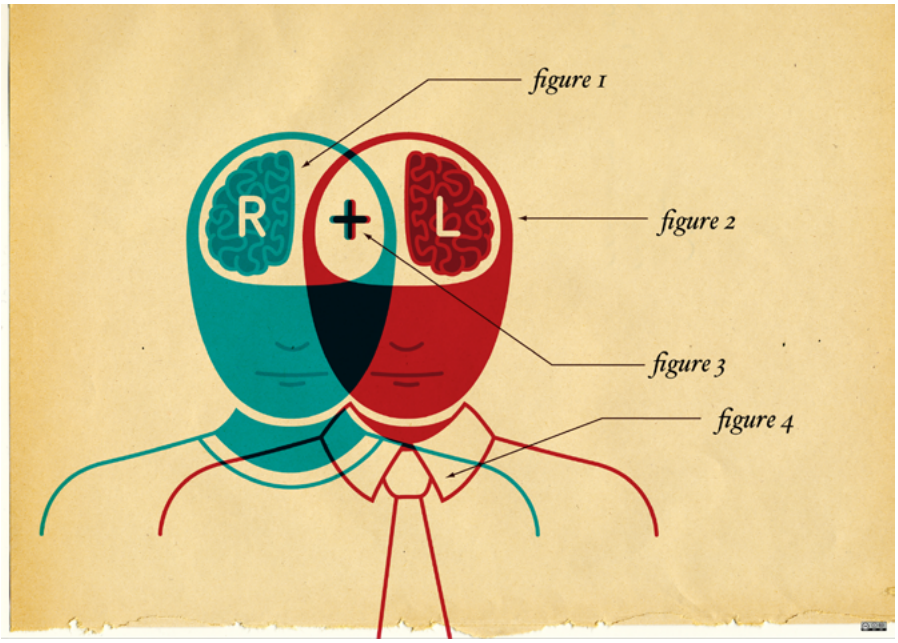


Photo credit: Open Source

What happens, for example, if a hotel's description is extra-long? How should the search results trim the text? There are many ways (patterns, of course) to solve this particular problem. Maybe the full

description deserves to appear on the results page. We might end with an ellipsis… or we might disregard text descriptions altogether.

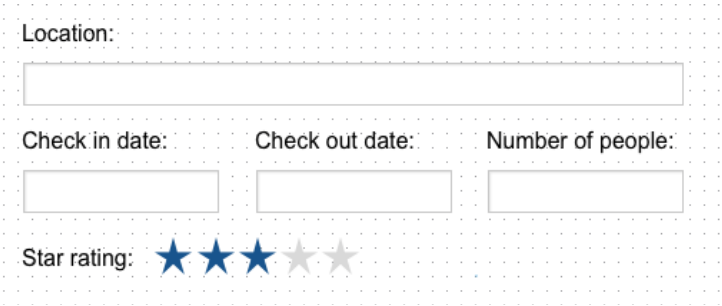
Mockups aren't much better. Comping up pages doesn't give a sense of the site's flow, robbing developers of the spirit behind your decisions. They can't see how a site hangs together – and therefore can't help you question the assumptions. Developers are a clever crowd, and we discount their brainpower at our expense.

User Testing: Insights You Can't Ignore

Few things frustrate developers like learning they've built the wrong product. That's why seeking outside opinions before coding is critical. Possibly the most important reason to prototype with patterns is to give users the chance to point out flaws in the design – before you spend time building the final product.

Remember that flaw in the process pattern we mentioned before?

By now you might have realized that a keyword search is pretty much useless. Real hotel finders let travelers search by availability, price and location. Users would point that out in a hurry, but not from static mockups. Verbally the process pattern is too straightfoward: search, choose, and select. *How* they search is as vital as the search results page's visual design. Maybe more so, since how they arrive at the results page depends on what they search for. But until they've tried it with a prototype, a user will easily gloss over such critical details.



Above: A more accurate hotel search form.

A user working through the hotel-finding app might find certain patterns are too simple. More likely they'll point out the need to search by availability on given dates, proximity to tourist attractions, smoking/non-smoking rooms, and other factors.

While those technically count as a search pattern, using a prototype would reveal necessary details. It's not enough to use *lorem ipsum*. Nothing beats testing the pattern in a real product.

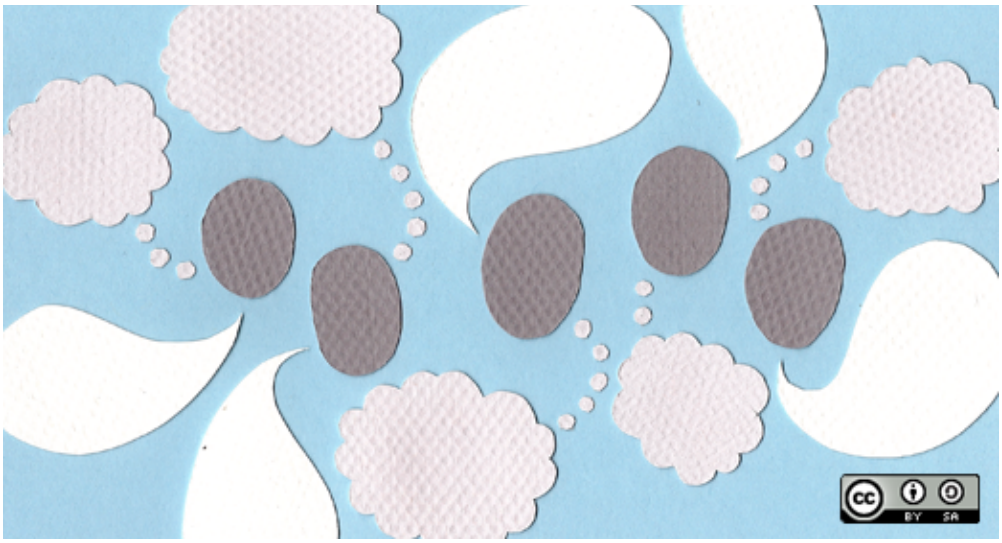


Photo credit: Open Source

There are other reasons to ask people not related to the project to test a prototype. Independently testing a product is a great chance to

find out if we're using the most appropriate pattern to solve a design problem. The trick is to get out of their way and:

- **Be prepared to abandon patterns.** Sometimes things just don't work out. Users ignore or misinterpret your carefully-crafted paths through a product. Acknowledging and fixing that is easier if your prototype is simple and flexible enough to edit in a hurry (which is why you should always test your lo-fi prototype first).
- **Watch for workarounds.** Users are natural process finders. As they become familiar with a prototype, they'll discover ways to use it that you never considered. If you stay out of the way, they'll often point out patterns and workflows that you can emphasize and fine-tune for the next iteration.
- **Consider alternatives.** Users don't always directly point out which patterns confuse them. They'll usually ask why and how, as in "why can't I… ?" and "how do I… ?" Listening to their concerns and observations is a great way to identify which workflow patterns aren't working, and which visual patterns aren't clear. Pay attention, and ask smart follow-up questions.

Usability testing is critical because other people trust that you've made the best decisions possible. Let's say developers trust your choices and build a keyword search for, in this example, a hotel's name and maybe amenities. Placing those amenities on the main search page for users to choose from might be a great idea – one that comes from spending time with actual data, as developers are wont to do.

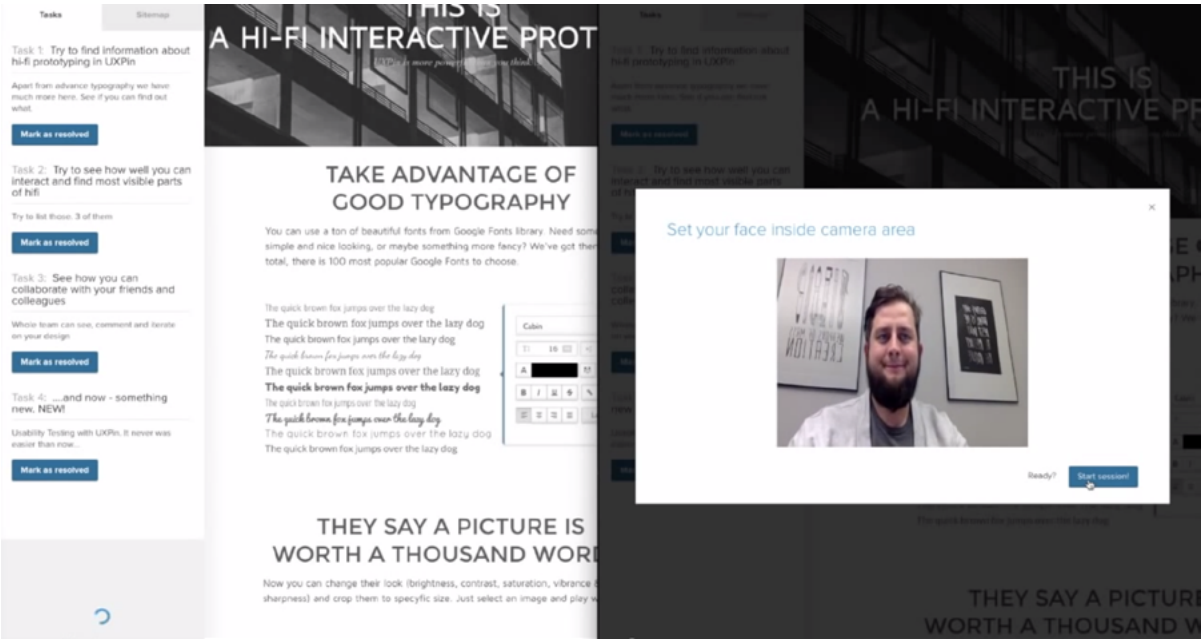


Photo credit: Usability Testing

Developers are smart people. They're like beta testers in that they can help during the planning stages, although instead of the practical parts, they think about the technical aspects of a site. But giving them responsibility to figure out things like search parameters only occupies valuable development time. They have to stop and think about the things you should have considered before dropping the project on their desk.

That's the power of prototyping first: decision-making upfront saves time for everyone in the long run.

Prototyping UI Patterns

If there's a pattern to using patterns, it lies in the tools designers use to create them.

Having a standard library of reusable, easy-to-access elements is vital because the faster we build prototypes, the faster we can iterate through them. Quick iterations in the beginning lead to more time for development. Development time translates to beating deadlines. And that means happier clients, stakeholders, and users.

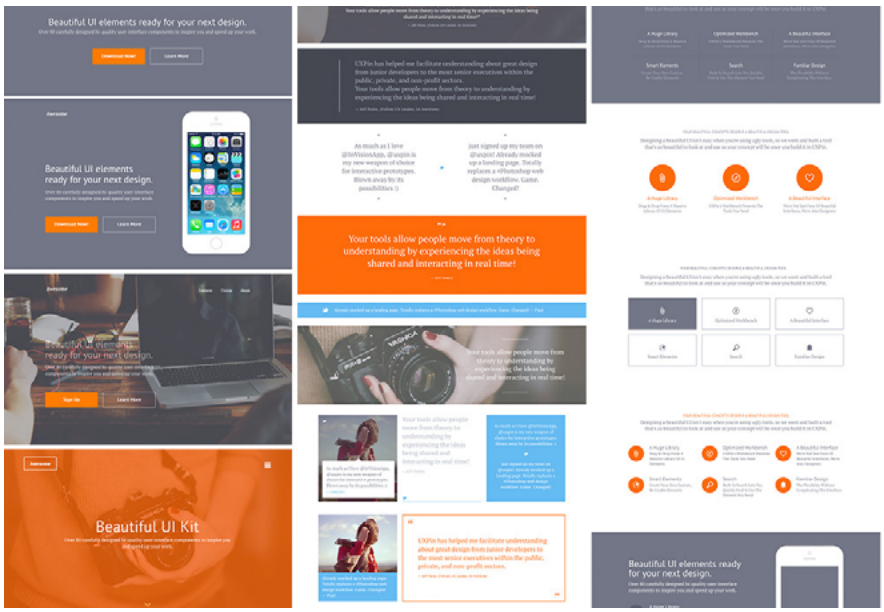


Photo credit: Free UI Kit

So it's not surprising that designers take full advantage of pre-made elements for low-fidelity and high-fidelity prototypes.

But pre-made patterns have their drawbacks, and overcoming their shortcomings takes effort and awareness of what is going on – and how ignorance of that process is actually strength.

Explaining the Gray Box

Those who use prototyping tools face a difficult challenge.

Elements in their designs must represent specific components like input form fields, text blocks, and images. They also need variety: not just "form fields" but text areas, radio buttons and single-line input fields; not just "text" but paragraph text boxes of any size and shape; placeholder images, imported photos and plain ol' "pic goes here" boxes.

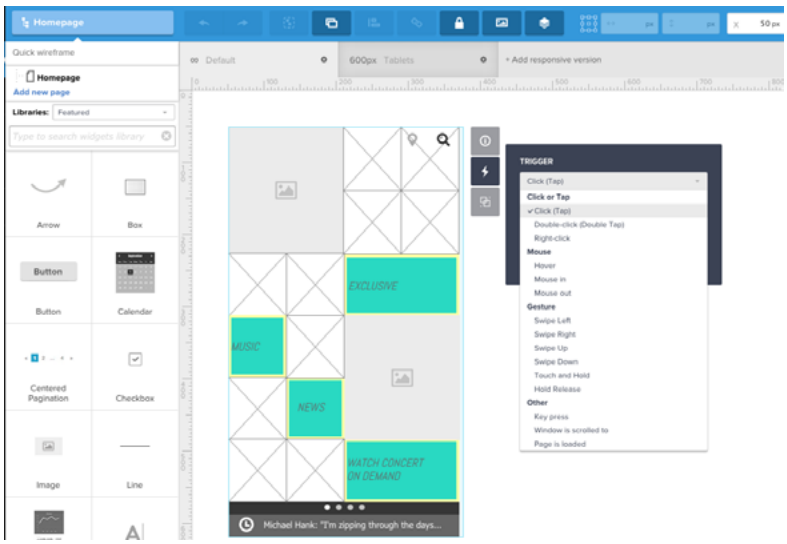
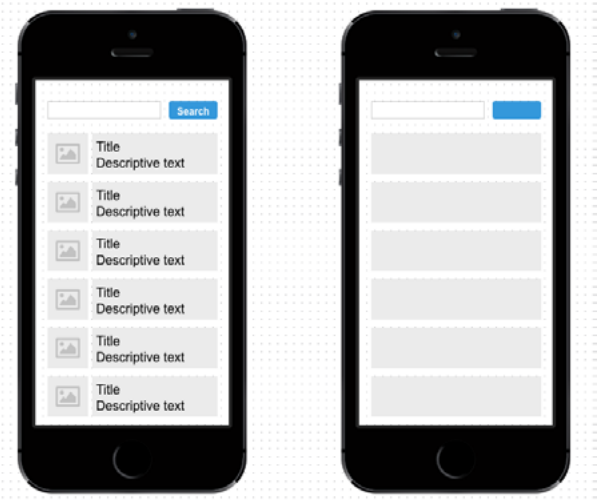


Photo credit: UXPin

The challenge is creating elements that clients and team members understand without dictating the final look-and-feel in a prototype. It's not about using the *best* fidelity – it's about using the right fidelity. Start with a lo-fi prototype so you can explore user flows and basic usability (with feedback from users and your team), then iterate into a hi-fi prototype after testing. Don't dive immediately into high fidelity, no matter how tempting the idea of visual finesse.



Above: A pattern in a lo-fi prototype with *just enough* detail (left) and somewhat lacking in detail (right) to indicate what's going on.

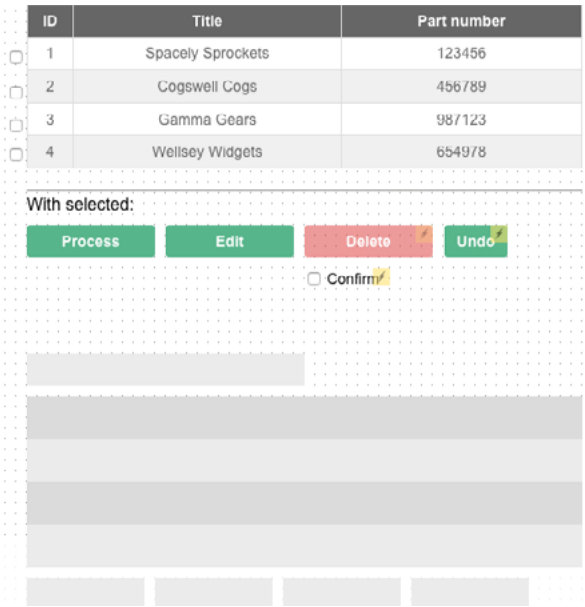
The easiest – and most common – solution is to start with grayscale boxes that may (or may not) get filled in later. Gray boxes are design-agnostic, meaning they could apply to anything. Boxes say "something will go here, we just don't know what yet" or "pay attention to the big picture, don't get bogged down in the details."

Thus most prototyping tools, including UXPin, come with many pre-made elements that could mean many things. Unadorned boxes, for example, can stand in for images or header backgrounds. Buttons

can let testers submit forms or signify calls to action. But sometimes gray rectangles are a little too vague. Even those responsible for the design can forget what every abstract shape means.

The opposite approach is to create more recognizable, less generic objects. Radio buttons and checkboxes don't have to resemble each other. Browser windows can have round corners on the outside, and sharp corners inside their frames. Text of varying sizes sets headings apart from paragraphs, and paragraphs apart from calls to action. The more specific an element is, though, the less likely it will get used. Not every every project will need, say, a calendar or video window.

The best prototypes bridge the gap that includes generic and specific objects. These objects become patterns as people learn to recognize them for what they are. Patterns emerge from understanding what, exactly, each element is without spelling it out. The prototype is therefore fast to create and provides just enough detail for users to focus on accomplishing tasks.

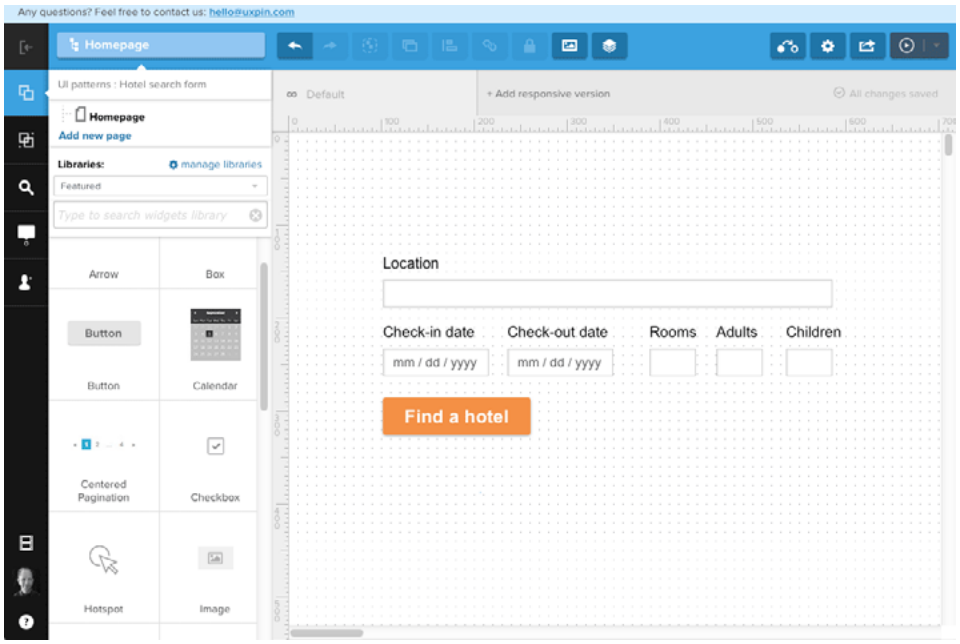


Above, a more-specific object (top) communicates its purpose better than an abstract set of unlabeled shapes (bottom).

Pattern Libraries Are Prototyping Shortcuts

Prototypes are built with a set of patterns, often called libraries, that contain elements designers don't have to invent. At the time of this writing, UXPin has more than 900 of them.

It's easy to drag a pattern into a design and discard it later. This allows designers to focus on solving problems without gnashing their teeth over visual design details. Instead they can design for user actions like whether or not to let travelers give hotels a star rating, or if that's overkill. Designers should make decisions, not patterns.



In programming terms, patterns are instances of a class; in practical terms, they're shapes in a sidebar you drag into your design. Make

as many as you need of the same thing. “What if we have a calendar here?” designers ask, and drag-n-drop one into place. Patterns are disposable. Prototyping tools make them inexpensive.

Let’s look how UI pattern libraries help you prototype faster while making the design easy to understand.

Reusable elements

The mockup above recreates a standard search pattern with a twist: This one’s geared for finding a hotel. It includes some technical (and practical) requirements like check/out dates, number of people and location.

Notice that the whole design uses only three kinds of elements: Text, input fields, and a single button. They’re instantly recognizable for what they are. Not only that, but they’re arranged almost identically. A text label is set atop each input field, making them easy to copy/paste throughout the prototype.

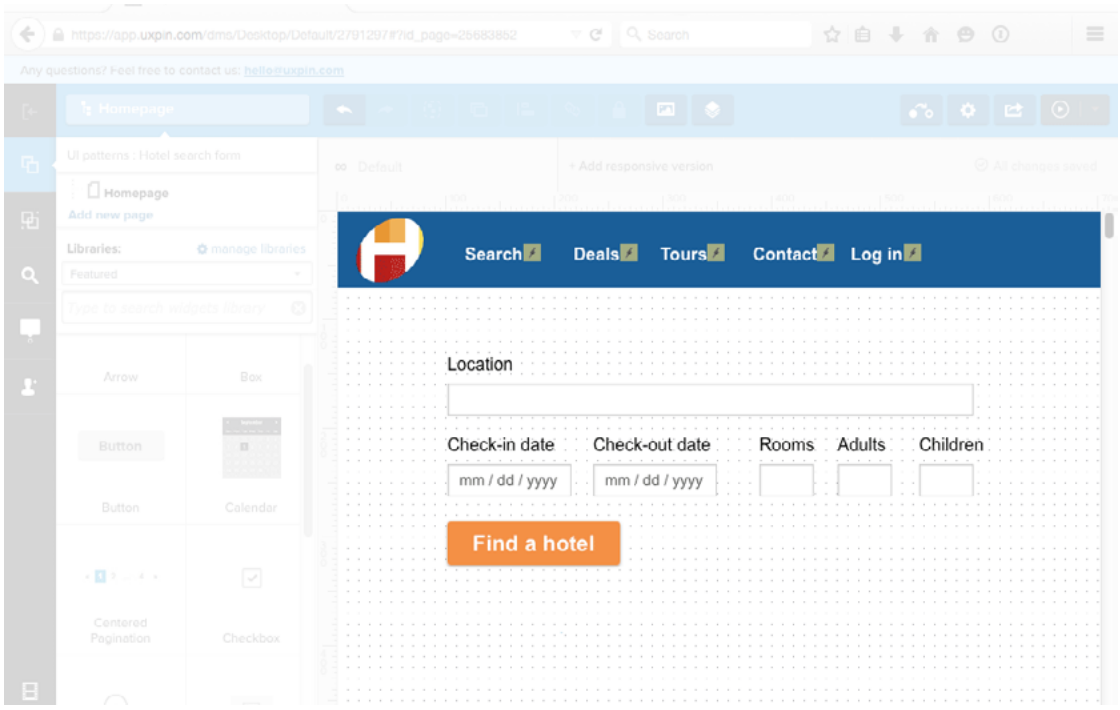
In the beginning stages of design, visual details don’t matter as much, so it’s more important that the design structure makes sense (which UI patterns help).

1. **Custom libraries**
   One-off elements are quick to create and customize. Whenever you need to repeat yourself precisely, though, you need some smarts.

Certain parts of a project tend to remain the same on every screen but, like anything in development, change over the course of the project. For example:

- Site-wide headers
- Primary navigation bars
- Tappable logos
- A standard copyright statement

Look sharp and you'll notice a pattern in these elements. Each is part of the overall website design and branding. In addition, some parts – particularly the navigation – may change on multiple wire-frames throughout a project.



Here, tappable links in a navigation bar cap the entire design. Users would expect such an element on every page of the site, reminding them which site they're on and letting them jump from page to page. And then things change.

Let's say you decide to add a new page to the primary navigation bar that's already on, say, 30 wireframes. To prevent the tedium of editing each page individually, "smart" elements update every page on which they're placed. Changing the navigation in one place changes them everywhere.

Using this approach – drag-n-drop elements and smart elements – setting up an entire flow is easy. For demonstration purposes, this quick example only mocks up two pages. But with a little planning, the process scales to projects of any size.

Patterns and Prototypes Work Together

Patterns and prototypes complete each other. You can't have one without the other… unless you don't mind creating everything from scratch. But even then you'll find yourself recreating what's come before.

Prototypes must communicate the solutions that designs bring. Their patterns must be recognizable as distinct elements – for example, that calendar widget – without declaring that its final look will be a particular style, color or typeface.

As you work, find a good tool that becomes an extension of yourself. The quicker you forget that you're using it, the quicker you'll iterate through the design process.

Applying UI Design Patterns

You've seen patterns before. You're probably looking at some now. And you're inundated with them every day: Layouts and icons, symbols and workflows. The bulk of web design is built on patterns.

Sometimes they're obvious. Design agency sites tend to mimic each other, following broad trends as technology advances and trends change. News sites that experiment with article teasers' formats still use them. And shopping carts tend to work the same way.

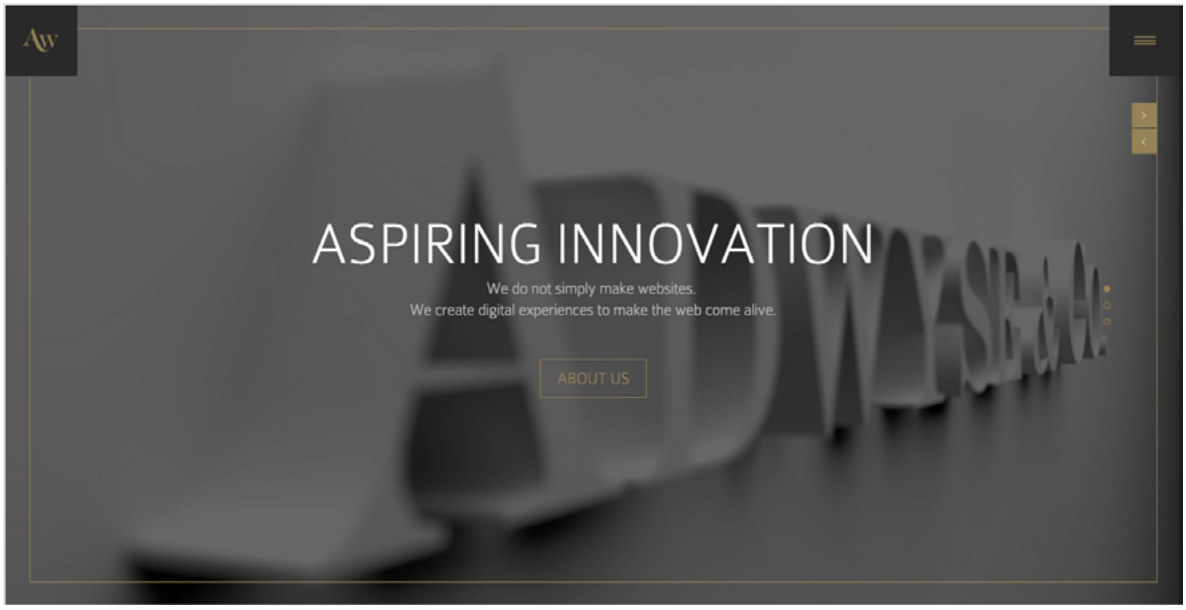


Photo credit: Adwyse

Other times they're more subtle, like variations in sites' footers, or the (growing) use of "hamburger" icons which, at the time of this writing, most readers don't quite get. Or scrolling sideways instead of down through a website. It's technically possible, but unexpected.

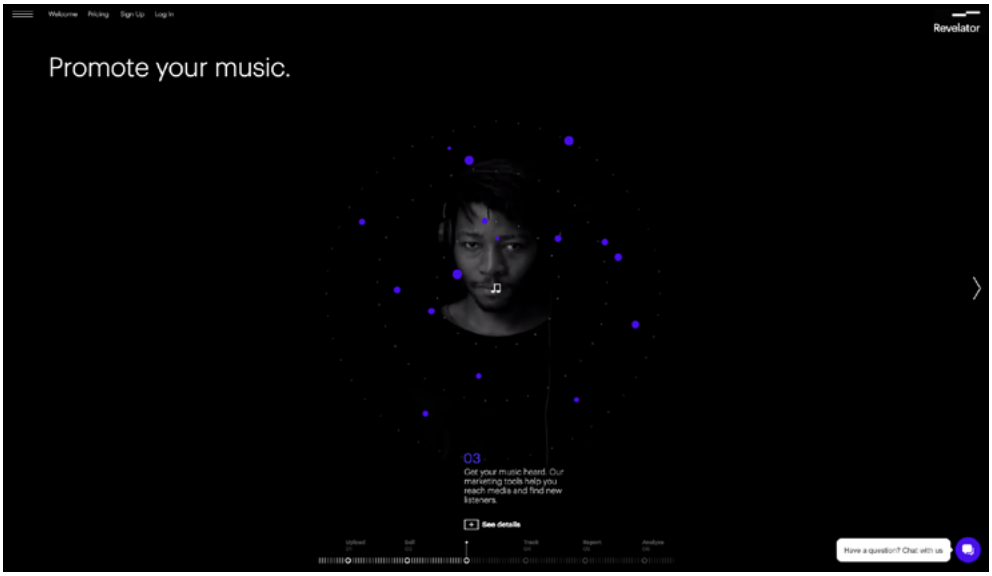


Photo credit: Revelator

Whether they're easy or not, it's important to recognize when you're using patterns in prototypes. Knowing conventions that users expect – and knowing when to break them – will make your designs easier to use and stronger overall.

Building a Pattern Library

Unfortunately, keeping patterns in your head isn't a great way to manage ideas. Assumptions creep in, and not having a visual reference makes patterns difficult to share. Creating a pattern library is so important that entire sites like UI-Patterns and PatternTap exist as design references. But you can also make your own.

Step 1: Look for design patterns in the wild. Here's one way how.

- Go to any website and look at the "big picture." Notice its overall layout: Do you see columns and sidebars?
- Set yourself a goal in a website that requires several steps. Jot down the steps you took to get to that goal.
- Look back at patterns involved in each step.
- Examine macros, shortcuts, templates and other techniques you've used across multiple projects. Take one of each and archive them.

That last item underscores the point. The key to pattern-hunting is to start with a running list of patterns: a repository of ideas from which you can pull as needed.

Common patterns include:

- Tabbed menus
- Off-canvas navigation
- Dominant headers above columns
- Pagination links
- Breadcrumbs
- Clickable logos in the upper left corner of a page
- Footers with navigation links
- Name/email fields above message text areas in contact forms
- Search fields at the top of a screen
- Horizontal navigation links

- Blue text, often with underlines, are clickable
- Large text is more important than small text

But don't stop there. Keep records of anything you see repeated across websites – on paper, or digitally with UXPin (by creating a "Pattern Library" project and dropping in screenshots) or even in a Dropbox folder.

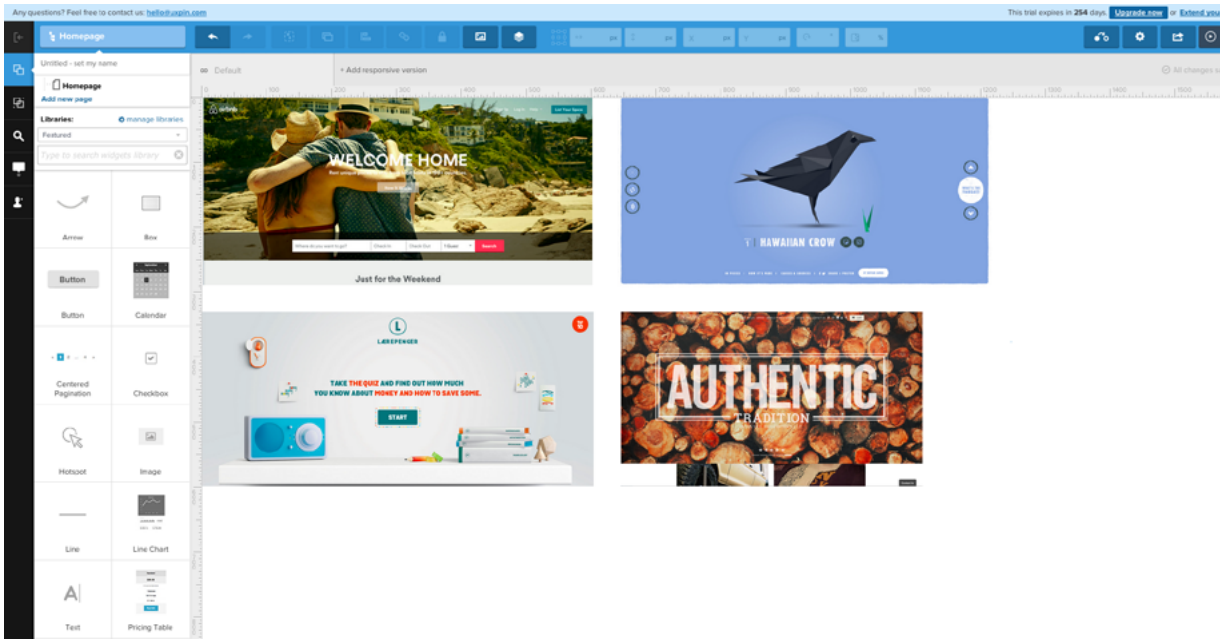


Photo credit: UXPin via Airbnb, Noveske, Species in Pieces, Laerepenger

Step 2: Start to use them in your work. Not the examples themselves, but the patterns behind them.

It's OK to use breadcrumbs. It's less OK to use red breadcrumbs separated by a particular icon that you especially liked somewhere on the web. Remember that patterns are ideas upon which you build your unique aesthetics. (That's not to say you must avoid every look-and-feel you've ever seen. Creating something totally original is nearly impossible today, so seek inspiration from your project's branding and don't worry if "it's been done.")

Step 3: Share your findings with your team.

This resource you're creating benefits everyone. Fellow designers can use patterns for the same reasons you do. Developers will come to recognize the spirit of your designs, not just the surface details. Stakeholders can better understand your design rationale by recognizing similar examples.

Finally, keep at it. Libraries don't happen overnight. As styles and trends change, so will your collection of examples – though we bet that the ideas will remain largely the same. It takes a game-changing UI to invent new patterns.

Riffing on Design Patterns

A pattern library is more than a scrapbook of interesting aesthetics. It's also a foundation on which to create your own.

As an example, let's look at a common pattern for search results. When people browse search results, they mentally eliminate choices that aren't compatible as they zero in on the perfect match. Normally, that means the primary CTA is to click on the link.

If you were building a results page for a hotel, you need to filter that pattern through user behavior. Instead of browsing down a list linearly, comparative shoppers might move back and forth between options (evaluating based on price, location, decor, etc.).

In the below example built in UXPin, what if you flipped the pattern around so the primary CTA was to remove links that users didn't want?

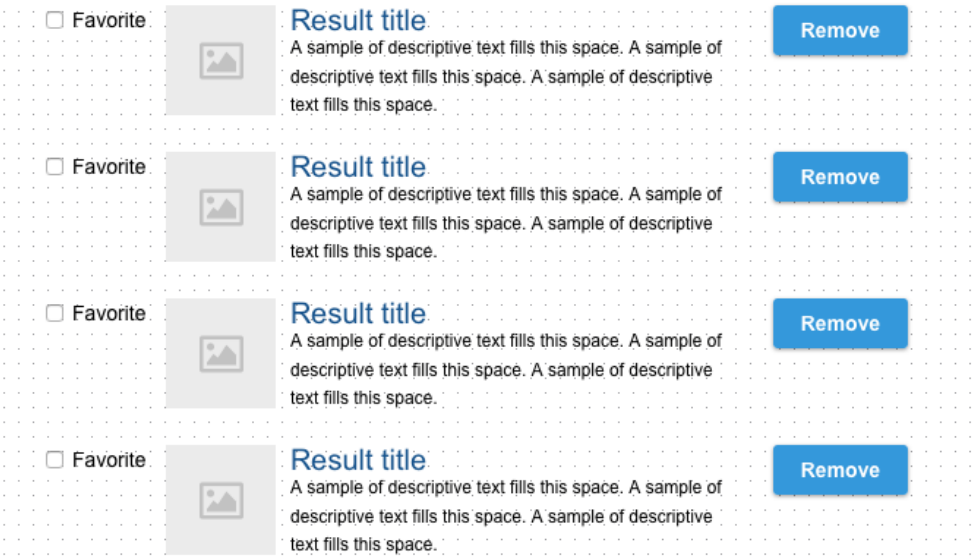


Photo credit: UXPin

You don't break the idea behind search results, but you riff on the idea, implementing new solutions based on the user context. You can't think sideways unless you fully understand the solution a pattern represents.

Of course, we don't know yet if this new pattern will work, but perhaps it's worth testing alongside the original pattern. It might fail horribly, or it might work better than expected because it's so unconventional.

Let's adopt a similar experimental mindset with breadcrumbs. They seem like a simple pattern without much room to grow… unless you get a little creative. What if:

- Their links were listed vertically, in a column of their own?
- We used icons instead of words?

- We put breadcrumbs at the bottom of the page instead of the top?
- Breadcrumbs included the next most likely link?
- Each link included a drop-down menu of its siblings?

Not every brainstormed idea is a winner, but may eventually lead to something great. Don't be afraid to think outside the pattern – just make sure you test it with at least 5 users before you fall in love with it.

Tweaking Pattern Styles

You've chosen a pattern. You've tested it out. Now it's time to give it a look. How does that work?

Falling into a rut is easy with patterns when they're thorough. But if we use good reasons to change them – when necessary – then our design will come out ahead. We'll describe a couple ways below.

1. **Incorporate brand colors and typography**
   Every project has certain attributes that make it seem like a coherent whole. Color and type choices are frequent tools to accomplish that, and easy to apply to patterns. You can use patterns to reinforce that brand:
   - Colors, like we said, but especially shades of the same hues to reinforce the overall look.
   - Distinctive shapes and geometry.

- Type families and various weights.
- Follows the page's layout grid as much as possible.
- Illustration style

2. Give it a little flair

As we mentioned before, breaking patterns (or at least bending them) will make certain use cases stand out from the rest. The more you use regular patterns, the more differences will stand out. The trick is to identify and question assumptions. For example:

- Must dates in a calendar reside in boxes? Or would circles work as well?
- When would you use serif typefaces in a sans-serif dominated design, or vice-versa?
- What if buttons had alternating round and sharp corners?
- What if testimonial quotes substituted for headlines?
- Should switches be horizontal or vertical?
- Do star ratings have to use stars?

3. Use them as-is

Alternatively, sometimes the simplest solution is best. Patterns that solve a problem do so because they're well-designed.

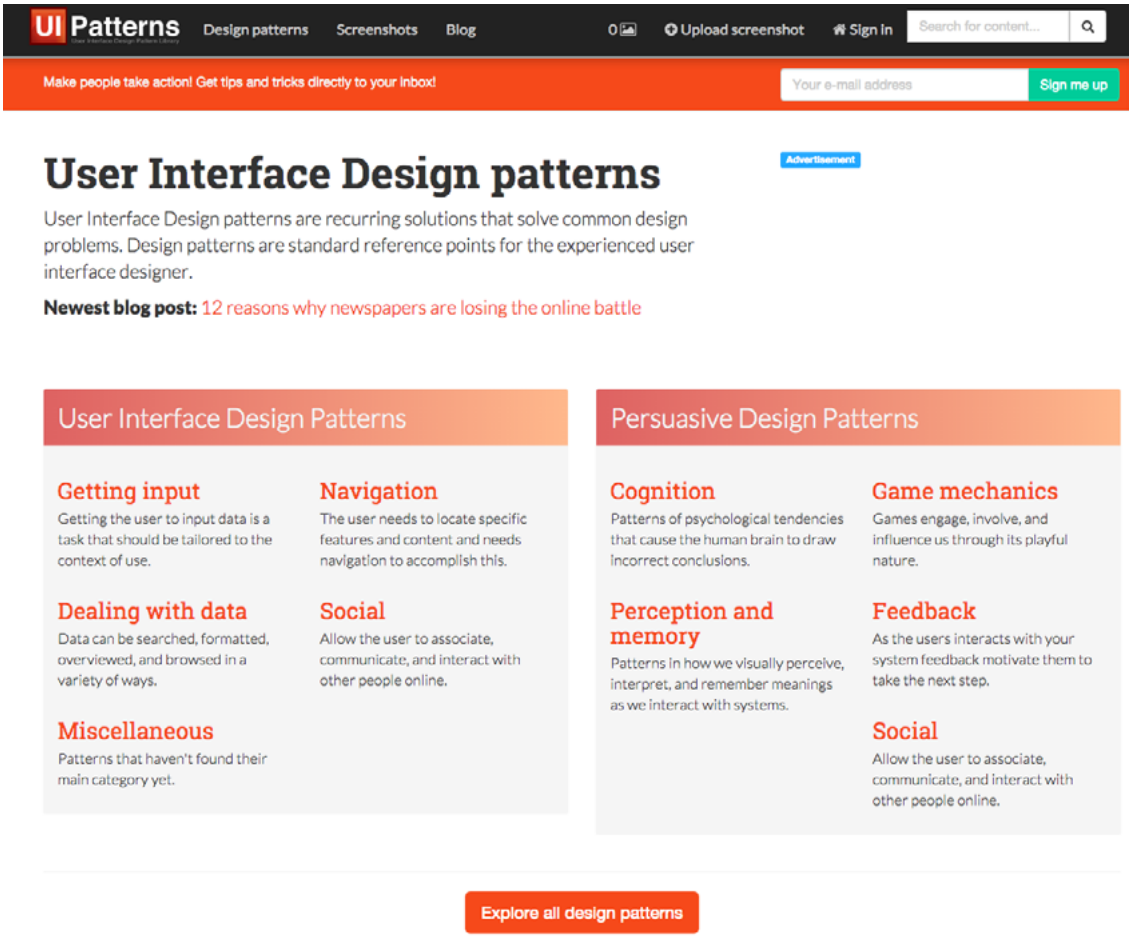


Photo credit: UI Patterns

Buttons, for example, may not need extra frills and colors. Users are accustomed to certain patterns looking a certain way, and if there's no good reason to change them, then you'd be smart to leave them alone. Good reasons include:

- Drawing attention to areas you want users to notice, like calls to action.
- Indicating that something is *less* important by making it resemble the background.

- Make something contextual.

Bad reasons include:

- Because it's "artistic." We should not break our own rules for the sake of art, but for communication.
- We just want to stand out. Don't waste energy satisfying your designer ego.

The most important reason to apply your project's style to a pattern is also the trickiest: not to look like a pattern while retaining its spirit. A table can look like anything, as long as it works like a table. Buttons still need a tappable quality about them. Links need to stand out from text. Otherwise you have free rein to riff on patterns with your own creative ideas.

Going forward

The web's history is full of patterns, from clickable corner logos to vertical scrolling to underlined links. Designers have learned to take advantage of how people recognize trends in visuals and interactions, and users have come to expect certain conventions in web UI design. It's a win-win situation that only gets better over time – even as it evolves.

The rest of this workbook explores common patterns and their variations. If you want to start practicing, feel free to get started with hundreds of elements built into UXPin. You'll find UI elements for Foundation, Bootstrap, all major mobile devices, tablets, and more added regularly. You can even create your own custom reusable patterns with Smart Elements.

Design better web experiences with UXPin (free trial)

Useful UI Pattern Examples

Some patterns are more common than others.

In our research for this e-book, we discovered that we could categorize patterns into both broad and specific groups. The more we thought about it, the more we realized that we'd been using certain types of patterns from the start.

Where would we be without navigation? Forms and getting user input are more common than ever as people use the web as a tool to do anything from ordering food to personal banking. And as websites become increasingly complex, we've needed new ways to view the data on screen.

Styles varied and applications got creative, but by their nature, patterns persisted across most every website we saw. Based on what we found, here are some handy examples for your reference.

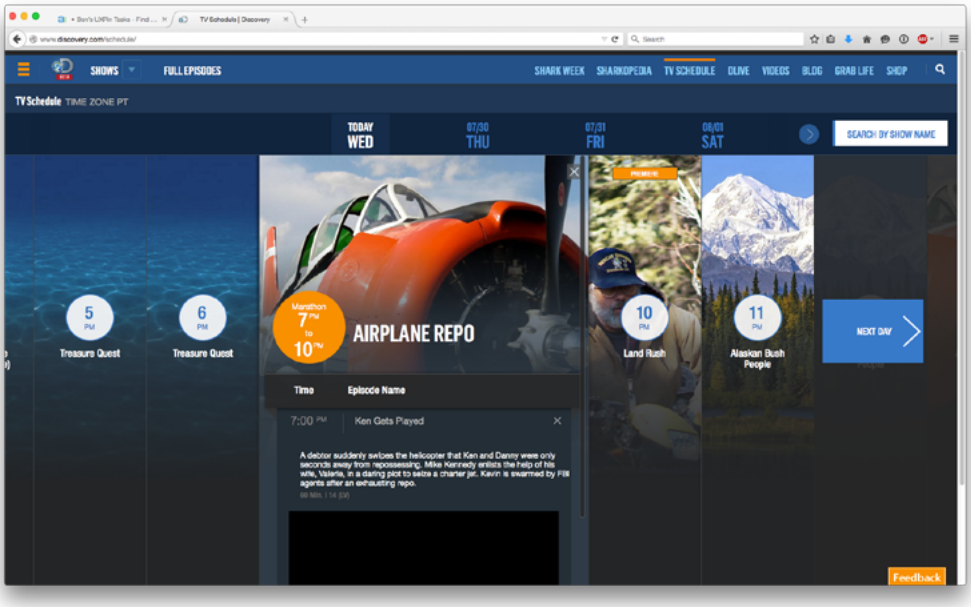
Of course, we're only exploring the core UI patterns to any site. If you'd like a visual exploration of almost every type of web UI pat-

tern, check out the free e-book *Web UI Design Patterns*. You'll find 63 examples explained in a problem/solution format so you know which ones to use.

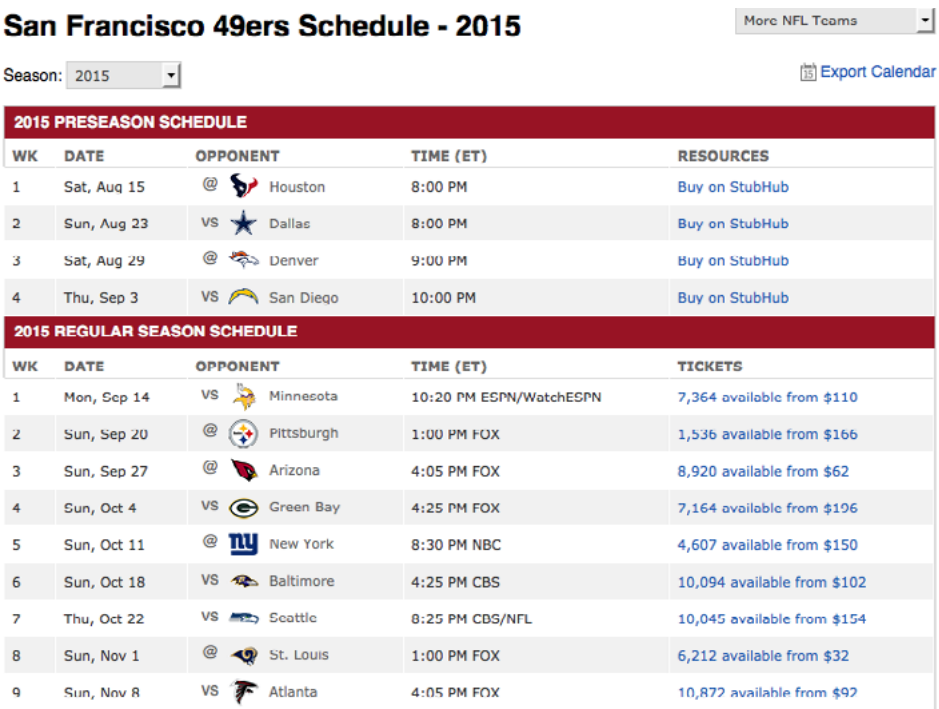
Formatting Data

Calendars

1. Discovery Channel



2. ESPN



San Francisco 49ers Schedule - 2015

More NFL Teams

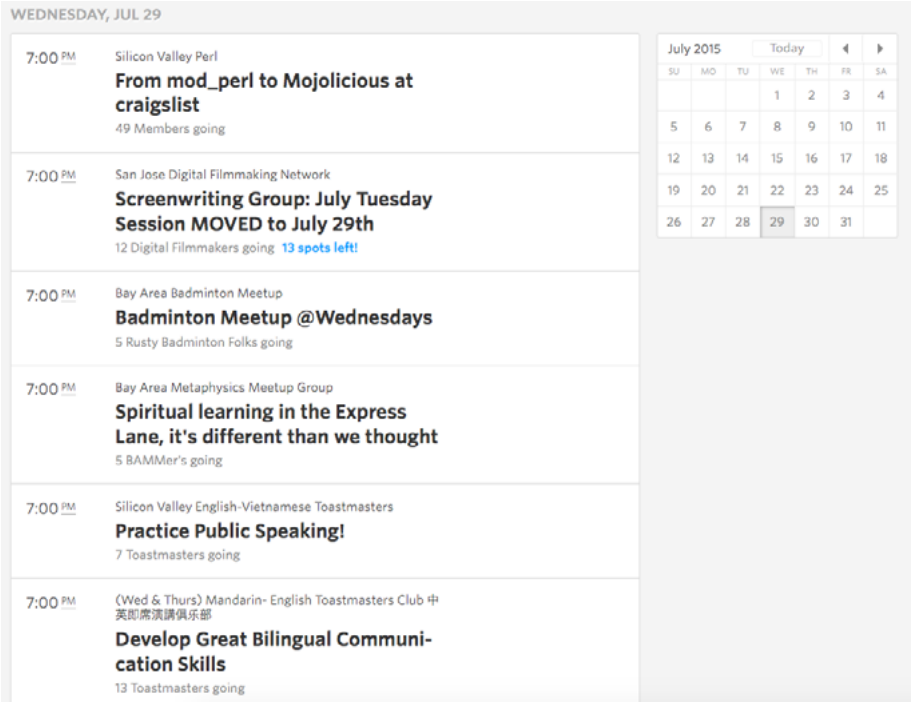
Season: 2015

Export Calendar

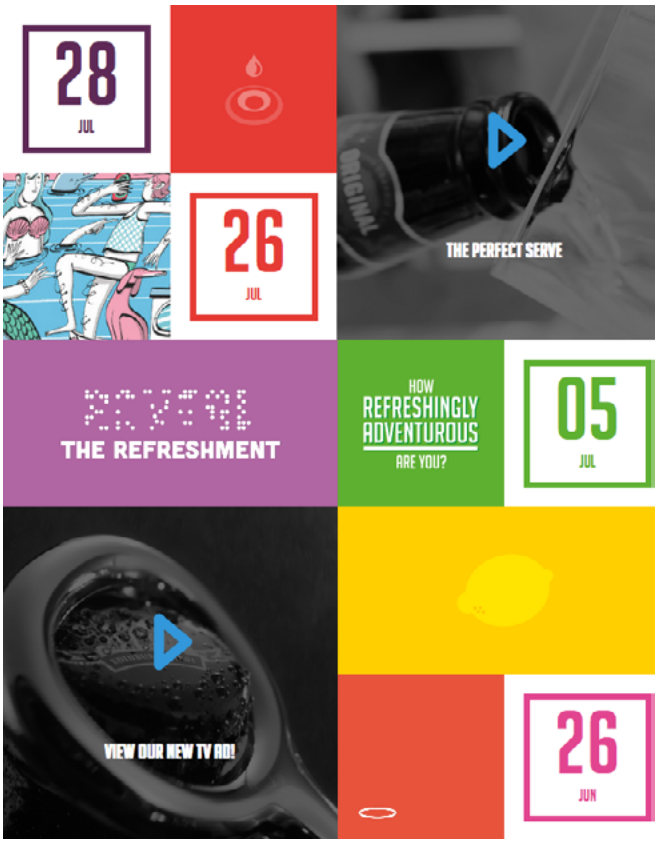
2015 PRESEASON SCHEDULE				
WK	DATE	OPPONENT	TIME (ET)	RESOURCES
1	Sat, Aug 15	@ Houston	8:00 PM	Buy on StubHub
2	Sun, Aug 23	vs Dallas	8:00 PM	Buy on StubHub
3	Sat, Aug 29	@ Denver	9:00 PM	Buy on StubHub
4	Thu, Sep 3	vs San Diego	10:00 PM	Buy on StubHub

2015 REGULAR SEASON SCHEDULE				
WK	DATE	OPPONENT	TIME (ET)	TICKETS
1	Mon, Sep 14	vs Minnesota	10:20 PM ESPN/WatchESPN	7,364 available from $110
2	Sun, Sep 20	@ Pittsburgh	1:00 PM FOX	1,536 available from $166
3	Sun, Sep 27	@ Arizona	4:05 PM FOX	8,920 available from $62
4	Sun, Oct 4	vs Green Bay	4:25 PM FOX	7,164 available from $196
5	Sun, Oct 11	@ New York	8:30 PM NBC	4,607 available from $150
6	Sun, Oct 18	vs Baltimore	4:25 PM CBS	10,094 available from $102
7	Thu, Oct 22	vs Seattle	8:25 PM CBS/NFL	10,045 available from $154
8	Sun, Nov 1	@ St. Louis	1:00 PM FOX	6,212 available from $32
9	Sun, Nov 8	vs Atlanta	4:05 PM FOX	10,872 available from $92

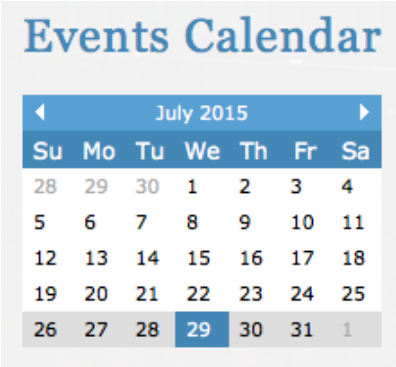
3. Meetup.com



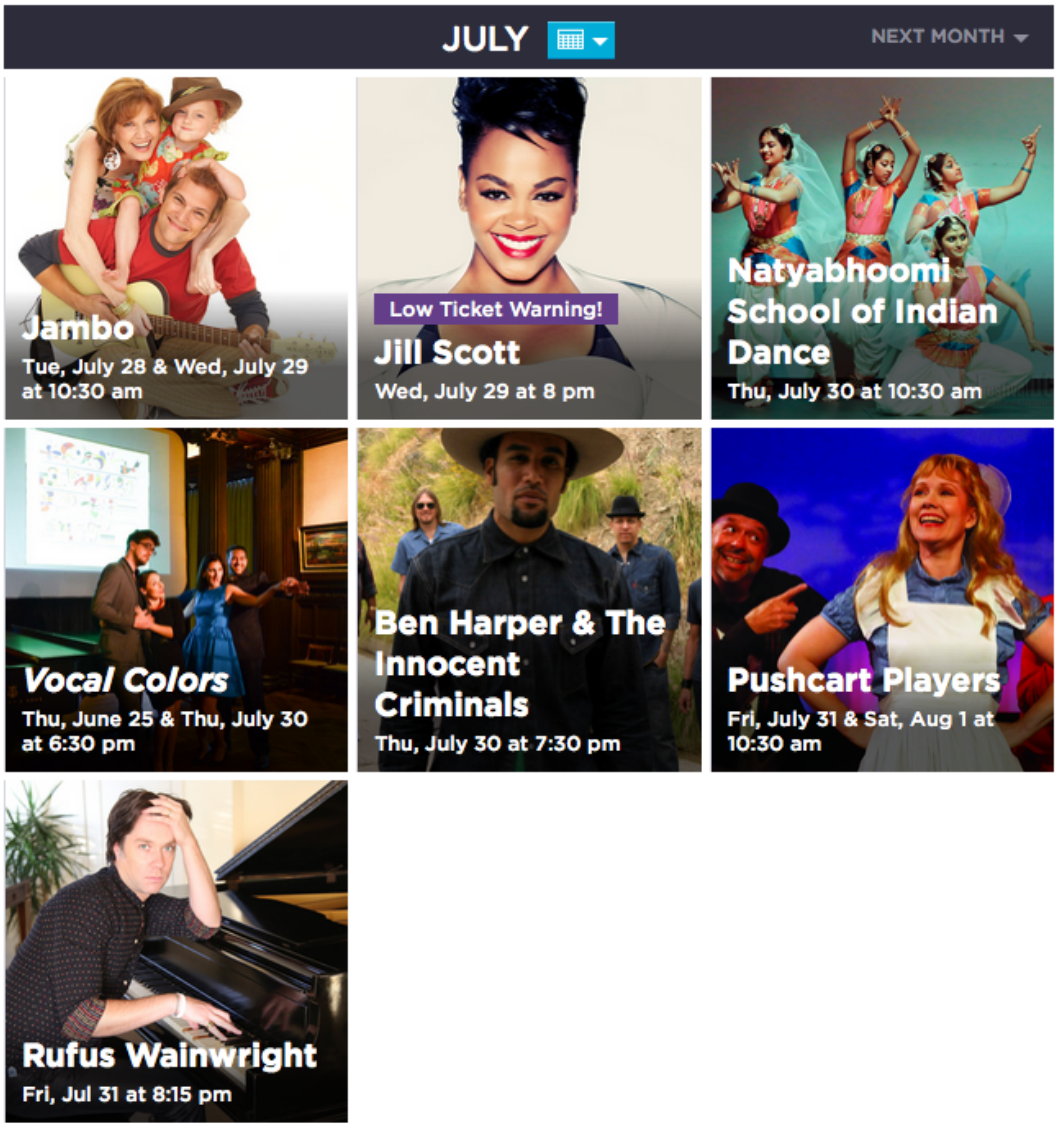
4. Crabbie's



5. University of North Carolina

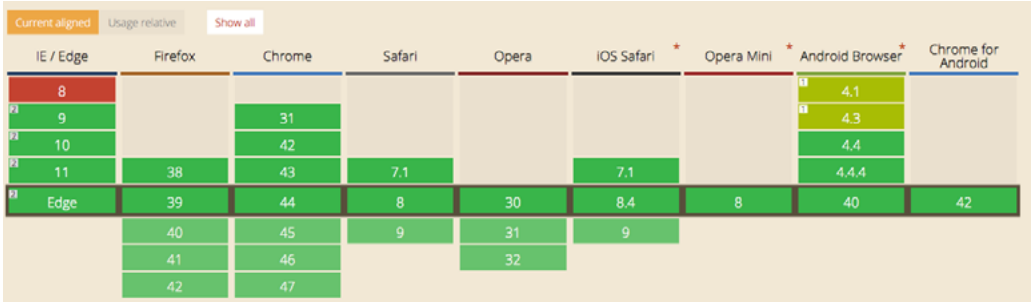


6. Wolf Trap



Tables

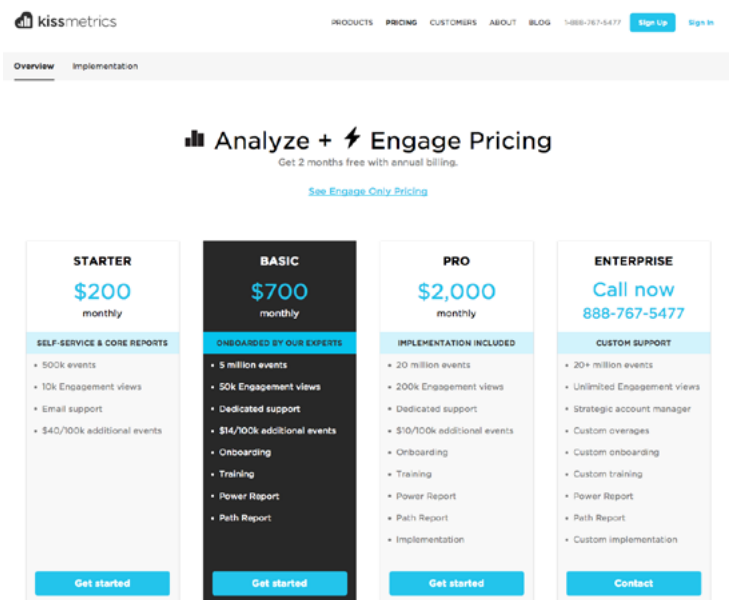
1. Can I Use



Current aligned | Usage relative | Show all

IE / Edge	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8							4.1	
9		31					4.3	
10		42					4.4	
11	38	43	7.1		7.1		4.4.4	
Edge	39	44	8	30	8.4	8	40	42
	40	45	9	31	9			
	41	46		32				
	42	47						

2. KISS Metrics



3. ESPN

2014-15 Pacific Standings

TEAM	W	L	PCT	GB	STRK
Golden State	67	15	.817	-	W4
Los Angeles	56	26	.683	11	W7
Phoenix	39	43	.476	28	L5
Sacramento	29	53	.354	38	W2
Los Angeles	21	61	.256	46	L3

Full Standings

4. Vox Media

Curbed	Curbed NY Editor	NY
	Curbed SF Editor	San Francisco office
Eater	Community Manager	New York
	Creative Producer, Digital Video	New York, NY
	Eater New York Reporter	New York
	Eater Social Media Editor	NY
	Video Creator, Social	NY
Polygon	Engagement Editor	NY
Racked	Essays and Op-Ed Editor	NY
	News Editor	NY
	Staff Writer, Pop Culture	NY

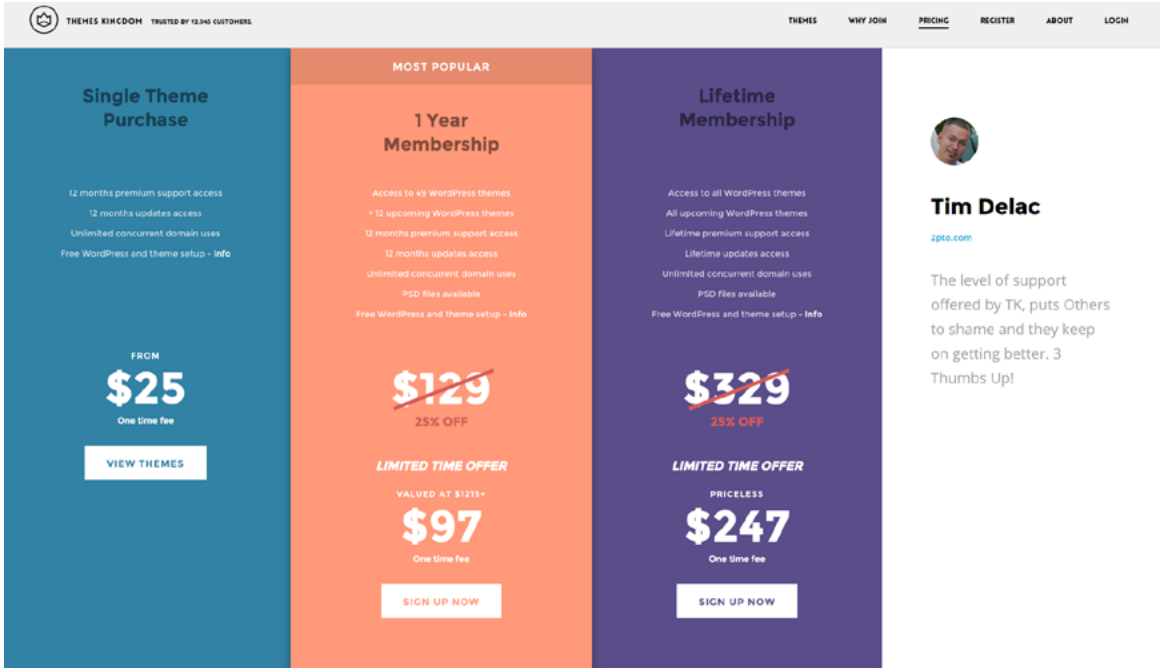
5. Wufoo

Plans & Pricing

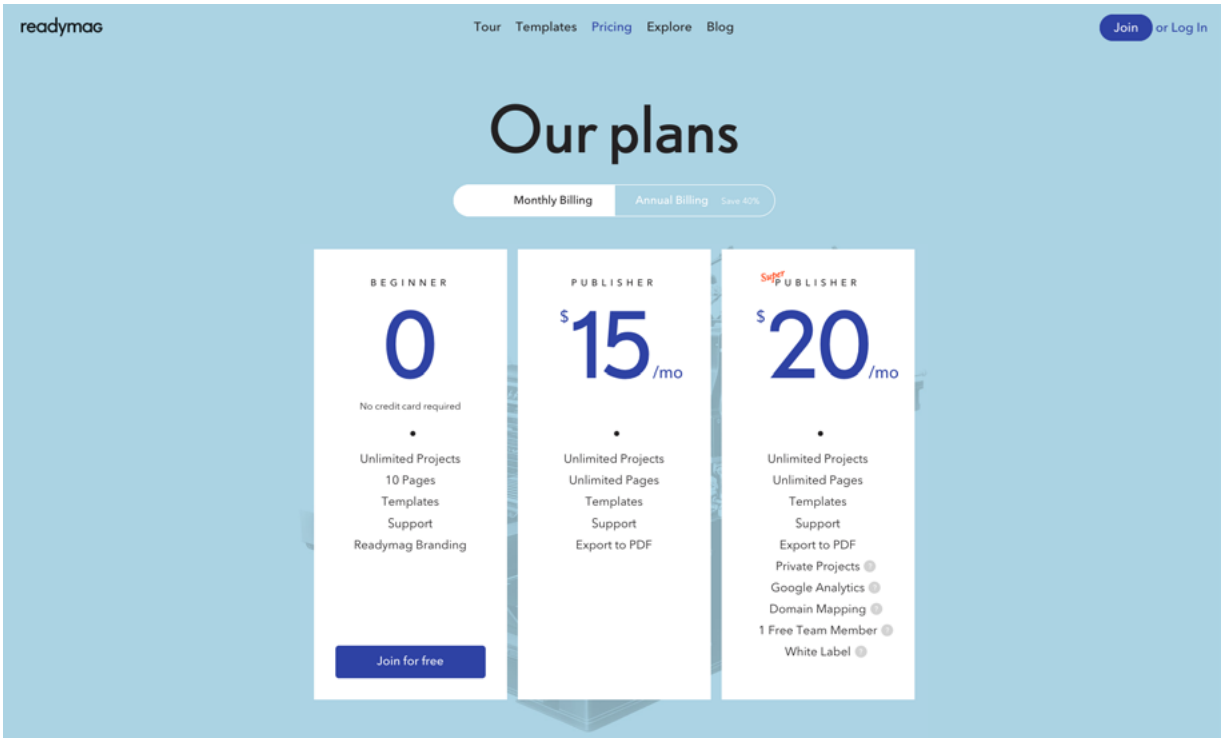
monthly annual

GRATIS	AD HOC	BONA FIDE	CARPE DIEM
FREE Forever	$14.95 month	$29.95 month	$69.95 month
	Go Annual, Save 25%! →	Go Annual, Save 25%! →	Go Annual, Save 25%! →
SIGN UP!	SIGN UP!	SIGN UP!	SIGN UP!
3 Forms	10 Forms	Unlimited Forms	✓
100 Entries	500 Entries	3,000 Entries	15,000 Entries
10 Fields	100 Fields	Max Fields*	✓
1 User	1 User	5 Users	20 Users
3 Reports	20 Reports	Unlimited Reports	✓
-	250MB Uploads	1GB Uploads	3GB Uploads

6. Themes Kingdom



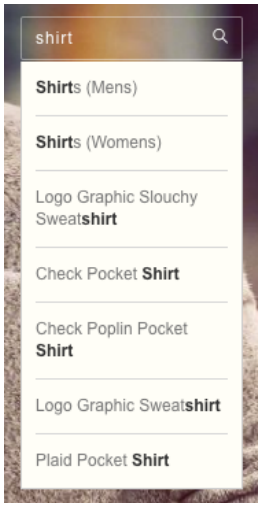
7. Readymag



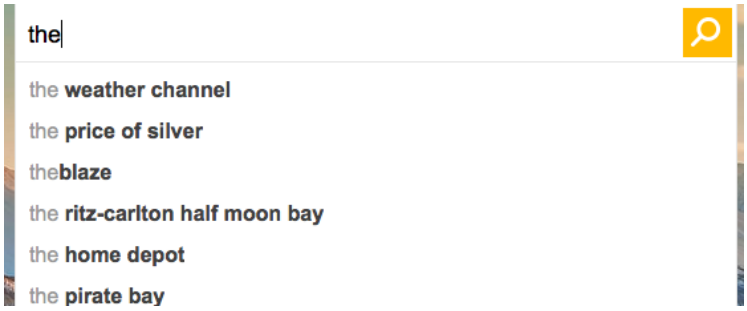
Getting input

Autocomplete

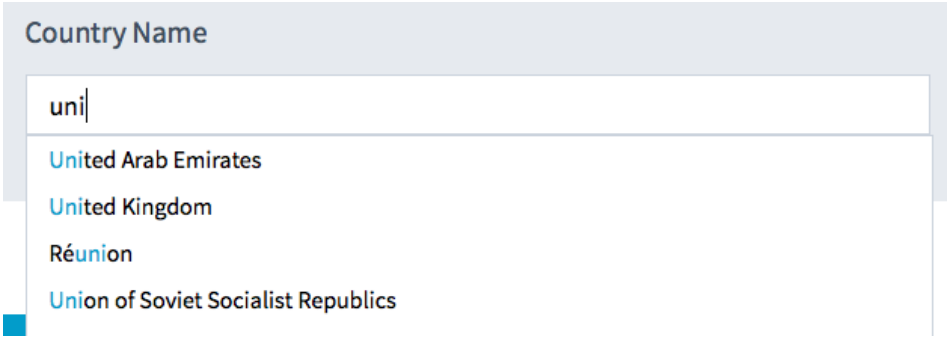
1. Abercrombie & Fitch



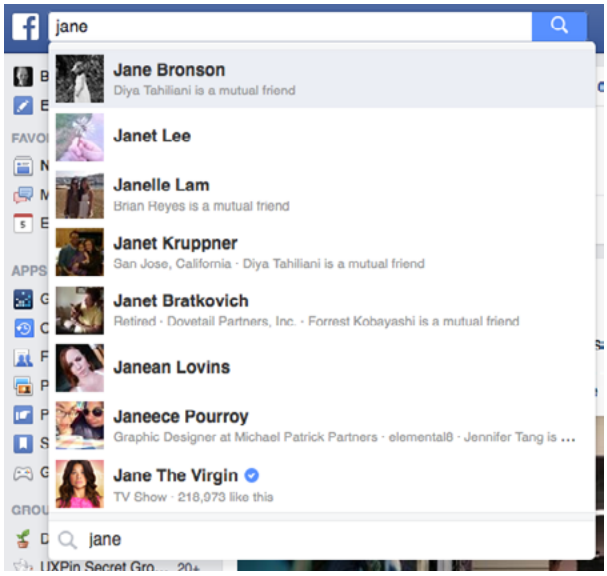
2. Microsoft Bing



3. Devbridge Group

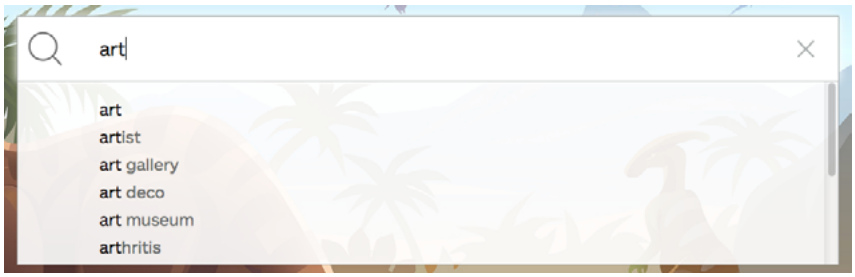


4. Facebook

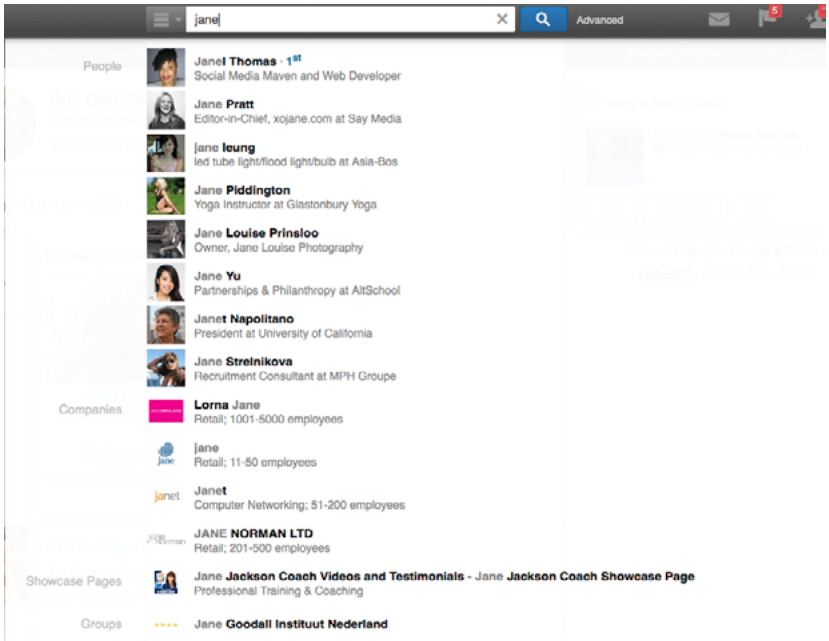


5. IMDb – Internet Movie Database

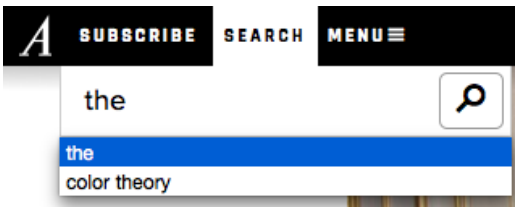
6. iStock



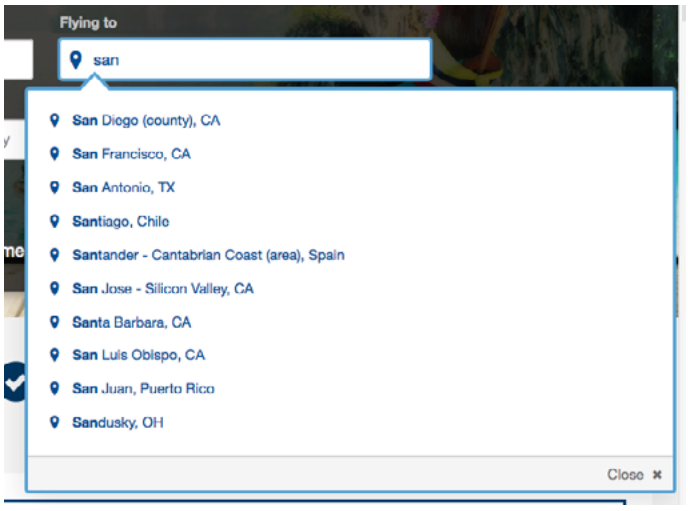
7. LinkedIn



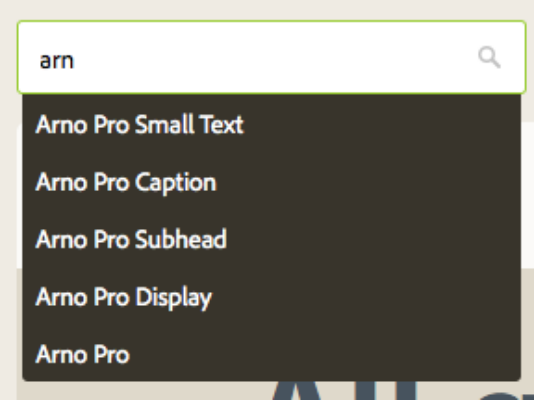
8. The Atlantic



9. Travelocity

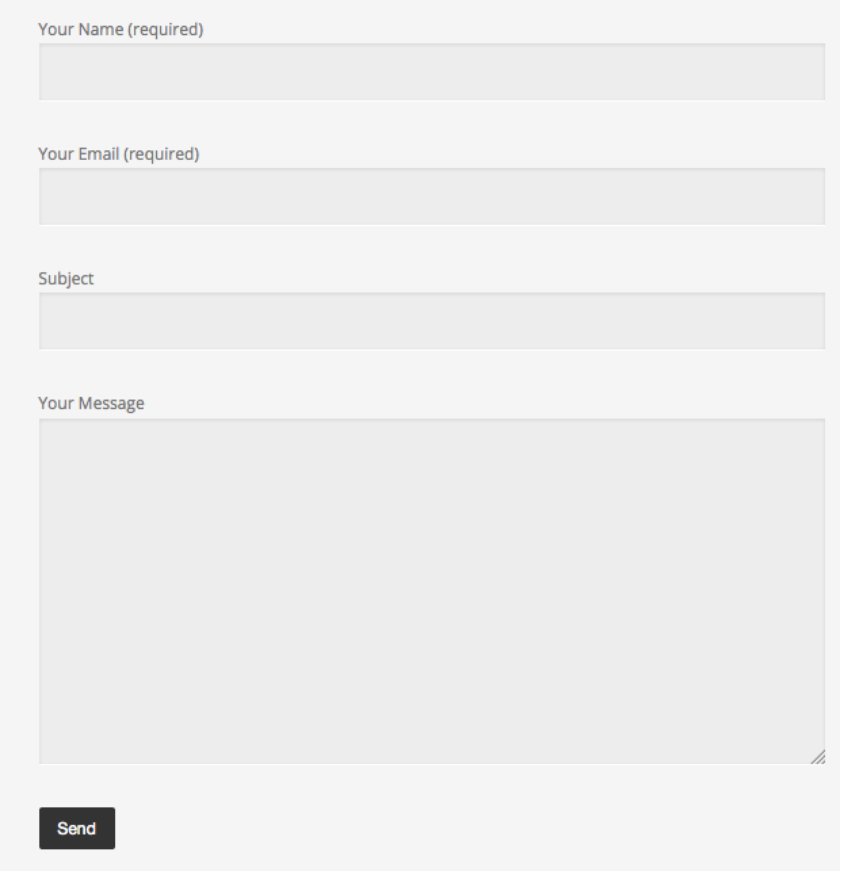


10. Adobe Typekit

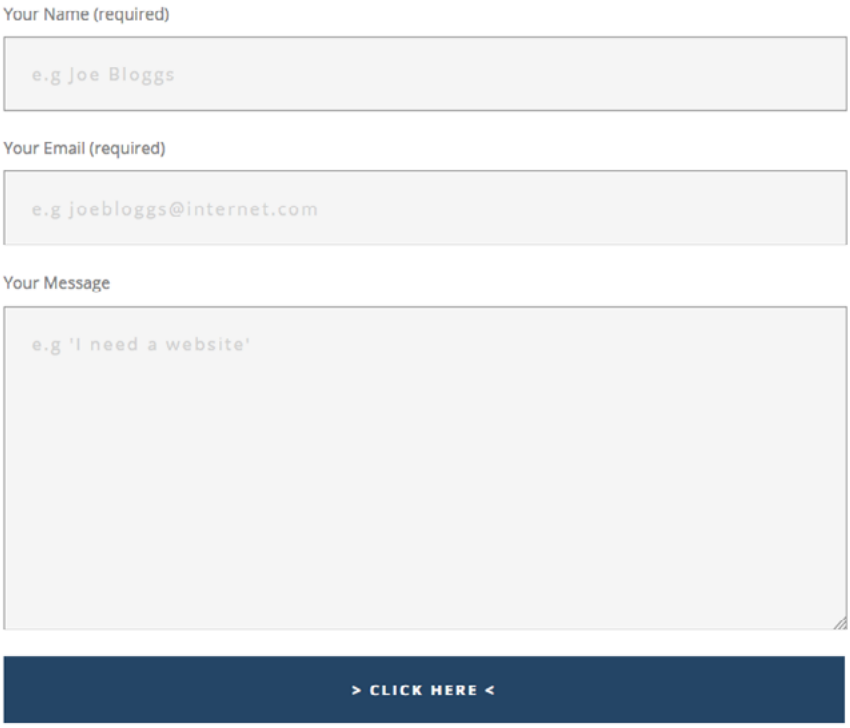


Contact forms

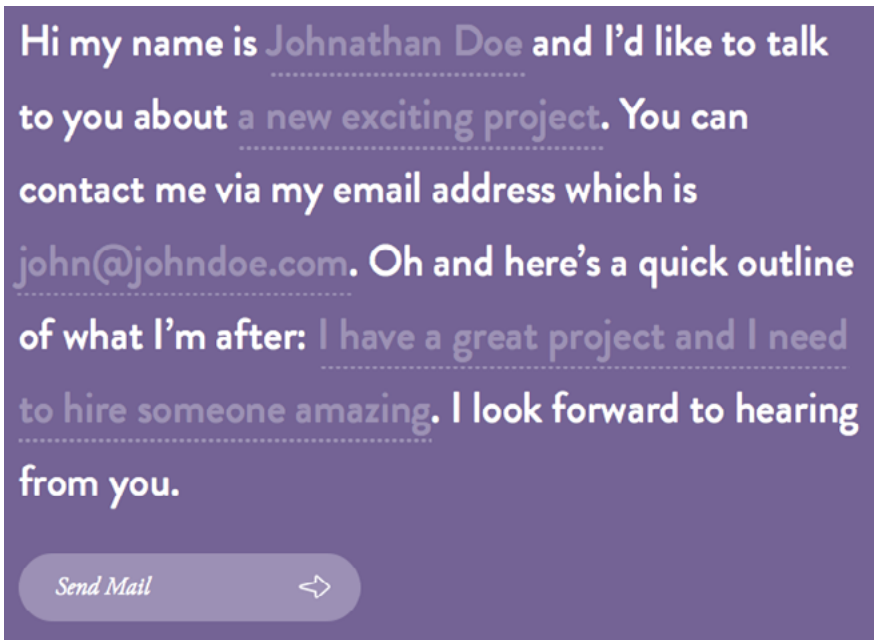
1. Blue Ant Design



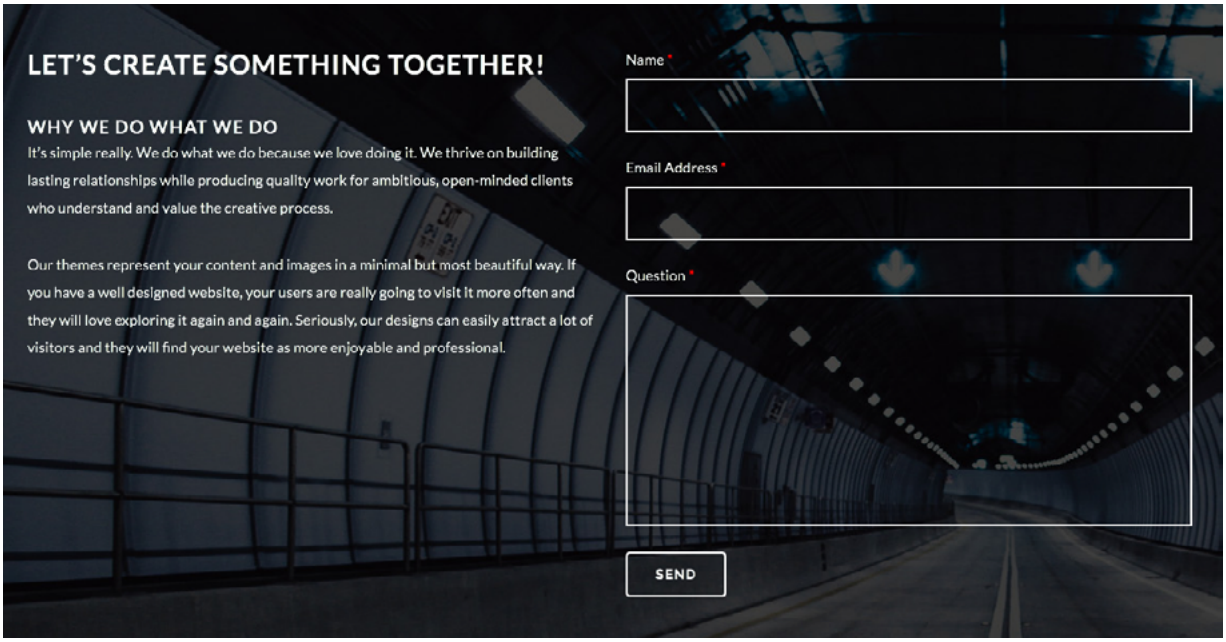
2. Canny Creative



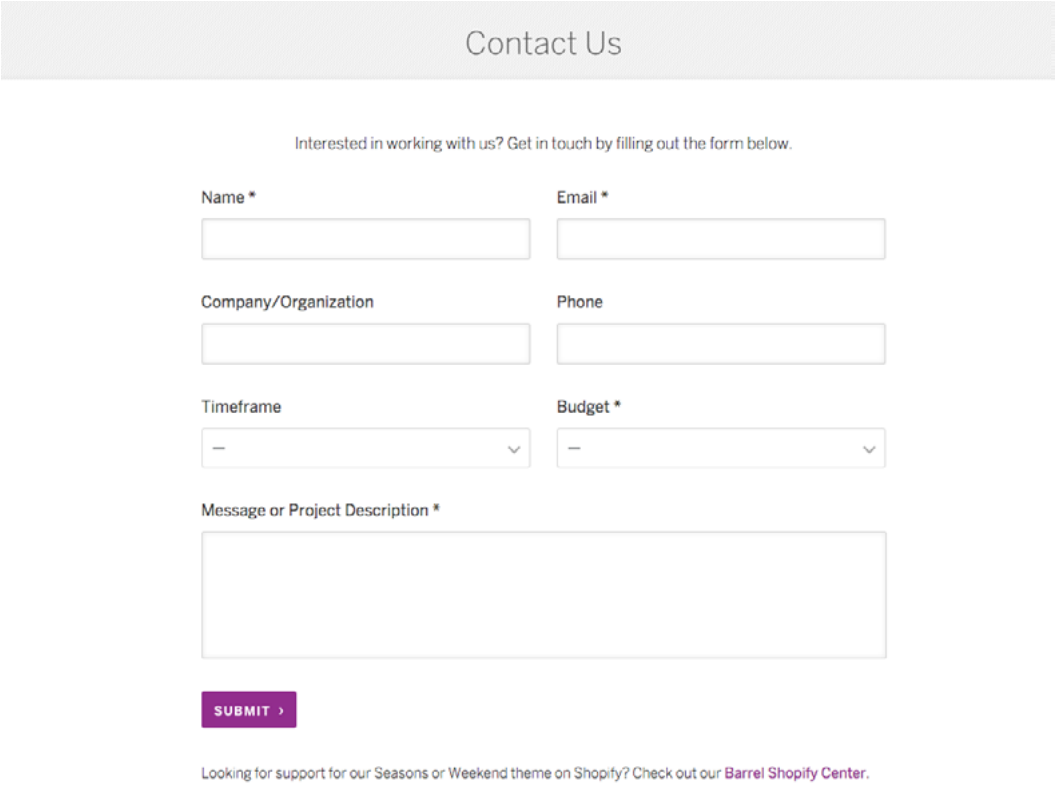
3. Tribal Media



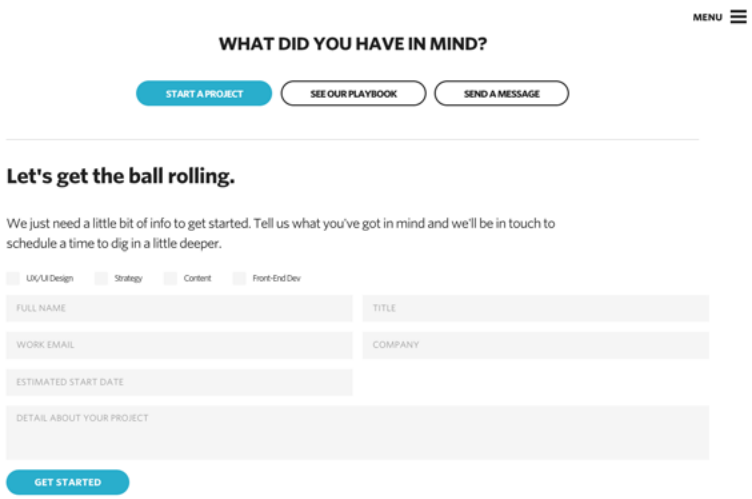
4. W Design Agency



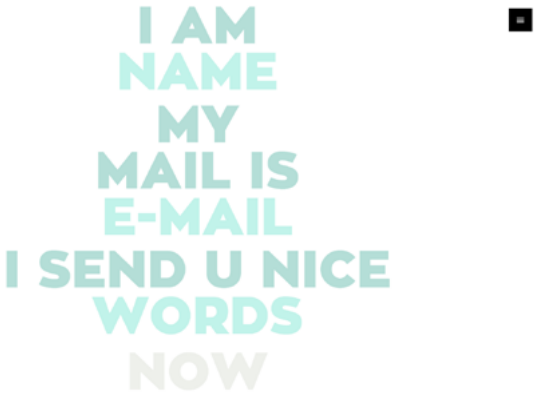
5. Barrel



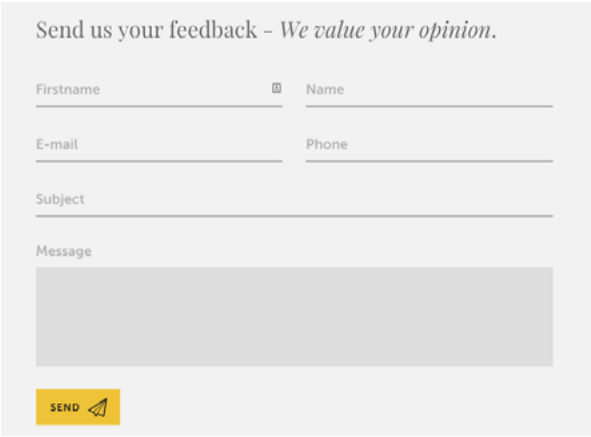
6. Digital Telepathy



7. Anakin



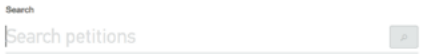
8. Weblounge



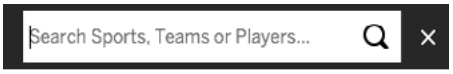
Search

1. A List Apart

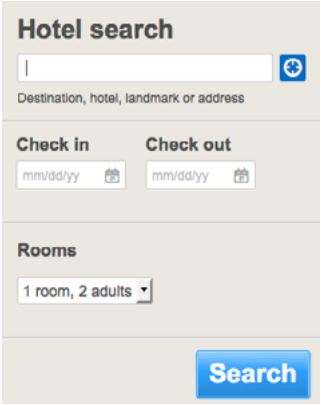
2. Change.org



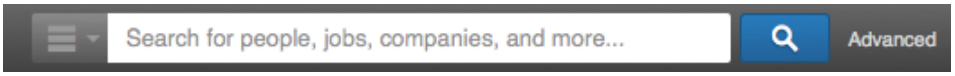
3. ESPN



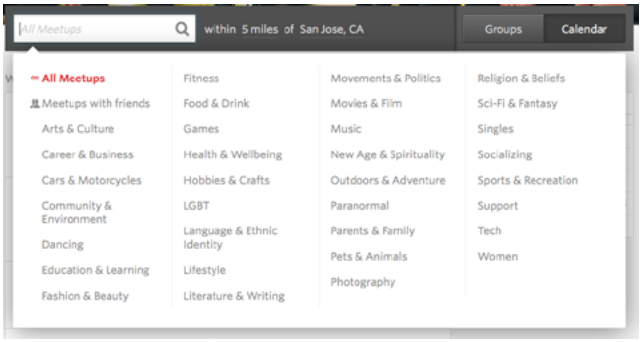
4. Hotels.com



5. LinkedIn

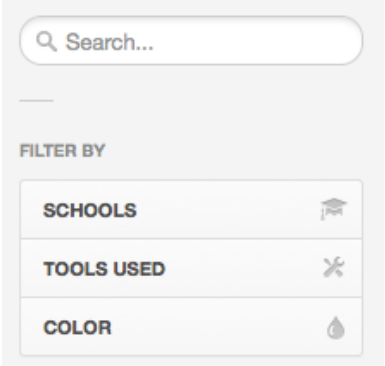


6. Meetup



7. Neon Roots

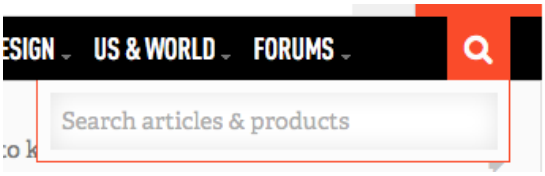
8. Behance



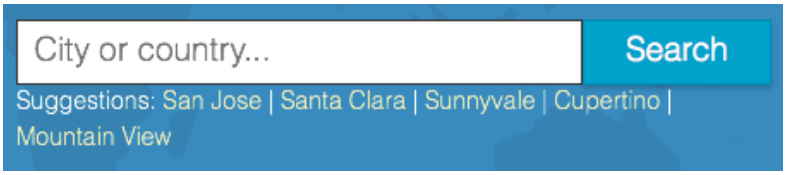
9. Site Inspire

10. TED

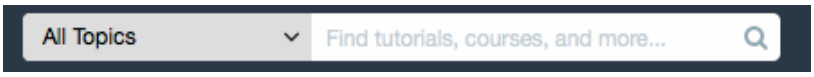
11. The Verge



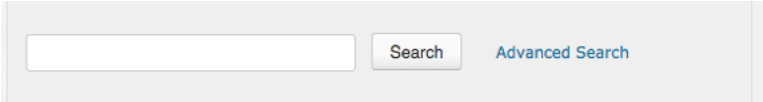
12. Timeanddate.com



13. Tuts+

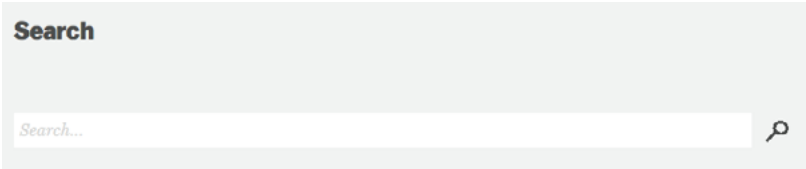


14. University of North Carolina

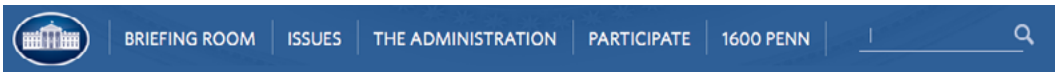


15. UXPin

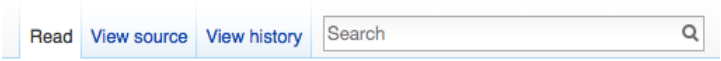
16. Vox Media



17. The White House



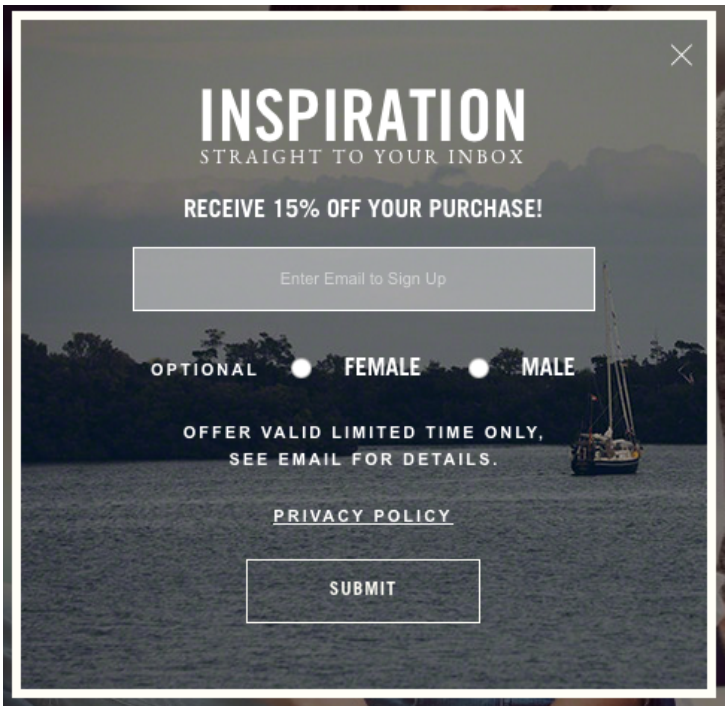
18. Wikipedia



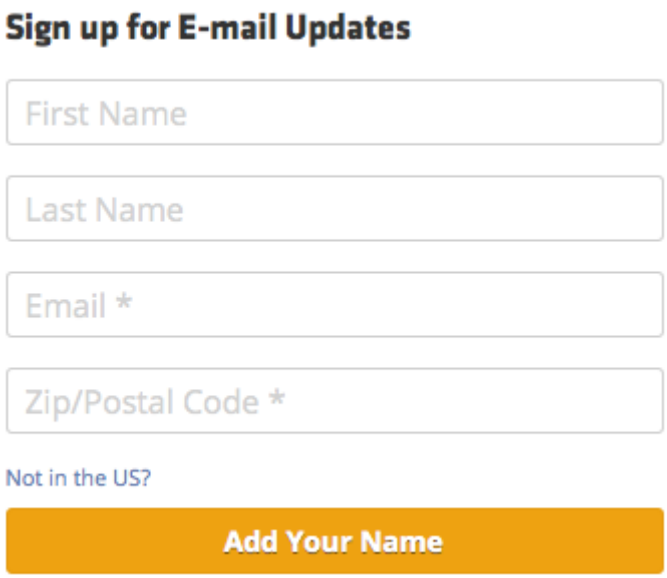
19. Wolf Trap

Signup forms

1. Abercrombie & Fitch



2. Action Network



3. AWeber

Select A Plan

Choose a monthly, quarterly, or annual plan.

First Month FREE
Then only $19 monthly!

First Quarter $32.67
Then Only $49 Quarterly

First Year $177.65
Then Only $194 Annually

Create Your Account

Your login name must be between 3 and 15 characters. You can include lowercase letters (a-z), numbers, hyphens ("-") and underscores ("_"), but **it must contain at least one letter**.

Login Name

First Name

Last Name

Email Address

Phone Number

Website URL (optional)

Company Name (optional)

Secure Website

After you fill out this form, you can begin to use your AWeber account immediately!

Who's Joining AWeber?

- Claudia in Tegucigalpa, Honduras (11 minutes ago)
- Kim in Bristol, VA (14 minutes ago)
- Thalia in Garland, TX (22 minutes ago)
- Donna in Pleasantville, New Jersey (26 minutes ago)

Enter Billing Information

Billing Address 1

Billing Address 2 (optional)

City

State / Territory

Postal Code

Country

Select One

Why do you need my credit card?

Credit Card Number

CVV Where is the CVV code?

Expiration Date

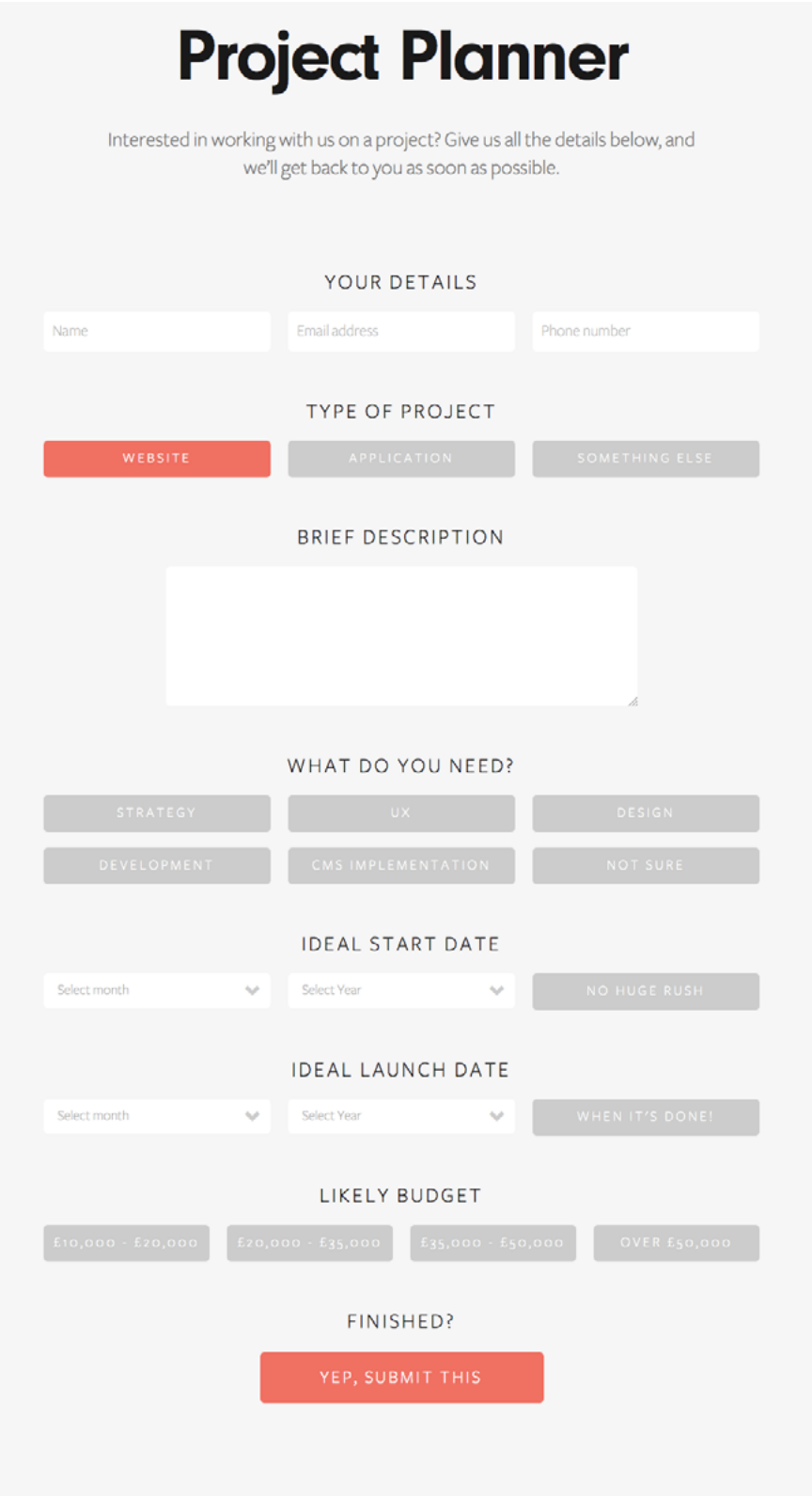
01 - January 2015

Have a gift card or promo code?

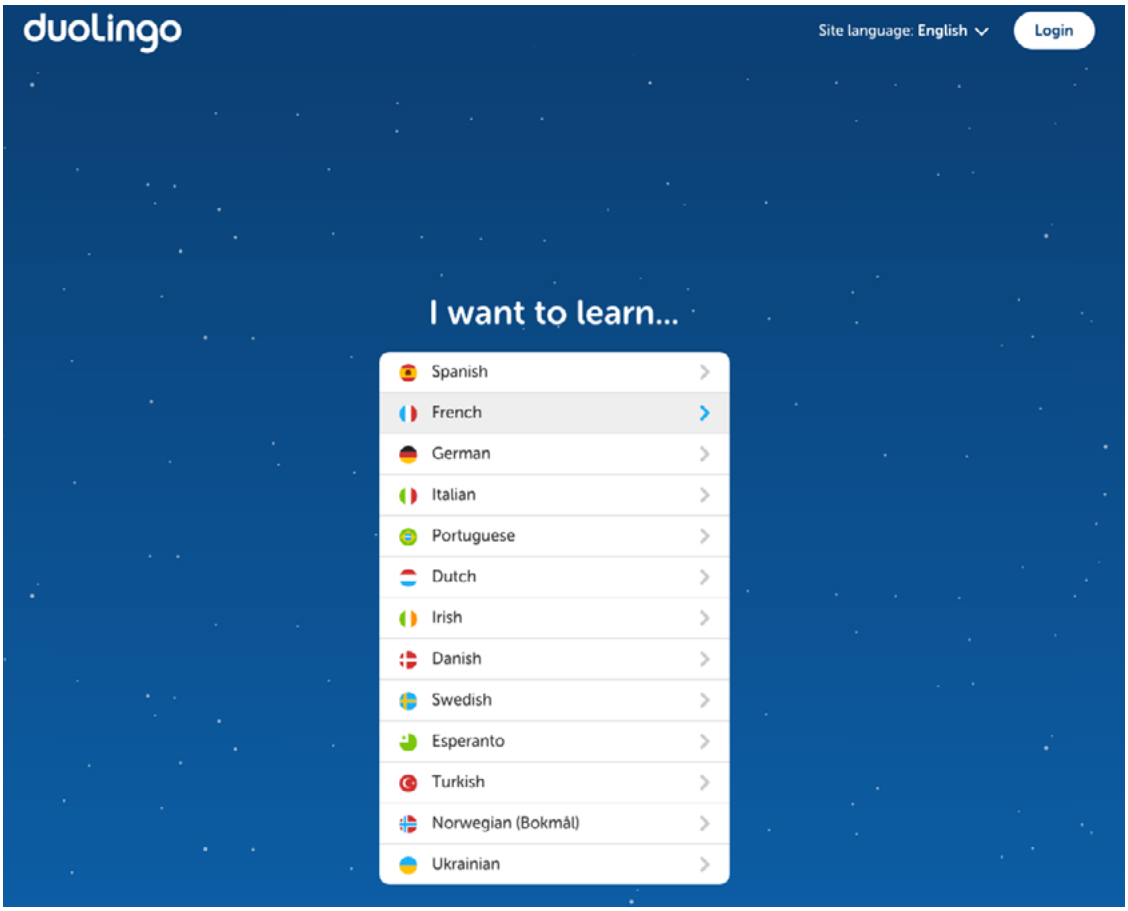
Service Agreement

Sending Spam through AWeber is strictly forbidden. By clicking the button below, you

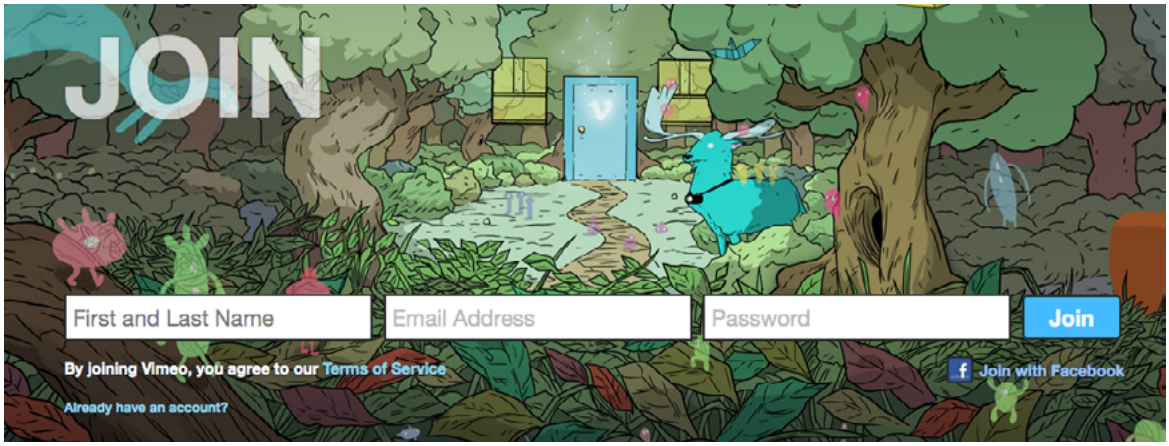
4. Buffalo Web Design



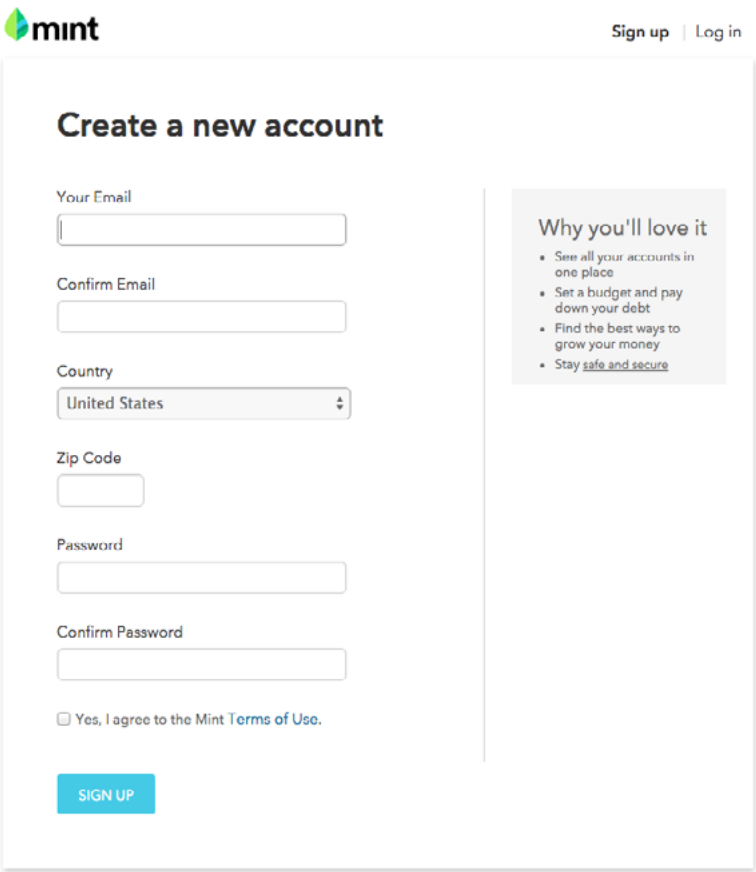
5. Duolingo



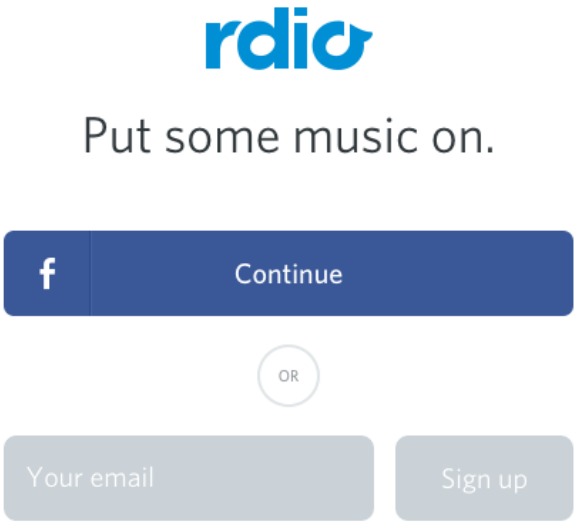
6. Vimeo



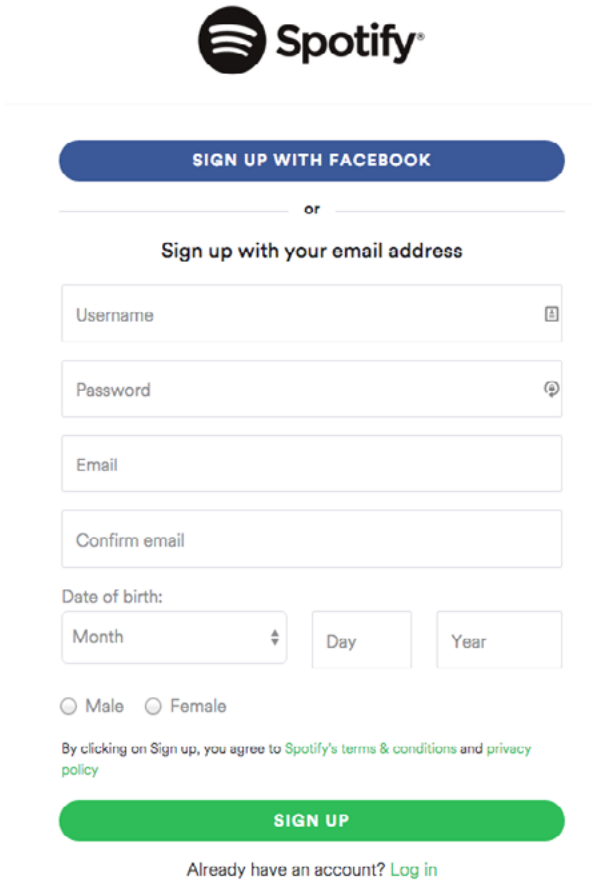
7. Mint


mint
Sign up | Log in
Create a new account
Your Email
Confirm Email
Country
United States
Zip Code
Password
Confirm Password
Yes, I agree to the Mint Terms of Use.
SIGN UP
Why you'll love it
See all your accounts in one place
Set a budget and pay down your debt
Find the best ways to grow your money
Stay safe and secure

8. Rdio


rdio
Put some music on.
Continue
OR
Your email
Sign up

9. Spotify

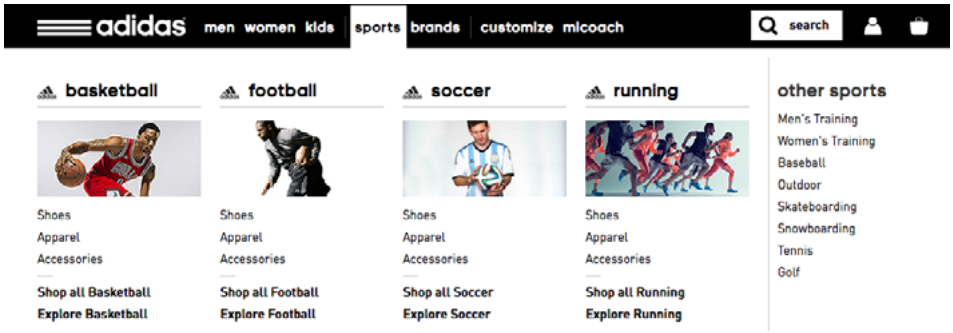


10. Squarespace

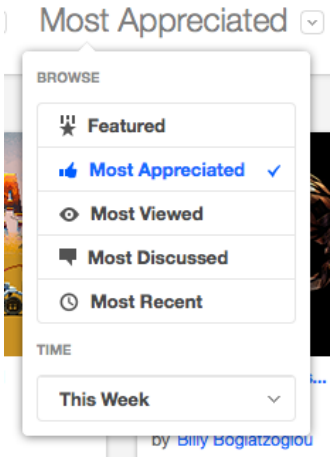
Navigation

Drop down navigation (mega menus)

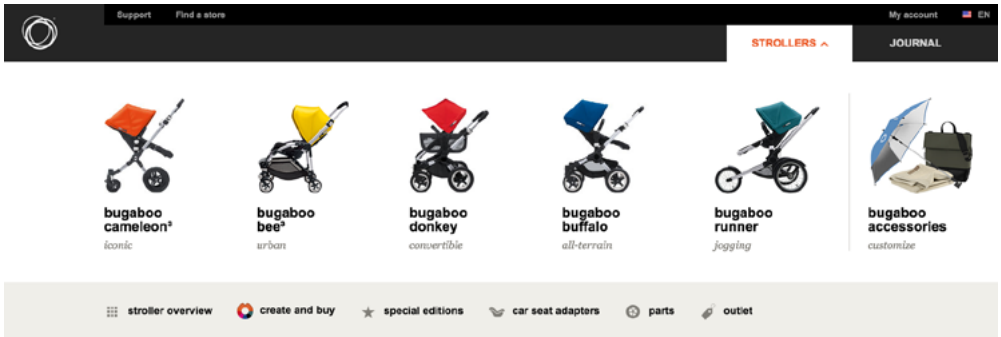
1. Adidas



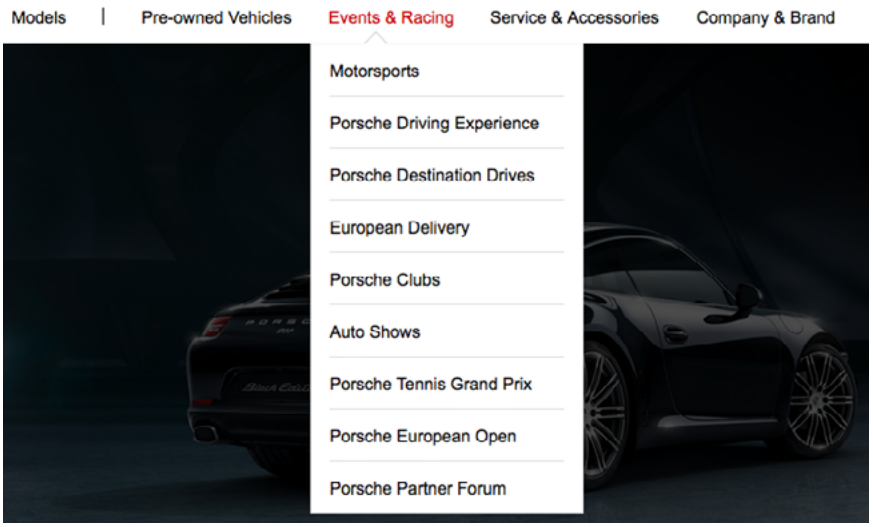
2. Behance



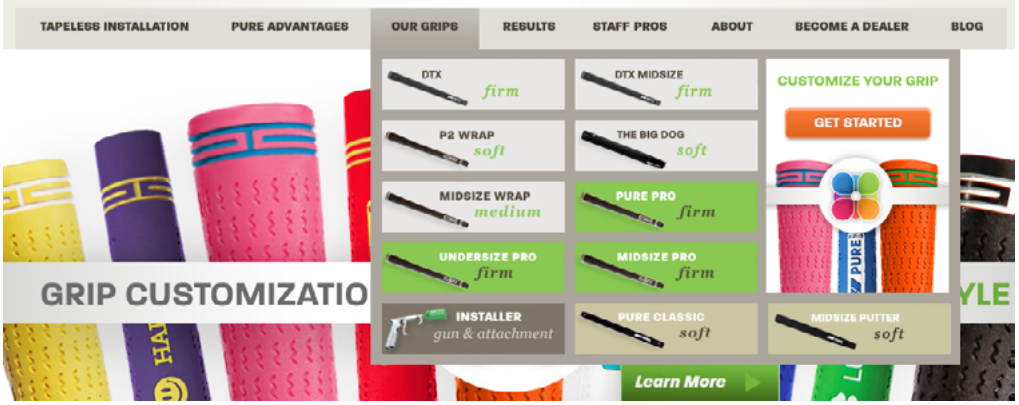
3. Bugaboo



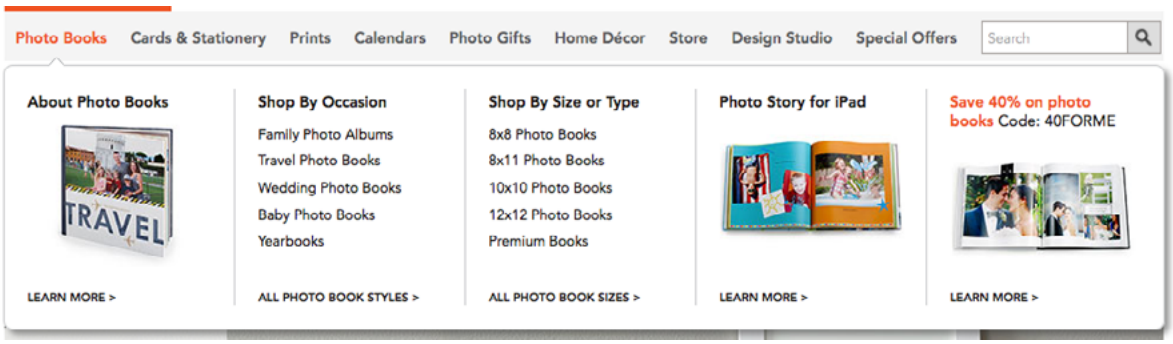
4. Porsche



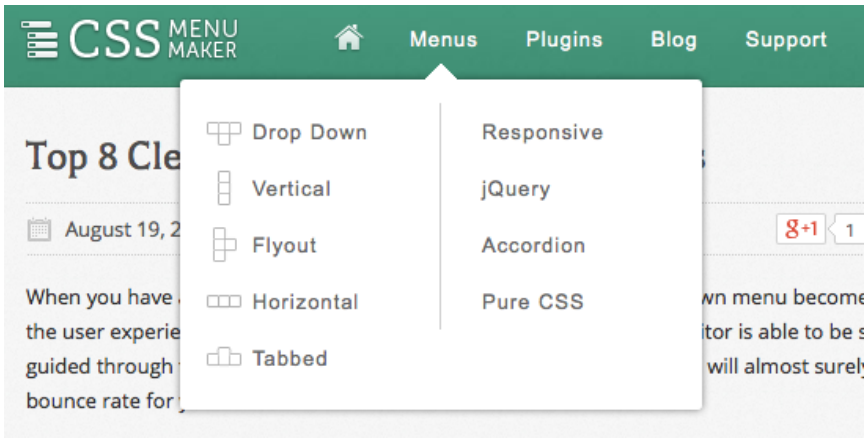
5. PURE Grips



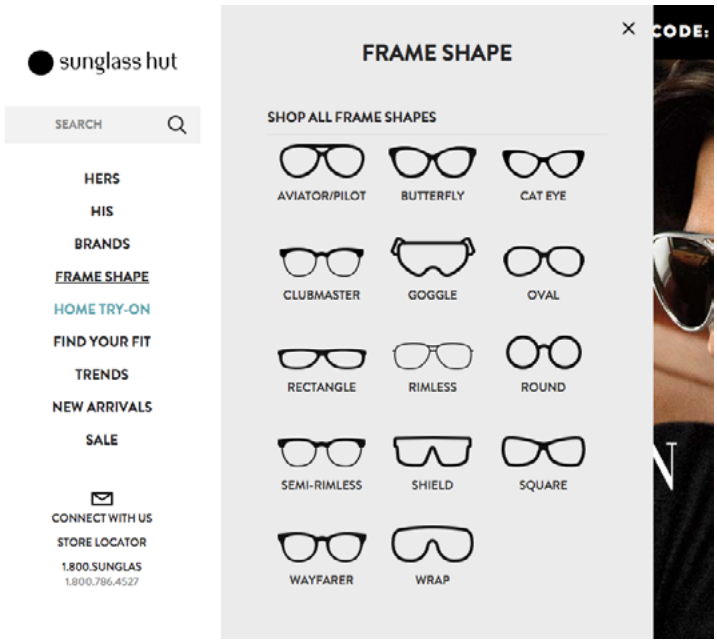
6. Shutterfly



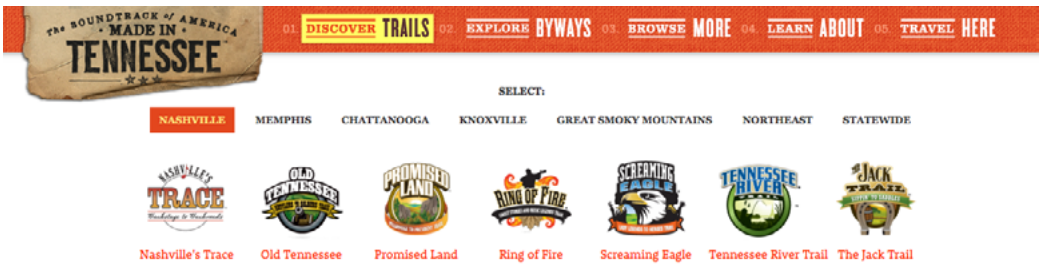
7. CSS Menu Maker



8. Sunglass Hut



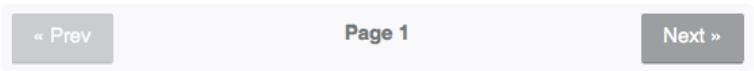
9. Tennessee Trails and Byways



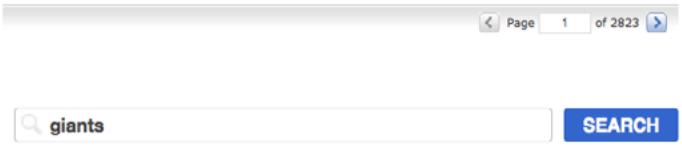
Pagination

1. A List Apart

2. Change.org



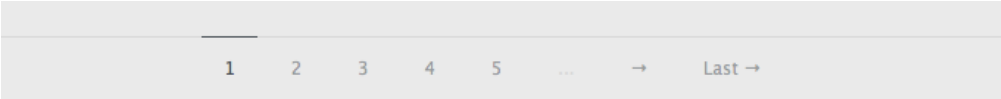
3. ESPN



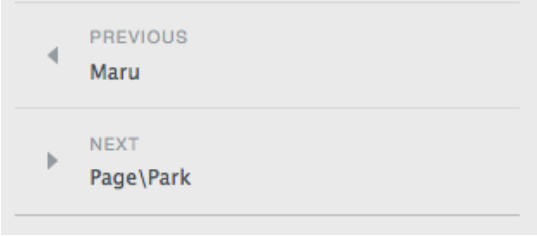
4. LinkedIn

5. Neon Roots

6. Site Inspire



7. Site Inspire

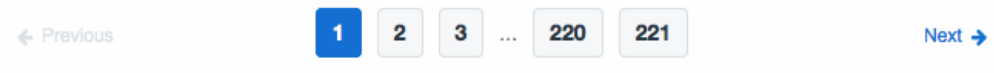


8. TED

9. The Verge

10. Crabbie's

11. Tuts+



12. University of North Carolina

13. University of Texas

14. Vox Media

15. Wolf **Trap**

Teasers

Article lists

1. AIGA

2. A List Apart

Issue № 423

Container Queries: Once More Unto the Breach
by Mat Marquis · June 30, 2015 · 23 Comments
Media queries have been the go-to tool in building responsive sites, allowing us to resize and recombine modules to suit multiple contexts, layouts, and viewports. But relying on fixed viewport sizes can quickly twist stylesheets into Gordian knots. We still need a future-friendly way to manage responsive CSS. Mat Marquis explores the problem and the progress toward the solution—and issues a rallying call.

Create a Content Compass
by Meghan Casey · June 30, 2015 · 10 Comments
Content projects need a sense of direction: something to help you and your team provide the right content to the right people at the right time. Enter the content compass—centered on your strategy and supported by your messaging—to keep your content efforts on track. In this excerpt from Chapter 11 of The Content Strategy Toolkit, Meghan Casey explains her methodology for developing a core strategy statement and messaging framework.

Issue № 422

The Homepage Exception
by Johanna Bates · June 16, 2015 · 29 Comments
Structured, automatic systems are great at managing content efficiently—but not so great at accommodating human changes in that content. On the other hand, free-for-all WYSIWYGs lead to inconsistency and breakdowns. Stakeholders and content administrators need flexibility and control, especially where the all-important homepage is concerned. What's a website to do? Johanna Bates suggests embracing a people-friendly homepage solution within our robot-driven architectures.

Understanding the Emotional Response
by Kelsey Lynn Lundberg · June 16, 2015 · 10 Comments
Validating emotions isn't a glorified psychological process; part of our work is to hear our

Recent Blog posts

Love Your CMS. (No, Really!)
On August 25, the newest ALA: On Air event tackles CMSes: customization, traditional versus "headless," backend UX, and more.
Eileen Webb · August 03, 2015

Developing Empathy
Empathy sounds great in theory, but how can we actually practice it for our users? Susan Robertson shows how to get started.
Susan Robertson · July 22, 2015

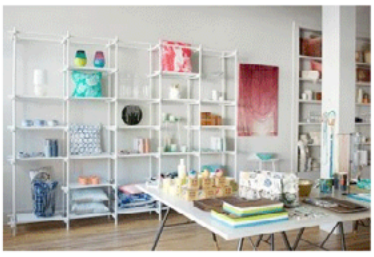
On Our Radar: Four-and-a-Horse Stars
The ALA staff is reading about icon font accessibility issues, talent incubators in Africa, the new WICG, WebAssembly, and more.
ALA Staff · July 10, 2015

Designing for Non-Native Speakers
Designing for non-native English speakers might be easier than you think. Here are three areas to focus on as you get started.
Senongo Akpem · July 08, 2015

Feedback Phases and Personas
These five feedback personas will help you get valuable feedback for your project, right when you need it most.
Anthony Colangelo · July 06, 2015

3. Architectural Digest

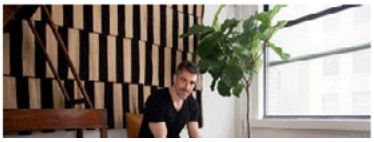
RUN IN PRINT
In a new partnership with Vans, British brand Eley Kishimoto takes its graphic prints to the street



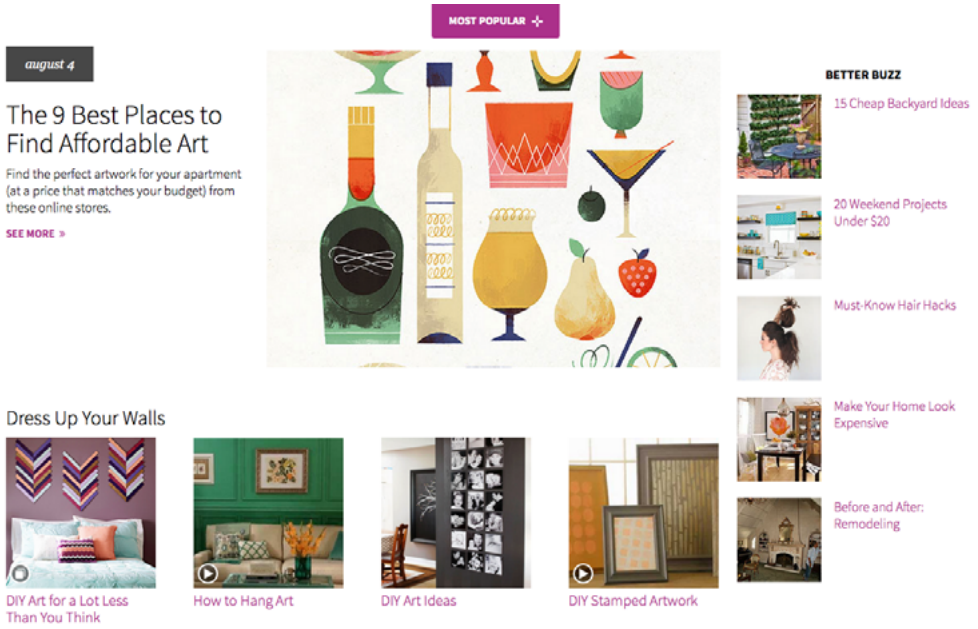
SHOP TALK
Burkelman—a new home design store—brings artisanal ceramics, textiles, and more to Cold Spring, New York

BESTOWING SHADE
John Robshaw debuts a line of patterned sunglasses with Italian brand Mondelliani

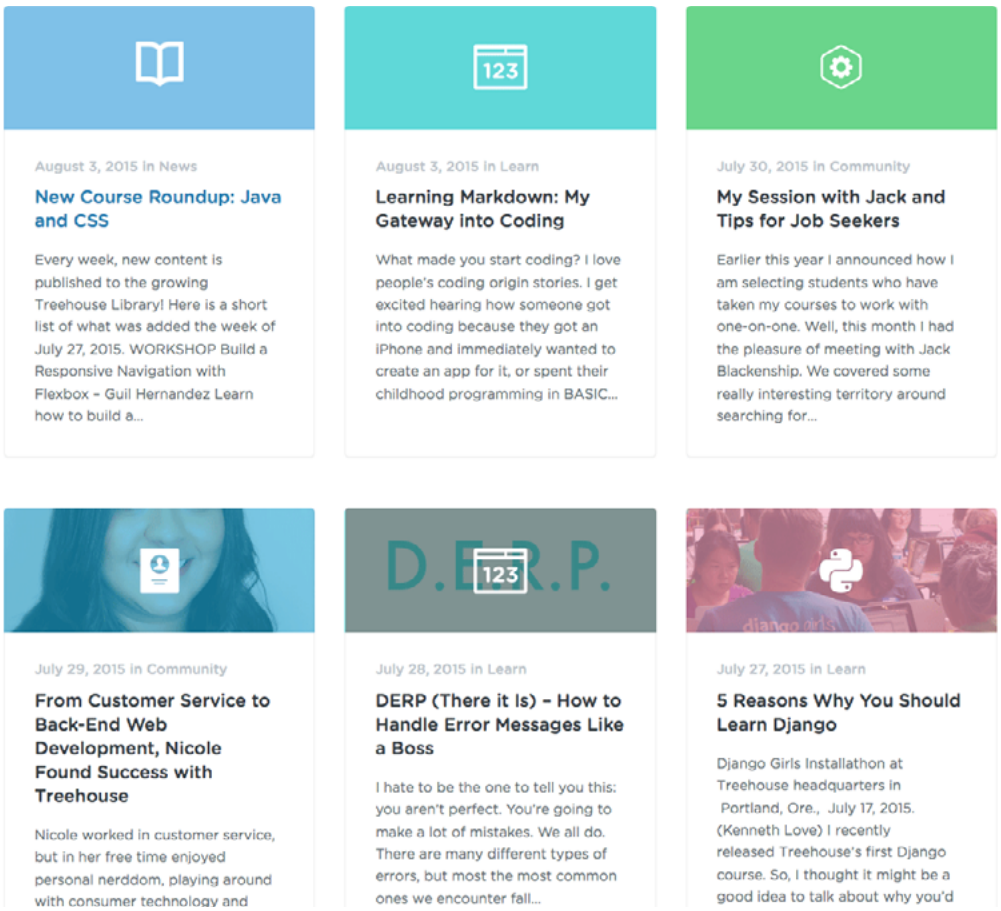
MADURA MOVES IN
Home brand Madura sets up shop in New York's Flatiron District



4. Better Homes and Gardens



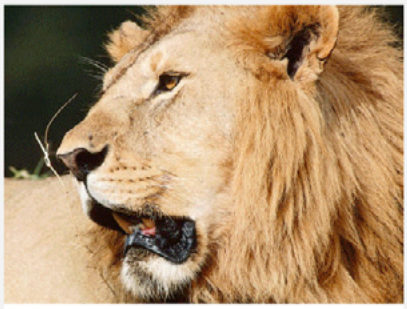
5. Team Treehouse



6. The Discovery Channel

Could Barnacles Hold the Key to Finding MH370? 4 AUG

Tiny crustaceans stuck to the wreckage washing up on Reunion Island could help scientists locate the rest of MH370, researchers claim. MORE



Three Major Airlines Ban Shipment of Hunting Trophies 4 AUG

Hunters looking to ship big game trophies via air now have three fewer options. MORE

NASA Crash-Lands Cessna Airplane to Test Emergency Response Systems 3 AUG

The 'violent but controlled' experiment yielded a gold mine of data about the durability of emergency transmitters. MORE

Aussie Daredevil Hacks Dirt Bike to Run on Water 3 AUG

Is a dirt bike still a dirt bike without the dirt? MORE

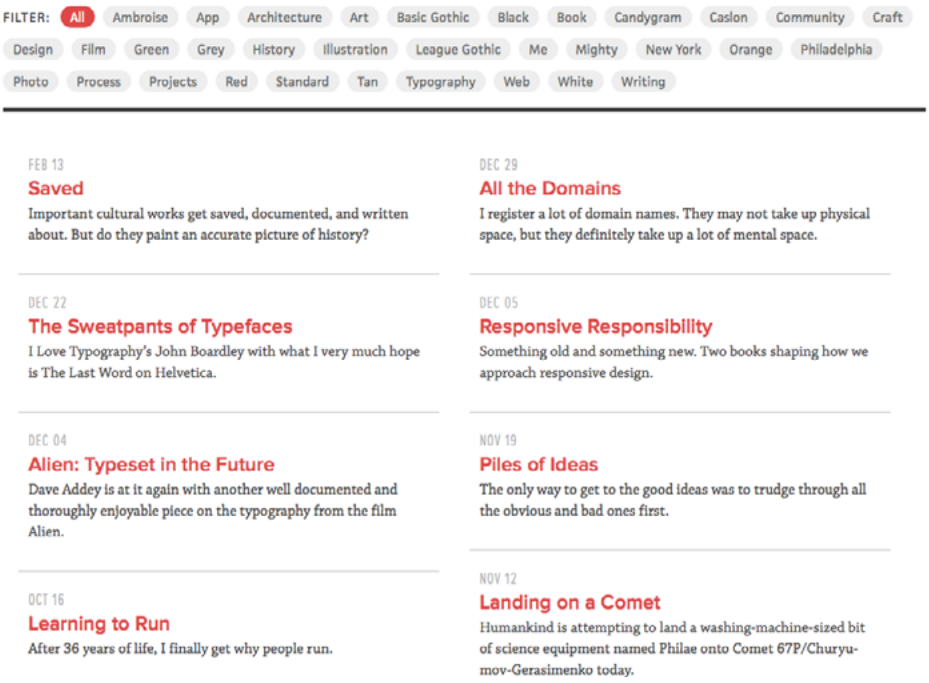
Zimbabwe Halts Lion, Leopard and Elephant Hunting in Hwange 3 AUG

Zimbabwean authorities 'will not hesitate to arrest, prosecute, and ban' any violators. MORE

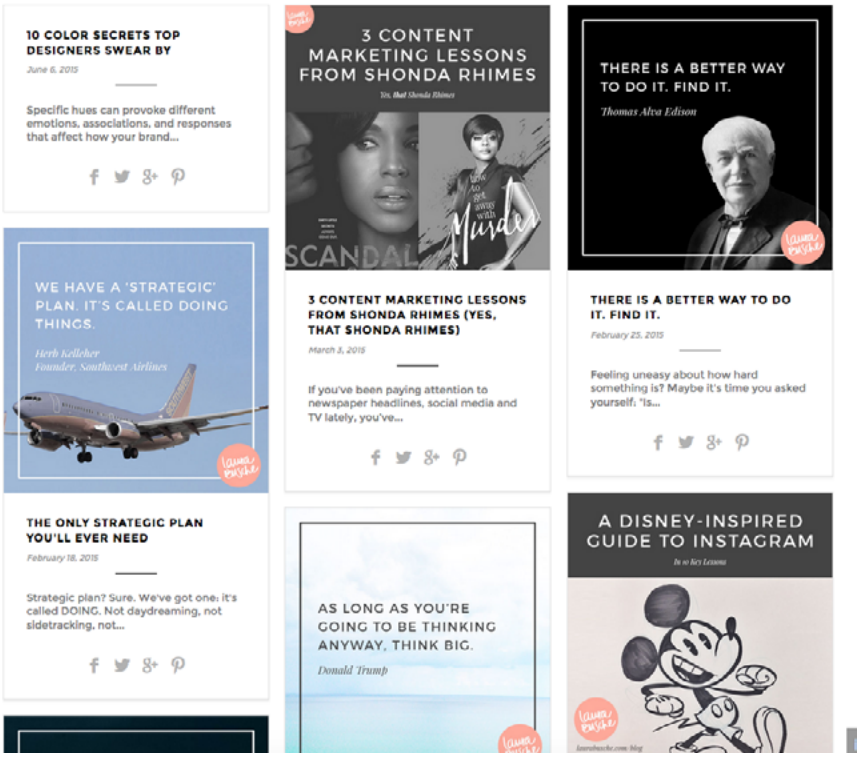
You've Got to See this Peacock Spider's Sick Dance Moves 3 AUG

Recently described peacock spider Maratus personatus is a beast on the dance floor. MORE

7. Jason Santa Maria

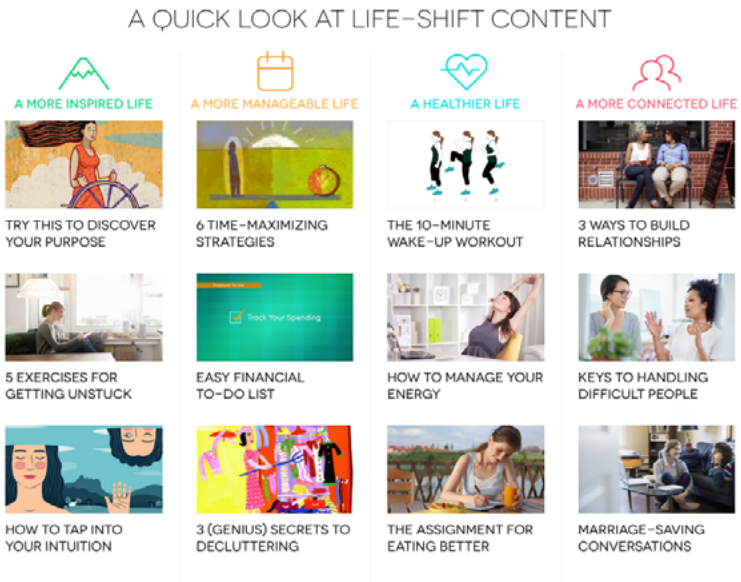


8. Laura Busche



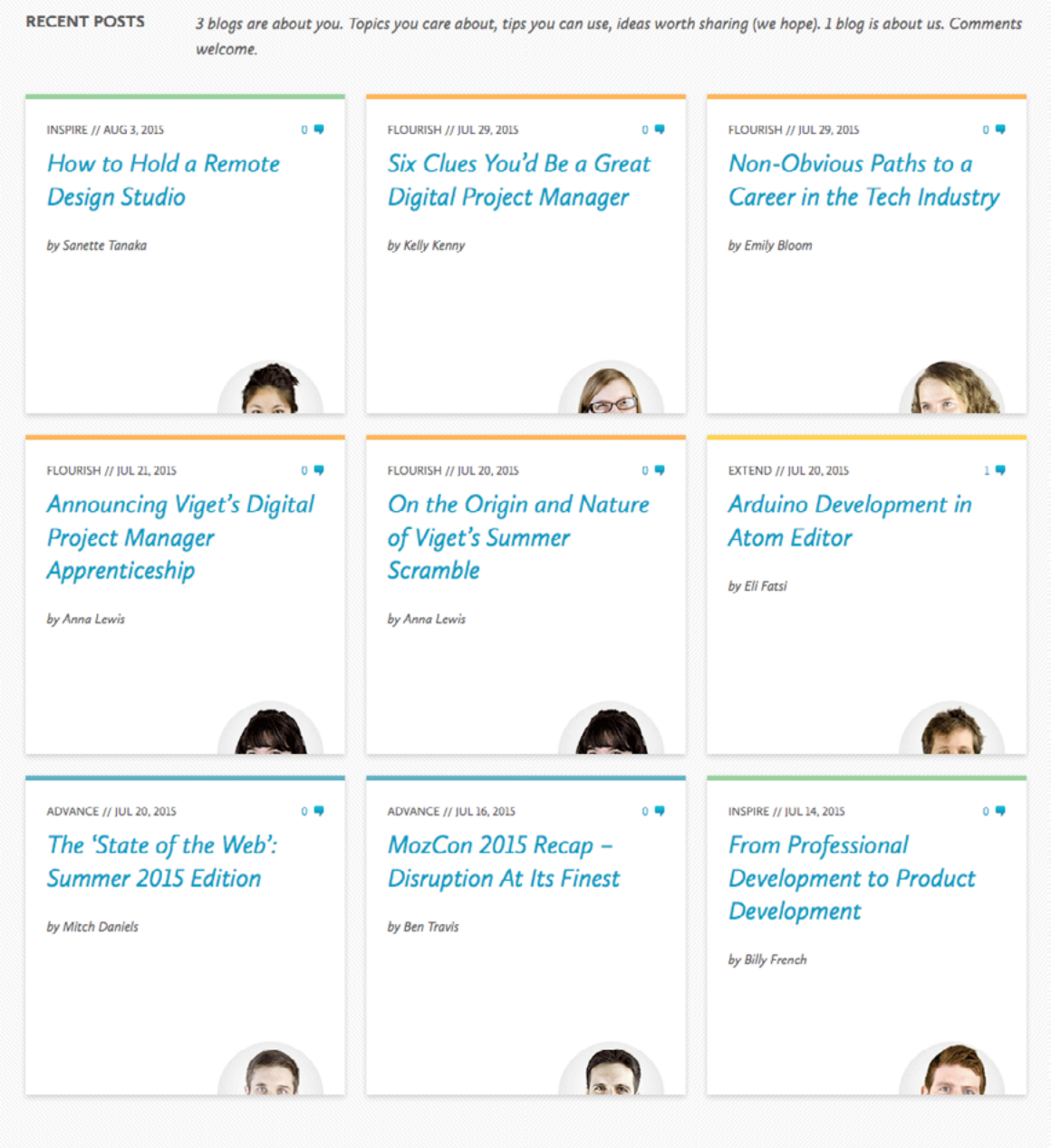
9. Mezzoblue

10. Oprah Winfrey



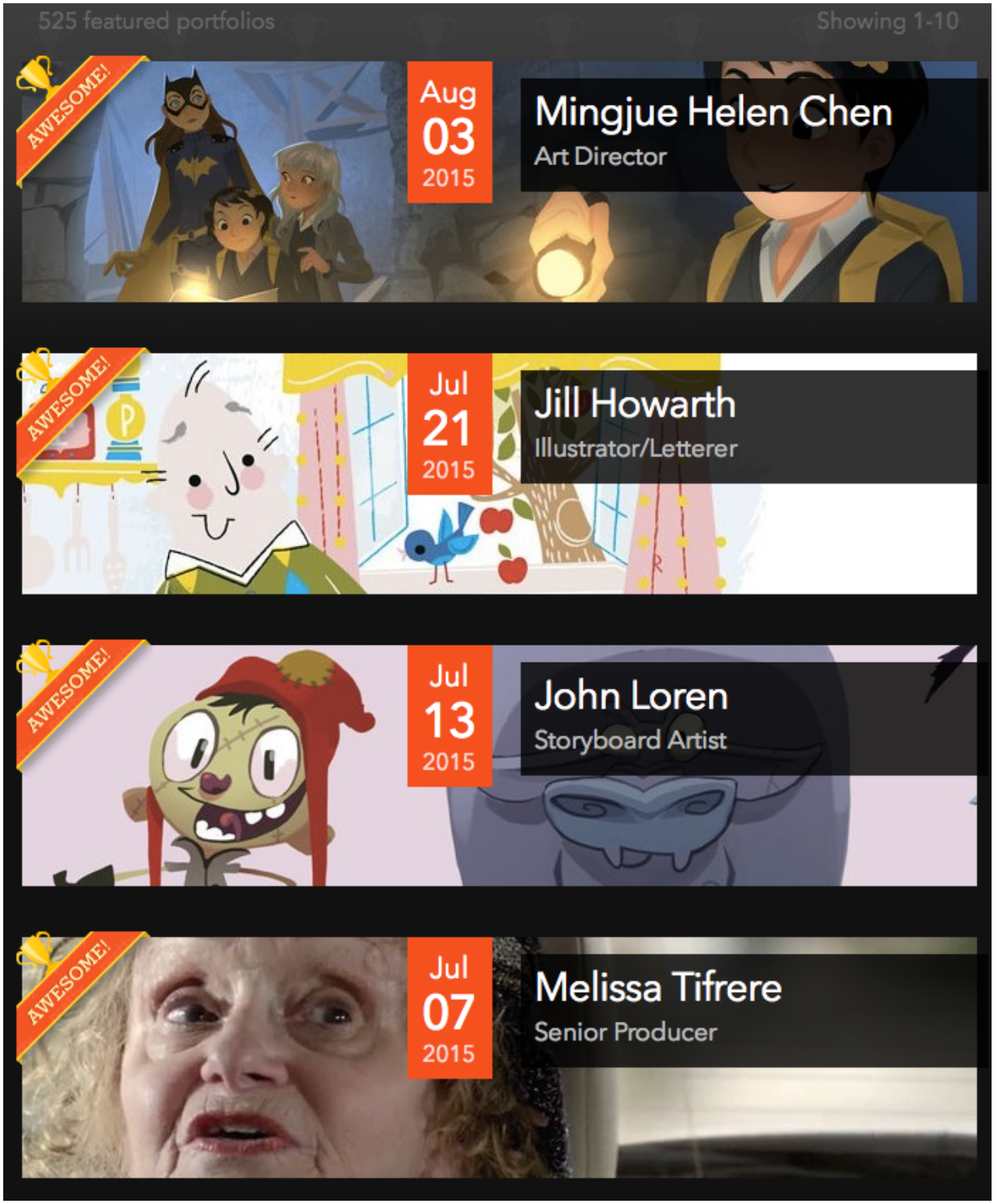
11. Quartz

12. Viget

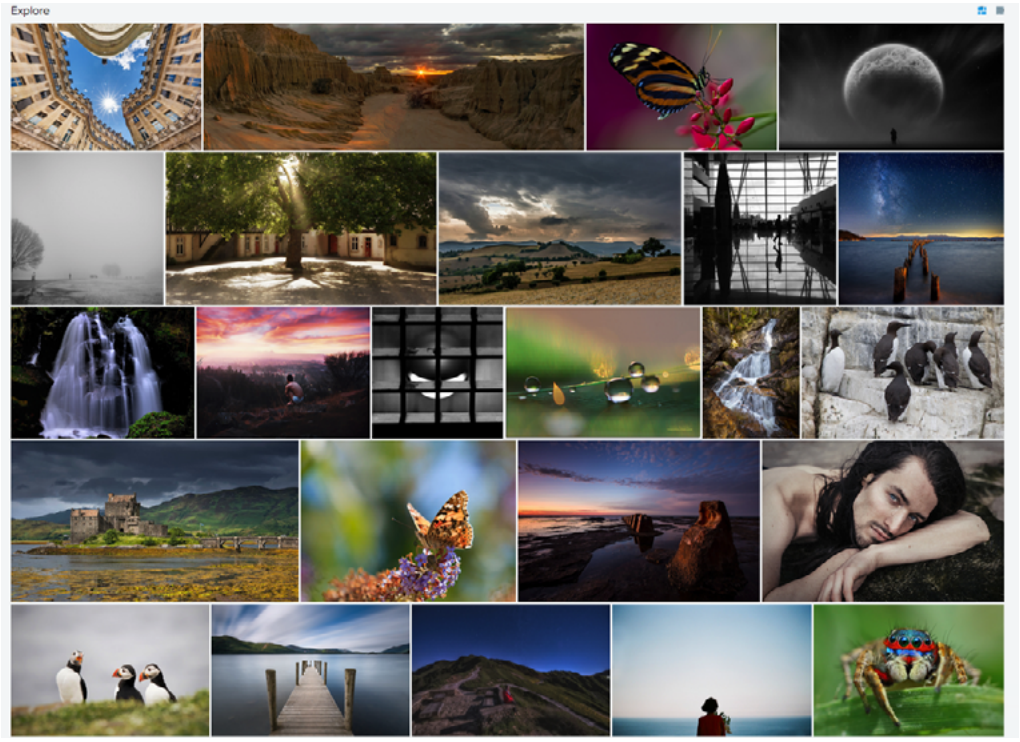

RECENT POSTS
3 blogs are about you. Topics you care about, tips you can use, ideas worth sharing (we hope). 1 blog is about us. Comments welcome.
INSPIRE // AUG 3, 2015
0
How to Hold a Remote Design Studio
by Sanette Tanaka
FLOURISH // JUL 29, 2015
0
Six Clues You'd Be a Great Digital Project Manager
by Kelly Kenny
FLOURISH // JUL 29, 2015
0
Non-Obvious Paths to a Career in the Tech Industry
by Emily Bloom
FLOURISH // JUL 21, 2015
0
Announcing Viget's Digital Project Manager Apprenticeship
by Anna Lewis
FLOURISH // JUL 20, 2015
0
On the Origin and Nature of Viget's Summer Scramble
by Anna Lewis
EXTEND // JUL 20, 2015
1
Arduino Development in Atom Editor
by Eli Fatsi
ADVANCE // JUL 20, 2015
0
The 'State of the Web': Summer 2015 Edition
by Mitch Daniels
ADVANCE // JUL 16, 2015
0
MozCon 2015 Recap – Disruption At Its Finest
by Ben Travis
INSPIRE // JUL 14, 2015
0
From Professional Development to Product Development
by Billy French

Galleries

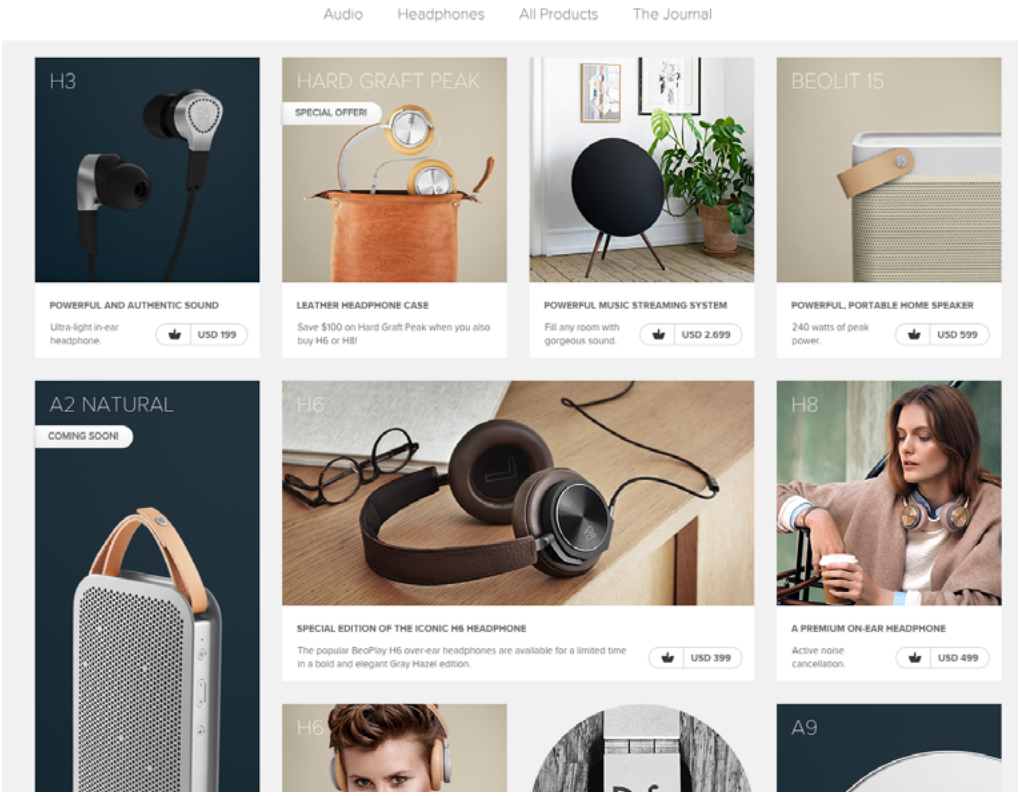
1. Carbon Made



2. Flickr

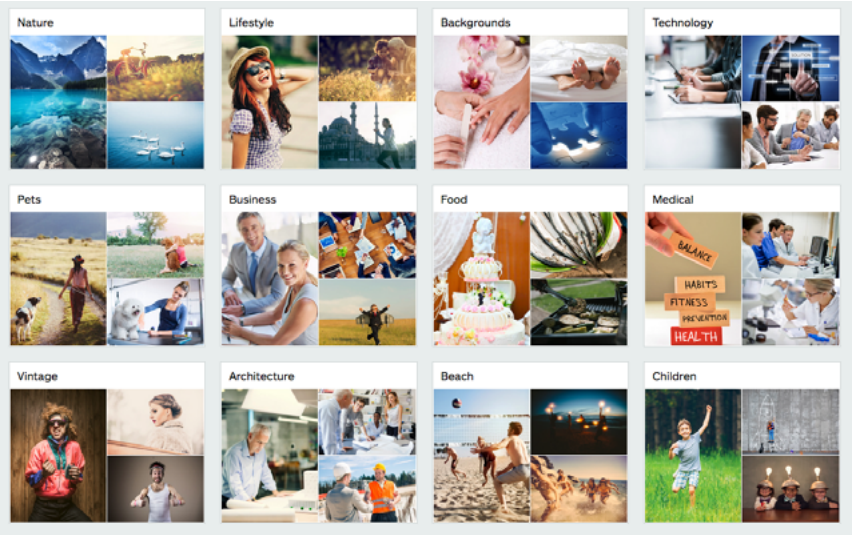


3. Beoplay



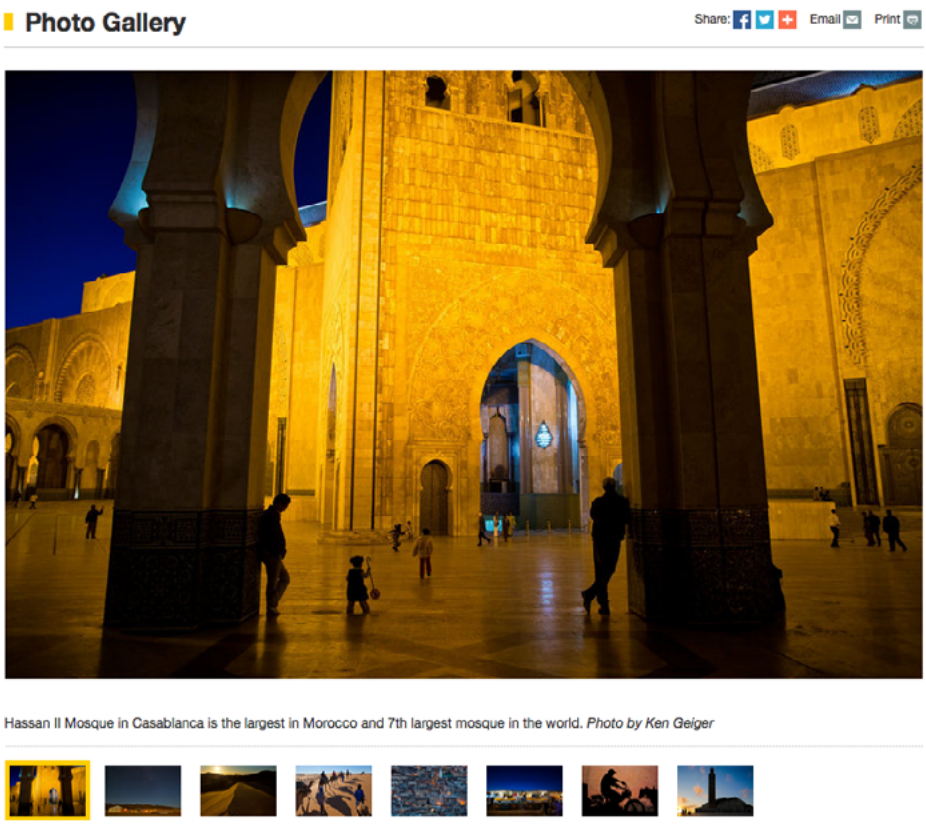
4. iOS Icon Gallery

5. iStock

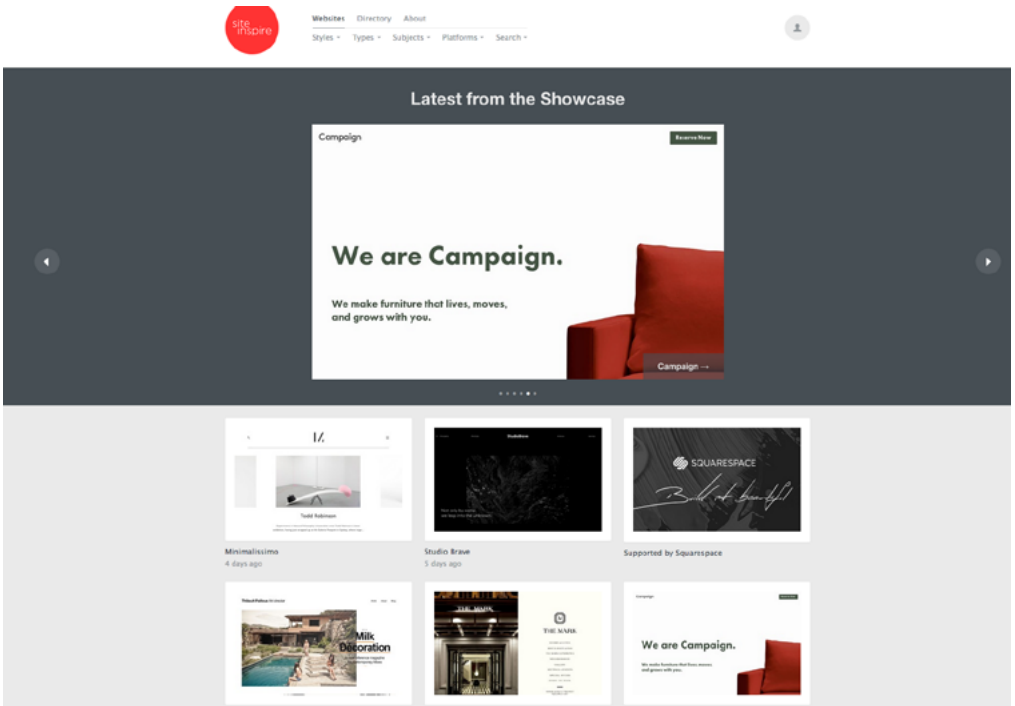


6. Mario Mashee

7. National Geographic Expeditions

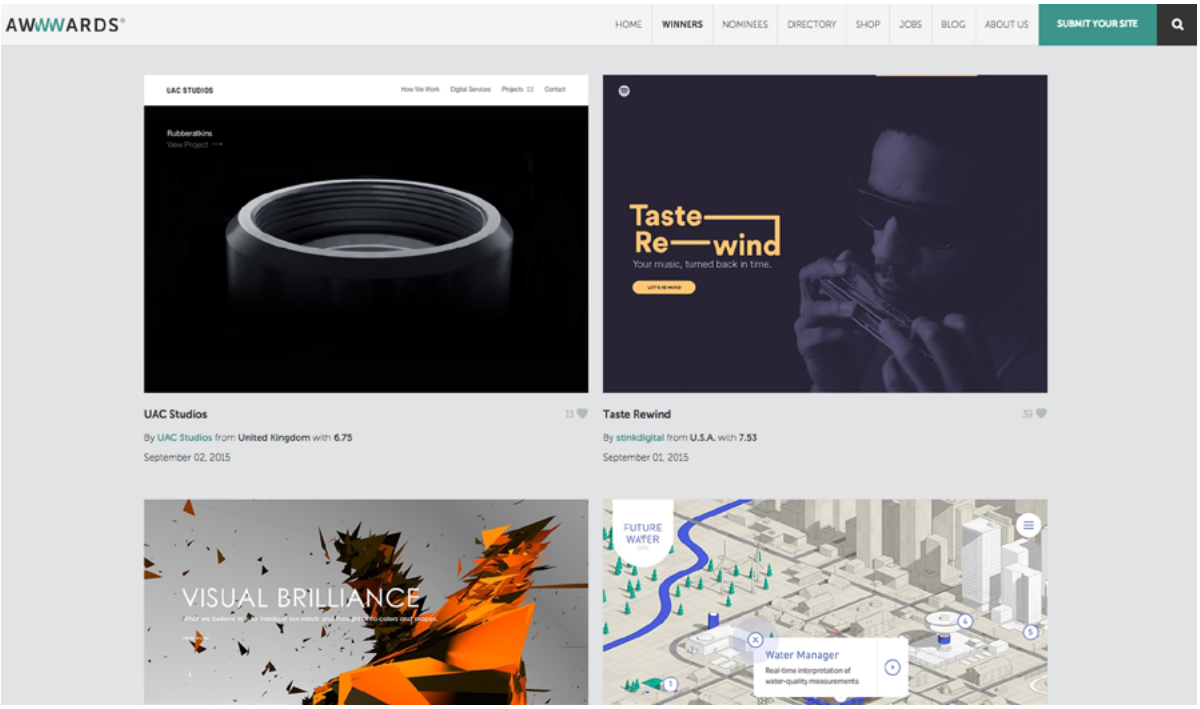


8. Siteinspire

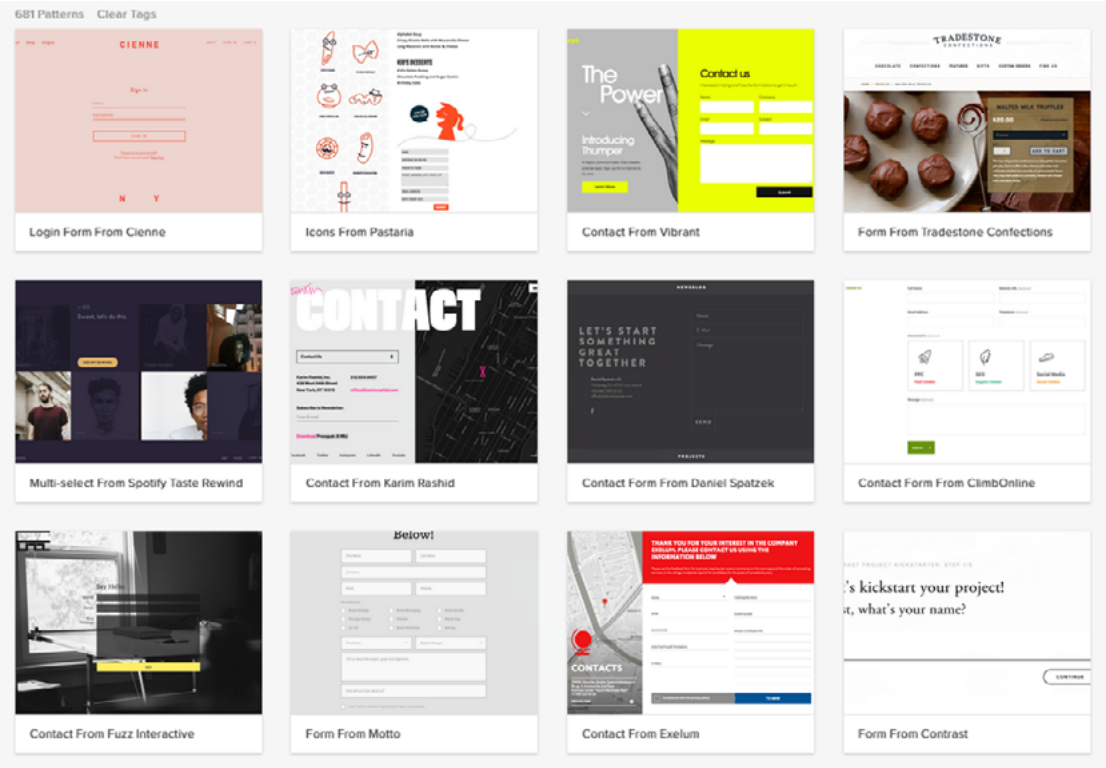


9. Sports Illustrated Photos

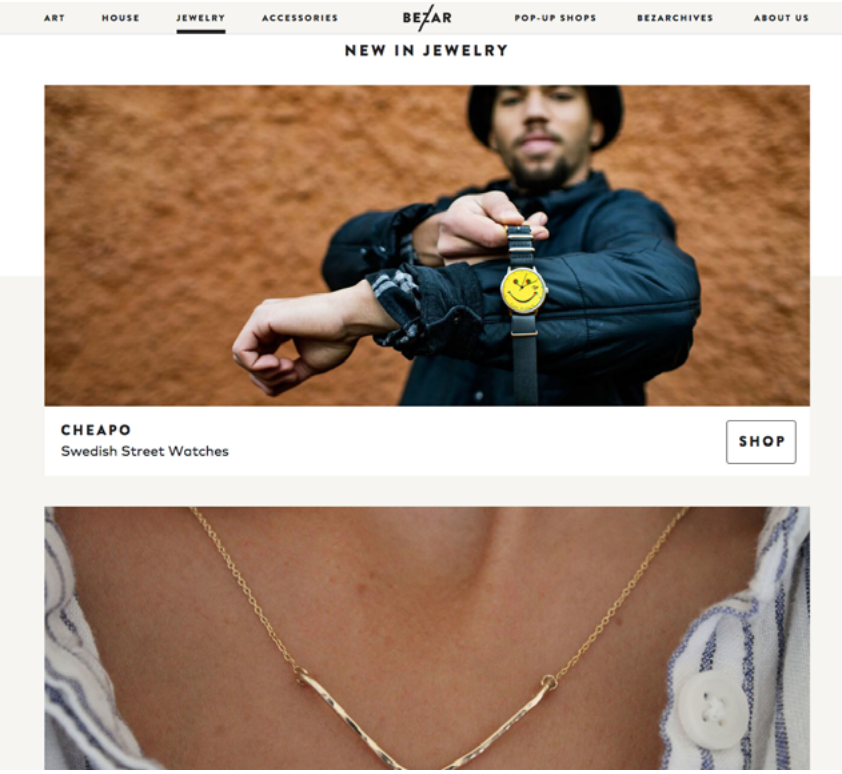
10. awwwards



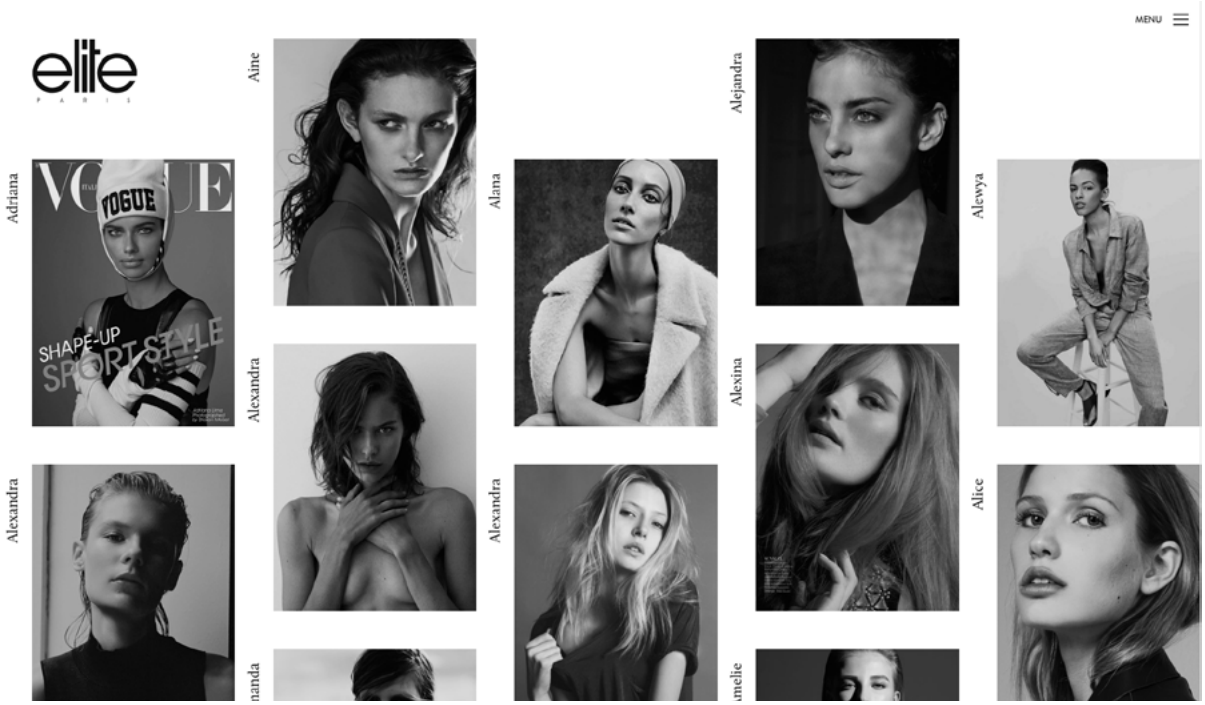
11. ZURB PatternTap



12. Bezar



13. Elite Model Management



UXPin

Everything you ever wanted in a **UX Design Platform**

- Complete prototyping framework for web and mobile
- Collaboration and feedback for any team size
- Lo-fi to hi-fi design in a single tool
- Integration with Photoshop and Sketch

Start using it now!

www.uxpin.com