

Navigation Design

High-level navigation 76

A counterpoint: Mark Hurst and the page paradigm? 78

Low-level navigation 81

Utilities 83

Button or link: “Your Account” utility link on Amazon.com? 84

Breadcrumbs 84

Site map 87

Associative links 88

The browser 91

Dynamic menus 93

Link hover effects 97



Navigation—arguably the most powerful metaphor of the web—has become so much a part of the online experience that we no longer even realize that *technically* we don’t “go” to a web site, but rather the site comes to us. But even though it is technically inaccurate, the concept of going to places on the web is more natural to us because we can form a conceptual model of navigating the web by likening it to something familiar—moving around in a physical space.

The implications of this metaphor in design are profound. In order to let the users be successful in applying the navigational model to browsing the web, the attributes of navigating a physical space—landmarks, paths, “you are here” signs, and even tracks (or breadcrumbs)—must be implemented in the web navigation model.

Components of navigation systems

It is important to remember that no matter what shape a link takes on, it is just a link—a pointer to another location within the site or the web. There are basically two major types of links: associative and structural.¹ Associative links point to more information about something that’s related to the user’s current context. Structural links, on the other hand, “systematically point to other levels of the site structure as well as to siblings or children in a hierarchy.”² Unlike associative links, structural links can enable the user to form a mental map of the site’s content architecture, not just its hypertext structure.

Depending on where and how it appears, a structural link can either stand on its own or be a part of high-level navigation, low-level navigation, utilities, breadcrumbs, a site map, etc. While structural links are what is commonly referred to as navigation, both types of links along with tools for faceted browsing, winnowing, searching, filtering, sorting, paging and browser-provided controls constitute a complete navigation system.

¹ While Jakob Nielsen identified a third type of a link—embedded—it is nothing more than a subtype of an associative link, as I will explain later.

² Nielsen, Jakob. “Designing Web Usability: The Practice of Simplicity” (New Riders, 2000), p. 195.

High-level navigation

Also known as global nav, high-level nav allows users to “move” within the site’s structure. It shows which section or category the users are “in” (a “you are here” indicator) and what other sections/categories are available directly from this page. Simple enough.

Here are a few attributes of a good high-level navigation tool:

- › **Consistent between pages:** use the same design, size, placement, etc. This is a no-brainer for all pages except the home page and the process pages like the checkout. However, there is merit to making the high-level navigation on the *home page* look as close to the high-level nav on the subpages as possible. It simply does not make sense to ask the user coming from the home page to adjust to a new high-level nav layout.

JCPenney.com’s home and “interior” pages are an extreme example of this. If you compare their home page and subpages you will notice that many of the same elements are in different locations, have a different visual treatment, or both:

JCPenney.com’s
home page:



A typical interior page at JCPenney.com:

This page shares most of its structural links with home page, but most of them have either “migrated” to a different location or taken on a different look. The two pages are so different, in fact, that it seems jcpenny.com’s designers tried to make them as different as they could on purpose.



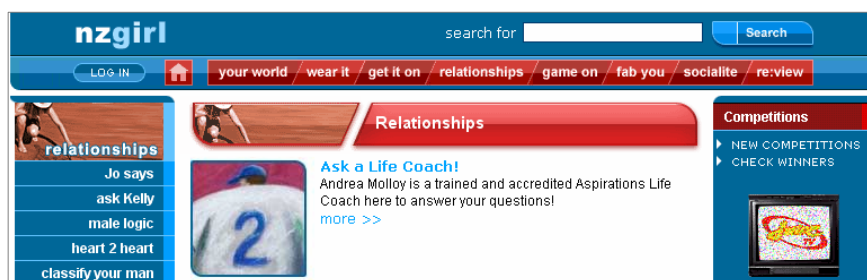
- › **Loads quickly.** This guideline applies mostly to sites that use Flash for navigation. Flash navigation may load quickly enough over a broadband connection, but over a slower connection the download delay can keep navigation from displaying until long after the rest of the page has loaded. Why is it important for navigation to load quickly? After all, isn't it the content that is most important to the users? The answer is two-fold:
 - quick-loading navigation is important for the same reason as quick-loading pages are: to create a smooth browsing experience, where system response times are short enough that the user's focus remains on the task.
 - one of the purposes of navigation on a web site is to show “what's there.” Sites that take too long to do this will aggravate their users.
- › **Has clear “you are here” indicators.** In most cases giving the current location a graphical treatment that is different from all other options in the same group (and removing the link) works well. “You are here”-indicated nav link must stand out, but at the same time look similar enough to its siblings. The indicator is clearly too subtle at bloomingdales.com (hint: it's “men”):



A better solution for Bloomingdales.com would have been to place a pointer next to the nav link. Here's one possibility:



Nzgirl.com paints itself into a corner as far as “you are here” signs go with alternating backgrounds for its primary nav links. As a result it chooses to avoid the signs altogether:



Tabs are arguably the best to use if clear “you are here” signs are required: if they're drawn correctly it is clear which one is selected (see the petsmart.com example on page 79 ahead). See Steve Krug's “Don't Make Me Think” for instructions on how to draw “tabby” tabs.

A counterpoint: Mark Hurst and the page paradigm?

Several years ago I read the “Holiday ’99 E-Commerce” report by Creative Good’s co-founder, Mark Hurst, where he introduced the page paradigm. It is based on the observation that “most customers go page by page through the shopping experience, not thinking too much about pages they’ve seen previously.” The paradigm states that “On any given page, customers will do one of two things: EITHER click on something that appears to take them closer to their goal, OR click the Back button.”

The paradigm, as it is defined by Hurst, is not surprising given the studies showing time and again that:

- most online user behavior is highly goal-driven;
- most users don’t make optimal decisions, but satisfice, choosing the first option that looks promising.

What is surprising, however, is that Mark’s observations led him to the conclusion that “navigation bars (‘navbars’), toolbars, and other features [...] are totally irrelevant to the customer.”

Does the fact that users pay little attention to navigation tools negate their usefulness? Absolutely not! In fact, one of the virtues of a well-designed navigation tool is unobtrusiveness: it fades from the users’ consciousness, letting them concentrate on the page’s main content and their task.

In order to understand the role of structural navigation keep in mind that what goes on in our subconscious as we move around the site is similar to what happens when we move around in physical space: we form a mental model of the site just like we form a model of our environment. We simply can’t avoid it—it is one of those innate human processes. Therefore, a set of navigational tools that clearly shows how the site is organized, where we are within it, and where we have been is instrumental in our understanding of the site’s information space.

I must point out that Mark’s unorthodox conclusion is not completely unfounded. Remember that it is based on user testing of eCommerce sites, where navigating often takes a back seat to another type of behavior: performing actions (or completing processes). On an eCommerce site, finding a product and going through the checkout are the corresponding examples. In fact, on most sites a user can find a product and place the order while avoiding structural navigation tools altogether: search for a product from the home page (or click a featured product to speed things up), click the one you want in the list of search results, place it in the cart, and complete the checkout. Judging from my own experience user-testing eCommerce sites I suspect that this what might have happened during at least some of Mark’s tests.

The bottom line, then, is that structural navigation tools are essential in letting the users make sense, or form a good conceptual model, of a web site’s information space.

- › **Reveals the site's hypertext structure and content architecture.** In most cases navigation naturally reveals both, because it is based on them. However, the visual appearance must reinforce the logical relationships within the site's hypertext and content. The design of petsmart.com's high-level nav clearly places visual and logical organizations at odds, as the caption on the right explains:



High-level nav at petsmart.com:

Because the tab and the area below it share the same background color, there is an implied logical connection between the tab title and the links below. However, only the “Answers” link goes to a page tailored specifically to each kind of pet; the others link to pages that are the same regardless of which pet tab is chosen. Remember false affordances from Chapter 1? This is an example.

One of the clearest ways to show hierarchical relationships is by using a tree, because trees are the most common way to visualize a hierarchy both online and off. In fact, the relationships implied by a tree presentation are so powerful they can indicate a hierarchical relationship even where the other visual cues fail to do so. Notice how indentation makes the hierarchical relationship among rides.org's nav items clearer:

Original:

Home
Employer Services
Commute Options
Carpool
Vanpool
Transit
Bike
Walk
Telecommute
Commute Incentives
Lots, Lanes & Links
Commute Research
Commute Calculator

Add indents and reinforce the “you are here” indicator.

Revised:

Home
Employer Services
Commute Options
› Carpool
Vanpool
Transit
Bike
Walk
Telecommute
Commute Incentives
Lots, Lanes & Links
Commute Research
Commute Calculator

Clarifying the hierarchy at rides.com.

The problem here is that, based on the visual cues alone, it is difficult to notice the relationship between “Commute Options” and the links below it. Indenting the six commute options links makes the relationship clearer: you don't even have to read the labels to tell how the blue links are related to the red link.

A further enhancement could actually come in the form of a de-enhancement: changing the ‘Carpool’ link's color and background back to match the other five items:

Commute Options
› Carpool
Vanpool
Transit
Bike
Walk
Telecommute

The point: don't layer many “you are here” indicators onto a single control. Use as few as possible while preserving clarity.

Of course, as Bob Baxley notes in “Making the Web Work,” trees have a few disadvantages such as (a) shifting visual relationships among the items (due to expanding/contracting nature of trees) and (b) limited scalability (due to the tree extending below “the fold”). While these disadvantages may make trees more appropriate for smaller sites, it is up to the designer to determine whether the clarity of a tree presentation outweighs its drawbacks.

- › **Is unobtrusive.** The purpose of high-level navigation is to get a user to the content or a beginning of a task. Once there, the user should be able to “tune out” navigation to better concentrate on the content or the task. Some studies show that users already tend to look at the page content rather than navigation when a new page loads. All we have to do here is help them by creating a clear visual separation between navigation and content. An older design of getconnected.com achieves this by setting the high-level navigation against a dark background, while keeping the main content area’s background white:

Phone detail page at getconnected.com:

The dark, desaturated colors used for the primary navigation background keep it visually separated from the content.

GETCONNECTED™
Compare & Shop for Great Deals

Internet | Wireless | Long Distance | Local | TV | Handhelds

Home : **Wireless Phones** : View Wireless Phone Detail

Nokia 3360 (AT&T)

Price With Plan: \$99.99
Total available rebates: **-\$130.00**
Your Price with Plan & Rebate: **Free**

› [view accessories for this phone](#)

Description
Introducing the Nokia 3360 phone. Change its style and attitude at a moment's notice. Personalize the way it rings. The way it looks. Change everything. Or not. It's all about you. One-button Navi Key that provides quick, one button access to features. On-screen help, large backlit screen. Phone book will store up to 100 names and numbers, 3 games, clock with alarm, paging. Versatile range of accessories.

Features

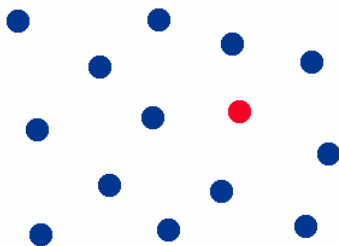
- Changeable front and back color covers
- Alarm clock
- 25 ring tones
- Up to 10 downloadable ring tones
- Create your own ring tone
- 3 games featuring vibra-shock
- Includes: Standard NiMH Battery, AC Adapter, Headset

Rebate Information

- Receive an \$80 mail-in rebate from AT&T when you purchase any AT&T Wireless service plan \$29.99 or higher with a 2-year contract. Or receive a \$40 mail-in rebate from AT&T when you purchase any AT&T Wireless service plan \$29.99 or higher with a 1-year contract. *Expires 1/26/02
- Receive a \$50 mail-in rebate from GetConnected when you purchase any phone and activate any service plan \$29.99 or higher through GetConnected. *Expires 11/30/01

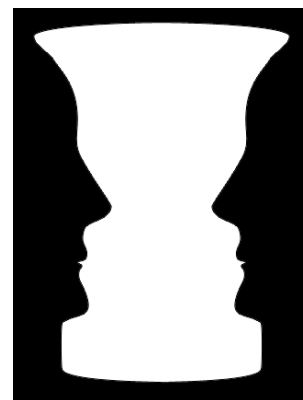
Specifications

Color	Black
Talk Time	210 min.
Web Features	No
Text Messaging	Yes
Internationally Compatible	No
Dimensions	4.5 in. tall
Weight	4.8 oz.
Mode	Tri-Mode
Standby Time	260 hrs.
Technology	TDMA 1900MHz



Contrast not only reinforces the logical distinction among navigation and the content, but also allows the user to locate each one very quickly. In fact, given enough contrast, the boundary between navigation and content area can be detected preattentively, that is, without the need for focused attention. This “preattentive processing” is what allows you to immediately detect the red dot in the image on the left.

In the case of getconnected.com nav design, there is an additional advantage. Believe it or not, there a Gestalt principle of figure-ground, particularly something called “figure-ground reversal,” at work here. Notice that all of navigation text is light on a dark background (with the exception of the breadcrumbs), while the body text is dark on a light background. Since the text letters are much smaller than the background on which they appear the Gestalt principle of figure-ground explains why they will always be perceived as “figure” while their background will always be “ground.”



“Figure-ground reversal”—where it is difficult to determine which area is figure and which is ground—is something that usually happens when the areas occupied by figure and ground are similar in size. The drawing on the right takes advantage of this ambiguity. The thing that interests us in the drawing is not merely that you can see either a vessel or two faces, but that you can’t see them at the same time (just as your attention can only focus on one thing at a time). That is, there is a perceptual shift that occurs as you alternate between seeing the faces and the vessel. Here’s what I mean:

This example illustrates the perceptual shift that occurs when you switch between reading white text on dark background and dark text on white background.

On getconnected.com this perceptual shift prevents the viewer from “accidentally” reading the navigation text until she makes a conscious decision to do so.

Low-level navigation

“If the purpose of high-level navigation is to allow users to establish a context, the purpose of low-level navigation is to allow them to get around inside that context.”³ The exact place in the site’s structure where high-level nav ends and the low-level nav begins is a gray area. For example, on an eCommerce site the links to products on a product list page can be considered low-level navigation. Here “the context” is the product category. On the other hand, if the context is the actual product, then the high-level nav are all the links you need to click to get to the product detail page, whereas the low-level nav are the links that let you find out more about that product.

³ Baxley, Bob. “Making the Web Work: Designing Effective Web Applications” (New Riders, 2003), p. 214. This book is by far the most comprehensive and best-organized treatment of web interface design I have seen. Highly recommended.

Regardless of where you draw the line, low-level navigation is logically closer to the actual content than the high-level nav. This difference should echo in the appearance of the two navs. For example, since contrast is often used to differentiate elements, the lack of contrast can be used to relate them.

Crutchfield.com's product detail page is a great example of this:

Product detail page at crutchfield.com:

The dark, desaturated colors used for the primary navigation background keep it visually separated from the content.

Another point to be made here is about the “you are here” indicator in the primary nav: the highlighted option looks so different from the other items in the same list that the connection between them is almost lost. A simple pointer would have worked better here.

High-level nav – more contrast with the page background.

Low-level nav – less contrast with the page background.

Here, the page background contrasts more with the high-level nav buttons (along the left edge) than with the low-level nav (the tabs below the product photo and the “buy” button). This creates a visual hierarchy that reinforces the logical relationship that exists among the different levels of navigation and the page content.

Another thing you’ll notice in the screenshot is the use of tabs for low-level nav. Tabs work well to show different views of the same product because this is similar to how they have been used in desktop software. The only thing I would change here is to make the tabs look a little “tabbier”:

“Tabbier” tabs at crutchfield.com:

I have “tuned” Crutchfield’s tabs to make it clearer which tab is selected and which are “hiding” behind the current view.

Amazon.com's low-level nav on the product detail page takes the form of a simple, boxed-in list of links down the left side of the page (the "book information" box). Using tabs here would have been particularly impractical because (1) tabs are already used for high-level nav, and (2) there is so much content at the top of the page that it would push the tabs below "the fold," drastically reducing their effectiveness.



Product detail page at amazon.com.

Utilities

Sometimes called courtesy navigation, utilities usually take the form of a small nav bar or a list of links consistently pointing to locations outside the site's primary structure. They are usually less prominent than high-level nav and appear even further away from the content area of the page—usually at the very top of a page—visually reinforcing their lack of a strong relationship to anything else on the page. You can clearly see the special appearance and placement of the utilities at crutchfield.com (previous page).

Amazon.com goes a step further in the design of its utility links: not only are they different and out of the way, but some of them even get a distinct visual treatment. For example, the "your account" link looks like a button and the shopping cart link is adorned with its very own shopping cart icon. These visual distinctions make the links easier to locate at a glance. I bet that after going to "your account" a few times you won't even need to read the words on the button. Instead you will use your spatial memory to locate the utilities on the page and will almost preattentively choose the correct button (because of its contrast with other links and buttony appearance).

Button or link: “Your Account” utility link on Amazon.com?

The classical distinction between links and buttons is straightforward: as Eric Eaton puts it in his book “Designing Web Site Interactive Elements,” “a button accomplishes something, while a link references something. Clicking on a button should result in an expected progression through a task, or the initiation of a process that is directly related to the interface.”

The important thing to realize here is that this definition is based on the distinction as seen from the point of view of the user. That is, from the user’s point of view, some work is done (be it on the part of the system or the user) when he clicks a button, whereas no work is done (merely a movement to a new page) when he clicks a link. Link and button labels reflect this as well: link labels should not contain verbs, whereas button labels must.

This definition naturally makes for some gray areas. Enter the “Your Account” button on amazon.com. Based on the button’s title alone the user might expect to simply move to a different area of the site, in which case it should be simple link. However, the activities in the “Your Account” section (tracking orders, changing settings, etc.) are totally different from browsing for products. Therefore, a case can be made that the button signals a departure from browsing for products, and the initiation of an account management *process*.

What do I think? The link to “Your Account” should *not* look like a button, just like most links that get you *to* the start of a process. However, there is merit to giving it a distinct visual treatment to make it easier to notice at a glance.

Breadcrumbs

Breadcrumbs is a generic term for a series of links attached to an object or a page that either (a) indicate how it fits into the site’s structure and/or (b) what the user clicked to get to the current page. Keith Instone, an information architect at IBM and a breadcrumb guru, has grouped breadcrumbs into the following three categories:

- › **Path breadcrumbs**, which are the only ones bearing a direct resemblance to their namesake—Hansel and Gretel’s breadcrumbs. Path breadcrumbs tell the user how he got to the current page by showing the names of the last handful of pages he visited. While this sounds like a useful tool in theory, path breadcrumbs are rarely used because:
 - browsers do a pretty good job of keeping track of the pages the user visits;
 - most users don’t even care where they’ve been on the site;
 - path breadcrumbs confuse users who are otherwise accustomed to seeing the overwhelmingly more popular location breadcrumbs.

Therefore, I advise against using path breadcrumbs.

- › **Location breadcrumbs**, which show the location of the current page within the site's structure. Because every item in the breadcrumb chain is linked, they also allow the user to move up to higher levels in the hierarchy. If a level in a web site's content hierarchy can be likened to a floor in a building, then the location breadcrumbs work like an elevator.

Design considerations for location breadcrumbs:

- Place them above the page content, but avoid the top edge of the page, especially above ad banners—no one will notice them there.
- Include a “home” link at the beginning of the breadcrumb chain. I prefer doing this instead of using the words “You are here,” because the home link not only identifies the chain as breadcrumbs almost as effectively, but is also a useful link in itself (see an example from yahoo.com on the next page).
- Separate the breadcrumb items using a symbol that visually reinforces the logical relationship among them. A greater than sign (“>”) or an arrow (“v”) work well for both hierarchies and faceted browsing systems.
- Prevent the chain from wrapping if possible, but don't eliminate the items or abbreviate them: this would make the breadcrumbs much more confusing than any amount of wrapping. If you can't prevent wrapping, control it by keeping the words in multi-word items together with the `<nobr>` tag.

Hungry for more breadcrumb design tips? You'll find a very comprehensive list in Steve Krug's “Don't Make Me Think.”

- › **Attribute breadcrumbs**. These are basically a special case of location breadcrumbs: they indicate the position of a unit of content—rather than an entire page—in a hierarchy. One situation where this can be particularly useful is on a search results page, where attribute breadcrumbs can point to the category where each matching item lives. This lets the user locate and explore the entire category to which the item of interest belongs.

Multiple chains of attribute breadcrumbs can be used for items that “live” in several categories at once. The items in Amazon.com's ambiguous, highly cross-listed product hierarchy lend themselves nicely to multiple attribute breadcrumb chains. For example, the product detail page for the item pictured on page 83 has two sets of attribute breadcrumbs:

Look for similar books by subject:

Browse for [books](#) in:

- [Subjects](#) > [Computers & Internet](#) > [Web Development](#) > [Internet Commerce](#) > [Web Site Design](#)
- [Subjects](#) > [Computers & Internet](#) > [General](#)

“Envisioning Information” has *nine* attribute breadcrumb chains (see page 66 in the previous chapter). In fact, because most products on

amazon.com “live” in several categories, there are no location breadcrumbs: which chain would you choose?

Unlike location breadcrumbs, attribute breadcrumbs still work—though not as well—if only a portion of the complete path to the top-level category is shown. This approach can save considerable screen “real estate” while still tapping into the awesome, yet inconspicuous cross-selling power of attribute breadcrumbs.

Yahoo.com takes an interesting approach to attribute breadcrumbs in the “Directory Category Matches” section of their search results pages. Instead of displaying the entire chain—which can take up two full lines as location breadcrumbs—they create a shorter one using the most salient two or three keywords. For example, here are some directory category matches for “movie reviews”:

Directory Category Matches for “movie reviews” at yahoo.com:

Yahoo.com’s designers have abbreviated the actual path to each category from the home page to arrive at these compact attribute breadcrumbs. Considering the alternative—a long and unwieldy chain—these breadcrumbs work well.

Search Results movie reviews Search Advanced Search About These Results powered by hp

Your search: **movie reviews** Search in: [The Web](#) | [Directory](#) | [News](#)

Directory Category Matches 4 - 19 of 19 [First Page](#) | [Previous 3](#)

- [Public Access TV Shows > Movie Reviews](#)
- [Movies and Film Reviews > Reviews for Parents](#)
- [Movies and Film Reviews > Capsule Reviews](#)
- [Home Video > Reviews](#)
- [Movies and Film Reviews > Review Hubs](#)
- [Movies and Film > Humorous Reviews](#)
- [Movies and Film Reviews > Counterpoint Reviews](#)
- [Hong Kong Movies > Reviews](#)
- [Movies and Film Soundtracks > Reviews](#)

BARNES & NOBLE
Shop at B&N.com
• [Bookstore](#)
• [Music](#)
• [DVD & Video](#)
• [Textbooks](#)
[Free Shipping on Orders of 2 or more. See details.](#)

“Movie and Film Soundtracks > Reviews” at yahoo.com:

Or is it “[Home > Entertainment > Movies and Film > Film Music > Soundtracks > Reviews](#)”?

It’s both! The two-link attribute breadcrumb chain from the “directory category matches” acts as a page title here. Meanwhile, the location breadcrumbs are doing what they are supposed to do: show the exact path to this page from home.

Yahoo! Directory Movies and Film Soundtracks > Reviews the Web just this category Search Advanced Search Help powered by hp

[Home](#) > [Entertainment](#) > [Movies and Film](#) > [Film Music](#) > [Soundtracks](#) > [Reviews](#)

Inside Yahoo!

- [Yahoo! Movies](#) - showtimes, trailers, photos, reviews, and more.

Site Listings

- [Cinemusic Online](#) - with composer interviews, soundtrack reviews, mystery clip contest, and more.
- [Film Score Rundowns](#) - provides cue-by-cue analysis of film and television scores. Organized by composer.
- [Filmtracks Modern Soundtrack Reviews](#) - reviews of current film scores, tributes to the best modern composers, and information about collecting soundtrack CDs.
- [Movie Music UK](#) - film music reviews, composer biographies and more.

More Yahoo!

[Featured Category: Anger Management](#)
Jack is back.

[Trailers and Clips](#)
get a sneak peek.

[Y! Shopping: Video and DVD](#)

[Terminator 3](#)
photos of Ah-nold and company.

[Y! Buzz Index: Movies](#)
popular searches

The only possible source of confusion here could be that the attribute and location breadcrumbs don't match, but from my own experience I can't see many people getting confused here. In fact, a user moving quickly through these pages will likely pay no attention to this discrepancy: as long as what they clicked matches where they ended up, they are happy. If the page is not what they thought it would be, they can always click "back," search again, or use the location breadcrumbs to go up a few levels.

Another thing to notice about yahoo.com's attribute breadcrumbs is that the entire chain rather than the individual element is linked. In general, this should be avoided because it reduces the ability of attribute breadcrumbs to allow the user to jump to any level in the breadcrumb chain. For example, on amazon.com each attribute breadcrumb link is individually linked for maximum flexibility.

Site map

A site map visualizes the organization of a site by showing the logical relationship among the site's top-level sections and subsections. While Mark Hurst would argue that "customers do not care about site maps,"⁴ I believe a site map is an important tool for a user who is lost or simply wants to get an overview the entire site.

If you decide to use a dynamically expandable outline-style (or tree-style) site map make sure that at least one level of categories is show by default. Here's an example of what *not* to do from usabilitysciences.com:



The site map at usabilitysciences.com: Completely collapsed site map + absent page title + no "you are here" indicators = where am I? In addition, notice how much more difficult the nav bar labels are to scan quickly because their text is slanted.

⁴ Mark Hurst. "Holiday '99 E-Commerce" report, www.creativegood.com.

Self-popping vs. user-prompted popup windows.

Users hate windows that pop up by themselves. What's more, users are so used to these things being useless that many will close the popup without reading its contents.

This does not mean that all pop up windows get booted by the user as soon as they appear. For example, I have never seen a user "kill" a window which appeared as a direct results of the user's action, especially when it was expected. The key here, then, is to make sure the user can anticipate the popup window and see the connection between the window and his own actions.

One of the most important things a site map should do, but rarely does, is to tell the user where he is on the site. This, of course, requires that the site map appears in the same context as the current page. Sending the user to a site map "page" means the user will leave the current context and will always be on the site map page when viewing it. The solution is to use a popup window for the site map: the user stays on the page (context is preserved) and the page's location can be easily passed to the site map window.

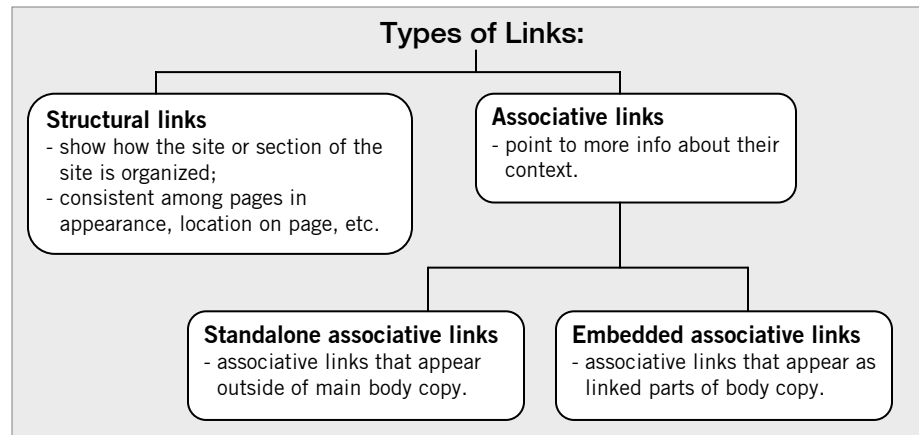
There are many more things to consider when creating site maps. You can find quite a handful of them in Jakob Nielsen's "Site Map Usability" alertbox at www.useit.com/alertbox/20020106.html.

Associative links

You can easily distinguish between structural and associative links by answering the question: does this link help me understand how this site is organized? If it does, then it is a structural link, if not—most likely the link's target is on a different "branch" of the hierarchy or a different site altogether—then you have an associative link. If the associative link is a part of a sentence or a sentence fragment, then it is an embedded associative link. If it is not, then it is a standalone associative link.

Types of links

Breaking structural links into standalone and embedded is not necessary since the vast majority of structural links are standalone. In fact, you should not embed structural links into body text as it will create inconsistencies in placement and appearance of these links among pages.



Here are a few tips for creating better associative links:

- › **Embedded vs. standalone.** The first thing you have to decide is which links should be embedded and which should be standalone. Making this decision in advance will enable you to make the most out of the advantages of each link type. So instead of simply taking a page's text content and arbitrarily linking juicy terms and passages, consider whether a link should be left inside a paragraph or pulled out of it.

Here are some advantages of standalone over embedded associative links:

- More layout flexibility: they can be placed in a box, grouped neatly, given a group title, etc.
- Their labels do not have to fit into the sentence where they appear, making labeling more flexible.
- They are easier to scan in large numbers, provided they are aligned properly. Here's an example of what *not* to do from lowes.com:

Hardware	Heating and Cooling
Cabinet Hardware , Child Safety and Protection , Door Hardware , Door Hardware Accessories , Fire Extinguishers & Escape Ladders , Flashlights , Garage Door Openers & Accessories , General Purpose Batteries , Ladders , House Numbers , Mailboxes and Accessories , Padlocks , Safes , Smoke and Carbon Monoxide Detectors , Window Hardware	Air Conditioner/Portable Fans , Ceiling Fans , Cleaners & Gaskets , Decorative Outdoor Heat , Electric Fireplaces , Electric Heaters , Evaporative Coolers , Evaporative Cooler Accessories , Fireplace Access. , Fireplace Grates , Furnace Filters , Gas Heater Accessories , Gas Logs Access. , Glass Fireplace Doors , Hearth Sets/Accessories , Heating , Heating Accessories , Kerosene Accessories , Log Racks , Portable Kerosene Construction Heaters , Portable LP Construction Heaters , Vent-Free Gas Fireplaces , Vent-Free Gas Heaters , Vent-Free Gas Stoves , Vented Gas Logs , Ventilation , Stove Boards , Wood Heaters

Product category links at lowes.com.

Had each link been placed on its own line and the list left-aligned, it would have made the entire list much easier to scan. In addition, individual link labels should not be split up by wrapping. Any space savings achieved here are negated by this awkward arrangement.

Note: even though these are actually structural links, I am using them to illustrate a point.

Embedded links have their advantages:

- They unclutter a page by making portions of body text serve double duty;
- They are explained by the body text that surrounds them.

While these factors can't solely dictate which type of link should be used—the main body content determines that to a great extent—you can use them to make choices in situations where either type will work. Here's an example:

• Buy a **Motorola V60i or 120t** and activate new service on any AT&T Wireless calling plan and you will be eligible for a **\$50.00 mail-in phone rebate** from AT&T Wireless. Please note: Once the rebate fulfillment form has been submitted, you can not return your Motorola V60i phone. **Click [here](#) to view the rebate form.** You will need Adobe Acrobat Reader to view and print PDF files. Click [here](#) to download a free copy of the Adobe Acrobat Reader. *Offer expires: 05/10/2003

A piece of a plan detail page at getconnected.com.

Taking the two links out of the dense body text and rearranging what's left yields:

• Buy a **Motorola V60i or 120t** and activate new service on any AT&T Wireless calling plan and you will be eligible for a **\$50.00 mail-in phone rebate** from AT&T Wireless. Please note: Once the rebate fulfillment form has been submitted, you can not return your Motorola V60i phone. *Offer expires: 05/10/2003

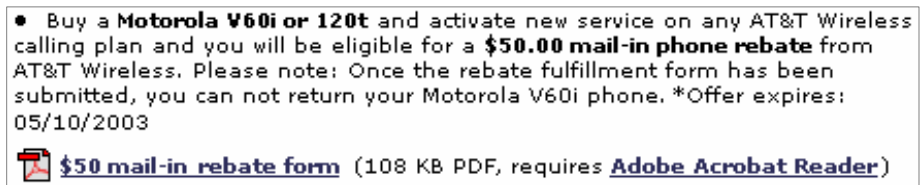
View [the rebate form](#) (requires [Adobe Acrobat Reader](#))

Taking the links out required renaming them so that they could stand on their own. In fact, “here” is about the worst label a link can have because it carries no information, making it necessary for the user to read the surrounding text to understand where the link goes.

Two other things to note in this example:



- Only the words “the rebate form” are linked. This is consistent with the convention of using nouns in link labels, but verbs in button labels. I could have also omitted “View,” but leaving in creates a prompt for the user to click the rebate form link;
- The context of the “Adobe Acrobat Reader” link makes it ok to omit any explanatory text—it is clear that the link leads to more information about the Reader. A link title⁵ can be used to explain that a copy of the Reader can be downloaded at the link’s destination in case the user fails to make the connection.

An even better approach in this particular case would be to clearly indicate that the “the rebate form” link’s target is a PDF document rather than an HTML page:



The Adobe Acrobat document icon, the file size, and document type tell the user that the link goes to a PDF document rather than a plain HTML page.

In addition to indicating the format of the document on the other side of a link, you can also mark the following two types of links to manage users’ expectations:

- *links that open in a new window* can be marked with a little window icon placed before or after a link:  [\\$50 mail-in rebate form](#). An alt tag-driven tooltip can be used to explain the icon to anyone seeing it for the first time.
- *links that lead off the site* can be marked with a similar type of icon: [Adobe Acrobat Reader](#) .

While these icons are certainly not universal, they will quickly become clear if applied consistently throughout the site.

⁵ A link title is a term used by Jakob Nielsen to refer to a tooltip that pop-ups up when a mouse pointer hovers over an anchor whose “title” attribute is specified.

Before leaving the discussion of links let me mention one more issue that affects both embedded and standalone links: wrapping of multi-word links. With embedded links, wrapping places the link pieces so far apart that they are likely to be interpreted as two separate links, especially if both pieces can be links by themselves. The Adobe Acrobat Reader link example in the bullet at the bottom of the previous page is an example of this. Legibility of standalone links suffers from wrapping as well, especially if there is a list of links with no leading to clearly separate each one. It took one user 15 seconds to realize that “Studies” was a part of the “Gay and Lesbian Studies” link rather than a category heading at powells.com (see illustration on the right). Fortunately for web designers it is easy to control wrapping in HTML by placing the link inside the `<nobr></nobr>` tags.

There are, of course, many more things to consider when creating text links. You can find many useful link tips in Eric Eaton’s “Designing Web Site Interface Elements.”

The browser

As a navigation tool, the browser is the one most familiar to the users because they “take it” to every site they visit. Browser controls not only help users get around a site, but also act as a fallback if the site’s navigation fails. This is why the sites that hide and disable browser features often suffer from poor usability. Here are a few examples of how hiding or changing various parts of the browser window impacts a user’s experience:

- hiding the toolbar hides the one thing that web users click most often: the “Back” button;
- hiding the menu bar prevents saving web pages and changing internet/browser preferences;
- hiding the address bar prevents the user from knowing where they are on the web as a whole and from going anywhere other than the places they’ve bookmarked;
- hiding the status bar hides the addresses of the links that the user points to with the mouse and the security status of the current page (the padlock icon for SSL-encrypted pages);
- making the browser window non-resizable prevents the user from resizing to better fit the content. For instance, the product view popup at 800.com was non-resizable, which meant I could not see the complete product photo even though I had more than enough screen real estate to display it without scrolling;



Book category list at powells.com.

Tight leading and history>us/word indentation conspired to make one user conclude that “Studies” is a category heading rather than the piece of the “Gay and Lesbian Studies” link.

The non-resizable product view popup window at 800.com

Had this window been made resizable I would have been able to see the entire picture without having to scroll.



Defenders of hiding browser controls may argue that doing so unclutters the interface: why show the controls that the user will not need? However, there are at least three problems with this reasoning:

- Users like to be in control of their experience. They want to be able to stop a slow page from loading or quickly leave an offensive site. Disabling browser controls is a sure way to take that control away from them;
- Then there's is something that Jodie Dagleish would call second-guessing the users' needs. How can you possibly predict which browser features the user will want to use?
- The third is a technical limitation: you can't hide browser features à la carte—you can only hide the entire toolbar, menu bar, etc.—so you'll invariably end up hiding at least some features that the user may need.

The point is that the tradeoff between looks and denying the users most familiar functions is never in the user's favor. Therefore, my advice is to leave the browser alone.

About the only case I can think of where hiding browser features is warranted is in a pop-up window. Even then, only a window with none or a very limited set of navigation options qualifies. The toolbar and the address bar can be hidden. But the menu bar should still be shown in case the user wants to increase the font size in the window or print it. However, no pop-up window should ever be made non-resizable. In addition, it usually helps to include a “close window” button. You can make the button larger than the title bar buttons ([- _ x]), which are often too small to comfortably hit with the mouse, even for users who know what those little suckers do.

Adding interactivity

Dynamic menus

Pop-up menus have been very successfully used in desktop applications to neatly organize and hide groups of similar commands. It is no wonder, then, that they have made their way into web interface design just about as soon as the first JavaScript-capable browsers appeared. Of course, most web applications don't have enough commands on a single screen to warrant hiding them in menus. Therefore, dynamic menus are used mostly for navigation.

In a nutshell, dynamic menus:

- provide shortcuts to deeper levels in the site's structure and content hierarchy, and enable users to move laterally through it without having to go up one or more levels;
- reveal more of the site structure without requiring the user to click through to the lower levels in the hierarchy. Dynamic menus enable users with slow connections—for whom purely exploratory clicks can quickly become a very time-consuming proposition—to get an idea of the site's offerings much faster.

The following several pages present and discuss the most common design issues you'll have to consider when designing popup menus. Keep in mind that, as with many interface design decisions, your particular situation will dictate the applicability of the advice that follows.

- › **Show on click or on hover?** Does the menu open when the user clicks or hovers over the menu title? Bob Baxley notes that hover-triggered pop-ups run the risk of misinterpreting the user's intention: the user may want to merely pass the mouse pointer over the menu title to get to another area of the screen or to browser controls.⁶ On the other hand, one of the most fundamental conventions of the web is that a click takes a user to a different page. Therefore, click-to-show dynamic menus break this convention. Add the fact that clicks are costly (in terms of time) and what you get is a de facto standard for dynamic menus to *open on hover*. In fact, most dynamic menus I have encountered work this way. This is an example of how a new medium changes the implementation standards of an old tool, because in desktop apps dropdown menus are opened by clicking the menu title.

⁶ Baxley, p. 211.

So what about popping up menus when the user just wants pass over the menu title to get somewhere else? Well, there at least two things you can do to prevent this:

- don't put dynamic menus between areas of page content: menus are best placed on the outer edges of a page;
- build in a slight delay between the time when the mouse pointer hovers over a menu title and when the menu pops up.

Microsoft.com's dynamic menus sport both of these features:

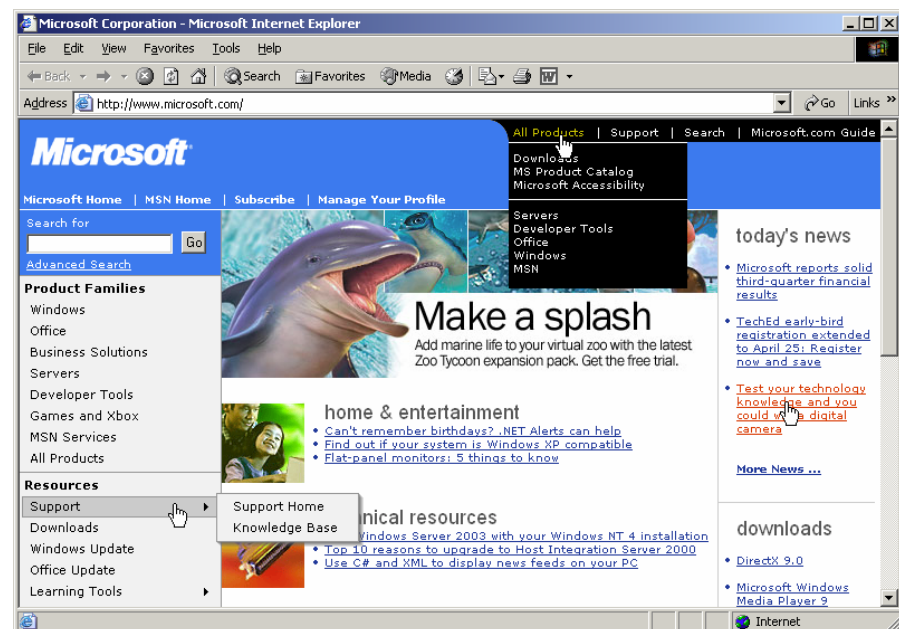
Dynamic menus at microsoft.com

See the little black triangles on the gray nav bar marking the nav items that have pop-ups? These pointers not only mark the items that have pop-ups—a must for delayed popup menus—but also tell the user where the popup will appear. Compare this with the absence of these clues on the black menu titles at the top of the page.

Actually, marking navlinks that are also menu titles is very important in setting user expectations. I have seen users make a decision to click a nav link (thinking that it is just a plain link), then point to it and click it so quickly that when the menu pops up, they couldn't react to it fast enough to stop their finger. Many of them then end up clicking the "Back" button and reading the menu contents; even if the same set of menus appears on the page where they landed by clicking the menu title!

Notice also, that the menu background color differs from the page background color. The gray menus also add shadows, creating an impression of three-dimensional space. While it is generally not a good idea to use 3-d on the web, in this case the 3-d treatment enhances the menus, making them easier to differentiate from the page itself. Of course dynamic menus aren't the only ones that benefit from this: 3-d can be also used with tabs to raise their "tabbiness" quotient.

Another interesting thing to note here is that the items within a single menu are not separated (for example, by lines). This is a good approach because it reduces the clutter by doing away with elements that don't carry any information. Hover-highlighting reassures the user that he successfully pointed at the item of interest before clicking it.



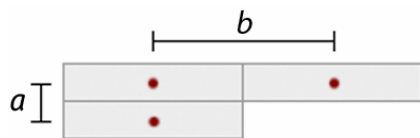
The gray menus on the left and the black menus on top are placed along the outer edges of the page, out of the way of the content. Gray menus also have about a ½ second delay. The only thing I would change here is add the same delay to the black menus: even though these menus are out of the way of most browser controls, a little delay wouldn't hurt here as well.

- **Hide on hover off or on click elsewhere?** Does the menu disappear when the user hovers off of the menu or clicks somewhere on the page? The answer to this question lies mostly in the way the menu opens: if it opens on hover then a user will expect it to close on hover *off*, but if it opens on click it should also hide on click off the menu.

Microsoft's gray menus not only show on slight delay, but also don't hide for about a ½ second after the mouse pointer is off the menu. If the user accidentally mouses off of the menu this gives the user a chance to move the mouse pointer back over the menu before it vanishes.

There is third alternative for hiding a menu that applies to menus that stay open as long as the you hold down mouse button while pointing at a menu item. Apple's Mac OS features this kind of menus. I advise against using this type of menu on the web simply because of the increased mechanical dexterity required to wield them: it is more difficult to point while holding down the mouse button.

- › **Arrange menu choices vertically or horizontally?** The factor that affects this decision the most is the distance a mouse pointer must travel to move between the menu items. To point to an item a pointer will, on average, be positioned at the middle of the menu item. In this case you can easily see why menu items will be faster to point to if they are arranged vertically rather than horizontally:



For items that are wider than they are tall the distance a pointer must travel between vertically arranged items (a) is shorter than between horizontally arranged ones (b). But don't make the menu items too narrow: their centers will be too close, requiring an unnecessary amount of pointing precision.

Therefore, I recommended vertical arrangement for menu items that are wider than they are tall. Not only will the items be faster to point to, there is also less chance for the pointer to veer off course on its way to the target.

There is one other good reason for vertical arrangement: if the items in the menu are left-aligned, they will be easier to scan quickly.

- › **Open down or to the side?** Obviously the arrangement of menu titles will dictate this. This is really a question of whether horizontal nav is better suited for dynamic menus than vertical. If you agree with me on the point in the previous bullet, it should be easy to see why a combination of a horizontal nav and down-popping menus are more usable: increased pointing speed and accuracy. Just don't make the menu items *too* small or they will be difficult to aim at with the mouse.
- › **Cascade menus or show one level only?** This one is a little more difficult. On the one hand, cascading several menus gives the user greater navigational flexibility: user can jump over one or more levels in the information hierarchy without actually visiting those levels. On the other hand, greater pointing precision will be required to successfully “navigate” down the menus. In this case, however, the functional benefits outweigh the increased pointing precision requirements. So go ahead and cascade those menus, but don't forget to run a user test with your target audience before you release. If you run a site about arthritis treatments your target audience will probably not want to have to point at anything at all.

- › **Animate menus?** While menu animation adds to a site’s “cool” factor, it will also slow down your users and make it more difficult for them to focus on reading the actual menu contents. Most types of sites do not benefit enough from the “coolness” of animated menus to warrant their use. While this may be hard for some designers to accept, especially if the client’s requirements call for an element of “cool,” try they must. I am not sure if this was the case for molecular.com, but they just don’t seem to get rid of animation in their navigation. In an earlier version of the site the sub-navigation buttons “flew in” from right to left as the user hovered over any of the six top-level nav items. In the current version the sub-navigation menu items successively fade in from left to right:



The results of a usability test run by Julie Ratner and the usability team at Molecular showed that users were put off by sliding menu items on the earlier version of the site.⁷ The fading menus fared better, supported by user comments like “The Flash is less intrusive [...] movement subdued and not quite as frantic” and “A much better—faster—feeling of being faster—tighter—not as distracting...”⁷ The common thread here is that users preferred less intrusive, less distracting menus.

There is one other thing about molecular.com’s menus worth mentioning: the fading menu routine runs not only when the user mouses around in the navigation area, but whenever a new page loads.⁸ This means that the user’s attention is briefly diverted from the page content to the nav area. Because the animation is not unobtrusive (an oxymoron?), this means that every time a page loads there is a totally unnecessary distraction that takes the user’s attention away from the page content.⁹

⁷ Julie Ratner, *Learning About the User Experience on the Web With the Phone Usability Method* in Julie Ratner (ed.), “Human Factors and Web Development” (Lawrence Erlbaum Assoc., 2003), pp. 141 – 143.

⁸ When the page loads not only do the sub-nav option fade in, but also the gray highlight for the current top-level nav item flies in from the left.

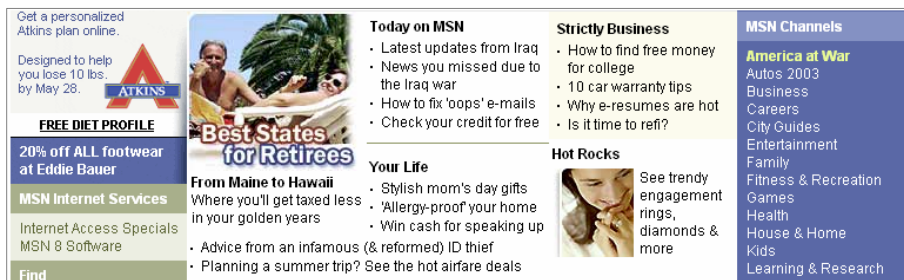
⁹ Windows XP taskbar animates whenever an application launches. This was driving me nuts until I finally found a way to turn off the animation.

Link hovereffects

Just before I wrote this section a colleague of mine asked me about link hovereffects—changing the appearance of links when a mouse pointer is hovered over them. Here’s an expanded version of my answer.

In general, statically underlined links [like these](#) are great for the obvious reason: you can clearly see that they’re clickable without having to interact with them. So if the links are already underlined and differ in color from the body text, then no hovereffects are necessary.

There are situations, however, when removing the underline and making it appear on hover is better. For example, pages that are extremely link-heavy look much less cluttered when their links lose the underline. Imagine how much busier MSN’s home page would look with all the links underlined:



A section of msn.com's home. Removing link underlines keeps the page from becoming too busy.

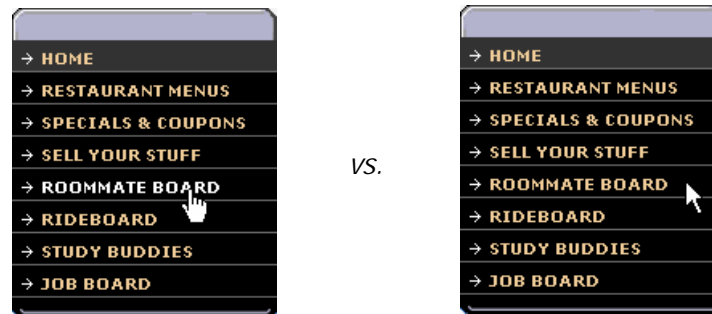
There are two other interesting aspects of human psychology and perception that support adding hovereffects to links:

- there is something magically empowering about hovering your hand over something and have it change without ever touching it. This can very well be a part of the reason why people like links that respond to mouse hovering.¹⁰
- as you point to things they become your locus of attention. In this case, your attention is focused not on the mouse pointer but on the link at which it is pointing. This means that a mouse pointer changing into a pointing hand is a weak a way of indicating clickability, because the change is so subtle: only the pointer's shape changes. This makes rollovers a welcome addition in situations where there may be confusion about what is clickable: the user doesn't have to take her attention off the link (to see if the mouse pointer changes shape) to find out if it's clickable.

¹⁰ I have seen this more than once in my own user testing as well as in tests carried out by others.

Hovereffects can also indicate more clearly when a mouse pointer has been successfully poised over a link—something that may not be clear if a non-underlined link is placed into a box: do you point to the word or anywhere in the box? High-level navigation at collegemenus.com is an example of this:

Only the text of the high-level nav items are collegemenus.com is clickable. This fact is made clear through hover-highlighting.



If you choose to use link hovereffects, here are two more tips to keep you on the right track:

- Make the change strong enough to be noticeable, but subtle enough not to distract. The high-level links at collegemenus.com achieve this balance. Microsoft.com's links' hover state is more prominent but works just as well (see page 94). However, the hover state of the links at powerwriting.com is way overdone and distracting:

Now you probably have questions about how Powerwriting works. So you'll find some [frequently asked questions](#) and specific answers in the FAQs.



Now you probably have questions about how Powerwriting works. So you'll find some [frequently asked questions](#) and specific answers in the FAQs.



Note, also, that the mouse pointer does not change in shape into a hand but gets a question mark attached to it instead. Nothing is gained by this except to raise two questions in the viewer's mind: "the pointer is not a hand; is this really a link?" and "what does the question mark mean? Is this a special kind of link?" To use Steve Krug's favorite expression: "Don't make me think!"

- If the links are already underlined don't remove the underline on hover or you'll make the user think again: "Hey, what's happening, I thought this was a link!"