

Form Design

Overview of the major input elements 19

Input type usage 22

Text vs. specialized input types: select, checkbox, and radio

Select vs. radio

Select vs. radio vs. checkbox for true/false questions

Checkbox vs. multiple select list box

Text vs. select pulldown

Password vs., well, password...

Form visual design 28

Sizing text fields

Making form elements noticeable

Labeling input elements

Labeling required fields

Button design 34

Using contrast

Button placement

Button labels

Disabling buttons

The “Reset” button

The “Submit” button

Grouping and differentiating tasks through button design

Adding interactivity 41

Input validation 41

Error message tips



Forms are without a doubt the biggest contributor to graphical user interface complexity. While links and buttons accept only one form of stimulus—a click—forms are designed to accept text in all of its variety. This makes much more complex interactions possible. More complex interactions, however, create more room for user error. After all, you can either click a link or not (whether or not it is the correct link is another question), but text characters present almost limitless input possibilities (only some of which are correct). The aim of this chapter is to present and discuss the wide gamut of issues that invariably arises in the design of online forms.

Overview of the major input elements

As you read this section keep in mind that even though form elements may look different and allow different types of interactions (checking something on/off, dragging, Ctrl-clicking, etc.), all of them send only one type of data to the web server: text. Therefore, the purpose of having specialized input elements is mostly to:

- *reduce input errors* by placing constraints on what can be entered and allowing users to select from a list of choices;
- *reduce the load on users' memory*. Offering users a list of choices means that they will only need to *recognize* the correct entry rather than *recall* it. This is, by the way, one of the main ways of making interfaces easy to learn.

Text

Text input type is the most flexible because it allows users to enter any string of characters and numbers. However, with flexibility comes the need for extensive error checking, especially with fields designed to take data that must be in a certain format (i.e. phones, credit card numbers, or email addresses).

Password

This input type prevents someone looking over the user's shoulder from seeing the password, because every character entered into a password field appears as an asterisk. It is identical to text in all other respects.

Textarea

Textarea is an extension of the text input type. While a simple text box is one line high—the input string only scrolling sideways—textarea can have many rows scrolling both sideways and vertically. The textarea’s wrap attribute determines whether the text scrolls sideways until a line break is inserted or wraps to the width of the text area.

Select

The select (dropdown) input type limits the number of input choices available to the user and lets them know what the choices are. This cuts down on errors and decreases memory load. While a regular select dropdown accepts only one selection, adding the “multiple” attribute allows users to select more than one item by ctrl-clicking. Multiple select usually takes the shape of a multi-item list box, where several choices are visible at once to make selecting multiple options easier.

Radio

The radio input type is similar in function to a select dropdown: each group of radio “buttons” can submit only one selection. The major difference between select and radio input types is that the radio lists all of the choices in plain sight, while a select dropdown usually hides all but one when closed.

Checkbox

A single checkbox lets the user mark an item or a statement as true or false. A group of checkboxes can also be used with a list of attributes of an entity to identify what is true about that entity. For example, a set of checkboxes can be used in a movie search tool to specify whether to search for VHS, DVD or both.

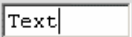
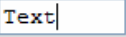
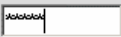

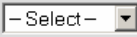

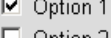
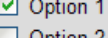
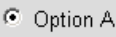
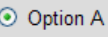

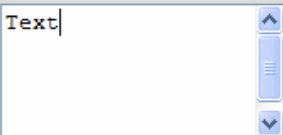
Missing in action: Combo box

Combo box is a truly useful input type that is sadly missing in HTML. Combo box is a combination of a list box and a single-line text box. A user who knows the name of the item to be selected can start typing it into the text box. At the same time, the list box will scroll to show the first item matching the initial characters. I have seen users use a select dropdown this way. Unfortunately, the dropdown will only jump to the first letter of the item.

For example, typing “ma” into a state select dropdown will not select “Massachusetts,” but “Alabama” instead, because the select dropdown only “remembers” the last keystroke. Having observed this behavior, mostly from advanced users, makes me wish combo boxes were a part of standard HTML.

Windows XP form elements: Change for change’s sake?

Windows XP has been deservedly called the “Fisher Price” version of Windows 2000 from the standpoint of UI. Bright colors, rounded shapes, drop shadows, and gradients now dominate the interface. Form elements are different too: many have lost most of their 3D treatment, and all are now colorized.

Element	In Win 9x/NT/2000	In Win XP
Text		
Password		
Select		
Checkbox		
Radio		
Textarea		

So what’s wrong with this picture? Two things:

- because the text boxes have lost most of their 3D “inset” look they are less noticeable, and can become visually “lost” on pages where the field and form background colors are the same;
- because the elements are now colorized, their colors may clash with other colors on the page where they’re placed.

Once again Microsoft designers decide to fix something that’s not broken... I do, however, give them credit for letting the user revert to a variant of the original Win 9x/NT/2000 scheme.

Input type usage

Form input elements may all look different, but not all of them are unique—there is significant overlap in functionality between them. For example, select pulldowns and radios do essentially the same thing: list choices only one of which can be selected at a time (with the exception of multiple select list boxes). However, a select is more appropriate in some situations, while radios are better in others.

Text vs. specialized input types: select, checkbox, radio

All input types pass plain text to the server. Therefore, you could always present you users with a bunch of text boxes, or even a single one. For example, amazon.com’s search box will accept a book’s name, author, ISBN, etc. Amazon.com gets away with this approach because it has a very robust search query processing “engine” capable of sorting through all these different keywords and yield meaningful matches. In determining whether to use a text box or a more restrictive form element you will have to consider a multitude of factors including your application’s input processing capabilities, sophistication of error checking, as well as user skill level and user performance requirements.

One click to pick a color here:



Choose Color:

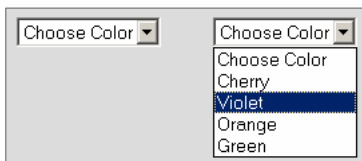
☐ Cherry

☐ Violet

☐ Orange

☐ Green

Two clicks to do the same here:



Choose Color

Choose Color

Cherry

Violet

Orange

Green

It takes just a mouse move and a click to select a color using radios. On the other hand, it takes a mouse move and click to open the pulldown, and then another move and click to select the color to do the same using a select pulldown.

In this particular example radios have another advantage over the pulldown: the color swatches placed next to the color names give the user an additional clue to help in decision-making. The user can pick a color visually. This is more natural than using a symbol—the color’s name.

Select vs. radio

The advantage of a select pulldown over a group of radio buttons is that it is compact. This makes it a good choice for presenting many options when screen space is limited. However, this also means that all of the options except one are hidden in a closed pulldown. Therefore, a pulldown is less effective in listing a set of unfamiliar options or as a navigation tool, because the user can’t predict what the options in the list are. On the other hand, a pulldown is a good way to present a list of states/provinces on an address form, because the user knows just from reading the label what will be in the list. Predictable pulldowns are more intuitive.

Of course, radios have their advantages:

- all options are visible at once;
- it takes half as many mouse moves and clicks to make a choice using a radio button than a pulldown. Consider the example on the left.

These advantages make radios more suitable for displaying a small number of choices with short labels such as colors or gender (Male/Female).

Select vs. radio vs. checkbox for true/false questions

There are at least three ways to ask a true/false question on a form:

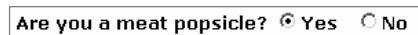
With a pulldown:



Are you a meat popsicle?

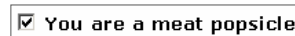
The pulldown menu is open, showing the following options: - Select -, Yes, and No. The 'Yes' option is currently selected and highlighted in blue.

With a radio:



Are you a meat popsicle? ☒ Yes ☐ No

With a checkbox:



☒ You are a meat popsicle

You can see that even though the three inputs accomplish the same goal, they do it differently. Of the three, radio and checkbox are both good alternatives here because they show all of the available choices. Checkbox approach is more compact, but a single checkmark does not indicate the two possible choices explicitly the way two radio buttons do. In addition, with radios the choice is *indicated* by the on state of the active radio button AND *reinforced* by the off state of the other button. In other words, there are more visual clues with a radio input than with the checkbox. Therefore, the radio approach is the clearest one of the three.

Checkboxes vs. multiple-option select

Here are, again, two different input types doing essentially the same thing: allowing the user to select any combination of attributes for a single entity. For example, a job search engine may use either one of these two input types to let users specify the types of positions they're looking for. Keep in mind, however, that while a multiple-option select does offer economy of space, it loses to checkboxes on these two counts:

- it is similar to the regular pulldown in that it hides some of the choices;
- it is more difficult to “operate” because ctrl-clicking to select multiple options requires using keyboard and mouse simultaneously. This is the domain of advanced users, and it is prohibitively difficult for users with motor disabilities.

A list of checkboxes, while less compact, is physically easier to use and shows all options at once. This makes checkboxes a clear choice for lists of up to a

couple dozen items. For example, compare the following sets of apartment features from springstreet.com:

Select amenities search at
springstreet.com:

The screenshot shows a search interface with a red sidebar on the left. The sidebar has two sections: 'NARROW SEARCH' at the top and 'SELECTED RENT & ROOMS' at the bottom. The main content area is divided into two columns: 'Apartment Features' and 'Community Features'. Each column contains a list of features with checkboxes next to them. Below the feature lists are two dropdown menus for 'Rent' (set to '\$0') and 'to' (set to 'no max'), and a 'Minimum # Bedrooms' dropdown (set to 'All Types'). A 'see results' button is located at the bottom right.

NARROW SEARCH

Apartment Features

- ☐ Air Conditioning
- ☐ Alarm System
- ☐ Dishwasher
- ☐ Fireplace
- ☐ Furnished Rentals
- ☐ Garbage Disposal
- ☐ High Speed Internet Access
- ☐ Microwave
- ☐ Patio/Balcony
- ☐ Short Term/Corporate
- ☐ Washer/Dryer Hookups
- ☐ W/D in Unit
- ☐ Wheelchair Access

Community Features

- ☐ Cable TV Available
- ☐ Fitness Center
- ☐ Garages/Covered Parking
- ☐ Gated Entrance
- ☐ Income Guidelines Apply
- ☐ Laundry Facility
- ☐ Pet-friendly*
- ☐ Public Transportation
- ☐ Senior Discounts Apply
- ☐ Spa/Hot Tub/Sauna
- ☐ Swimming Pool

*Communities that are pet-friendly accept some pets.

Rent: \$0 to: no max

Minimum # Bedrooms: All Types

see results

SELECTED RENT & ROOMS

Same thing using list boxes.

If you find that you absolutely have to use a multi-select pulldown, make sure the list box displays at least:

- four options at a time to make selecting easier;
- a third of the items in the list to make scrolling easier.

The screenshot shows the same search interface as the previous one, but with list boxes instead of checkboxes for the features. The sidebar and the 'Rent' and 'Bedrooms' dropdowns are the same. The 'Apartment Features' list box shows the first five items: Air Conditioning, Alarm System, Dishwasher, Fireplace, and Furnished Rentals. The 'Community Features' list box shows the first five items: Cable TV Available, Fitness Center, Garages/Covered Parking, Gated Entrance, and Income Guidelines Apply. The 'see results' button is still present.

NARROW SEARCH

Apartment Features

- Air Conditioning
- Alarm System
- Dishwasher
- Fireplace
- Furnished Rentals

Community Features

- Cable TV Available
- Fitness Center
- Garages/Covered Parking
- Gated Entrance
- Income Guidelines Apply

Rent: \$0 to: no max

Minimum # Bedrooms: All Types

see results

SELECTED RENT & ROOMS

Text vs. select pulldown

The advantage of a pulldown list over text is straightforward: the user doesn't have to remember the choices, but can use recognize the correct one when he sees it. For novice users making a selection using a pulldown is also faster than using text, provided that the number of choices is low and the text string to be entered is long. On the other hand, expert users often prefer simple text boxes. From the standpoint of data storage and processing, pulldowns allow for more standardized data sets that can be manipulated easier.

Sometimes, however, using plain text fields instead of pulldowns is better even for novice users: usually when the number of choices is relatively high, their text is short, and the choices are straightforward. For example, it is better to use text fields rather than dropdowns for entering dates in numeric format: it is much faster to key in a two- or four-digit number than to select one from a list of twelve, thirty, or a hundred items.

Password vs., well, password...

Password characters are masked with asterisks to protect the password from prying eyes and inspire a sense of security. However, the fact that the user can't see the password characters can cause usability problems.

One problem that came up during a usability test of guess.com occurred when the user was trying to confirm a newly created password by re-entering it. This is a very common task, and on most sites it goes off without any problems. Here, however, other site features conspired to make it very difficult for the user to fix the problem. Here's what happened:

When the user got to the checkout, she was asked to enter a username and a password before she could continue (See illustration on the next page). After making up a username, she tried hitting the "tab" key to move to the password box, but hit "caps lock" instead and didn't notice it. After concluding that she could not "tab" on the site (she commented on this in the post-test questionnaire), she clicked into the password box and entered a password in all uppercase letters without realizing it—the password was masked by asterisks. The site didn't accept her login, because she entered a username and password into the login boxes for returning customers. (D'oh!) However, the account creation fields on the same page were now prepopulated with her new username and password. When she got to the bottom of the page after entering her address, she dutifully entered her password into the "Password again" box. But this time the "caps lock" was off—she turned it off when she was entering her address. When she submitted the form the site objected again, saying that her passwords didn't match. So she retyped her password into the "Password again" box and resubmitted the form. After repeating this two more times and getting the same result she finally told me that the site was broken and that she had no idea how to fix the problem. When asked what she would do at that point, she said she would go to a different site or to the mall.

GUESS Women Men Girls Boys Baby Girls Baby Boys Models & More GUESS, Inc.

Passwords must match.
Your password reminder cannot match your password.

Checkout

Your Shopping Bag contains:

Item	Remove	Qty	Price	Subtotal
Denim Boot T-shirt (100% Cotton)	Remove	1	\$24.00	\$24.00
TOTAL:				\$24.00

Discounts will appear on final checkout page.
If you make any changes, be sure to click on the **APPLY CHANGES** button.

APPLY CHANGES **CONTINUE SHOPPING**

A Little Something Extra!
Make your order complete with one of these items or scroll down to complete your checkout.

Five Piece Grooming Kit
\$38.00

[Add Item to Bag](#) [More Info](#)

GUESS Spring 2002 Journal
\$3.00

[Add Item to Bag](#) [More Info](#)

GUESS Body Spray
\$35.00

[Add Item to Bag](#) [More Info](#)

Evo Square Tote
\$55.00

[Add Item to Bag](#) [More Info](#)

READY TO PURCHASE?

Scroll down and complete the following forms. If you've already saved a Username and Profile, log in below and we'll fill in your forms for you!

Username: Password: **SUBMIT**

Your Profile

Billing Address:

Address provided must EXACTLY match your credit card billing information.

First Name:

Last Name:

Email:

Daytime Phone:

Address:

City:

State:

Zip/Postal Code:

Country:

(We can accept orders within the 48 continental United States, Alaska, and Hawaii.)

Shipping Address:

Do NOT complete if your Billing and Shipping Address are the same.

First Name:

Last Name:

Daytime Phone:

Address:

City:

State:

Zip/Postal Code:

Country:

(We can accept orders within the 48 continental United States, Alaska, and Hawaii.)

Coupon Code:
If you have a promotional code you'd like to use, enter it below. Only one code is permitted per order, as offers cannot be combined. Your updated balance will appear during the final stage of your checkout.

Send As A Gift

☐ **Send this as a gift.**
We'll remove any price tags and include a gift receipt. If you'd like, write a brief message and we'll include it with your order.

GIFT MESSAGE:

Shipping Preference

☐ **UPS Next Day Air @ \$14.95**

☐ **UPS 2nd Day Air @ \$10.95**

☒ **UPS Ground \$5.95**

Please note: P.O., box, APO, and FPO addresses will be shipped via USPS Ground (\$5.95). No deliveries will be made on Saturday or Sunday.

Let Us Save Your Profile (optional)

It will make your next checkout faster, and you can get updates on sales, contests and more. Simply create a Username and Password you'll remember. For security reasons, your credit card information will NOT be saved.

Username:

Password:

Password Again (to verify):

Password Reminder (A sentence that will remind you of your password.):

Goodbye

Birth day: / /

Gender (optional):

☐ I'd like to receive news from GUESS? about new sales and contests.

☐ Enter me automatically in any future contests.

Next >>

[Sign In](#) [Your Shopping Bag](#) [Checkout](#) [Site Map](#) [Size Chart](#) [Jeans Guide](#) [Store Locator](#) [Help](#)

What happened here? Well, let's count the culprits:

- the returning customer login boxes were too prominent causing the user to start filling them out first;
- the new password field (at the bottom of the page) was prepopulated with the contents of the returning user password field;
- it was never mentioned that password was case-sensitive;
- when passwords didn't match, the system cleared only the password re-entry, but not the new password box.

Based on the observations from this and other studies, here are few tips for making passwords more user-friendly:

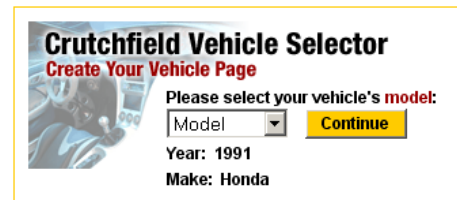
- ✧ **Make it easy for the user to verify a new password immediately after creating it.** Placing the two fields close together will increase the chance that the user will fill them in together without giving the sequence a second thought;
- ✧ **Clear both the "password" and "verify password" fields if the passwords don't match.** This way if the user makes an error when creating or verifying a password, both will have to be entered again. While this creates more work for the user, it ensures that the password is created and verified in "one shot," thus reducing the risk of a mismatch.
- ✧ **Make passwords case-insensitive.** There are two advantages here:
 - reduced memory load: a user has to remember only the password characters, but not their case;
 - fewer login errors in case a user creates a password in all uppercase and later logs in using all lowercase instead, or vice versa.

The Login/shipping address/billing address/gift options/shipping options/account creation page at guess.com. If you can overlook the fact that so many things are crammed onto the single page, you're still left with the password-related problems experienced by the user in my test.

Take advantage of conventions to prevent error

Before leaving this section I want to mention one more thing: convention. You may think that form input types are always used as their creators intended. So did I until I came across one form designed by an interactive agency that was hired by a client of mine to design its web site. The graphic designer decided to use checkboxes for a group of attributes where only one could be selected at a time. In other words, the designer used checkboxes in place of radios. Because the client already signed off on the design I had to talk with several decision-makers to get the “ok” to change the checkboxes back to radios. In the end the client had to pay to fix what should have been done correctly in the first place.

Another group of convention violators are sites where making a selection in a pulldown or clicking a checkbox / radio button submits the form. This behavior is often unexpected because only buttons and links ever should and, usually do, initiate actions. Auto-submitting select pulldowns, where making a change submits the form, are especially popular. I once user-tested crutchfield.com’s vehicle selector, which used auto-submitting dropdowns in conjunction with a submit button. That is, you could either make a selection in the dropdown or click the button to submit the dropdown. After making a selection in the dropdown one user first reached for the “Continue” button, but then noticed that something was happening: the little globe in the upper right corner of the browser started rotating. So he patiently waited for the next screen to appear. Unfortunately, the 56k modem’s response delay instilled enough doubt in his mind that he decided to click the “Continue” button after a few seconds of waiting. Ironically, this only delayed the next page. After repeating this routine *several times* he said he was confused about what was going on with the dropdown and whether he also had to click the “Continue” button.



Crutchfield Vehicle Selector
Create Your Vehicle Page

Please select your vehicle's model:
Model

Year: 1991
Make: Honda

Crutchfield's Vehicle Selector. The source of confusion here was the combination of a modem-delayed action of the auto-submitting dropdown and the presence of the “Continue” button. This issue did not come up when the site was tested over a broadband Internet connection.

In addition to being unconventional auto-submitting pulldowns also make selection errors more costly. The problem is that pulldowns are “twitchy” to begin with:¹ they appear and disappear easily, making it more difficult for users to make the correct choice on the first try. Making them auto-submitting means that users who make a wrong selection must wait for an irrelevant page to load and then go back to the previous page to try again.

¹ Steve Krug, “Don’t Make Me Think! A Common Sense Approach to Web Usability” (Steve Krug, 2000), p. 115.

Form visual design

Even though the standard HTML form elements are already “designed” and look very similar from site to site (which is a good thing), the layout is still in the designer’s hands. Robin Williams’ four design principles are especially useful here.² Let me illustrate them with an example from aa.com:

American Airlines home page.

On travelocity.com background contrast alone sets the dates section of the form apart, eliminating the need for non-information-carrying lines.

- **Contrast.** If two elements are not exactly the same, then make them really different using size, color, position, white space, etc. Contrast plays an important role on the aa.com home page. The form headings are prominent due to their high contrast with the rest of the form elements and the background. The forms are clearly separated from each other and the rest of the page by lines and contrast with the page background (light blue vs. white). This makes it easy to see each form’s boundaries.³ Each form appears as a cohesive unit. Since the forms represent different activity areas, these design features help the user easily locate and focus on one task and ignore the others.
- **Repetition.** Repeat certain form element attributes like size, font, or color to develop organization and strengthen unity on the page. In our example, the repetition of form headings, colors, fonts, element widths and buttons creates a coherent overall look. On complex, multi-part forms, contrast and repetition of form section headings can help the user quickly scan the entire form for relevant sections. On the following form, contrasty headings break up the form into chunks, making it easier to complete:

² Robin Williams, “The Non-Designer’s Design Book”, pp. 15, 27, 43, 53.

³ It is usually better to group items using white space or background shading rather than boxing them in with lines. Lines don’t carry information and detract from the other elements that do. Sometimes, as in the case of aa.com, the shortage of space make lines necessary.

✓ Review My Cart **2 Provide My Information** 3 Finalize My Order CONTINUE ▶

AT&T Wireless - mLife National Plan - Digital \$39.99

Current Customer ☐ Yes, I am currently an AT&T Wireless Customer.
☒ No, I am not currently an AT&T Wireless Customer.
☒ I have read and accept the [Terms and Conditions](#)

Service Address and Contact Info [Why do we need this info?](#)

First Name
 Last Name
 Street Address No PO boxes
 Address Line 2
 City State Zip Boston, MA 02114
 Contact Phone Contact Phone MUST be a landline.
 Email Address
☒ I want to receive the FREE GetConnected "Great Deals" Newsletter

Credit Information [Why do we need this info?](#)

Wireless service providers perform routine credit checks using your social security number, date of birth, and driver's license. This information is secure and will **NOT** be shared with anyone except for the service provider. [more detail](#)

Social Security # MUST be for the legal name above
 Date of Birth MM/DD/YYYY
 Card Type ☒ VISA ☐ MasterCard ☐ Discover ☐ American Express
 Card #
 Security # [Where is this located?](#)
 Expiration Date Month Year
 Cardholder's Name

NOTE: You can also place your order with us by calling 800-775-2506

◀ BACK CONTINUE ▶

Contrasty section headings split this form into easily digestible chunks (getconnected.com). About the only thing I would change on this form in respect to contrast would be to add a slight background shading to each of the three containers to (a) make the form fields pop out a little more and (b) strengthen the section groupings:

✓ Review My Cart **2 Provide My Information** 3 Finalize My Order CONTINUE ▶

AT&T Wireless - mLife National Plan - Digital \$39.99

Current Customer ☐ Yes, I am currently an AT&T Wireless Customer.
☒ No, I am not currently an AT&T Wireless Customer.
☒ I have read and accept the [Terms and Conditions](#)

Service Address and Contact Info [Why do we need this info?](#)

First Name
 Last Name
 Street Address No PO boxes
 Address Line 2
 City State Zip Boston, MA 02114
 Contact Phone Contact Phone MUST be a landline.
 Email Address
☒ I want to receive the FREE GetConnected "Great Deals" Newsletter

Credit Information [Why do we need this info?](#)

Wireless service providers perform routine credit checks using your social security number, date of birth, and driver's license. This information is secure and will **NOT** be shared with anyone except for the service provider. [more detail](#)

Social Security # MUST be for the legal name above
 Date of Birth MM/DD/YYYY
 Card Type ☒ VISA ☐ MasterCard ☐ Discover ☐ American Express
 Card #
 Security # [Where is this located?](#)
 Expiration Date Month Year
 Cardholder's Name

NOTE: You can also place your order with us by calling 800-775-2506

◀ BACK CONTINUE ▶

- **Alignment.** Nothing should be placed on a page arbitrarily, but aligned with something else on the page. Alignment creates lines for our eyes to follow, making it easy, for example, to scan a list of items. Alignment can also help visually connect related items, especially when they are also in close proximity. In the "Make a Reservation" form on aa.com (previous page), the "Departure Date:" and "Return Date:" fields are aligned with "From:" and "To:" fields, respectively. Here, the alignment visually reinforces the logical connection between the elements within each pair. Alignment also makes the "Search by" radio labels easier to scan.
- **Proximity.** Items related logically belong together visually. On aa.com the radio buttons that belong to the same group are placed tightly together, letting the user know they are all part of the same group. I don't understand why fossil.com decided to group gift certificate number with the credit card info fields so tightly (see right).

Getconnted.com could improve their form too by grouping the credit card pictures and radios better.

Payment info section in the checkout at fossil.com. Why is gift certificate grouped so closely with credit card fields?

PAYMENT INFORMATION *Under 18, must have parental or guardian consent

Acct. Number* Credit Card* Visa

Expiration* Oct 2007 Gift Certificate Number

Use white space make it clearer which radio goes with which card:

Card Type ☒ VISA ☐ MasterCard ☐ Discover ☐ American Express

Better yet, use natural mappings to align the radios with pictures:

Card Type ☒ VISA ☐ MasterCard ☐ Discover ☐ American Express

Sizing text fields

Text field size is highly suggestive of the length of input strings it is supposed to accept. Therefore, text fields should be sized in proportion to the length of the strings that go into them. For example, search boxes on many sites are not large enough to accept a typical search query. This makes it difficult for the user to review and edit the search string.⁴ The reverse is often true as well: text boxes that should be small are often too wide.

Here are a couple of rules of thumb for sizing text fields.

- Fields that take fixed-length inputs like zip, middle initial, or phone number parts should be just wide enough to comfortably fit the entry. For these types of inputs the field size itself provides an additional clue for the users to make sure they're using the correct field.
- Fields that accept a limited number of known strings should be wide enough to comfortably fit the longest entry but not wider. The date, month and year parts of a complete date is one example.

Some designers are tempted to make the ends of text fields line up, because they make a form look... well... all lined up. Unfortunately, doing this often results in grossly oversized fields. During a test of abercrombie.com, one test user tried entering his last name into the field just under the first name field without reading the field label, assuming it was the last name field. However, after the user typed in the first letter the field refused to accept any more characters. At first he thought there was something wrong with the keyboard... It took him a few clicks and re-entries to realize that the input length was for some reason limited to one character. Only then did he read the field label which was asking for a middle initial instead of the last name. Had this field been sized in proportion to the length of its intended input, the field size would have immediately suggested to the user that this was not the field for something as long as a last name.

ENTER YOUR BILLING ADDRESS BELOW	
First Name	<input type="text" value="John"/>
Middle Initial	<input type="text" value="D"/>
Last Name	<input type="text"/>
Address	<input type="text"/>

A part of the address form at abercrombie.com.

The middle initial field is too wide for its intended input. In addition, the field labels are placed too far away from the boxes, causing the user to not notice the label for the second box.

⁴ Jakob Nielsen and Marie Tahir, "Homepage Usability: 50 Websites Deconstructed" (New Riders, 2002), pp. 63, 69, 103, 129, 139, 145, 157, 167, 173, 181, 223, 249. Thirty six out of fifty sites reviewed had search boxes, and thirteen (about 36%) of those were cited with having search boxes that are too small.

In the following example from getconnected.com the fields were also made to align on purpose, but they were combined so that the middle initial and zip code fields didn't get oversized. There was a price to pay here, however: there is no way to tell whether the first or the last name should be entered into the first "Name" box without reading the hint to the right of the third name field.

Billing Address

Title:

*Name: First, Middle Initial, Last Name

Company Name:

*Street Address 1:

Street Address 2:

*City/Town:

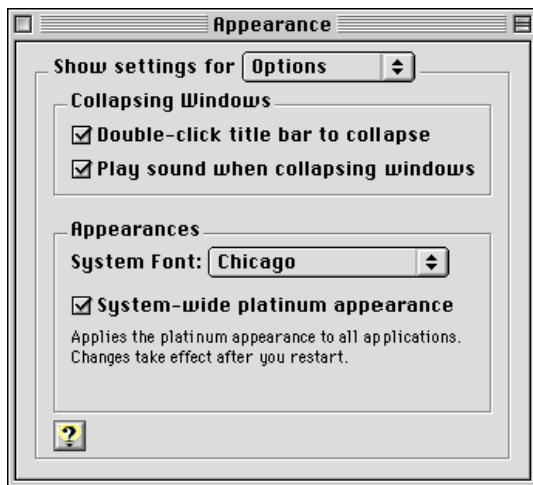
*State/Zip:

Several fields on this form from [Getconnected.com](http://getconnected.com) are not labeled directly. Therefore, the user must play a matching game: matching the labels with the fields to which they refer (zip), or rely on blue hint text (name).

Making form elements noticeable

Here are two tips for making form elements stand out:

- › **Contrast them with the background.** This seems straightforward enough, yet the designers of Mac OS 8 interface chose to camouflage form elements making them the same color as the background of many dialog boxes:



A sample dialog box from Mac OS 8 (platinum appearance enabled).

Lack of contrast between form elements and the background makes the elements less prominent. The 3D appearance of form elements is the only thing that saves this design.

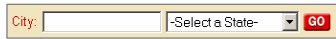
- › **Make them three-dimensional.** Standard HTML form elements are natural attention-getters due to their unique three-dimensional appearance. This illusion of depth makes them stand out from the otherwise flat surface of a web page. Windows XP form elements have a flat look which means they have to rely on contrast with their background for visual prominence.

The image shows a search box from nybor.com. It has a light blue background. The label 'keywords:' is in a small font. There's a text input field. To the right of the input field is a 'search' button with a magnifying glass icon.

Search box at nybor.com.

The box looks flat and, if not for the colored background, would get visually "lost".

Labeling input elements



Search box at springstreet.com.

The label for the State dropdown is the first option in the dropdown.

This is about the only case where a field label can successfully be a part of the element. Be aware, however, that as soon as a selection is made in the dropdown, the label disappears.

There are a few general things to keep in mind when labeling input elements:

- Be consistent with font and placement of labels. If you vary either, make sure there's a good reason to do so;
- Leave enough white space around label-field pairs to relate them;
- Make labels look different from hints (discussed below) by using a different font size, color, etc. Keep in mind, however, that there is a difference between using just enough contrast to make different elements look different and using so many colors, fonts, shapes, and sizes that the entire thing becomes visually overwhelming.

Hints

While lengthy instructions don't work on the web, well-placed hints can be very effective, especially on sites that sell complex, configurable items like computers or insurance. Let's briefly look at three types of hints: a sample input, valid data format, and an explanation of the main field label.

Sample input:
(amazon.com)

In amazon.com's case the year input box is not labeled at all, so the only thing the user has to go on is the sample input.

Valid data format:
(progressive.com)

The valid data format hints at progressive.com pull double duty as field labels. This works fine here, because progressive.com caters to American users. However this approach may present problems on sites whose audience includes international users. Many of them enter dates in a different format and may not understand what "mm," "dd," and "yyyy" stand for.

Explanation of the main field label:
(progressive.com)

Explanation is included for a good reason: the question itself doesn't mention non-drivers, roommates, or drivers. In general, however, if a field label needs additional explanation, the label itself may be worded poorly. It is better to have a clear label with no additional hints.

Labeling required fields

While there are many ways to mark required form fields, here are the three I've seen used the most:

- › **Place an asterisk next to the field label.** This is the most common way of marking required fields. It is also one of the clearest, second only to explicitly labeling fields as “required.” If you choose to use asterisks, here are a couple of tips to get the most out of them:
 - make sure you don't use asterisks for footnotes, comments or any other purpose;
 - use color to make the asterisk more noticeable. Small quantities of color can be very effectively used for emphasis. Asterisks present the perfect opportunity to exercise this power of color.
- › **Make the field label bold.** This method is not as common as marking a required field with an asterisk. If you decide to use this method keep in mind that if bold text is used elsewhere on the page (i.e. for emphasis) it will dilute the effect of bolding the required fields. For example, bolding the required field labels on nordstrom.com's 2002 form would not be very effective because of all the other text on the form that is also bold (top picture on the right).
- › **Label the field literally, using words like “required” or “optional.”** This is the clearest way to label a required field, because the label is self-explanatory. However, this approach also results in more overall clutter, especially if most fields are required and the “required” label is used. To illustrate, here's a piece of widescreenreview.com's subscription form:

(REQUIRED) Billing Name (as seen on credit card):

Company:

(REQUIRED) Street Address:

(REQUIRED) City: (REQUIRED) State/Province:

(REQUIRED) Zip: (REQUIRED) Country:

(REQUIRED) Telephone: Extension:

Fax: (REQUIRED) E-mail:

WSR could have uncluttered the form by marking the optional instead of the required fields or by using a simple red asterisk to mark the required fields. Better yet, why not do both!

SHIPPING DESTINATION

☐ Use this address for billing

* Indicates Optional Field

Title* Ms.

First Name MI*

Last Name

Street Address/P.O. Box

Please note: we are unable to ship shoes to APO/FPO or P.O. Boxes

Apt. No./Additional Address Info*

City (no abbreviations)

U.S. State Virginia

Zip Code or Postal Code

Country **USA**

Orders can be billed internationally.
We can only ship to U.S. destinations.

Daytime Phone*

E-mail Address*

[E-mail usage policy](#)

Shipping address form at nordstrom.com in 2002.

Asterisks here mark optional fields—precisely the opposite of what most other sites do.

Update: When I checked Nordstrom.com's site a year later (in May of 2003) asterisks marked the required fields:

Default Shipping Address

*First Name M.I.

*Last Name

*Address

*City

*State/Province

*Zip/Postal Code

*Country United States

We are unable to ship orders internationally at this time. Other exceptions apply to shoes, fine jewelry and gift card purchases. Please see our [Shipping Methods & Charges](#) for details.

*Required information.

The address form is much improved in 2003. Better field alignment, better field label placement, and less clutter overall. And the cute truck icon stayed!

Tips for helping users keep their fingers on the keyboard.

- set the cursor's focus on the first field in the form that requires user input. For example, when Yahoo! Mail login page loads, the cursor focus is set on the first of the two login fields (the username). This eliminates an extra click.
- set the "tabindex" attribute of all of the inputs on the form so that tabbing from one to another would guide the user logically through the entire form. I have been asked to implement this by my clients for intranet sites. In addition, several users in my usability tests "tabbed" between form fields as well. Both Netscape and Internet Explorer will base the tab sequence on the order in which the elements are coded in HTML. But sometimes this order is not the most logical, so the browser needs a little help. While in the past only Internet Explorer supported tabindex, the newer versions of Netscape (6 and up) now do too.

Button design

Two reasons I like standard HTML buttons:

- They **look like physical buttons**: they have that "push me" three-dimensional look. This makes them easy to notice, even on busiest forms.
- They **behave like physical buttons**: the button appears depressed when the mouse button is pressed, and bounces back when the mouse button is released. This responsiveness becomes important on the web where there's often a delay between the time when a user clicks a button and the time when the page changes in response to the click. In this case the only thing that tells the user that the click has been registered successfully is the fact that the button physically reacted to it. On an eCommerce site, giving the "Place Order" button this kind of behavior can reassure the users that they successfully activated the button, stopping them from clicking the button repeatedly.

If HTML button are so great, why not use HTML buttons more often?

Design flexibility. HTML buttons are just not very flexible in their appearance. Of course, this can be seen as a good thing—less chance for the designer to mess it up. Most of the time, however, an HTML button just looks out of place (ugly?) no matter how much you tweak it. This is why graphical buttons are used most of the time.

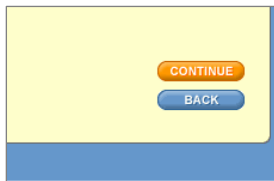
Even though most buttons out there are graphical, there is no reason why they can't mimic the three-dimensional look *and* the responsiveness of standard HTML buttons. While making a button look "buttony" is easy,

making them *behave* like buttons is little trickier. It usually involves some JavaScript (or Flash) programming to capture a mouse press and release. But it can and should be done. So next time, while programming that slick rollover, make those static buttons more responsive too.⁵

Using contrast

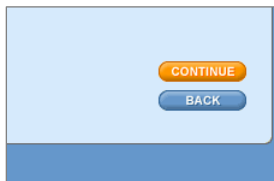
In addition to 3D appearance, contrast is key to making a button noticeable. Contrast of color (hue, lightness, *and* saturation) makes buttons on aa.com (on page 28) literally jump off the page. The contrast is so strong, in fact, that the designers got away with giving the buttons a minimal 3D treatment.

Contrast can also be a powerful tool to influence how the viewer perceives compositional elements, or, in our case, buttons. For example, you can use contrast to make the default button more prominent. Below is a crop from one of orbitz.com's checkout pages.



“Buttonz” at orbitz.com (a crop from the original checkout page). Follow the shots in this section to see how contrast, placement and labeling can improve these buttons.

The default button—“Continue”—is orange, while the “Back” button is blue. Since orange is a “warm” color and blue is a “cool” color, the “Continue” button moves forward in the viewer’s eye, while the “Back” button recedes. The effect of page background color, however, is at odds with this. The background is a lighter shade of orange. This makes the “Continue” button less prominent, because it does not contrast with the background as much in hue. Had the background color been “cooler,” it would have reinforced the contrast between the two buttons and made the “Continue” button “pop” off the page:

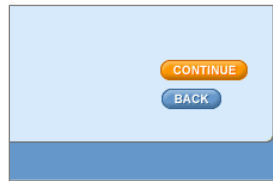


Re-designing the orbitz.com buttons: step 1. Make the background cooler to make the default button stand out. Using a unique hue for the “Continue” button only enhances the effect. Note also, that the contrast of luminance (lightness, value) between the continue button and the background is for the most part preserved. I could, however, make the continue button’s face a little darker so that the white text is more readable.

Contrast is not limited to contrast of color. You can also create contrast using size, white space, position, texture, etc.

⁵ For Internet Explorer you can add this event handler to the submit button’s code (input type=“image”): `onMouseDown=“this.src=‘button_down_state_image_filename’”`

Let's enhance the contrast between the buttons in our example by making the "Back" button smaller, creating a contrast of size:

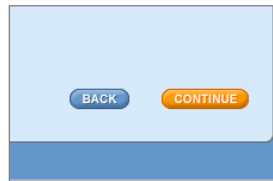


Re-designing the orbitz.com buttons: step 2.

The "Back" button is less important than "Continue." It can be smaller.

Button placement

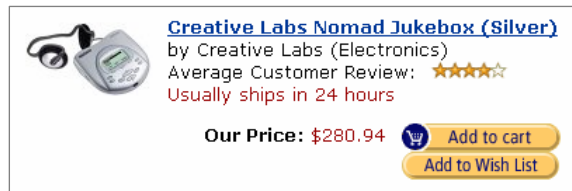
Even though conventional button placement is at the bottom of the form, longer forms can benefit from having the submit button repeated at the top of the form as well. Center and right alignment works for most buttons, but buttons that take the user back in a multi-step process (like the checkout) should be left-aligned or to the left of the continue button. The reason behind this is simple: in western cultures people read from left to right. This makes right the forward direction. Applying this principle to our orbitz.com button example, we should place the "Back" button to left of "Continue":



Re-designing the orbitz.com buttons: step 3.

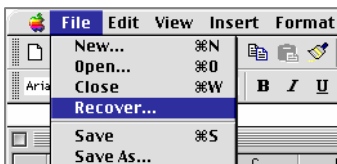
Place the "Continue" button to the right of the "Back" button. This creates a cultural clue for customers whose goal it is to go forward.

You should also avoid placing buttons too close to each other for the simple reason that this makes the user work extra hard to point to the correct button. The two buttons in the following example are too close:



A product listing on amazon.com.

The centers of "Add to cart" and "Add to Wish List" buttons are too close together. The user must be extra careful with the mouse pointer to hit the correct button.



A Macintosh menu item at the edge of the screen requires less pointing precision. This effectively increases its size compared to items that "float" in the middle of the screen.

In fact, the same law that makes small buttons and buttons placed too close together more difficult to "hit" with the mouse pointer explains why an Apple Macintosh menu is easier to point to: the menu is at the edge of the screen so the mouse pointer doesn't have to stop precisely over the menu item—the pointer will automatically stop over the menu item even if the mouse keeps moving, pushing the pointer off the top edge of the screen.⁶

⁶ Jef Raskin, "The Humane Interface" (Addison-Wesley, 2000), p. 94. By the same token, placing an item in the corner of the screen makes it even easier to point to.

Button labels

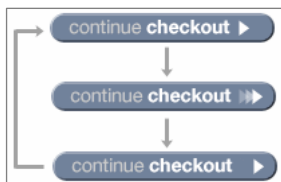
Since buttons, unlike form fields, can initiate actions, they should be labeled with an “action” label that includes a verb (or an adverb/preposition that implies an action). A button’s label and context should be sufficient enough for the button to be clear—the button should not have to rely on instructions elsewhere on the page to tell the user what it is for. A button’s label doesn’t have to be limited to text only. Process movement buttons like “Back” and “Continue” can benefit from simple arrows pointing to the left and right, respectively. Here’s how our orbitz.com buttons would look with arrows:



Re-designing the orbitz.com buttons: step 4.

The arrows serve as an additional clue for the user. Note that even though the label of the “Back” button does not include a verb, an action is implied by the adverb back.

Bloomingdales.com takes this one step further by making the arrow animate:

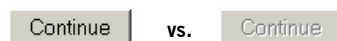


The animated “continue checkout” button at bloomingdales.com (some time back in 2001). The animated arrow is cute, but can provide the wrong initial clue for a seasoned user: things that animate are usually advertisements.

I am not sure that animation is necessary here. Many web users are conditioned to ignore anything that animates, because it is usually an advertisement. Overdesigning buttons by including over-stylized text and elaborate icons/graphics can also lead users to mistake them for advertisements, especially if the buttons are large and don’t look three-dimensional.

Disabling buttons

Never hide buttons. Disable them instead. Another reason why I like standard HTML buttons: it is clear when they’re disabled:



Disabled states for graphical buttons are usually created by making the regular button lighter/desaturated in color or by converting it to grayscale. Sometimes, however, what looks like a disabled button is really not and vice versa. The “New Flight Search” and “Cancel” buttons on the “Select your departure date” page from Travelocity.com are lighter than the buttons used



Buying choices box on a product detail page at amazon.com. The “Choose a store” button’s label is inaccurate, given what happens when the button is pressed. From the point of view of the user, there is nothing to choose from at this point: no stores are visible. A better descriptive label would have been “See stores in your area,” because the user is presented with the list of stored from which he can then choose. The point here is that a good button label should accurately indicate what will happen once it is activated.

In this case, however, there are other factors at play that affect the final choice of the label:

- the context is already explained by the keywords “in-store pickup” and “enter you ZIP code;”
- there is a limited amount of physical space for a very descriptive button.

Taking these factors into consideration, a simple label like “Go” would have been my final choice here.

elsewhere on the site. They're also lighter than any other blue elements on the page. This makes them look disabled. They are not.

Flight: Thu, May 02

Option 1

Select Flight: Delta Air Lines flight
Depart: Los Angeles, CA (LAX)
Arrive: Boston, MA (BOS)
Meals: Breakfast

Option 2

Select Flight: Delta Air Lines flight
Depart: Los Angeles, CA (LAX)
Arrive: Boston, MA (BOS)
Meals: Dinner

Regular blue travelocity.com buttons are more prominent (the blue is more saturated/darker).

Home Guides & Advice Flights Lodging Cars/Rail Vacations Cruises Deals/Rewards My Stuff

Round Trip/One Way Multiple Destinations Flight Arrival/Departure Info Dream Maps Deals

Select your departure date

Please select a departure date from those highlighted to confirm that the fare is available. If it is not offered on the date you want:

- [Try a different airline or another low fare](#)
- [Get the specific date you want](#), even if the fare increases

Boston, MA to Los Angeles, CA on America West Price: USD 258.00*

Click on your preferred departure date

April							May							June						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
14	15	16	17	18	19	20	19	20	21	22	23	24	16	17	18	19	20	21	22	
21	22	23	24	25	26	27	26	27	28	29	30	31	23	24	25	26	27	28	29	
28	29	30											30							

■ Fare offered ■ Fare not offered

New Flight Search Cancel Go To Home Page

Select departure date at travelocity.com.

The default response for the user on this page is to select a date by clicking on a day in the calendar. In light of this, the designers probably wanted to de-emphasize the two large buttons below the calendar by desaturating and lightening their color to make them less prominent. This is justified because by clicking either button the user would abandon the purchase of the selected flight. Unfortunately, this is not enough! The buttons are still too prominent: they're too large with too much white space around them. Why not make both buttons smaller and place "New Flight Search" in the left column?

The "Reset" button

Most usability specialists don't recommend using the "Reset" button at all. The rationale is straightforward: it is faster to correct a couple of incorrect fields than to erase the entire form and start over. In addition, an extra button at the end of the form clutters up the page and adds more work for the user who must now spend time and energy to decide which button should be pressed.⁷ Therefore, I agree with the gurus: you can easily omit the "Reset" button in most cases.

⁷ Jakob Nielsen. "Reset and Cancel Button" alertbox (April 16, 2000). <http://www.useit.com/alertbox/20000416.html>.

If you do use a “reset” button, be sure to give it an accurate label. For example, a “reset” button is not a “clear” button: it resets all field contents to what they were when the form loaded. This means that the fields that were prepopulated and were erased by the user will become prepopulated again when the user clicks “Reset.”

“Reset” is also not a “Cancel” button. On the web, the browser’s “Back” button is the equivalent of a “Cancel” button. In fact, the best name for the “Reset” button is... “Reset” or maybe “Reset All.” You should also place “Reset” far enough from the “Submit” button and make it less prominent. This will prevent a user from accidentally hitting “Reset” instead of “Submit.”

The “Submit” button

Every form, no matter how short, should have a submit button. Some sites ignore this simple rule by autosubmitting single-input forms (especially pulldowns). The only exception to this recommendation are select pulldowns on high-use, quickly-responding intranet forms and only in cases where the negative consequences of choosing the wrong option are negligible.

Label the “Submit” button with a clear action label the user will understand. “Submit” is for web designers and programmers. The exact label will be dictated by the context and intended audience. For example, “Go” is good for a search button (provided the query input box is labeled “Search:”), while “Place Order” is good for a button that commits a customer’s order at the end of a checkout. Notice that verbs are used in button labels in both cases.

I like the “GO” button in the aa.com example (page 30). It’s large, contrasts well with the background, has a 3D treatment, and is the only button on the form. This makes clicking it a “no-brainer” step for users who are finished filling out the form and want to continue.

Grouping and differentiating tasks through button design

Sometimes a single page on an transactional web site has to support several tasks. For example, the “Manage Addresses and 1-Click Settings” page at amazon.com enables a customer not only to add and edit addresses, but also turn 1-Click ordering on and off. These two tasks are differentiated using button design: the buttons for address operations are white, while the “Turn 1-Click on” button is yellow. This difference immediately suggests to the users that the buttons belong to different groups of tasks.

“Manage Addresses and 1-Click Settings” page at amazon.com.

The address-related buttons all have the same design. This design both relates them and differentiates them from other types of tasks on this page.

Of course, it would also help if the “Continue shopping” button was a link, just like the Find [answers](#) link below the “Turn 1-Click On” button.

[Account Maintenance](#) > **Manage Addresses and 1-Click Settings**

Click the Edit button next to any piece of information below that you'd like to adjust. Need to change an address on an open order? Go to your [Order History](#).

When your changes are complete, [Continue shopping](#)

Click here to add a new address [Enter a New Address](#)

1-Click ordering is turned **OFF** for John Doe.

[Turn 1-Click On](#)

Questions about 1-Click? Find [answers](#)

1. John Doe
123 Anystreet
Anytown, California
90210
Albania
Phone: (000) 000-0000

[EDIT](#)

[Delete Address](#)

[Click here](#) to make this your 1-Click default address.
([what this means](#))

1-Click settings and Credit card preferences for address # 1

Include in 1-Click dropdown: Yes

How address appears in dropdown: John Doe

Shipping method: International Priority (Averages 1-4 business days)
*** We selected a shipping method for you. Please change it if you prefer a different shipping method. ***

Credit card: ***You will need to add a credit card number in order to use this address for 1-Click orders.***

[EDIT](#) 1-Click settings for this address.

[Enter a New Address](#)

Another case where differentiating tasks using buttons can be effective is on a shopping cart page. The shopping cart page at amazon.com illustrates how button design helps users understand the hierarchy of tasks. The buttons for “local” shopping cart actions (“Save for later,” “Delete,” and “Update”) have a lighter background and are smaller than the “Proceed to checkout” button. They are also placed close to the items to which they apply.

Amazon.com shopping cart.

The size and color of the shopping cart action buttons relates them and sets them apart from the other buttons on the page.

[Continue shopping](#) in All Products search results

[Proceed to checkout](#) Returning customer? [Sign in](#) to turn on 1-Click ordering.

Shopping Cart Items--To Buy Now	Qty.	
Homepage Usability: 50 Websites Deconstructed Jakob Nielsen, Marie Tahir, Paperback Usually ships in 24 hours	1	List Price: \$39.99 Our Price: \$27.99 You Save: \$12.00 (30%) Save for later Delete
		<input type="checkbox"/> Add gift-wrap/note. (Learn more)
Adobe Master Class: Web Site Redesigns Darcy Dinucci, Paperback Usually ships in 24 hours	1	List Price: \$40.00 Our Price: \$28.00 You Save: \$12.00 (30%) Save for later Delete
		<input type="checkbox"/> Add gift-wrap/note. (Learn more)
Subtotal: \$55.99		
Make any changes above? Click Update		

Of course, the button design also helps relate the shopping cart tasks simply because their buttons have the same look. On a page with two or more button designs, like these two examples from amazon.com, the user will see the buttons with the same design as belonging to related tasks.

The underlying principle behind using buttons to relate and differentiate tasks is simple: things that are related logically should be related visually.

Adding interactivity

Electronic forms are more flexible than paper forms because they can interact with the user. Interactive elements such as pulldowns and radios are already a part of the convention. Client-side scripting can increase this level of interactivity by having the form elements change, appear or disappear based on user selections. The question here is: what level of added interactivity is appropriate?

It is easy to hide irrelevant form elements until a user selection somewhere on the form makes them relevant. But while doing this may reduce clutter, it can also create confusion when those elements do appear. The confusion will be joined by aggravation, if expecting to fill out a short form a user ends up completing a much longer form as new elements keep popping up.

Therefore, it is better to disable (“gray out”) irrelevant form elements than to hide them. This approach sets correct expectations about the amount of work required to complete a form, and lets the user anticipate additional input requirements, eliminating the surprise (and confusion) that results when those elements appear “out of nowhere.” If you do choose to hide and display form elements make sure the “apparitions” are minor and do not involve more than one or two fields located close to the form element which triggered them.

Using DHTML, you can track whether all required fields have been filled in and display a “one or more fields need to be filled in” type of message next to the “Continue” button as long as there are required fields that are still blank. This is unobtrusive and warns the user that an error may occur before it does, which cuts down on anxiety level. You should not, however, disable the “Continue” button until all fields are filled in, as this will confuse more users than it will help.

Input validation

There are two ways a form can be validated (or error-checked): on the client side and on the server side. Client side checking is great for testing whether a required field has been filled in or selected, or whether the input string is in a correct format. Server side validation is required for things that can’t be checked for on the client side. For example, approving credit card numbers requires a server-side check. Because both client- and server-side errors can be validated on the server side, server-side validation is used more often.

There are two basic ways that error messages can be *presented* to the user: in a modal alert box⁸ or on the form itself. Let's call the latter "in-line error messages." Each has its advantages and disadvantages:

- An alert box gets great visibility: both the error state of the system and the error message are nearly impossible to miss. However, as soon as the user closes the alert box, the error message disappears. The user must memorize the message in order to be able to correct the error. In cases where the alert box contains several messages the user must remember them all—not good.
- In-line error messages (the ones that are placed on the form itself) are not as visible as an alert box—they can get lost below "the fold" on a long form, especially if the system's error state is not clearly shown above "the fold." On the other hand, the user does not have to remember what the errors are and where on the form each of them occurred.

"What about speed?" you may ask, "Aren't alert boxes faster because they avoid a trip to the server?" It is true that alert boxes have been used to display the messages for client-side validated errors, while server-side validated error messages are usually displayed directly on the form. However, this doesn't have to be the case. With a little scripting you can place client- and server-side validated errors all in an alert box or all on the form. There's even a way to place client side-validated error messages on the form itself without reloading it.⁹ This means that the "alert box and client-side validation" *versus* "in-line and server-side validation" dichotomy is no longer valid.

So which presentation is better: alert box or in-line? From my own experience I can't say that one method is always better than the other: I have observed users have problems with both implementations and also be successful with both. In general, I recommend using the in-line presentation for both client- and server-validated messages. However, there are a few situation where programming simplicity makes alert boxes an option:

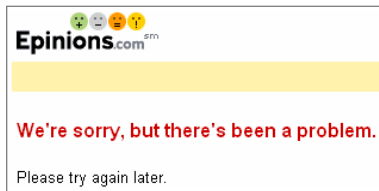
- when there are very few form elements (fewer than 7 or so) and none require a server-side check *and/or*
- when user testing shows no more than two user errors per form, one error or less being the average.

⁸ A modal alert box on a form will not allow you interact with the form until you respond to the message in the box by clicking the appropriate button. The simplest alert box will have only one button: "OK."

⁹ While hiding layers works in any self-respecting browser these days, in IE this can be done with the `innerText` attribute of the `` tag. This is different from simply hiding `<div>` layers because the page content will actually wrap around the span tags.

Error message tips

- › **Avoid generic error messages like the plague.** I once searched for a Nikon digital camera on Epinions.com and this error message:



Even though this was a rare system error (I never saw it again), these generic error messages are not helpful and should not have been created in the first place.

In addition to being useless, generic messages can lead a user to conclude that the entire site is “broken.” In one of my tests, a user arrived at that exact conclusion after trying a few different things to fix the same error she was getting. To be more specific, the site didn’t clearly show which fields were required and displayed a generic message at the top of the page that said “Information missing below. Please correct and re-submit.” To make things worse, the error checking script was validating one field at a time and the fields with errors were not marked. This meant that the user had to re-submit the form for every erroneous field she thought she corrected to find out whether it was the one with the error. After a few unsuccessful attempts she finally gave up. This is, then, the sad story of how a generic error message conspired with a few other usability quirks to prevent the customer from completing the checkout, costing the site a sale.

- › **Include a sample input in the error message itself** for inputs that require the data to be in a certain format like date, phone, etc. Do this even if the format is already included as a hint. If the user made the error, chances are the hint went unnoticed and additional help is required.

Tips for in-line error messages:

- › **Include a short, but prominent message at the top of the page** telling the user that there are errors. This way error messages hiding below “the fold” will not be ignored. Make the message very visible using a unique color (red works best), contrast, white space, etc. While Amazon.com’s error message stands out due to its color, its placement—in-line with other text on the page—makes it seem like the message text is a part of the instructions (which most users don’t read):

Error messages at amazon.com. An error message placed next to the field where the error occurred is much more visible than the string of messages in-line with the instructions at the top of the page.

[Account Maintenance](#) > [Manage Addresses and 1-Click Settings](#) > [Add New Address](#)

Future mailing labels will appear exactly as you enter them below. This change will not affect orders currently being processed. To change the address on an open order, select that order from your [Order History](#). Your request is missing information or needs correcting. *** Please fix the areas indicated below. *** When you are done, please press the button the Confirm button to send your information again.

Add an address

Full Name:

Address Line 1:

Address Line 2:

City:

State/Province/Region: *** The state and zip code you provided do not match. ***

ZIP/Postal Code:

I recently tested sears.com's guided search tool. One of the more interesting observations was that the blue "your search options" bar was more prominent than the error message just above it: all of the users noticed it before they saw the error message.

This error message at sears.com was not as noticeable as the contrasty box just below it. This isn't bad at all, however. The emphasis is on what's good—on the fact that there *are* products matching some of the criteria. Of course, it is nice that the error message itself is polite and clearly states what's wrong.

SEARS Where else?
 Sears Homepage

Search by Keyword or Item #

[Auctions on eBay](#) | [Hot Buys](#) | [Shop by Brands](#) | [Clearance Center](#) | [Catalog Quick Order](#)

[Gift Registry](#) [Gifts](#)

Search by Feature ▶ Portable Power Tools: Drills, Cordless

0 products match all of your search options. Some of your choices have matches, which are outlined below.

Your search options:	# of matching products
Brand: "DeWalt, Milwaukee"	12 ▶ View these 12 products
Cordless Drill Type: "3/8 in. 12.0 to 14.4 Volt"	2 ▶ View these 2 products
Price: "\$ - \$120"	0

▶ **Please Select a Brand:** (Select all that apply)

☐ No Preference
 ☐ Black & Decker
 ☐ Bosch Tools
 ☒ DeWalt
 ☐ Panasonic

☐ Chicago Pneumatic
 ☐ Craftsman
 ☒ Milwaukee

▶ **Select by Type:**

☐ No Preference
☐ 4.8 to 9.6 Volt
☐ Combo or Value Kit
☐ Hammer Drill or Driver
☐ Impact driver
☐ Mini Driver
☐ One-half in. 12.0 to 14.4 Volt
☐ One-half in. 14.5 Volt and Above
☐ Right Angle or Close Quarter
☐ Three-Eighths in. 9.7 to 11.9 Volt
☒ Three-eighths in. 12.0 to 14.4 Volt
☐ Three-eighths in. 14.5 Volt and Above

Q. Where can I learn more about Portable Power Tools?

A. The Buying Guide.

Learn tips on purchasing, how it works, product options and more.

- › **Put error message next to the form element where the error occurred.** This makes it easier to find the source of the error and the place to fix it. Amazon.com's error message "The State and zip code you provided do not match" (at the top of previous page) is a good example.
- › **Use a unique text font/color for error messages.** During a test of guess.com one user confused a success message ("This item has been added to your shopping bag") for an error message, because both messages were rendered using the same font.

Five Pocket Easy Pant
100% cotton. Low waist. Zip fly. Front slash pockets. Back patch pockets. Slightly flared leg. Easy fitting. Machine wash. Domestic and/or imported. Shown in Shattered Italy Wash.

Y2166096 \$68.00

Please fill in the required information.

Choose Size
Choose Color

1 Quantity

Five Pocket Easy Pant
100% cotton. Low waist. Zip fly. Front slash pockets. Back patch pockets. Slightly flared leg. Easy fitting. Machine wash. Domestic and/or imported. Shown in Shattered Italy Wash.

Y2166096 \$68.00

This item has been added to your shopping bag.

Choose Size
Choose Color

1 Quantity

Error and success messages look the same at guess.com. In addition, the error messages are not very helpful because they are too generic.

Update: When I visited the site in May of 2003, the size and color selection errors were validated on the client side and presented in an alert box. This is a better approach here, because the form is so short and alert box response time is lightning-fast. In addition, the error message itself is more helpful and even the cursor's focus is set on the first incorrect field when you close the alert box. Kudos!

Tips for error message alert boxes (*Remember: small forms / few error only*):

- › **Include the label of the form element in the error message.** This makes it easier for the user to locate the form element once the alert box disappears;
- › **List all error messages in a single alert box, displaying it when the form is submitted.** This is a better alternative to validating each form element individually when it loses focus. I have seen users make an error, move to the next field, but then notice the error and correct it before submitting the form. Popping an error message before they have a chance to correct themselves would only aggravate them. This is also better than validating a field at a time when the form is submitted. The user will not know how many fields are wrong and will get frustrated when, after being sure that she corrected the only error, she sees another error message. There are fewer surprises with a single alert box.
- › **Consider setting the cursor's focus on the first form element where an error occurred,** making it easy for the user to locate the element once the alert box is closed. The only case where this is not recommended is if the form is so long that it scrolls *and* the field labels are placed above the form fields. This way when the form element gets focus the window will scroll up enough to reveal the form element but not its label. In this case either place labels to the side of the form fields or don't use the focusing trick.