

Creating Usable Forms

by **Mary Masterilli**

If a form is difficult to fill in, requires too much information, or is seen as invasive or annoying, users will simply avoid it. In this article we look at design principles for making your forms more functional, accurate and easy to use.

Page Focus

When a user loads your form into their Web browser, they have to click on it before they can start typing their input. With the help of a simple JavaScript command, you can place the user's cursor anywhere you want on the form, such as the first field. They use this on [Google](#) with superb results: as soon as the page loads, your cursor sits blinking in the search box, ready for your query. So simple you don't even notice it.

Tab Index

You can navigate forms using your tab key to move between fields. By default, the cursor will tab through the elements in the order in which they appear in the source code. However, this may not always be the best way to move through a form. For instance, you might have a two-column layout where you want to move down the left column first, then the right. With the "tabindex" property you can set any order you choose, even skipping fields that are not required input. This can help users fill in a form more quickly and efficiently.

Splitting Forms

Large forms that require large amounts of information should be split into several screens. This makes the task less daunting for your users and increases the likelihood of compliance. When you split a form into separate pages, always indicate to the user how far through the form they are.

Labels

Before the <label> tag (new to IE 4+ and Netscape 6+) you had to click on the form element to select it. This new tag allows you to click the text beside the element, which makes selecting an element much easier.

The <label> tag can increase the usability of a form, not only for users with screen readers, but for those of us with less than perfect mouse skills, especially laptop users. This makes for much friendlier radio buttons and checkboxes, too.

Reset Buttons

Reset buttons are too often mistaken for submit buttons. If you've ever clicked 'reset' when you meant to 'submit,' thus erasing all your data, then you know how frustrating this can be. It probably made you angry at the company that published the form. It may even have caused you to abandon your transaction.

Do your users a favor and kill the reset button. It's unnecessary on most forms. If the user wants to change a detail, they can go back and edit the relevant field. There is rarely an instance where the user will want to start over from scratch - but plenty of risk they will do this by accident.

Splitting Inputs

As with other constraints, over-splitting inputs can frustrate users. For example, breaking a name field into 'first name' and 'last name' requires two separate inputs rather than one. Or splitting a telephone field into area code, exchange and number. These are usability issues that frustrate and annoy users, yet both are commonly seen on forms. Forcing users to enter what to them is a single datapoint into multiple fields is an unnecessary delay.

Required Fields

Requiring specific input may serve to prevent erroneous form submissions, but is it serving your purposes at the expense of your users? Requiring a phone number when the user has clearly stated their preference for email contact doesn't help anyone. Such stringent requirements often lead to annoyance, and any form that requires an unreasonable amount of data only encourages users to enter "valid" but false information. How many times have you signed up for a site using junk data such as user@aol.com? Come on, be honest.

If you need accurate data to fulfill an order, for example, then you must minimize chances that users will enter incorrect information. The answer is to carefully consider which fields must be required, and which can be optional.