

Completely Accessible Standards Compliant Web Form -Brian Hogan

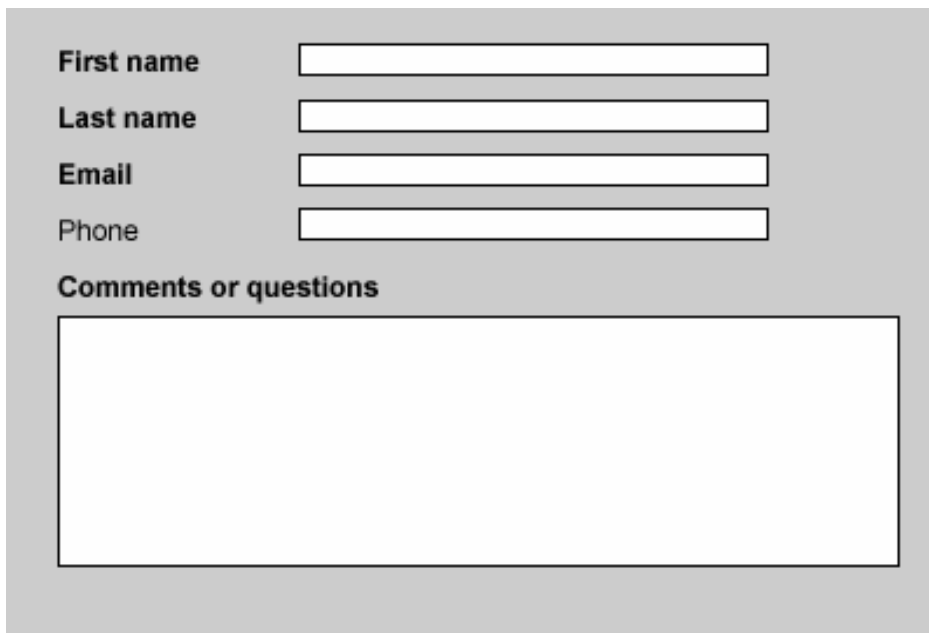
I have spent a good part of my life developing web pages, and since I have a vision disability, I have always been very interested in assisted and adaptive technology. For many of my users with similar impairments, web forms have traditionally been a problem. That problem was magnified when many web designers began using tables to lay out web forms, thus causing screen reading software to trip over the complex layouts.

While working on a client's web site, I set out to try to make a web form to collect questions and comments that didn't require tables. Quite by accident, I discovered the solution on the very first attempt.

The Form In Theory

For this form, I needed to collect the person's first name, last name, email address, and their comments. I wanted to optionally collect a phone number.

My desired result was to have the following look to my form.



The image shows a web form layout on a light gray background. It consists of four input fields stacked vertically, each with a label to its left. The labels are 'First name', 'Last name', 'Email', and 'Phone'. Below these four fields is a larger text area with a label 'Comments or questions' above it. The input fields are simple white rectangles with black borders. The text area is a larger white rectangle with a black border.

This was something I knew I could easily achieve using the standard table approach, but I just thought that I should challenge myself a little

Having done many two-column CSS-only layouts, my first thought was that I should use the same approach to lay out the form. I would use several layers of nested div tags to group the rows of the form together, thus simulating a table.

Red: `<div class="form">`

Blue: `<div class="row">`

Green: `<div class="leftcol">`

Pink: `<div class="rightcol">`

The CSS Styles

I had my idea, and so I set to work on my style sheet. I like to keep my layout separate from my style, so first, I worked with the styles that would perform my layout.

```
.form{
    display:block;
    width:70%;
    float:left;
}

.form .row{
    display:block;
    clear:both;
}

.form .row .leftcol{
    display:block;
    width:30%;
    float:left;
}

.form .row .rightcol{
    display:block;
    width:60%;
    float:left;
}
```

Next, I worked on the style sheet for the look for the form. I added a class called "Instructions" that I could use if I wanted to include some instructions on the form. I also opted to use the style sheet to control the length of my textarea tag.

```
.form{
    text-align:left;
    background-color:#D9E2F9;
    border:1px solid #000000;
    padding:1%;
}

.form .instructions{
    display:block;
    float:left;
    margin-bottom:10px;
}

.form textarea{
    width:100%;
}
```

The Web Form

Now it was time to do that all-important markup of the content. I wanted to make sure that the markup made sense to people who used screen readers as well as those who use their cell phone to browse the web.

I started with one row. (I don't usually use this much white space in my code, but this is done here to make it easier to understand.)

```
<div class="form">

    <div class="row">

        <div class="leftcol">
            <b>First name</b>
        </div>

        <div class="rightcol">
            <input type="text" name="fname" />
        </div>

    </div>

</div>
```

This one row could then be used as my template for the entire form.

```
<div class="form">

    <div class="instructions">
        You can use this form to send us an email message. Please enter
        your first name, last name, and email address. You may also give us
        your phone number, but it is not required. Use the space on this
        form to provide us with feedback on our site or to inquire more
        about our services.
    </div>

    <div class="row">

        <div class="leftcol">
            <b>First name</b>
        </div>

        <div class="rightcol">
            <input type="text" name="fname" />
        </div>

    </div>

</div>
```

```

</div>
<div class="row">
  <div class="leftcol">
    <b>First name</b>
  </div>
  <div class="rightcol">
    <input type="text" name="fname" />
  </div>
</div>
<div class="row">
  <div class="leftcol">
    <b>Last name</b>
  </div>
  <div class="rightcol">
    <input type="text" name="lname" />
  </div>
</div>
<div class="row">
  <div class="leftcol">
    <b>Email address</b>
  </div>
  <div class="rightcol">
    <input type="text" name="email" />
  </div>
</div>
<div class="row">
  <div class="leftcol">
    Phone number
  </div>
  <div class="rightcol">
    <input type="text" name="phone1" />
  </div>
</div>
<div class="row">
  <b>Comments or questions</b><br />
  <textarea rows="5"></textarea>
</div>
</div>
</div>

```

Since the section that included my textarea box didn't need two columns, I simply wrapped the description and the textarea box in a div with the style for the row.

```

<div class="row">
  <b>Comments or questions</b><br />
  <textarea rows="5"></textarea>
</div>

```

One of the first things you'll notice is that I didn't really reduce my file size or amount of characters. In fact, with all the extra style information, I may have added more characters and made my file a little bigger. However, I did adhere more strictly to the standards, which state that I should use tags for their intended purpose. Since a div tag is designed to designate a section, this is completely standards-compliant.

If I look at my output of my HTML without applying the style sheets, you will notice that I see a completely readable and usable form.

You can use this form to send us an email message. Please enter your first name, last name, and email address. You may also give us your phone number, but it is not required. Use the space on this form to provide us with feedback on our site or to inquire more about our services.

First name

Last Name

Email address

Phone number

Comments or questions

Once the styles are applied, however, things change. I now have what looks like a table-based web form, made completely with styles.

You can use this form to send us an email message. Please enter your first name, last name, and email address. You may also give us your phone number, but it is not required. Use the space on this form to provide us with feedback on our site or to inquire more about our services.

First name

Last Name

Email address

Phone number

Comments or questions

Cleaning Up

One thing I noticed right away though was that while my form looked great in IE, there wasn't enough spacing between rows in Firefox. To correct this, I added the following style information to my layout style sheet.

```
.form .row{
    margin-bottom:10px;
    float:left;
    width:98%;
}
```

However, adding this completely screwed up my output in Internet Explorer. I could have spent hours trying to find out why this was, but I decided that since it already looks better on a browser that I hate but everyone in the world uses, I should just use one of those evil CSS hacks to correct the problem.

```
html>body .form .row{
    margin-bottom:10px;
    float:left;
    width:98%;
}
```

The **html>body** hack should be pretty familiar to most people. Internet Explorer does not understand what the > symbol means, so it ignores the entire definition. The > symbol means "child of". Firefox understands this fine, and it's completely valid. The above definition means

"Apply this style to an element with a class of "row" that is a descendant of an element with class "form" that is a descendant of the HTML tag "body" that is a child of the HTML element "html"

Extra Accessible

I could at this point add the **label** tags to my form to make it easier, but in my tests with screen reading software and with blind users, there hasn't been any benefit with this example. I'll let you decide if you want to add them.

Finishing Up

I'm sure others will have their own spin on what I've done, and I welcome any comments or suggestions on this article.