

Creating a Survey Form in HTML

Post navigation

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
Start Page: p://www.w3.org/TR/html4/strict.dtd"
>
<html lang="en">
<head>
  <title>Survey Form</title>
</head>
<body>
  <form action="#" method="post">
  |
  </form>

</body>
</html>
```

[← Previous](#)[Next →](#)

Surveys are fun! You get one in your email or regular mail or someone on the street just randomly hands you one. *My opinions matter, my life has meaning* you think to yourself, and you begin to fill out the survey.

The beginning questions are easy. You write in your name, your address and your telephone number with ease. You are confident.

But then you start getting to the hard questions. Do you completely agree with something or only partially agree with it? What is partially agreeing? How can you endorse something when you aren't even aware of the right words to use? Are you even good enough to have your opinions matter? Maybe Uncle John was right and...

You know what, I am going down a whole other road. How about we focus on the topic, okay? That'll show Uncle John and his fancy friends down at the trailer park.

Today's tutorial is focused on how to create a survey form in HTML. Fun, right? Let's get started!

To create a form we are going to use a standard HTML 4.01 document. In order for my form to look nice we are also going to write the form within a table. If you do not know how to create a table you can always check out [this earlier tutorial](#).

Step 1:

To create a form, we use the `<form>` tag with the action and method attributes. The action attribute designates where the form data should go and the method attribute determines the way in which the data will be sent.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
Start Page: p://www.w3.org/TR/html4/strict.dtd"
>
<html lang="en">
<head>
  <title>Survey Form</title>
</head>
<body>
  <form action="#" method="post">
  |
  </form>

</body>
</html>
```

There are two methods for sending form data: get and post. The important thing to note is that get is not remotely secure. All it does is attach the data to the url where you are sending the data. Not secure even remotely.

Post is a lot more secure because it sends the data within the body of the HTML. I recommend using post in most situations.

Step 2:

Let's create our first form item within a table. First we need to establish we are using a table with the `<table>` tag and designate a table row with `<tr>`. The form will go within a `<td>` tag.

The actual form input box starts with the tag `<input>`. In this situation there are two attributes to a form: type and name. Type indicates what type of input the field will be. There are numerous types of input boxes and we will review a few of them later. Name labels the data that you are sending out so that it can be processed. In this situation we are using the "text" input type so the user can input any text.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
Start Page p://www.w3.org/TR/html4/strict.dtd"
>
<html lang="en">
<head>
  <title>Survey Form</title>
</head>
<body>
  <form action="#" method="post">
    <table>
      <tr>
        <td>Customer First Name:</td>
        <td><input type="text" name="first"/></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

The first input field should output like this in the browser:

Customer First Name:

Step 3:

Now that we've created the first form, let's create the rest of the text forms. Once we have finished doing that, the code will now appear like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"
>
<html lang="en">
<head>
  <title>Survey Form</title>
</head>
<body>
  <form action="#" method="post">
    <table>
      <tr>
        <td>Customer First Name:</td>
        <td><input type="text" name="first"/></td>
      </tr>
      <tr>
        <td>Customer Last Name:</td>
        <td><input type="text" name="last"/></td>
      </tr>
      <tr>
        <td>Customer Email Address:</td>
        <td><input type="text" name="email"/></td>
      </tr>
      <tr>
        <td>Customer Phone Number:</td>
        <td><input type="text" name="phone"/></td>
      </tr>
      <tr>
        <td>Customer Street Address:</td>
        <td><input type="text" name="address"/></td>
      </tr>
      <tr>
        <td>Customer City:</td>
        <td><input type="text" name="city" /></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Putting all of the input fields in rows should make the survey look nice and neat.

Customer First Name:	<input type="text"/>
Customer Last Name:	<input type="text"/>
Customer Email Address:	<input type="text"/>
Customer Phone Number:	<input type="text"/>
Customer Street Address:	<input type="text"/>
Customer City:	<input type="text"/>

Step 4:

There are, of course, other types of input fields we can create. While we can have the user write in the state they live in, we can also use a dropdown menu and present a series of options to the user.

To do this, we use the select tag with the name attribute. The select tag creates the dropdown menu and the name labels the data.

In order to create the options on the dropdown menu, we use the option tag with the value attribute and enter a state name. Before we end the tag, we must give the tag a name that will output to the browser.

```
<td>Customer State:</td>  
<td><select name="state">  
  <option value="Alabama">AL</option>  
</select>
```

Once we have done this, AL will appear in the dropdown menu and the user can select it.

```

<td>Customer State:</td>
<td><select name="state">
  <option value="Alabama">AL</option>
  <option value="Alabama">AL</option>
  <option value="Alaska">AK</option>
  <option value="Arizona">AZ</option>
  <option value="Arkansas">AR</option>
  <option value="California">CA</option>
  <option value="Colorado">CO</option>
  <option value="Connecticut">CT</option>
  <option value="Delaware">DE</option>
  <option value="DC">DC</option>
  <option value="Florida">FL</option>
  <option value="Georgia">GA</option>
  <option value="Hawaii">HI</option>
  <option value="Idaho">ID</option>
  <option value="Illinois">IL</option>
  <option value="Indiana">IN</option>
  <option value="Iowa">IA</option>
  <option value="Kansas">KS</option>
  <option value="Kentucky">KY</option>
  <option value="Louisiana">LA</option>
  <option value="Maine">ME</option>
  <option value="Maryland">MD</option>
  <option value="Massachusetts">MA</option>
  <option value="Michigan">MI</option>
  <option value="Minnesota">MN</option>
  <option value="Mississippi">MS</option>
  <option value="Missouri">MO</option>
  <option value="Montana">MT</option>
  <option value="Nebraska">NE</option>
  <option value="Nevada">NV</option>
  <option value="New Hampshire">NH</option>
  <option value="New Jersey">NJ</option>
  <option value="New Mexico">NM</option>
  <option value="New York">NY</option>
  <option value="North Carolina">NC</option>
  <option value="North Dakota">ND</option>
  <option value="Ohio">OH</option>
  <option value="Oklahoma">OK</option>
  <option value="Oregon">OR</option>
  <option value="Pennsylvania">PA</option>
  <option value="Rhode Island">RI</option>
  <option value="South Carolina">SC</option>
  <option value="South Dakota">SD</option>
  <option value="Tennessee">TN</option>
  <option value="Texas">TX</option>
  <option value="Utah">UT</option>
  <option value="Vermont">VT</option>
  <option value="Virginia">VA</option>
  <option value="Washington">WA</option>
  <option value="West Virginia">WV</option>
  <option value="Wisconsin">WI</option>
  <option value="Wyoming">WY</option>
</select>

```

In order to fully populate the list,

we keep using the option tag for every state.

Customer First Name:

Customer Last Name:

Customer Email Address:

Customer Phone Number:

Customer Street Address:

Customer City:

Customer State:

✓ AL

AL

AK

AZ

AR

CA

CO

CT

DE

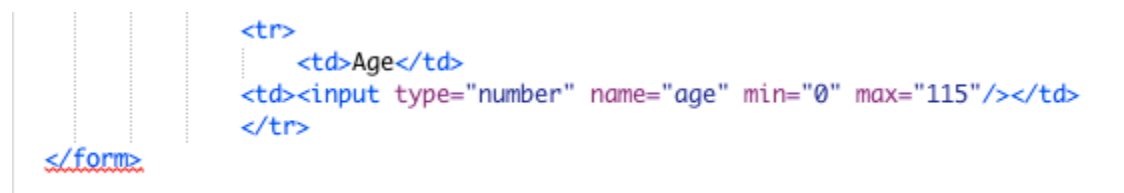
DC

The user should be able to select from a list of states.

Step 5:

Another form field we can use is the “number” type. The number type is an HTML5 input that displays differently in different browsers. In some browsers it will bring up a box that allows you to increase and decrease the number. Unfortunately, in some browsers all it will do is bring up a blank text field.

To use the number type, we use the `<input>` tag with the `type` attribute and select “number” as the value. A couple of attributes that we can use with the number type is “min” and “max” which will allow us to set the range of numbers the user can choose from.



If the user’s browser supports HTML5, this is what should appear:

Customer First Name:	<input type="text"/>
Customer Last Name:	<input type="text"/>
Customer Email Address:	<input type="text"/>
Customer Phone Number:	<input type="text"/>
Customer Street Address:	<input type="text"/>
Customer City:	<input type="text"/>
Customer State:	<input type="text" value="AL"/>
Age	<input type="text" value="37"/>

Step 6:

Another great input type is the date type. In HTML5, this type brings up a calendar that users can select from in compatible browsers. This is accomplished with the value “date” in the type attribute.

```
<tr>  
  <td>DOB:</td>  
  <td><input type="date" name="DOB" id="DOB" /></td>  
</tr>
```

This really does look nice on browsers like Opera:

Customer First Name:
Customer Last Name:
Customer Email Address:
Customer Phone Number:
Customer Street Address:
Customer City:
Customer State:
Age
DOB:



While it's not so great on some (yet):

Customer First Name:
Customer Last Name:
Customer Email Address:
Customer Phone Number:
Customer Street Address:
Customer City:
Customer State:
Age
DOB:

Step 7:

A really useful feature of HTML forms is the radio button input. This input will allow us to define a series of values and allow the user to select one. To accomplish this, you use the type “radio” and the “value” and “name” attributes. The value attribute defines the choice the user is selecting and the name refers to the radio buttons. We must keep the name the same so that the HTML knows that the buttons are in the same group and the user can only choose one.

```
<td>What Type of Crust Do You Prefer?</td>
<td>
  <input type="radio" value="Thin Crust" name="crust">Thin Crust
  <input type="radio" value="Thick Crust" name="crust">Thick Crust
  <input type="radio" value="Deep Dish" name="crust">Deep Dish
</td>
```

This is how the radio button looks on the browser. Remember that the user can only choose one item.

Customer First Name:	<input type="text"/>
Customer Last Name:	<input type="text"/>
Customer Email Address:	<input type="text"/>
Customer Phone Number:	<input type="text"/>
Customer Street Address:	<input type="text"/>
Customer City:	<input type="text"/>
Customer State:	<input type="text" value="AL"/>
Age	<input type="text" value=""/>
DOB:	<input type="text" value="Month/Day/Year"/>
What Type of Crust Do You Prefer?	<input type="radio"/> Thin Crust <input type="radio"/> Thick Crust <input checked="" type="radio"/> Deep Dish

Step 8:

There might be another time when you want the user to be able to select multiple items from a list. In this case, we would use a checkbox.

A checkbox is created in the same way that radio buttons are created, just substituting checkbox for radio.

```
<tr>
<td>Pizza Toppings You Like</td>
<td>
<input type="checkbox" value="Pepperoni" name="Toppings"/>Pepperoni<br/>
<input type="checkbox" value="Mushroom" name="Toppings"/>Mushroom<br/>
<input type="checkbox" value="Sausage" name="Toppings"/>Sausage<br/>
<input type="checkbox" value="Hamburg" name="Toppings"/>Hamburg<br/>
<input type="checkbox" value="Peppers" name="Toppings"/>Peppers<br/>
<input type="checkbox" value="Anchovies" name="Toppings"/>Anchovies<br/>
</td>
</tr>
```

This input will allow the users to select from a list of different items.

Customer First Name:	<input type="text"/>
Customer Last Name:	<input type="text"/>
Customer Email Address:	<input type="text"/>
Customer Phone Number:	<input type="text"/>
Customer Street Address:	<input type="text"/>
Customer City:	<input type="text"/>
Customer State:	<input type="text" value="AL"/>
Age	<input type="text"/>
DOB:	<input type="text" value="Month/Day/Year"/>
What Type of Crust Do You Prefer?	<input type="radio"/> Thin Crust <input checked="" type="radio"/> Thick Crust <input type="radio"/> Deep Dish
	<input checked="" type="checkbox"/> Pepperoni
	<input checked="" type="checkbox"/> Mushroom
	<input type="checkbox"/> Sausage
	<input type="checkbox"/> Hamburg
	<input type="checkbox"/> Peppers
	<input checked="" type="checkbox"/> Anchovies
Pizza Toppings You Like	

Step 9:

To complete the form, we need to add buttons for the user to submit the form. We are also going to add a button to reset the information. This is accomplished with the input types “reset” and “submit”.

```

        <td><input type="reset" /></td>
        <td><input type="submit" value="Send Info" />
    </table>
</form>

```

Don't forget to end your table and form, otherwise you might end up with a mess.

In the end, we are left with a nice looking and functional survey form:

Customer First Name:	<input type="text" value="Brett"/>
Customer Last Name:	<input type="text" value="Lassoff"/>
Customer Email Address:	<input type="text" value="blasoff@yahoo.com"/>
Customer Phone Number:	<input type="text" value="(860)550-0000"/>
Customer Street Address:	<input type="text" value="2 Charley Rd."/>
Customer City:	<input type="text" value="Hampton"/>
Customer State:	<input type="text" value="CT"/>
Age	<input type="text" value="32"/>
DOB:	<input type="text" value="1/31/80"/>
What Type of Crust Do You Prefer?	<input type="radio"/> Thin Crust <input checked="" type="radio"/> Thick Crust <input type="radio"/> Deep Dish
	<input checked="" type="checkbox"/> Pepperoni <input checked="" type="checkbox"/> Mushroom <input type="checkbox"/> Sausage <input type="checkbox"/> Hamburg <input type="checkbox"/> Peppers <input checked="" type="checkbox"/> Anchovies
Pizza Toppings You Like	
<input type="button" value="Reset"/>	<input type="button" value="Send Info"/>