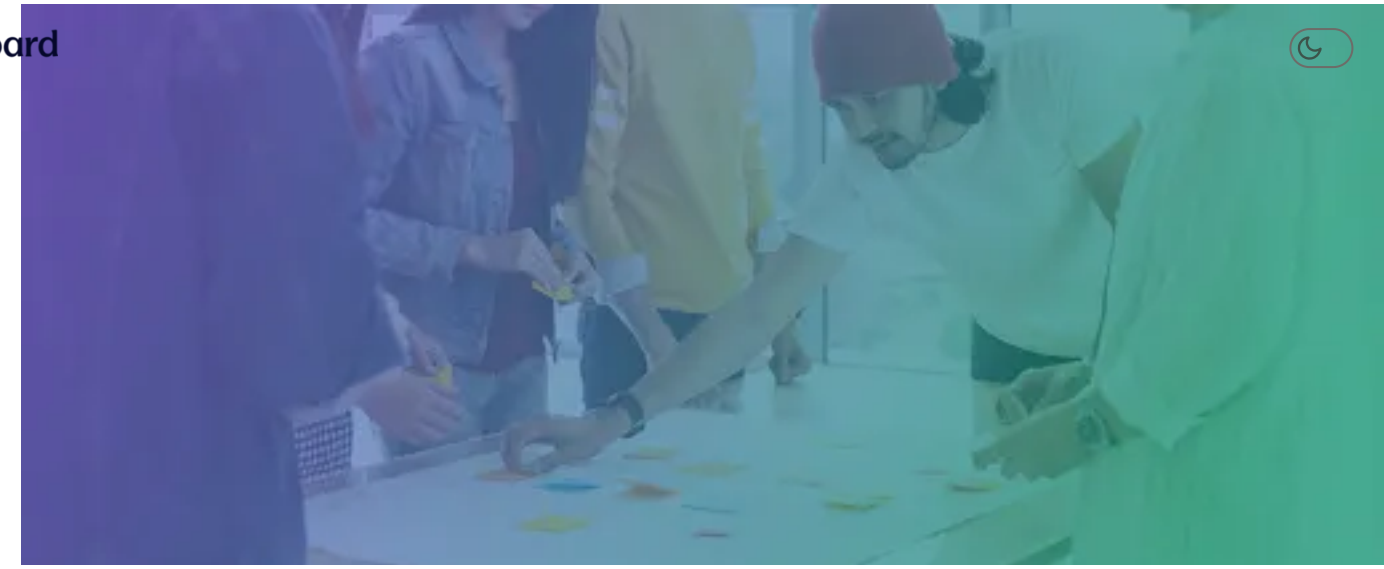
Springboard



HTML: Tables and Forms

Demo code.zip 3.5KB

Goals

- Create tables in HTML
  - Headers, rows, and cells
- Write forms in HTML
  - Textual & numeric inputs
  - Checkboxes and radio buttons
  - Longer textual areas
  - Form Validations

Tables

Tables are...a lot of work:

- ***table***: entire table
- ***thead*** and ***tbody***: header area and body area
- ***tfoot***: footer area
- ***tr***: Individual rows in table
- ***th***: Individual header cells
- ***td***: Individual content cells

- ***caption***: For captioning entire table

Simple Tables

```
<table> <tr>
<td>Cat</td> <td>4</td>
</tr> <tr>
<td>Octopus</td>
<td>8</td> </tr>
</table>
```

Simple table of animals & # legs:

Cat	4
Octopus	8

Header Cells

```
<table> <tr>
<th>Animal</th> <th># Legs</th> </tr> <tr>
<td>Cat</td> <td>4</td>
</tr> <tr>
<td>Octopus</td>
<td>8</td> </tr>
</table>
```

Table with header cells:

Animal	**# Legs**
Cat	4
Octopus	8

Header cells use ***th***, rather than ***td*** *(table data)*

Table Structure

We've seen how to structure basic tables, but we're missing some important pieces.

It's best to add elements for header, body, and footer to your table:

- screen readers can show users where different sections start/end
- it's easier to parse out different sections in code
- printouts of tables can repeat header cells/footer tags on each page

Table Headers and Bodies

You can better differentiate sections of table with tags: ***thead*** (head area), and ***tbody*** (body)

```
<table> <thead> <tr>
<th>Animal</th> <th>#
Legs</th> </tr>
</thead> <tbody> <tr>
<td>Cat</td> <td>4</td>
</tr> <tr>
<td>Octopus</td>
<td>8</td> </tr>
</tbody> </table>
```

Table with clear sections:

Animal	# Legs
Cat	4
Octopus	8

The table section elements (***thead***, ***tbody***, ***tfoot***) were added to HTML standard long after the simple table elements were added — so many existing examples of tables or explanations of tables may not have them.

Adding Footer Section

To clearly indicate which cells are footer cells, put them in a ***tfoot*** section:

```
<table> <thead> ...
</thead> <tbody> ...
</tbody> <tfoot> <tr>
<th>Totals</th>
<td>12</td> </tr>
</tfoot> </table>
```

Table with footers sections:

Animal	# Legs
Cat	4
Octopus	8
Totals	12

Table Captions

You can add a caption with a ***caption*** element:

```
<table>
<caption>Animals and
Leg Counts</caption>
<thead> ... </thead>
<tbody> ... </tbody>
<tfoot> ... </tfoot>
</table>
```

Table with Captions:

Animals and Leg Counts

Animal	# Legs
Cat	4
Octopus	8
Totals	12

Advanced Table Concepts

There's even more to learn with tables!

- making cells span more than one row
- making cells span more than one column
- identifying *groups* of cells

You can learn more about these features at MDN

Forms

Forms in HTML present opportunities for users to enter data

The forms won't do much unless you also add in...

- JavaScript to process the form data in the browser
- A server to send the form data to

Form Element

The ***form*** element wraps around all inputs in a form:

```
<form> ... individual inputs on form </form>
```

Form Attributes

Often the ***form*** element has other attributes that specify where and how data gets sent to a server

This is outside our scope for now, but here's a quick example:

```
<form action="/survey-submission" method="post">
```

This would send the form data to a server-side page called "survey-submission", using the "post" method, which tells servers to store the data.

> **Note: JavaScript and Forms**
> Technically, if the browser will use JavaScript directly to process the form input, it often doesn't need to wrap the inputs in a form tag — but it's still common and a good idea to do so.

Textual Elements

A common way to collect data is using plain text fields.

These are created via the ***input*** element:

```
<form> <p>First Name: <input name="first-name"> </p> </p>
... other fields ... </form>
```

- Note that ***input*** doesn't have a closing tag
- The name attribute won't change anything visually. The name value is used to "label" the data when it is sent to a server (or client-side JS).

Textual inputs are the most common & default — but there are many other types

The ***input*** element takes a ***type*** attribute,

To manually make a text field (the default if you leave off ***type***):

```
<input type="text" name="first-name">
```

Other Options

The ***input*** accepts many other attributes, here are a few:

minlength

Minimum number of characters to be valid

maxlength

Maximum number of characters to be valid

placeholder

Description that appears when input is empty

required

Whether this input is required

This is an incomplete list, featuring some of the most popular attributes — for a full listing, see MDN.

Here's a text input with multiple attributes:

```
<input type="text" name="first-name" minlength="2"
maxlength="20" placeholder="Enter first name" required>
```

Most browsers will not let you submit a form that has invalid inputs.

Other Types

Let's take a look at other ***input*** types:

- ***button***: clickable buttons
- ***checkbox***: checkboxes that can be turned on/off
- ***color***: for picking colors from a color picker
- ***date***: date picker
- ***email***: text input with email validation
- ***number***: enter numbers with up/down controls
- ***password***: enter passwords that do not show on screens
- ***radio***: pick from choices ("radio buttons")
- ***url***: entering URLs and validating them

This is a small sampling of the options — for a full list, see MDN.

Let's look at some of these in detail

Checkbox

Checkboxes can be turned on/off:

```
<form> <input type="checkbox" name="extra-sprinkles"> Add
extra sprinkles? </form>
```

If that checkbox is checked when form is submitted, JavaScript or server will be informed of that.

Email

Enter email address *(like text, but checks if email is valid-looking)*

```
<form> Enter your email: <input type="email" name="user-
email"> </form>
```

Password

Enter password *(like text, but doesn't appear on screen)*

```
<form> Enter password: <input type="password" name="new-
password"> </form>
```

Radio

Radio buttons typically show a series of choices, and you pick just one:

```
<form> Type of bicycle: <input type="radio" name="bike-
type" value="road"> Road <br> <input type="radio"
name="bike-type" value="mtn"> Mountain <br> <input
type="radio" name="bike-type" value="hybrid"> Hybrid <br>
<form>
```

We use the same ***name*** for all of these choices — we vary the ***value***

This makes these "one choice out of several" — JavaScript/server or server is sent the value so they know which was choice

Form Submission

In order for a form to be submitted, you need to add one of these:

- ***input*** with type of ***submit***
- ***button*** with type of ***submit***

Input Type of Submit

```
<form> First Name <input name="first-name"> <br> Last Name:
<input name="last-name"> <br> <input type="submit"
value="Go!"> </form>
```

This shows a button labeled “Go!” which, when clicked, submits form

Button Type of Submit

```
<form> First Name <input name="first-name"> <br> Last Name:
<input name="last-name"> <br> <button type="submit">Go!
</button> </form>
```

This *also* shows a button labeled “Go!” which, when clicked, submits form

This way is newer and more flexible — you can add other HTML elements inside button

Form Submission

Handling forms isn’t something you can do in HTML directly; JavaScript or servers do this.

Until you learn how to handle forms, browsers will just redirect you to the same page, but the URL will now reflect the form data

So, our first-name/last-name form would show a URL like:

`example-form?first-name=Jane&last-name=Smith`

Looking at that URL and looking for the first name (“Jane”) and the last name (“Smith”) can indicate that your form works, and is ready to be hooked up to JavaScript or a server.

Advanced Form Inputs

There are additional types of form fields that don't use the ***input*** element:

- drop down menus
- larger text blocks
- submission of files

Drop Down Menus

Drop down menus (where you can pick exactly one choice) use ***select*** and ***option*** elements:

```
Pick an animal: <select name="animal"> <option
value="cat">Cat</option> <option value="dog">Dog</option>
<option value="porcupine">Porcupine</option> </select>
```

That results in a drop-down menu with options of cat/dog/porcupine.

It starts with "cat" selected — if you want to start with an empty choice, add an option with ***value*** of `""`, and add some text like "Select choice"

Larger Text Areas

The ***input*** type of ***text*** accepts one line of text.

If you want larger text fields that can also include linebreaks, use ***textarea*** instead:

```
<textarea name="biography"></textarea>
```

Notice that this has a starting and ending tag.

This allows you to enter default text (for example, a generic bio)

You can set height/width of box with ***rows*** and ***cols*** attributes:

```
<textarea name="biography" rows="10" cols="40"></textarea>
```

File Submission

File submission requires that your server is set up to receive the file.

It requires you to change the ***form*** tag and have an ***input*** type of ***file***:

```
<form method="post" enctype="multipart/form-data"> Choose file to upload: <input type="file" name="my-file"> <button type="submit">Upload File</button> </form>
```

This submits the form in a *special* way so that it includes the file data.