

# HTML Tables & Form

Abhi Chakraborty ( 17th June 2022 )

# Tables

## Basics

- Table-related tags are meant for tabulating data.
- (Older HTML documents tend to use `<table>` for formatting the document to divide the document into columns/sections, which should be avoided. Use style sheet for formatting instead.)
- The basic unit of a table is a *cell*. Cells are grouped into *row*.
- Rows are grouped to form the *table*.
- This corresponds well to the "row-centric" approach in the display.

# Tables

## Essential tags

- <table>...</table>: contains the entire table.
- <tr>...</tr>: contains a row.
- <th>...</th> and <td>...</td>: contain a header cell and a data (detail) cell respectively.

# Tables

## Additional tags

- <caption>...</caption>: specifies a caption.
- <thead>...</thead>, <tbody>...</tbody>, and <tfoot>...</tfoot>: for marking out the table header, body and footer.
- <colgroup>...</colgroup> and <col>...</col>: for applying styles to column group and column respectively.

# Example

```
<table>
  <caption>Price List</caption>
  <tr>
    <th>Fruit</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>$0.50</td>
  </tr>
  <tr>
    <td>Orange</td>
    <td>$0.65</td>
  </tr>
</table>
```

# Table <table>...</table>

- Function: Set up a table, consisting of rows of cells.
- Three optional presentation attributes, `border="n"` (specifies the width of borders, in pixels), `cellspacing="n"` (specifies the spacing between cells, in pixels), and `cellpadding="n"` (define the spacing between the content of the cell and its boundaries, in pixels), are often used in older HTML pages but now deprecated.
- The now-preferred approach is to use CSS (again! but coming soon!).

# Table Row , Header & CellSpan

- Table Row `<tr>...</tr>` :
- Function: Set up a row inside a table, consisting of cells.
- Table Header Cell `<th>...</th>`, Table Data Cell `<td>...</td>` :
- Function: Set up each individual cell of a row (of a table). `<th>...</th>` defines a header cell (usually displayed in bold with center alignment) and `<td>...</td>` defines a body cell.
- An empty cell is typically marked as `<td>&nbsp;</td>`.
- Cell Span: Attributes "rowspan" and "colspan" :
  - On a `<td>` or `<th>` cell, we can use the attribute `rowspan="numOfRows"` or `colspan="numOfColumns"` to span the cell to occupy multiple rows or columns. The subsequent `<td>` or `<th>` cells will adjust their positions accordingly.

# Example 1

```
<table>
  <tr>
    <td>11111</td>
    <td>22222</td>
    <td>33333</td>
  </tr>
  <tr>
    <td>44444</td>
    <td rowspan="2">55555</td>
    <td>66666</td>
  </tr>
  <tr>
    <td>77777</td>
    <td>88888</td>
  </tr>
</table>
```

## Example 2

```
<table>
  <tr>
    <td colspan="2" rowspan="2">11111</td>
    <td>22222</td>
  </tr>
  <tr>
    <td>33333</td>
  </tr>
  <tr>
    <td>44444</td>
    <td>55555</td>
    <td>66666</td>
  </tr>
</table>
```

# Table Header , Body & Footer

- Table Header : <thead>...</thead>
- Table Body : <tbody>...</tbody>
- Table Footer : <tfoot>...</tfoot>
- Function: Used to define a header, body and footer sections for a table.
- Browser may duplicate the header or footer when breaking the table across multiple pages (in print-out).
- They can also be used to apply styles to each of the sections.

# Column Group <colgroup>...</colgroup> and Column <col>

- An HTML table is row-centric. It consists of rows of cells. Nonetheless, you can also identify the columns via <colgroup> or <col>.
- But the features are really limited. Only a few properties could be applied on columns: border, background, width, visibility and display.
- Function: <colgroup>...</colgroup> can be used to group a set of columns, so that styles can be applied to all the columns in the group.

# Column Group <colgroup>...</colgroup> and Column <col>

- Similarly, <col> can be used to identify a column for applying styles
- The attribute `span="numOfColumns"` specifies the number of columns belonging to this <colgroup> or <col> declaration
- A <col> not nested under <colgroup> is automatically wrap under a <colgroup>.

# Example 1

```
<table>
  <!-- col group 1 spans 3 columns -->
  <colgroup span="3" style="background-color:lightgrey"></colgroup>
  <!-- col group 2 spans 1 columns -->
  <colgroup span="1"></colgroup>
  <tr>
    <td>Col 1 is in the group</td>
    <td>Col 2 is in the group</td>
    <td>Col 3 is in the group</td>
    <td>Col 4 is NOT in the group</td>
  </tr>
  <tr>
    <td>Col 1 is in the group</td>
    <td>Col 2 is in the group</td>
    <td>Col 3 is in the group</td>
    <td>Col 4 is NOT in the group</td>
  </tr>
</table>
```

# Example 2

```
<table>
  <!-- colgroup 1 consists of col 1 and col 2 -->
  <colgroup>
    <col style="background-color:lightyellow" />
    <col style="background-color:white" />
  </colgroup>
  <!-- colgroup 2 consists of col 3 and col 4 -->
  <colgroup style="background-color:lightgrey" >
    <col span="2" />
  </colgroup>
  <tr>
    <td>Col 1 in the group 1</td>
    <td>Col 2 in the group 1</td>
    <td>Col 3 in the group 2</td>
    <td>Col 4 in the group 2</td>
  </tr>
  <tr>
    <td>Col 1 in the group 1</td>
    <td>Col 2 in the group 1</td>
    <td>Col 3 in the group 2</td>
    <td>Col 4 in the group 2</td>
  </tr>
</table>
```

# HTML Forms

- An **HTML** (HyperText Markup Language) **form** is a document that contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc., to store the information of a user on the web server.
- An HTML form is used to enter data that is to be sent to the server for processing, such as name, email address, password, phone number, etc., using interactive controls.
- The <form> element of HTML is used to create HTML form which is used for user to input information.
- The <form> element contains different types of input elements. Some of the form elements are: text fields, checkboxes, radio buttons, submit buttons, etc.

# Form Elements

| Tag        | Description  |
|------------|--|
| <form>     | This element defines an HTML form for user input                                       |
| <input>    | This element defines fines an input control  |
| <textarea> | This element defines a multiline input control (text area)                             |
| <label>    | This element defines a label for an <input> element                                    |
| <fieldset> | This element groups related elements in a form   |
| <legend>   | This element defines a caption for a <fieldset> element                                |
| <select>   | This element defines a drop-down list  |
| <optgroup> | This element of HTML <form> defines a group of related options in a drop-down list     |
| <option>   | This element defines an option in a drop-down list                                     |
| <button>   | This element defines a clickable button  |
| <datalist> | This element of HTML <form> specifies a list of pre-defined options for input controls |
| <output>   | This element defines the result of a calculation.                                      |

# HTML Form Attributes

The **HTML form Attribute** specifies that the element can contain one or more forms. The various `<form>` attributes are defined below:

- The Action Attribute :
- As the same specifies, the action attribute defines the action to be performed when the form is submitted
- When the submit button is clicked by the user, the form data is sent to a file that is on the server
- Let us understand with an example below: the form data is sent to a file named “action\_sample.php”. The file consists of a server-side script that handles the form data:

# Action Attribute Example

On submit, send form data to “action\_sample.php”:

```
<form action="/action_sample.php">
  <label for="first_name">First name:</label><br>
  <input type="text" id="first_name" name="first_name" value="Chris"><br>
  <label for="last_name">Last name:</label><br>
  <input type="text" id="last_name" name="last_name" value="Darwin"><br><br>
  <label for="age">Age:</label><br>
  <input type="text" id="age" name="age" value="27"><br><br>
  <label for="country">Age:</label><br>
  <input type="text" id="country" name="country" value="India"><br><br>

  <input type="submit" value="Submit">
</form>
```

# The Target Attribute

- As the name indicates, the target attribute specifies where the response is displayed after the user submits the form.
- The target attribute can have one of the following values as mentioned in the table below
- The default value of the target attribute is `_self` which means that the response will be displayed in the current window
- `<form action="/action_sample.php" target="_blank">`
- The submitted result of above query will open in a new browser tab.

# The Target Attribute

| <b>Value</b> | <b>Description</b>   |
|--------------|--|
| _blank       | This value means the response is displayed in a new window or tab.         |
| _self        | This value means the response is displayed in the current window.          |
| _parent      | This value means the response is displayed in the parent frame.            |
| _top         | This value means the response is displayed in the full body of the window. |
| framename    | This value means the response is displayed in a named iframe.              |

# The Method Attribute

- The method attribute is used to specify the HTTP method to be used when submitting the form data
- GET method is the default HTTP method used when submitting form data
- The form-data is sent as URL variables (with method="get") or as a HTTP post method(with method="post")
- Let us take an example that uses the GET method when submitting the form data
- <form action="/action\_sample.php" method="get">
- An example using he POST method when submitting the form data is as below
- <form action="/action\_sample.php" method="post">

# Important points to remember when using the GET method

- The form data is appended to the URL with the Get method, in name/value pairs
- We should never use the Get method to send sensitive data that is not meant to be exposed. It is because the submitted form data is visible in the URL. So the sensitive information will be exposed.
- The length of a URL when using the Get method is limited to 2048 characters
- GET method should be used for non-secure data, like query strings in Google

## **Important points to remember when using the POST method are**

- Post method appends the form data inside the body of the HTTP request. The form data that is submitted is not shown in the URL.
- Post method can be used to send large data; it has no size limitation
- Form submissions that are done with the Post method cannot be bookmarked
- Note: We should always use the Post method if the form data contains any sensitive information that is not meant to be exposed

# The Autocomplete Attribute

- As the name autocomplete specifies, the browser automatically fills the values based on what the user has filled earlier when this attribute is on.
- With the help of the autocomplete attribute, users can specify whether a form should have autocomplete on or off
- Example of autocomplete attribute is with autocomplete ON is:
- <form action="/action\_sample.php" autocomplete="on">

# The Novalidate Attribute

- The Novalidate attribute of HTML form is a boolean attribute. As we know, boolean means 2 values, either yes or no.
- So when the novalidate attribute of the HTML form is present, it indicates or specifies that the form-data should not be validated when submitted.
- An example of a form with a novalidate attribute is as below
- <form action="/action\_sample.php" novalidate>

# Below is the list of all <form> attributes

| Attribute      | Description  |
|----------------|--|
| accept-charset | This attribute is used to define or specify the character encodings used for form submission                                 |
| Action         | This attribute is used to define or specify where to send the form data when a form is submitted                             |
| autocomplete   | This attribute is used to tell whether a form should have autocomplete on or off   |
| enctype        | This attribute is used to tell how the form-data should be encoded when submitting it to the server (only for method="post") |
| method         | This attribute is used to tell the HTTP method to use when sending form-data   |
| Name           | This attribute is used to tell the name of the form  |
| novalidate     | This attribute is used to tell that the form should not be validated when submitted  |
| Rel            | This attribute is used to tell the relationship between a linked resource and the current document                           |
| Target         | This attribute is used to define where to display the response that the user receives after submitting the form              |



Thank You  
Keep Bashing The Walls

Abhi Chakraborty ( 17th June 2022 )