

HTML: Part 2



Roadmap

- Tables
- Forms
- CSS



Tables

- Tables are defined with the `<table>` tag
- Tables are useful for displaying large row/column data sets
- Tables are drawn by rows
- A row is created with `<tr>` element
 - Inside row: `<th>` element is used for table header
 - `<td>` is used for table data
 - A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.
- NOTE: tables are sometimes used for web page layout, but this is not proper semantic HTML and should be avoided



Table example

```
<table border="1">
<tr> <th>Heading 1</th>
      <th>Heading 2</th> </tr>
<tr> <td>row 1, cell 1</td>
      <td>row 1, cell 2</td> </tr>
<tr> <td>row 2, cell 1</td>
      <td>row 2, cell 2</td> </tr>
</table>
```

Heading 1	Heading 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Table with border-collapse

Heading 1	Heading 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

- If you do not specify a border for the table, it will be displayed without borders
- By default, the overall table has a separate border from each cell inside
- The **border-collapse** property merges these borders into one
 - You can define this as an attribute as

```
<table style="border-collapse: collapse">
```

- All standard CSS styles can be applied to a table, row, or cell



Table headers and Captions

```
<table style="border-collapse: collapse" border="1">  
  <caption>This is the header of the table</caption>  
  <tr><th>Column 1</th><th>Column 2</th></tr>  
  <tr><td>Row 1, Column 1</td><td>Row 1, Column 2</td></tr>  
  <tr><td>Row 2, Column 1</td><td>Row 2, Column 2</td></tr>  
</table>
```

This is the header of the table	
Column 1	Column 2
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

- **th** cells in a row are considered as headers; by default, they appear bold
- A **caption** at the start of the table labels its meaning



Table Spans and Paddings

- A cell can span more than one row or column
 - To make a cell span more than one column, use the **colspan** attribute within `th` or `td`
 - E.g., `<th colspan="2">Telephone</th>` means this particular cell will span two columns
 - To make a cell span more than one row, use the **rowspan** attribute within `th` or `td`
 - E.g., `<th rowspan="2">Telephone</th>` means this particular cell will span two rows
- **Cellpadding** specifies the space between the cell content and its borders

```
padding-top: 10px;  
padding-bottom: 20px;  
padding-left: 30px;  
padding-right: 40px;
```



Table Spans - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one column,
  use the colspan attribute.</p>

<table style="width:30%">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
</body>
</html>
```

Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name		Age
Jill	Smith	43
Eve	Jackson	57

Table Spans - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row,
  use the rowspan attribute.</p>

<table>
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
</table>
</body>
</html>
```

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Jill
Phone	555-1234
	555-8745

Table Spans - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row,
  use the rowspan attribute.</p>

<table style="width:30%">
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
</body>
</html>
```

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Jill
Phone	555-1234
	555-8745

HTML Form

- An HTML form is used to collect user input
- The user input is most often sent to a server for processing
- To create an HTML form:
`<form>...</form>`
- The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.



HTML Form

➤ Text input

- Single and multi-line texts
- Masked characters (password)

➤ Choices

- Radio buttons
 - Select one (of multiple options)
- Checkboxes
 - Select one or more options
- Drop-down list



HTML Form – Text input

```
<form action="">
```

User Name:

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

```
<br><br>
```

Password:

```
<input type="password" name="pass">
```

```
</form>
```

User Name:

Password:

HTML Form – Text input

```
<form action="">  
  <textarea name="description" rows="20" cols="30">  
    Description of your online business ideas  
  </textarea>  
  <br><br>  
  <input type="submit">  
</form>
```

Description of your online
business ideas

Submit

HTML Form – Radio buttons

➤ Radio button

- let a user select ONLY ONE of a limited number of choices
- name attribute: sent to server with value of selected option. Same name for all options
- value: option value
- checked: indicate which value should be selected when the page load

☒ Red
☐ White
☐ Navy

```
<form>
  <input type="radio" name="colour" value="red"
checked="checked"> Red<br>
  <input type="radio" name="colour" value="white"> White<br>
  <input type="radio" name="colour" value="navy"> Navy
</form>
```

HTML Form – Checkboxes

➤ Checkboxes

- `<input type="checkbox">` defines a checkbox
- Let a user select one or more options of a limited number of choices

☐ I have a bike

☐ I have a car

☐ I have a boat

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike">
    I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car">
    I have a car<br>
  <input type="checkbox" name="vehicle3" value="Boat">
    I have a boat<br><br>
</form>
```

HTML Form – Drop down lists

- Drop down list box
 - `<select>` is used to create drop down list box
 - `<option>` specify options user can select from
 - Value attribute is sent to server along with name
- If the list is short, and user needs to see all options quickly, go for radio buttons
- If the list of options is long, go for drop down list

```
<form >  
  <select name="colour">  
    <option value="red">Red</option>  
    <option value="white">White</option>  
    <option value="navy">Navy</option>  
    <option value="black">Black</option>  
  </select>  
  <br><br>  
  <input type="submit">  
</form>
```



Red ▼

Submit

Form – Submit

➤ Submit

- Input type="submit"
- Send form to server
- Value: display text to appear on a button (e.g. subscribe, send, submit, etc.)
- If you omit the submit button's value attribute, the button will get a default text

Please enter your email to subscribe

Subscribe

```
<form action="subscribe.php" method="post">  
  Please enter your email to subscribe<br>  
  <input type="text" name="email"></input><br>  
  <input type="submit" value="Subscribe">  
</form>
```

HTML Form - The Action Attribute

- Form elements require action attribute
- Value of action is a URL for a page on the server which receive the form information when it is submitted
- If the action attribute is omitted, the action is set to the current page
- E.g.

```
<form action="booking.php" method="get">
```



HTML Form - The Method Attribute

- The method attribute specifies the HTTP method to be used when submitting the form data.
 - ✓ The form-data can be sent as URL variables (with method="get")
 - ✓ or as HTTP post transaction (with method="post").
- The default HTTP method when submitting form data is GET.

```
<form action="/action_page.php" method="get">
```

<https://www.w3schools.com/html/default.asp>

HTML Form - The Method Attribute

➤ GET method:

- Appends the form data to the URL, in name/value pairs
- `www.test.com/demo.php?name1=value1&name2=value2`
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- GET is good for non-secure data, like query strings in Google

➤ POST method:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data

<https://www.w3schools.com/html/default.asp>

