

# HTML Table

An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

# HTML Table

## **Cellpadding and Cellspacing Attributes**

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The *cellspacing* attribute defines space between table cells, while *cellpadding* represents the distance between cell borders and the content within a cell.

## **Colspan and Rowspan Attributes**

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

# HTML Table

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Table Colspan/Rowspan</title>
  </head>
  <body>
    <table border = "1">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
      <tr>
        <td colspan = "3">Row 3 Cell 1</td>
      </tr>
    </table></body>
</html>
```

# HTML Table

## Tables Backgrounds

You can set table background using one of the following two ways –

- **bgcolor** attribute – You can set background color for whole table or just for one cell.

- **background** attribute – You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

# HTML Table

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Table Background</title>
  </head>

  <body>
    <table border = "1" bordercolor = "green" bgcolor = "yellow">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
      <tr>
        <td colspan = "3">Row 3 Cell 1</td>
      </tr>
    </table>
  </body>
</html>
```

## Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Table Width/Height</title>
  </head>

  <body>
    <table border = "1" width = "400" height = "150">
      <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>
  </body>
</html>
```

# Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table.

```
<table border = "1" width = "100%">
  <caption>This is the caption</caption>
  <tr>
    <td>row 1, column 1</td><td>row 1, columnn 2</td>
  </tr>
  <tr>
    <td>row 2, column 1</td><td>row 2, columnn 2</td>
  </tr>
</table>
```

## **Table Header, Body, and Footer**

Tables can be divided into three portions – a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are –

**<thead>** – to create a separate table header.

**<tbody>** – to indicate the main body of the table.

**<tfoot>** – to create a separate table footer.



```
<table border = "1" width = "100%">
  <thead>
    <tr>
      <td colspan = "4">This is the head of the table</td>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <td colspan = "4">This is the foot of the table</td>
    </tr>
  </tfoot>

  <tbody>
    <tr>
      <td>Cell 1</td>
      <td>Cell 2</td>
      <td>Cell 3</td>
      <td>Cell 4</td>
    </tr>
  </tbody>

</table>
```

## Nested Tables

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag `<td>`.

A border is set using the CSS **border** property:

```
table, th, td {  
    border: 1px solid black;  
}
```

To left-align the table headings, use the CSS **text-align** property:

```
th {  
    text-align: left;  
}
```

# CASCADING STYLE SHEETS (CSS)

Cascading style sheets provide a means to apply a presentation to an HTML structure by defining how HTML elements are displayed. By using CSS, you can set background and fore-ground colors, margins, borders, fonts, positions, and much more. You can apply a style to an element that causes it to render differently. ***A style is a rule that describes how to format a specific part of an HTML document.*** A style sheet is a set of style rules. You can create a style and apply it to many elements based on a selector. You use a *selector to locate and select elements based on tag name, class name, ID, and more.* You can create a style that works with images, and you can create a style that works only with hyper-links. You can also create a named style that you can apply to any element. The reusability is powerful.

# Cascading

The most powerful feature of CSS is *cascading*. The browser could be reading multiple style sheets for an HTML page, in which each style sheet could have style rules that affect the effective style of the element. As each style is read, it modifies the effective style of the element. Style settings that were applied from one style sheet can be overwritten by styles that are subsequently evaluated. The effective style cascades from one style sheet to the next style sheet, possibly being overwritten along the way.

# Defining and applying a style

A style rule, or style, is composed of two parts: the *selector*, which locates the elements in the HTML document that will be styled, and the *declaration block*, which contains the formatting instructions (declarations). Multiple declarations are always separated with a semicolon. A declaration comprises a CSS property, followed by a colon, followed by a value. The following is an example of a style rule that locates the `<body>` element and sets the background color to white and the font color to gray.

```
body { background-color: white; color: gray; }
```

In this example, the selector is `body` and the declaration block is contained within the curly braces. There are two declarations, each terminated with a semicolon. The first declaration specifies the CSS `background-color` property followed by a colon separator and the property value of `white`.

# Adding comments within a style sheet

You can add comments within a style sheet by using the `/*` characters to start the comment and the `*/` characters to end the comment.

Comments may also span multiple lines, as shown in the following example.

```
/* This is the style for the body element */  
body { background-color: white;  
/* The rgb value is #ffffff */  
color: gray;  
/* This is the font color */ }
```

# Creating an inline style

All elements have a global attribute called `style` that can be used to provide an *inline style*. *Because an inline style is defined on the element to which you wish to add styling, you don't need a selector; you just need to specify the declaration block. The following is an example of an inline style on the `<body>` element that sets the background color to white and the font color to gray.*

```
<body style='background-color: white; color: gray;*>  
</body>
```

# Creating an embedded style

*The <style> element is used to create an embedded style sheet within your HTML document. CSS selectors assign the style definitions to elements on the page. The following is an example of an embedded style sheet with a style rule that locates the <body> element and sets the background color to white and the font color to gray.*

```
<!DOCTYPE html>
```

```
<html xmlns='http://www.w3.org/1999/xhtml'>
```

```
<head> <title></title>
```

```
<style>
```

```
body { background-color: white; color: gray; } </style> </head>
```

```
<body>
```

```
</body>
```

```
</html>
```



# Creating an external style sheet

Instead of creating the same embedded styles in every HTML document, the best approach is to create an external style sheet file that you can link to all your pages. You can use the `<link>` element, as shown in the following example.

```
<!DOCTYPE html>
<html xmlns='http://www.w3.org/1999/xhtml'>
<head> <title></title>
  <link rel='stylesheet' type='text/css'
href='Content/default.css' />
</head>
<body> </body>
</html>
```

open the default.css file and add the following.

```
body { background-color: white; color: gray; }
```