



NOTES

Table in HTML



Introduction to the Table

A table is a structural data set of rows and columns. A table allows you to quickly and easily look up values that indicate some kind of connection between different types of data.

For example: Storing tabular data like a person's age, a day of the week, or the timetable for a local swimming pool.

Let's understand with a simple example.

Web dev team			
Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	
team of three developers 🧑🧑🧑			

Suppose, we want to display the above table in HTML. To construct tables in HTML, we use **<table>**, **<tr>**, **<th>**, and **<td>** tags.

Now let's understand all the table tags and create the above table step by step.

- **<table>**: The content of every table is enclosed by these two tags **<table></table>**
- **<td>**:

The smallest container in the table is a table cell, which we can create by using the **<td>** tag which denotes table data.

Example: Create multiple table data cells **<td>** and also add borders using table tag **border** attribute.

JavaScript

```
<table border="1">  
  <td> john </td>  
  <td> 25 </td>  
  <td> New York </td>  
</table>
```

Browser output:

john	25	New York
------	----	----------

cell cell cell

You can see that the data cells are not placed under each other, They are automatically aligned with each other on the same row. **<td>** create one single cell and together they create one-row. And the border attribute in HTML tables specifies the width of the border around table cells.

It's worth noting that the use of CSS's border property is generally preferred over the border attribute for greater control and flexibility in styling. We will be learning the css border property in the css section.

- **<tr>** : This element defines a row of cells in a table

JavaScript

```

<table border="1">
  <tr>
    <td>john</td>
    <td>25</td>
    <td>New York</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>30</td>
    <td>Los Angeles</td>
  </tr>
  <tr>
    <td>Michael</td>
    <td>22</td>
    <td>Chicago</td>
  </tr>
</table>

```

Browser output

john	25	New York
Jane	30	Los Angeles
Michael	22	Chicago

- **<th>**: This tag defines a header cell in an HTML table. The text in <th> elements are bold and centered by default.

JavaScript

```

<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>city</th>
  </tr>
  <tr>

```

```

...
</tr>
<tr>
...
</tr>
<tr>
...
</tr>
</table>

```

Browser output:

Name	Age	city
john	25	New York
Jane	30	Los Angeles
Michael	22	Chicago

To create the **designation** section and the **footer** section of the table, we have to use the “**rowspan**” and “**colspan**” attributes, respectively.

“rowspan”:

This **attribute** allows a cell to occupy and cover multiple rows, making it taller than a single row which means spanning a cell against several rows. To use **rowspan**, we need to include this attribute within **<td>** or **<th>** of the cell you want to span multiple rows. The value of **rowspan** specifies how many rows the cell should span.

Now let's add the designation header and fourth cell (“**td**”) of the second row (“**tr**”), which cover the last three rows,

```

JavaScript
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>city</th>
    <th>Designation</th>
  </tr>
  <tr>
    <td>john</td>
    <td>25</td>
    <td>New York</td>
    <td rowspan="3">developer</td>
  </tr>
  .....
</table>

```

Browser output

Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	

“colspan”: This allows a cell to span across multiple columns, making it wider than a single column which means spanning a cell against several columns. To use colspan, we need to include this attribute within <td> or <th> of the cell you want to span multiple columns. The value of colspan specifies how many columns the cell should span.




Let's create the footer for the table by adding one row (“tr”) at the bottom of a table whose cell (“td”) occupies four columns.

JavaScript

`</table border="1">`

....

`<tr>` `<td colspan="4">team of three developers 🧑🧑🧑</td>``</tr>``</table>`

Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	
team of three developers			  

Let's give some padding to each cell("td") by using cellpadding attribute

cellpadding":

creates a gap between content and cell borders, making the table look tidier and easier to read.

JavaScript

`<table border="1" cellpadding="10">` `<tr>` `<th>Name</th>` `<th>Age</th>` `<th>city</th>` `<th>Designation</th>`

```

    </tr>
    .....
</table>

```

The cellpadding attribute is set to 10, which means there will be 10 pixels of space between the content inside each cell and the cell borders. You can adjust the value to control how much space you want between the content and the cell borders.

Browser output:

Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	
team of three developers 🧑💻🧑💻🧑💻			

We can also add space between the table cells(“td”) by using Table attribute called **cellspacing**.

cellspacing:

This attribute specifies the amount of space we want between the cells of the table.

JavaScript

```

<table border="1" cellpadding="10" cellspacing="10">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>city</th>
    <th>Designation</th>
  </tr>

  .....

</table>

```

cellspacing attribute is set to 10, which means there will be a 10-pixel gap between all the cells of the table. You can adjust the value to control how much space you want between the cells.

Browser output

Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	
team of three developers 🧑💻🧑💻🧑💻			

- **<caption>:** <caption> tag in HTML is like a title for a table. It goes right after the <table> tag but before the data rows. It helps explain what the table is about, making things clear for people who see it.

Think of it as a label that makes your table organized and easy to understand.

JavaScript

```
<table border="1" cellpadding="10" cellspacing="10">
  <caption>Web dev team</caption>

  <tr>
    <td>john</td>
    <td>25</td>
    <td>New York</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>30</td>
    <td>Los Angeles</td>
  </tr>
  <tr>
    <td>Michael</td>
    <td>22</td>
    <td>Chicago</td>
  </tr>
</table>
```

Output:

Web dev team		
john	25	New York
Jane	30	Los Angeles
Michael	22	Chicago

<thead>, <tbody> and <tfoot>:

As you can see, we created the same table as shown in the introduction. Let's structure the content in the table using <thead>, <tbody> and <tfoot>. These elements are used to provide semantic meaning to different parts of the table.

JavaScript

```
<table border="1" cellpadding="10" cellspacing="10">
  <caption>Web dev team</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>city</th>
      <th>Designation</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>john</td>
      <td>25</td>
      <td>New York</td>
      <td rowspan="3">developer</td>
    </tr>
    <tr>
      <td>Jane</td>
      <td>30</td>
      <td>Los Angeles</td>
    </tr>
    <tr>
      <td>Michael</td>
      <td>22</td>
      <td>Chicago</td>
    </tr>
    <tr>
      <td colspan="4">team of three developers </td>
    </tr>
  </tbody>
</table>
```

Output:

Web dev team			
Name	Age	city	Designation
john	25	New York	developer
Jane	30	Los Angeles	
Michael	22	Chicago	
team of three developers			

THANK YOU

