

Defining Styles for Tables

In this lesson, we'll define our tables in style.

WE'LL COVER THE FOLLOWING



- The style explained
- **Listing 3-15:** Define alternating row colors for a table in CSS
- **Listing 3-16:** Using the `<colgroup>` and `<col>` tags



table **<style>** in **HTML5**



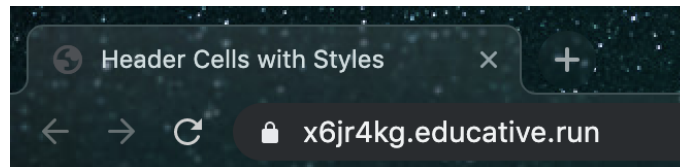
 **NOTE:** This course dedicates several chapters to styling with CSS, but here you will learn a few details in advance.

HTML markups prior to HTML5 defined dozens of attributes for table-related tags that were responsible for setting up the visual properties of tabular data.

For example, the `<table>` tag had attributes, such as `align`, `bgcolor` (background color), `cellpadding` (space between the cell wall and the cell content), `cellspacing` (space between cells), `summary`, and `width` to let designers style the visual appearance. Other tags had their own styling

attributes, too.

HTML5 got rid of these attributes and allows setting up table appearance only through styles. So, when you need to apply table styles, use CSS style sheets. Let's assume, you would like to add styles to the table defined in Listing 3-14 so that it is rendered as shown in the image below:



XOR		Input	
		false	true
Output	false	false	true
	true	true	false

Table with styles

One possible solution is the set of style rules in this code snippet:

style.css

```
<style>
body {
  font-family: Tahoma, Arial, sans-serif;
}

table {
  background-color: aliceblue;
  border: 4px solid dimgray;
  border-collapse: collapse;
}

td, th {
  padding: 4px 8px;
  text-align: center;
  border: 2px solid dimgray;
}

th {
  font-weight: normal;
  background-color: cornflowerblue;
  color: white;
}
```

```
td {
  color: dimgray;
  font-weight: bold;
}

.origin {
  background-color: navy;
  color: lightgoldenrodyellow
}
</style>
```

Without going into details about CSS, here are a few things to understand how styling works:

When a part of your HTML document is rendered, the browser applies the visual properties of the element being rendered.

By default, each element has a default style or a set of visual properties that can be modified by a style associated with that element, and with the style attribute. Certain visual properties are inherited by nested elements.

The style explained

The style sheet in the code snippet above does the following things:

1. The body rule (**lines 2-4**) sets the default font face of the entire page to *Tahoma*, or *Arial* provided *Tahoma* is not found, or to the default *sans-serif* font, if neither font is available. The `font-family` property is inherited in every nested element of `<body>`, so it will be the font of all elements, unless one of them overrides this setting.
2. The table rule (**lines 6-10**) sets up the background color of the table and applies a thick border with a solid line around it. The `border-collapse` property is set so that the table borders are collapsed into a single border.
3. There are two rules (**lines 12-16**) defined for table cells (`<td>`), the first is related to `<th>` tags as well, the second applies to `<td>` only. Both rules are applied to table cells, so each `<td>` cell has a light background and boldfaced font.
4. Similarly to `<td>`, there are two rules (**lines 18-22**) applied for `<th>` and as a result, header cells have a darker background with the light text color.
5. The field containing `"XOR"` is marked with the `class="origin"` attribute, so both the `td` (**line 24**) (this cell is a normal cell) and the `.origin` rule

(lines 29-32) is applied. So, this cell gains a dark background and a yellowish text color.

With CSS, you can easily define alternating row colors for a table, as shown in Listing 3-15.

 **NOTE:** You can find the completed sample in the Exercise-03-16 folder.

Listing 3-15: Define alternating row colors for a table in CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>Alternating Table Rows</title>
  <style>
    body {
      font-family: Verdana, Arial, sans-serif;
    }

    table {
      border-collapse: collapse;
    }

    td, th {
      padding: 4px 8px;
      border: 1px solid dimgray;
    }

    th {
      font-weight: normal;
      background-color: navy;
      color: white;
    }

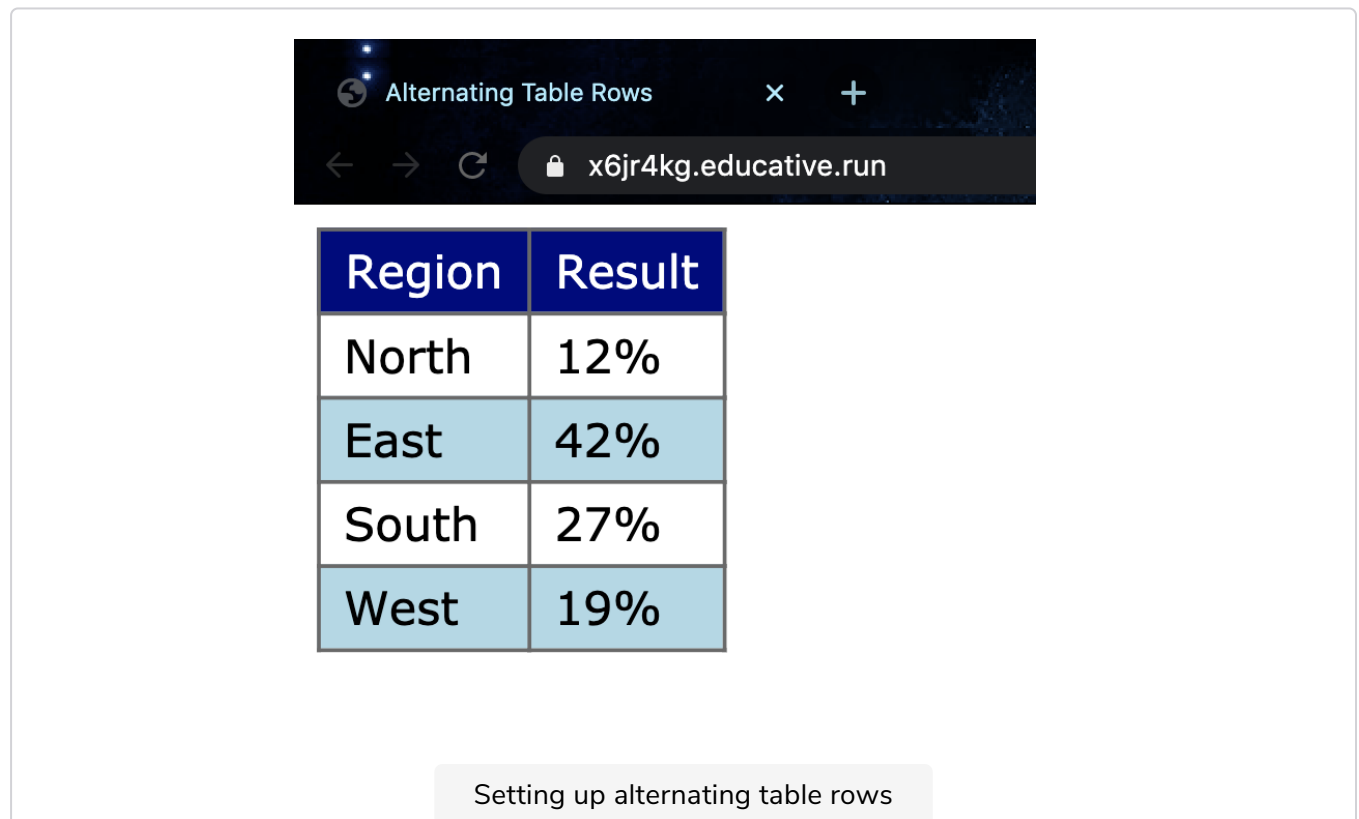
    tr:nth-child(odd) {
      background-color: lightblue;
    }

  </style>
</head>
<body>
  <table>
    <tr><th>Region</th><th>Result</th></tr>
    <tr><td>North</td><td>12%</td></tr>
    <tr><td>East</td><td>42%</td></tr>
    <tr><td>South</td><td>27%</td></tr>
    <tr><td>West</td><td>19%</td></tr>
```

```
</td><td>West</td><td>19%</td></tr>  
</table>  
</body>  
</html>
```

The `tr:nth-child(odd)` rule is applied for all table rows with an odd row index (1, 3, and 5 in this case). The first row has `<th>` elements, so the `th` rules override the style of cells set by `tr`.

The result of this markup is shown in the image below:



In HTML, the table structure is built up from rows (`<tr>`) nesting cells (`<td>` or `<th>`), so it's a bit more difficult to set up alternating columns.

One solution is to assign a different class attribute to each cell in alternating columns. But this adds a lot of extra markups and is pretty error-prone.

To cope with this situation, HTML defines two tags, `<colgroup>` and `<col>`, that can be used as shown in Listing 3-16:

Listing 3-16: Using the `<colgroup>` and `<col>` tags

```
<!DOCTYPE html>  
<html>  
<head>
```

```

<head>
<title>Alternating Table Columns</title>
<style>
  body {
    font-family: Verdana, Arial, sans-serif;
  }

  table {
    border-collapse: collapse;
  }

  td, th {
    padding: 4px 8px;
    border: 1px solid dimgray;
  }

  th {
    font-weight: normal;
    background-color: navy;
    color: white;
  }

  .result {
    background-color: lightblue;
  }

</style>
</head>
<body>
<table>
  <colgroup>
    <col span="2" />
    <col class="result" />
  </colgroup>
  <tr>
    <th>Region</th><th>Code</th>
    <th>Result</th>
  </tr>
  <tr>
    <td>North</td><td>N</td>
    <td>12%</td>
  </tr>
  <tr>
    <td>East</td><td>E</td>
    <td>42%</td>
  </tr>
  <tr>
    <td>South</td><td>S</td>
    <td>27%</td>
  </tr>
  <tr>
    <td>West</td><td>W</td>
    <td>19%</td>
  </tr>
</table>
</body>
</html>

```

This markup creates a table with three columns. The `<colgroup>` tag is a container that holds `<col>` tags. Each `<col>` defines the visual properties of a

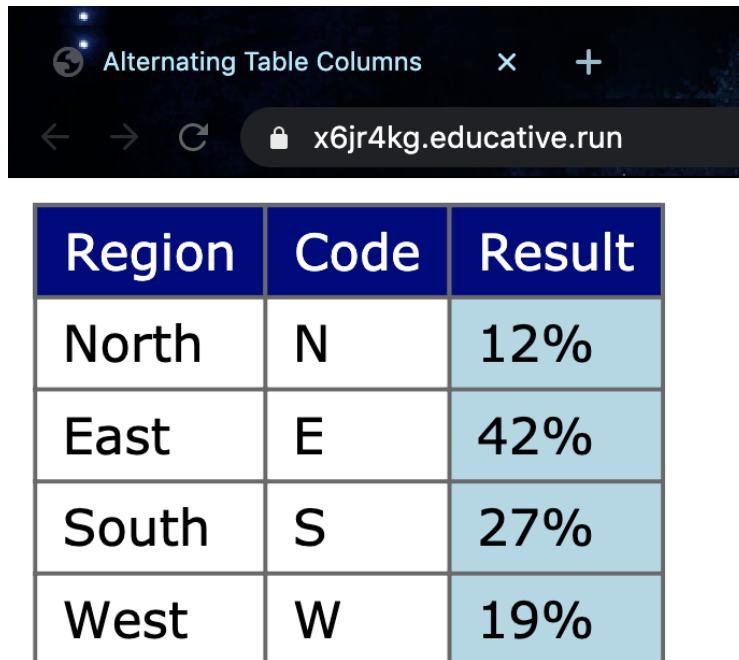
single column or a span of columns.

In Listing 3-16, the following definition is used:

```
index.html
<colgroup>
  <col span="2">
  <col class="result">
</colgroup>
```

The first `<col>` tag defines the style for the first two columns because the `span` attribute is set to 2. This element does not have any attributes, meaning that the first column should be displayed with their calculated style, without any deviation.

The second `<col>` is applied for the third column, and it says that the `.result` rule should be applied. This rule declares that the background color of this column should be light blue.



The screenshot shows a web browser window titled "Alternating Table Columns" with the URL "x6jr4kg.educative.run". The browser displays a table with the following data:

Region	Code	Result
North	N	12%
East	E	42%
South	S	27%
West	W	19%

Using `<colgroup>` and `<col>`

NOTE: The visual properties of `<col>` are applied only to the table cells, but not the cells' content. So, you cannot set text properties such as color, alignment, etc. through `<col>`.

In the *next lesson*, we'll get to know how to add captions to our tables.