Changing Attributes and Element Style

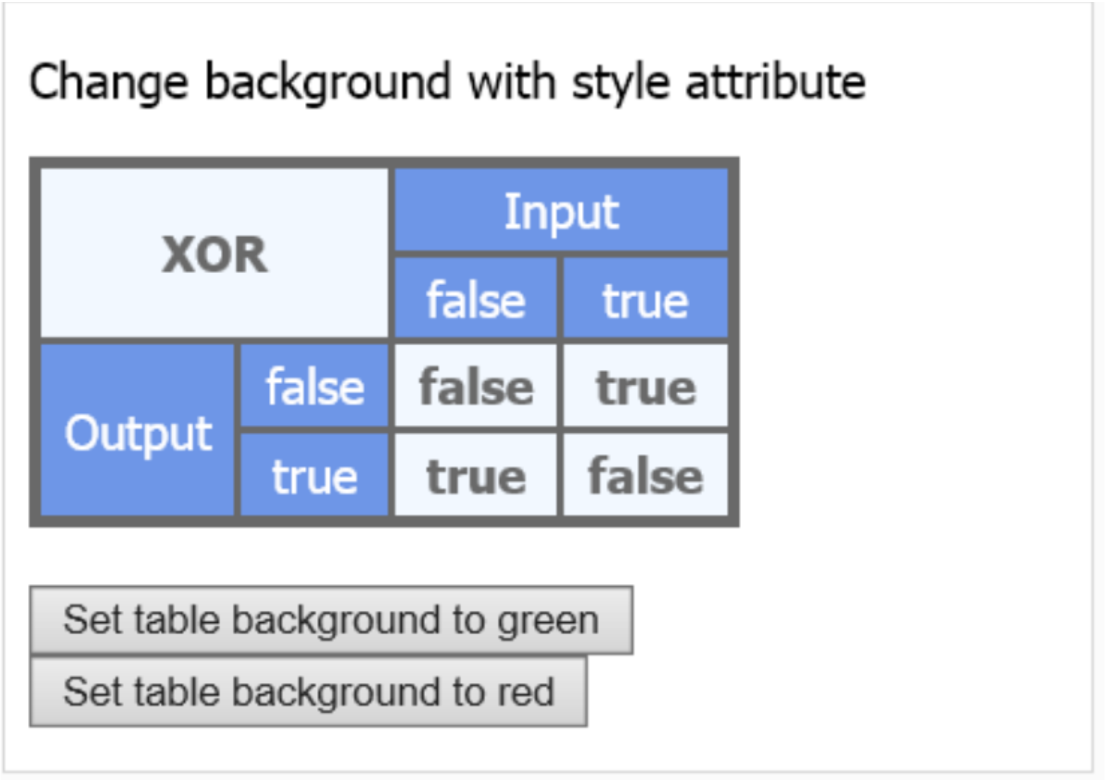
In this lesson, we change the style of attributes and elements.

WE'LL COVER THE FOLLOWING

- Listing 6-12: Exercise-06-13/index.html
- Listing 6-13: Using the DOM to change style

Modifying the document tree allows you to change not only elements, but also attributes (with the `setAttribute()` method). It provides a couple of ways to change the appearance of the web page on-the-fly.

In Chapter 4 (Listing 3-14), you displayed a table that defines the logical XOR operation, as shown in the image below:



Change background with style attribute

XOR		Input	
		false	true
Output	false	false	true
	true	true	false

The definition of logical XOR operation

Listing 6-12 adds two buttons to the markup that can be used to change the

Listing 6-12 adds two buttons to the markup that can be used to change the table background from its original color to green and red.

Listing 6-12: Exercise-06-13/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Change style with DOM</title>
  <style>
    body {
      font-family: Tahoma, Arial, sans-serif;
    }

    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;
    }

    table {
      background-color: aliceblue;
      border: 4px solid dimgray;
      border-collapse: collapse;
    }
  </style>
</head>
<body>
  <p>Change background with style attribute</p>
  <table id="table" border="1">
    <tr>
      <td rowspan="2" colspan="2">XOR</td>
      <th id="input" colspan="2">Input</th>
    </tr>
    <tr>
      <th headers="input" scope="col">false</th>
      <th headers="input" scope="col">true</th>
    </tr>
    <tr>
      <th id="output" rowspan="2">Output</th>
      <th headers="output" scope="row">false</th>
      <td>false</td>
      <td>true</td>
    </tr>
    <tr>
      <th headers="output" scope="row">true</th>
      <td>true</td>
      <td>false</td>
    </tr>
  </table>
  <br />
  <button onclick="toGreen()">
    Set table background to green
  </button>
  <br />
  <button onclick="toRed()">
    Set table background to red
  </button>
  <script>
    function toGreen() {
      var tableDef = document
        .getElementById('table');
      table.setAttribute('style',
        'background-color: lightgreen');
    }

    function toRed() {
      var tableDef = document
        .getElementById('table');
      table.style.backgroundColor = 'orangered';
    }
  </script>
</body>
</html>
```

The script in this code uses the `style` attribute to set the table background to green and red. The `toGreen()` method utilizes the `setAttribute()` method to set the style attribute's value to `“background-color: lightgreen”`.

In contrast, `toRed()` leverages a simpler way: it uses the `style` property of the object that represents the table node to access the `style` attribute, and uses its `backgroundColor` sub-property to change the table background.

If you have an inline style sheet, you can use the DOM to change a style. Listing 6-13 shows how you can carry out this task.

Listing 6-13: Using the DOM to change style

```
<!DOCTYPE html>
<html>
<head>
  <title>Change style with DOM</title>
  <style>
    body {
      font-family: Tahoma, Arial, sans-serif;
    }
    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;
    }

    table {
      border: 4px solid dimgray;
      border-collapse: collapse;
    }
  </style>
  <style id="tableStyle">
    table {
      background-color: aliceblue;
    }
  </style>
</head>
<body>
  <p>Change style with CSS manipulation</p>
  <table border="1">
    <tr>
      <td rowspan="2" colspan="2">XOR</td>
      <th id="input" colspan="2">Input</th>
    </tr>
    <tr>
      <th headers="input" scope="col">false</th>
      <th headers="input" scope="col">true</th>
    </tr>
    <tr>
      <th id="output" rowspan="2">Output</th>
      <th headers="output" scope="row">false</th>
      <td>false</td>
      <td>true</td>
    </tr>
    <tr>
      <th headers="output" scope="row">true</th>
      <td>true</td>
      <td>false</td>
    </tr>
  </table>
  <br />
  <button onclick="toGreen()">
    Set table background to green
  </button>
  <br />
  <button onclick="toRed()">
```

```
    Set table background to red
  </button>
  <script>
    function toGreen() {
      setBackgroundTo('lightgreen');
    }

    function toRed() {
      setBackgroundTo('orangered');
    }

    function setBackgroundTo(color) {
      var styleDef = document
        .getElementById('tableStyle');
      styleDef.innerHTML = "table { "
        + "background-color: "
        + color + "; }";
    }
  </script>
</body>
</html>
```

In this listing, `toGreen()` and `toRed()` delegate the task to the `setBackgroundTo()` method.

It navigates to the `<style`> node with `“tableStyle”` identifier and directly changes its HTML content. The inline style sheet contains two definitions for the table rule, this code sets the second one to use the background color passed to `setBackgroundTo()`. CSS merges the two table rules and applies this merged result to the table.

In the *next lesson*, we will start with global aspects of the document.

See you there!