Leveraging Page Inspector

In this lesson, we'll understand how to use a web browser's page inspector to analyze our HTML, CSS, and JavaScript code.
Let's begin!

You can learn a lot from looking into the **source code** of existing web pages.

Practically all web browsers support viewing that. However, when using JavaScript, the DOM of the page may *change at run time* and its source may not reflect the current structure of the page. Most browsers provide tools built directly into the browser to allow developers to examine the momentary state of the page structure. However, these tools are a bit different depending on your preferred browser.

In this exercise, you will learn to examine the DOM of the current page using the Developer Tools window of Chrome. Microsoft Edge and Firefox also have their own tools to discover the DOM.

EXERCISE 1-8: Discovering the changes of the document object model

To get acquainted with this great tool, follow these steps:

Step 1:

Display the page created in Exercise-01-07 provided below. If you’re using Chrome, you can go to step 3.

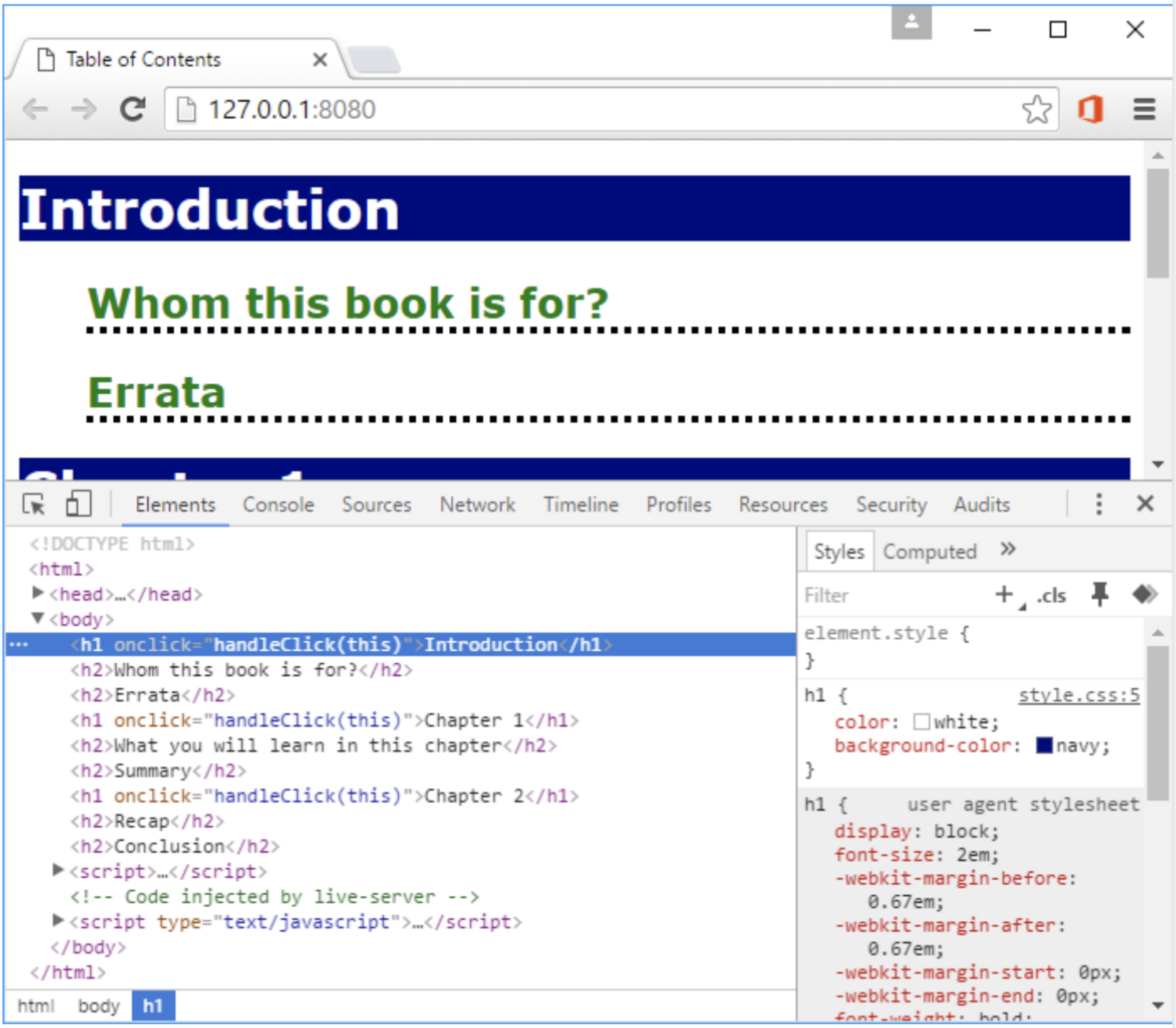
```
<!DOCTYPE html>
<html>
<head>
  <title>Table of Contents</title>
  <link href="style.css" rel="stylesheet" />
</head>
<body >
  <h1 onclick="handleClick(this)">Introduction</h1>
  <h2>Whom this book is for?</h2>
  <h2>Errata</h2>
  <h1 onclick="handleClick(this)">Chapter 1</h1>
  <h2>What you will learn in this chapter</h2>
  <h2>Summary</h2>
  <h1 onclick="handleClick(this)">Chapter 2</h1>
  <h2>Recap</h2>
  <h2>Conclusion</h2>

  <script>
    function handleClick(node) {
      var value = node.getAttribute('class') || '';
      value = value === '' ? 'clicked' : '';
      node.setAttribute('class', value);
    }
  </script>

</body>
</html>
```

Step 2:

Start Chrome, and click on the `“Your app can be found at: ”` link given below the live-coding widget to display the page.

Step 3:

To display the Developer Tools pane, press `F12` or `Ctrl+Shift+I`. When you select the Elements tab, the **DOM view** appears, displaying the structure of the **index.html** page. This view shows that the `<h1>` node does not have a class attribute as also shown below:



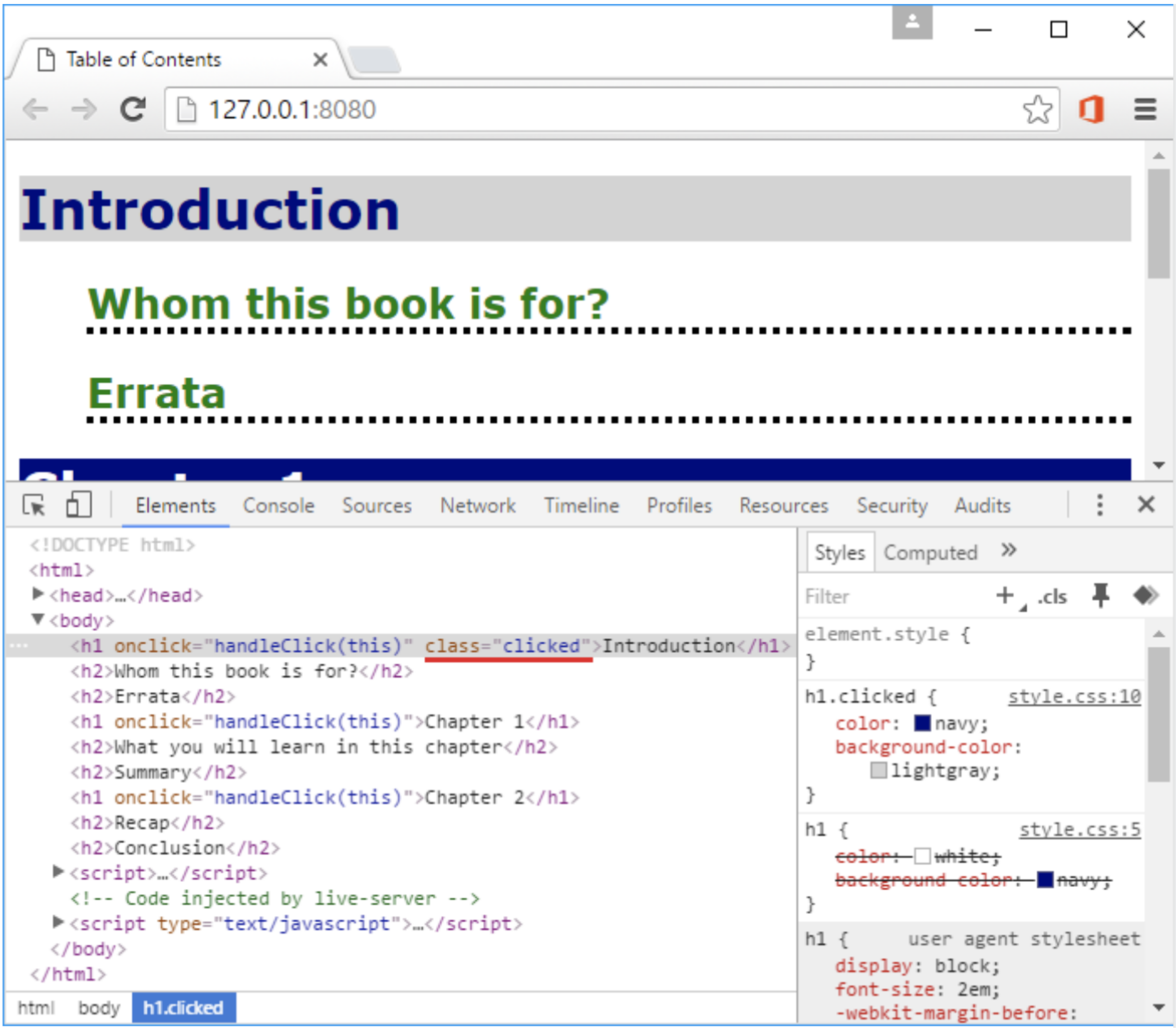
```
<!DOCTYPE html>
<html>
 ▶<head>…</head>
 ▼<body>
    <h1 onclick="handleClick(this)">Introduction</h1>
    <h2>Whom this book is for?</h2>
    <h2>Errata</h2>
    <h1 onclick="handleClick(this)">Chapter 1</h1>
    <h2>What you will learn in this chapter</h2>
    <h2>Summary</h2>
    <h1 onclick="handleClick(this)">Chapter 2</h1>
    <h2>Recap</h2>
    <h2>Conclusion</h2>
   ▶<script>…</script>
    <!-- Code injected by live-server -->
   ▶<script type="text/javascript">…</script>
  </body>
</html>
```

```
element.style {
}
h1 {                          style.css:5
  color: □white;
  background-color: ■navy;
}
h1 {          user agent stylesheet
  display: block;
  font-size: 2em;
  -webkit-margin-before:
     0.67em;
  -webkit-margin-after:
     0.67em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  font-weight: bold;
```

The DOM of the page

Step 4:

Now, in the page and not in the DOM pane, click the “Introduction” heading. This click action, as you learned in the previous exercise, adds a class attribute to the corresponding `<h1>` element with a value of “clicked”. With the help of the DOM view you can check that this change of structure really happens. The figure below shows that the attribute has been added to the element.

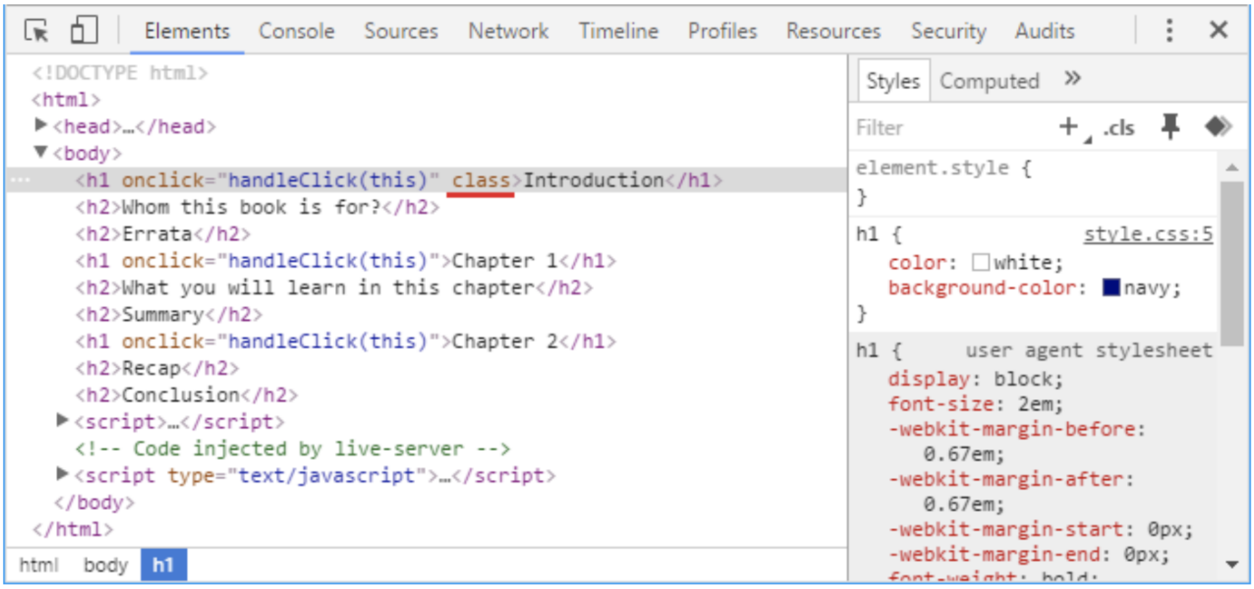


```
<!DOCTYPE html>
<html>
 ▶<head>…</head>
 ▼<body>
     <h1 onclick="handleClick(this)" class="clicked">Introduction</h1>
     <h2>Whom this book is for?</h2>
     <h2>Errata</h2>
     <h1 onclick="handleClick(this)">Chapter 1</h1>
     <h2>What you will learn in this chapter</h2>
     <h2>Summary</h2>
     <h1 onclick="handleClick(this)">Chapter 2</h1>
     <h2>Recap</h2>
     <h2>Conclusion</h2>
   ▶<script>…</script>
     <!-- Code injected by live-server -->
   ▶<script type="text/javascript">…</script>
   </body>
</html>
```

```
element.style {
}
h1.clicked {                    style.css:10
   color: ■navy;
   background-color:
       □lightgray;
}
h1 {                            style.css:5
   ~~color: □white;~~
   ~~background-color: ■navy;~~
}
h1 {          user agent stylesheet
   display: block;
   font-size: 2em;
   -webkit-margin-before:
```

The class attribute has been added to the <h1> element

Click the “Introduction” header again. Now you can check that the class attribute is toggled from clicked to empty, as shown in the figure below:



```
<!DOCTYPE html>
<html>
▶<head>…</head>
▼<body>
    <h1 onclick="handleClick(this)" class>Introduction</h1>
    <h2>Whom this book is for?</h2>
    <h2>Errata</h2>
    <h1 onclick="handleClick(this)">Chapter 1</h1>
    <h2>What you will learn in this chapter</h2>
    <h2>Summary</h2>
    <h1 onclick="handleClick(this)">Chapter 2</h1>
    <h2>Recap</h2>
    <h2>Conclusion</h2>
  ▶<script>…</script>
    <!-- Code injected by live-server -->
  ▶<script type="text/javascript">…</script>
  </body>
</html>
```

```
element.style {
}
h1 {                          style.css:5
   color: □white;
   background-color: ■navy;
}
h1 {       user agent stylesheet
   display: block;
   font-size: 2em;
   -webkit-margin-before:
      0.67em;
   -webkit-margin-after:
      0.67em;
   -webkit-margin-start: 0px;
   -webkit-margin-end: 0px;
```

The class attribute has been set to empty

In the *next lesson*, we’ll get acquainted with jQuery.