

Hands-On Projects

Before you begin working on the Hands-on Projects, be certain to extract the data files for this book, which you can download from Cengage's website at <http://www.cengage.com>, as explained in the preface to this book.

Hands-On Project 1-1

In this exercise you will use `document.write()` statements in a script section to add financial planning tips to a web page.

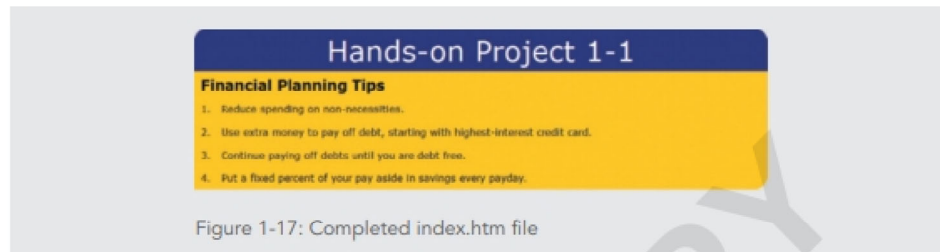
1. In your text editor, open **index.htm** from the HandsOnProject 1-1 folder in the Chapter01 folder. Enter your name and today's date where indicated in the comment section in the document head.
2. Just above the closing `</head>` tag, add the following element to link the file containing the Modernizr library:

```
<script src="modernizr.custom.05819.js"></script>
```

3. Within the article element, below the `h2` element, enter `<script>`, insert a blank line, and then enter `</script>` to create a new script section.
4. Within the script section you created in the previous step, enter the following comment and `document.write()` statements. These statements create an ordered list element containing financial planning tips.

```
1 // create ordered list
2 document.write("<ol>");
3 document.write("<li>Reduce spending on non-↵
4 necessities.</li>");
5 document.write("<li>Use extra money to pay off debt, starting↵
6 with highest-interest credit card.</li>");
7 document.write("<li>Continue paying off debts until you are↵
8 debt free.</li>");
9 document.write("<li>Put a fixed percent of your pay aside in↵
10 savings every payday.</li>");
11 document.write("</ol>");
```

5. Save your work, and then open **index.htm** in a web browser. A numbered list containing four items should be displayed below the `h2` heading "Financial Planning Tips," as shown in Figure 1-17.



6. Use the W3C Markup Validation Service to validate the **index.htm** document, and then, if necessary, fix any errors that the document contains.
7. Close your web browser window.

Hands-On Project 1-2

In this exercise, you will create a web page that uses variables to display information about high-speed Internet plans offered by an Internet service provider.

1. In your text editor, open **index.htm** from the HandsOnProject 1-2 folder in the Chapter01 folder. Enter your name and today's date where indicated in the comment section in the document head.
2. Just above the closing **</head>** tag, add the following element to link the file containing the Modernizr library:

```
<script src="modernizr.custom.05819.js"></script>
```

3. Within the **article** element, below the **h2** element, enter **<script>**, insert a blank line, and then enter **</script>** to create a new script section.
4. Within the script section you created in the previous step, enter the following comment and **var** statements. These statements declare variables containing the names and speeds of the high-speed Internet plans.

```
1 //initialize variables for service names and speeds
2 var service1Name = "Basic";
3 var service2Name = "Express";
4 var service3Name = "Extreme";
5 var service4Name = "Ultimate";
6 var service1Speed = 5;
7 var service2Speed = 25;
8 var service3Speed = 50;
9 var service4Speed = 150;
```

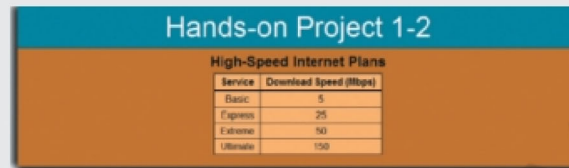
5. In the `table` element, within the `thead` section, locate the first row. Within the first `td` element, enter the following code to create a `script` element containing a `document.write()` statement to write the name of the first type of service:

```
<script>
    document.write(service1Name);
</script>
```

6. Repeat Step 5 for each of the remaining table cells, using the following content:

```
<script>
    document.write(service1Speed);
</script>
<script>
    document.write(service2Name);
</script>
<script>
    document.write(service2Speed);
</script>
<script>
    document.write(service3Name);
</script>
<script>
    document.write(service3Speed);
</script>
<script>
    document.write(service4Name);
</script>
<script>
    document.write(service4Speed);
</script>
```

7. Save your work, and then open **index.htm** in a web browser. A table containing two columns and five rows should be displayed below the `h2` heading "High-Speed Internet Plans," as shown in Figure 1-18.
8. Use the W3C Markup Validation Service to validate the **index.htm** document, and then, if necessary, fix any errors that the document contains.
9. Close your web browser window.



The screenshot shows a web browser window with a blue header bar that says "Hands-on Project 1-2". Below the header is a table titled "High-Speed Internet Plans". The table has two columns: "Service" and "Download Speed (Mbps)". The rows are: Basic (5), Express (25), Extreme (50), and Ultimate (150).

Service	Download Speed (Mbps)
Basic	5
Express	25
Extreme	50
Ultimate	150

Figure 1-18: Table contents written by JavaScript statements

Hands-On Project 1-3

In this exercise, you will enhance a form to display an alert box when a user clicks the Submit button.

1. In your text editor, open **index.htm** from the HandsOnProject1-3 folder in the Chapter01 folder. Enter your name and today's date where indicated in the comment section in the document head.
2. Just above the closing `</head>` tag, add the following element to link the file containing the Modernizr library:

```
<script src="modernizr.custom.05819.js"></script>
```

3. Locate the final fieldset element, with the id `submitButton`, just before the closing `</form>` tag. Within the input element that creates the Submit button, add the following event handler:

```
onclick="alert('Thanks for your order!');"
```
4. Save your work, open **index.htm** in a web browser, and then click the **Submit button**. As shown in Figure 1-19, an alert box should open, displaying the text "Thanks for your order!" Click the **OK** button to dismiss the alert box.

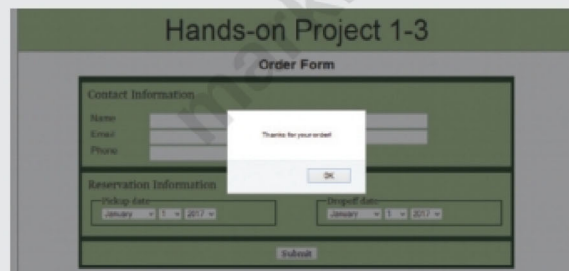


Figure 1-19: Alert box

5. Use the W3C Markup Validation Service to validate the **index.htm** document, and fix any errors that the document contains.
6. Close your web browser window.

Hands-On Project 1-4

In this chapter you learned how to dynamically change an image using the `getElementById()` method to set the value of the `src` attribute for an `img` element. You can apply this same process to many other web page elements. In this project, you'll use an event handler to automatically fill text boxes in a form with preassigned values based on the option button that a user clicks.

1. In your text editor, open **index.htm** from the HandsOnProject 1-4 folder in the Chapter01 folder. Enter your name and today's date where indicated in the comment section in the document head.
2. Just above the closing `</head>` tag, add the following element to link the file containing the Modernizr library:

```
<script src="modernizr.custom.05819.js"></script>
```

3. Open **index.htm** in your browser, and familiarize yourself with the layout of the form.
4. Return to your text editor, locate the first `fieldset` element, with the id `addroptions`, and then find the first `input` element, with the id `homeoption`. Within the `input` element that creates the first option button, add the following event handler:

```
1 onclick="document.getElementById('streetinput').value = ↵
2     '1 Main St.';
3     document.getElementById('cityinput').value = ↵
4     'Sicilia';
5     document.getElementById('stateinput').value = 'MA';
6     document.getElementById('zipinput').value = '02103';"
```

5. Within the same `fieldset` element, locate the second `input` element, with the id `workoption`. Within the `input` element that creates the second option button, add the following event handler:

```
1 onclick="document.getElementById('streetinput').value = '15↵
2     Columbine Ln.';
3     document.getElementById('cityinput').value = 'Crab↵
4     City';
5     document.getElementById('stateinput').value = 'MA';
6     document.getElementById('zipinput').value = '02104';"
```

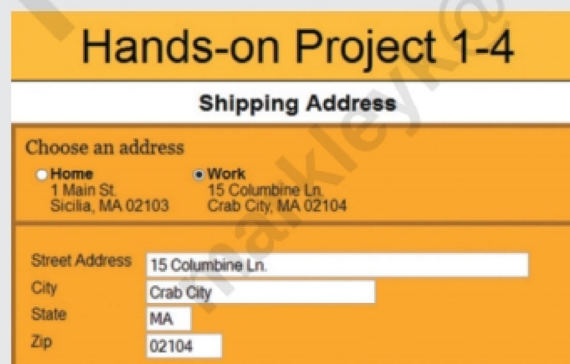
6. Save your work, open **index.htm** in a web browser, and then click the **Home** option button. The Street Address, City, State, and Zip fields should be filled with the information shown beneath the word "Home," as shown in Figure 1-20.



The screenshot shows a web form titled "Hands-on Project 1-4" with a sub-header "Shipping Address". Under the heading "Choose an address", there are two radio buttons: "Home" (selected) and "Work". Below the "Home" radio button, the address "1 Main St., Sicilia, MA 02103" is listed. Below the "Work" radio button, the address "15 Columbine Ln., Crab City, MA 02104" is listed. Below the address selection, there are four input fields: "Street Address" containing "1 Main St.", "City" containing "Sicilia", "State" containing "MA", and "Zip" containing "02103".

Figure 1-20: Fields populated with home address info

7. Click the **Work** option button. The contents of the Street Address, City, State, and Zip fields should be replaced with the information shown beneath the word "Work," as shown in Figure 1-21.



The screenshot shows the same web form as Figure 1-20, but with the "Work" radio button selected. The address "15 Columbine Ln., Crab City, MA 02104" is now listed under the "Work" radio button. The input fields below are updated: "Street Address" contains "15 Columbine Ln.", "City" contains "Crab City", "State" contains "MA", and "Zip" contains "02104".

Figure 1-21: Fields populated with work address info

8. Use the W3C Markup Validation Service to validate the **index.htm** document, and fix any errors that the document contains.
9. Close your web browser window.

Hands-On Project 1-5

In this project, you'll fix errors in JavaScript code to make a broken program work.

1. In your text editor, open **index.htm** from the HandsOnProject 1-5 folder in the Chapter01 folder. Enter your name and today's date where indicated in the comment section in the document head.
2. Just above the closing `</head>` tag, add the following code to link the file containing the Modernizr library:

```
<script src="modernizr.custom.65897.js"></script>
```

3. In the `article` element, examine the contents of the `script` element, which contains several errors.
4. Delete the characters `*/` from the third line of the script section, and then at the end of the first line, after the text "Add content to web page," type the `*/` characters. Now only the first line of the script section is a comment.
5. In the second line of the script section, change the word `Document` to all lowercase.
6. After the opening parenthesis, replace the apostrophe character (`'`) with a double quote (`"`) to match the closing character at the end of the line.
7. After the closing parenthesis, add a semicolon (`;`). Your corrected script section should match the following:

```
1 <script>
2 /* Add content to web page */
3 document.write("<p>We are closed for the holiday and will
4     reopen at midnight tonight.</p>");
5 </script>
```

8. Save your work, and then open **index.htm** in your browser. The `p` element is displayed on the web page, as shown in Figure 1-22.

Hands-on Project 1-5

We are closed for the holiday and will reopen at midnight tonight.

Figure 1-22: Text created by script section

9. Use the W3C Markup Validation Service to validate the **index.htm** document, and fix any errors that the document contains.
10. Close your web browser window.

Case Projects

Before you begin working on the Case Projects, be certain to extract the data files for this book, which you can download from Cengage's website at <http://www.cengage.com>, as explained in the preface to this book. Save any files you create for the Case Projects in the correct folder in your data files.

Individual Case Project

The Individual Case Project for each chapter in this book will build on a website that you create on a subject of your choice. To begin, choose a topic for your website. This can be a topic related to your major, or a personal interest or activity. Plan a website containing at least four pages with a common layout and navigation system. Note that you'll add pages to your site in later chapters, so ensure that your navigation system can support additional content. Add a `script` element that links to the `modernizr.custom.05819.js` file in each HTML document in your website. Ensure that all of your web pages pass validation.

Team Case Project

Throughout the Team Case Projects in this book you will continue to work on a website on a subject chosen by your team. Working in a team of 4–8 people, discuss and agree on a topic for your website. This may be a topic related to your major, another area of study, your college or university, or a shared interest. Work together to plan a website containing, at a minimum, a number of pages equal to the number of group members, and to create a common layout and navigation system. Note that you'll add pages to your site in later chapters, so ensure that your navigation system can support additional content. Decide as a group who will create which page, and create the pages individually. Add a `script` element that links to the `modernizr.custom.05819.js` file in each HTML document you create. When you've finished creating the individual pages, ensure they pass validation, and then work together to assemble the resulting website, identifying and fixing any issues as a group.