

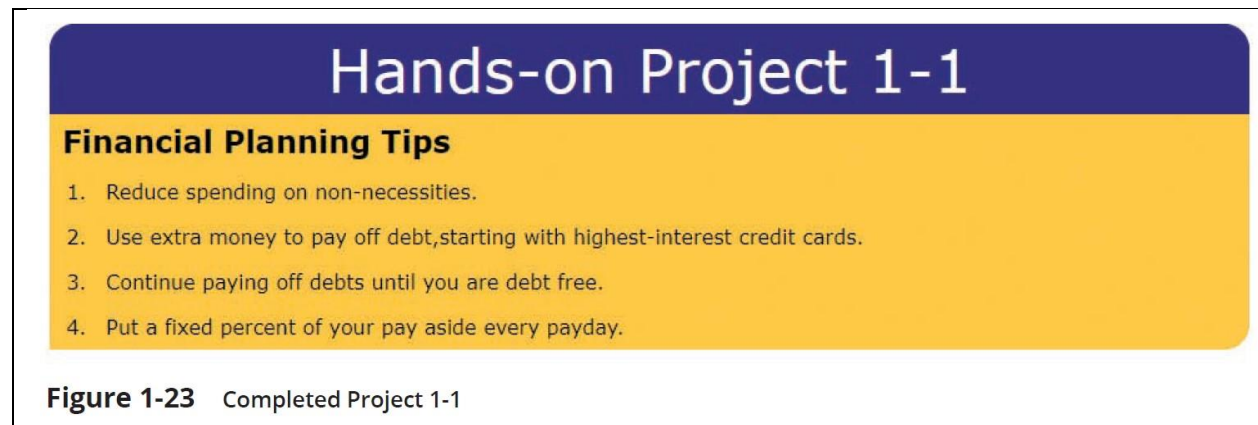
Hands-On Project 1-1

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

Do the following:

1. In your text editor, open **project01-01_txt.html** from the **HandsOnProject_01** folder. Enter your name and the date in the comment section of the document head.
2. Save the file as **project01-01.html**.
3. Within the article element, directly below the h2 element, enter the opening and closing tags of a script element on separate lines.
4. Within the script insert a JavaScript line comment containing the text **create ordered list**.
5. Below the line comment, insert multiple `document.write()` commands to write the following HTML code for an ordered list:

```
<ol>  
  <li>Reduce spending on non-necessities.</li>  
  <li>Use extra money to pay off debt,  
  starting with highest-interest credit cards.</li>  
  <li>Continue paying off debts until you are debt free.</li>  
  <li>Put a fixed percent of your pay aside every payday.</li>  
</ol>
```
6. Save your work and open **project01-01.html** in your web browser. Verify the content of the page resembles that shown in Figure 1-23. A numbered list containing four items should be displayed below the h2 heading "Financial Planning Tips," as shown in Figure 1-23.



Hands-On Project 1-2

In this project, you will create a web page that uses variables to display information about high-speed Internet plans offered by an Internet service provider. The completed page is shown in **Figure 1-24**.

Do the following:

1. In your text editor, open **project01-02_txt.html** from the **HandsOnProject_01** folder. Enter your name and the date in the comment section of each document and save them as **project01-02.html** and **project01-02.js**, respectively.
2. Below the comment section in the **project01-02.js** file, declare the following variables with indicated initial values:
service1Name 5 "Basic", service2Name 5 "Express", service3Name 5 "Extreme",

service4Name 5 "Ultimate", service1Speed 5 "0 Mbps", service2Speed 5 "100 Mbps", service3Speed 5 "500 Mbps", and service4Speed 5 "1 Gig".

3. Save your changes to the file.
4. Return to the **project01-02.html** file in your code editor. Directly above the closing `</head>` tag, insert a script element to load the `project01-02.js` source file. Do not add either the `async` or `defer` attributes to the script so that the code in the external file is loaded immediately as the web page is parsed by the browser.
5. Go to the first table row of the `tbody` section of the web table. Within the first `<td>` tag, insert a script to write the value of the `service1Name` variable. Within the second `<td>` tag, insert another script to write the value of the `service1Speed` variable.
6. Repeat Step 5 for the two cells in each of the next three table rows in the `tbody` section, writing the values of `service2Name` and `service2Speed` variables through the `service4Name` and `service4Speed` variables.
7. Save your work and then open **project01-02.html** in your web browser. Verify that the content of the page resembles that shown in Figure 1-24.

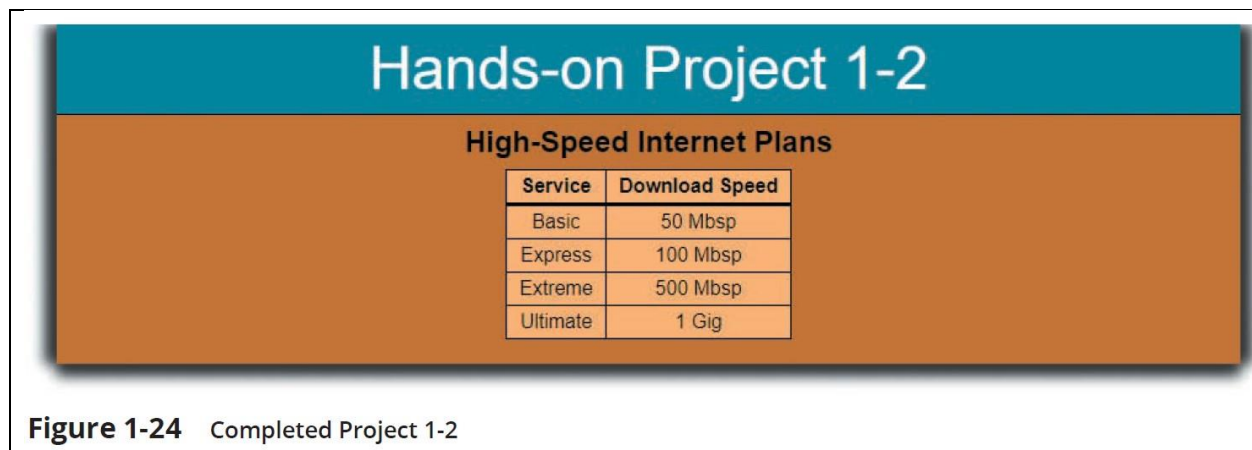


Figure 1-24 Completed Project 1-2

Hands-On Project 1-3

In this project, you will explore how to write text to a specific element in your web page in response to the `onclick` event handler. To complete the exercise, you will apply the following JavaScript expression:

```
document.getElementById('id').innerHTML = 'text';
```

where *id* is the value of the *id* attribute for the page element and *text* is the text of the content to be written into the element. You will use this expression to enhance a web form by displaying the message "Thank you for your order" when the user clicks the Submit button. **Figure 1-25** shows the completed web page.

Do the following:

1. Use your code editor to open **project01-03_txt.html** from the **HandsOnProject_01** folder. Enter your name and the date in the comment section of the document and save it as **project01-03.html**.
2. Scroll down to the bottom of the file and locate the `input` element for the Submit button.
3. Add an `onclick` event handler to the `<input>` tag that changes the `innerHTML` value of the page element with the id "submitMsg" to the text message **Thank you for your order**. (Note: The entire JavaScript expression should be enclosed within a set of double quotation marks, but the id and the text

message should be enclosed within single quotes.)

4. Save your changes to the file and then open **project01-03.html** in your web browser. Click the Submit button and verify that the text “Thank you for your order” appears on the bottom of the page.

The screenshot shows a web browser displaying a form titled "Hands-on Project 1-3" and "Order Form". The form is divided into two main sections: "Contact Information" and "Reservation Information". The "Contact Information" section has three input fields: "Name", "Email", and "Phone". The "Reservation Information" section has two date pickers: "Pickup date" and "Dropoff date". Each date picker consists of a month dropdown (showing "January"), a day dropdown (showing "1"), and a year dropdown (showing "2022"). Below these sections is a "Submit" button. At the bottom of the form, the text "Thank you for your order" is displayed in yellow.

Figure 1-25 Completed Project 1-3

Hands-On Project 1-4

You learned how to dynamically change an image using the `getElementById('id').src` expression along with the `onclick` event handler. In this project you will use the `onclick` event handler to automatically fill delivery address input boxes with preassigned values using the expression:

```
document.getElementById('id').value = variable;
```

where *id* is the value of the *id* attribute of a web form element and *variable* is the variable value to write into the element. A preview of the completed project is shown in **Figure 1-26**.

Do the following:

1. Use your code editor to open **project01-04_txt.html** and **project01-04_txt.js** from the **HandsOnProject_01** folder. Enter your name and the date in the comment section of each document and save them as **project01-04.html** and **project01-04.js**, respectively.
2. Go to the **project01-04.js** file in your code editor. Below the comment section declare the following variables and initial values: `homeStreet 5 "1 Main St."`, `homeCity 5 "Sicilia"`, `homeState 5 "MA"`, `homeCode 5 "02103"`, `workStreet 5 "15 Oak Ln."`, `workCity 5 "Central City"`, `workState 5 "MA"`, `workCode 5 "02104"`.
3. Close the file, saving your changes.
4. Go to the **project01-04.html** file in your code editor. Directly below the closing `</head>` tag insert a `script` element accessing the `project01-04.js` file. Do not include the `defer` or `async` attributes so that the code in the external file loads as the HTML is loaded.

5. Directly below the closing `</div>` tag for the Home address, insert a `script` element. Within the script, insert commands to write the following two lines of HTML code:

```
homeStreet <br>
homeCity, homeState homeCode
```

where `homeStreet`, `homeCity`, `homeState`, and `homeCode` are the variables you defined in Step 2. (Hint: You will have to use the add operator (`+`) to combine the variables with the literal text strings in these two lines of code.)

6. Directly below the closing `</div>` tag for the Work address, insert another script element. Within the script, insert commands to write the following two lines:

```
workStreet <br>
workCity, workState workCode
```

where `workStreet`, `workCity`, `workState`, and `workCode` are once again the variables you defined in Step 2.

7. Go to the `input` element with the id "homeoption". Within the `<input>` tag insert an `onclick` event handler that contains the following four JavaScript commands: (a) Set the value of the element with the id "street" to the value of the `homeStreet` variable; (b) Set the value of the element with the id "city" to the `homeCity` variable; (c) Set the value of the element with the id "state" to the `homeState` variable; (d) Set the value of the element with the id "code" to the value of the `homeCode` variable.
8. Go to the `input` element with the id "workoption". Repeat the previous step except store the values of the `workStreet`, `workCity`, `workState`, and `workCode` variables.
9. Save your changes to the file and then load **project01-04.html** in your web browser. Verify that the contents of the page resemble that shown in Figure 1-26 and that you can switch the address information at the bottom of the page between home and work by clicking the corresponding option buttons.

The screenshot shows a web form with an orange header bar containing the text "Hands-on Project 1-4". Below the header is a white bar with the text "Shipping Address". Underneath is an orange box with the text "Choose an address". Inside this box are two radio buttons: "Home" (unselected) and "Work" (selected). Below the radio buttons are two columns of text: "1 Main St. Sicilia, MA 02103" under "Home" and "15 Oak Ln. Central City, MA 02104" under "Work". Below the orange box is another orange box containing four input fields: "Street Address" (containing "15 Oak Ln."), "City" (containing "Central City"), "State" (containing "MA"), and "Zip" (containing "02104").

Figure 1-26 Completed Project 1-4

Hands-On Project 1-5

Learning to locate and fix errors is an important skill for anyone programming in JavaScript. You have been given a web page containing several errors that need to be fixed. When fixed, the page will display the content shown in **Figure 1-27**.



Figure 1-27 Completed Project 1-5

Do the following:

1. Use your code editor to open **project01-05_txt.html** from the **HandsOnProject_01** folder. Enter your name and the date in the comment section of the document and save it as **project01-05.html**.
2. In the head section of the document there is a script that declares and initializes the `reopenDate` variable. There are two errors in this code. Fix both errors.
3. Scroll down to the script embedded within the `article` element. The code contains a total of four errors. Locate and fix the errors.
4. Save your changes to the file and then open the file in your web browser. Verify that the page resembles that shown in Figure 1-27.

NOTE

1. Use the [W3C Markup Validation Service](#) to validate the `index.html` document, and then, if necessary, fix any errors that the document contains.
2. Make sure you organize the content of the project in different folders: images, styles, and scripts folders.