- The Device panel enables you to run the application by using simulations of different screen sizes and orientations.
- The Live DOM panel enables you to select an element and apply style rules to it.

Lesson review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. You would like to create a Windows 8 application by using Blend and HTML5, JavaScript, and CSS3. Which feature of Blend enables you to pause an application when it reaches a desired state so you can set the style rules for the page and its controls while in this state? (Choose all that apply.)

   A. Assets panel

   B. Projects panel

   C. Visual Design

   D. Interactive mode

2. On which panel can you see a hierarchically structured view of the DOM?

   A. Live DOM

   B. Projects

   C. Assets

   D. Device

3. Which panel can you use to access a list of the HTML elements, controls, and media that can be added to an HTML page that is open in the artboard?

   A. Projects

   B. Assets

   C. Device

   D. Live DOM

Practice exercises

If you encounter a problem completing any of these exercises, the completed projects can be installed from the Practice Exercises folder that is provided with the companion content.

Exercise 1: Hello World with Visual Studio Express 2012 for Windows 8

In this exercise, you create a simple HTML5 and JavaScript Hello World application by using Visual Studio Express 2012 for Windows 8. This practice, like all Hello World applications, is intended to get you started by creating a minimal application. In later exercises, you get more exposure to Visual Studio. In this exercise, you use HTML5 to display "Hello Visual Studio for Windows 8" on the screen.

1. If you haven't installed Visual Studio Express 2012 for Windows 8, do so now. You can download this from the Microsoft website.
2. Start Visual Studio. Click File and choose New Project to display the New Project dialog box. Navigate to Installed | Templates | JavaScript | Windows Store. Select Blank App.
3. Set the name of your application to **HelloVisualStudioForWin8**.
4. Select a location for your application.
5. Set the solution name to **HelloVisualStudioForWin8Solution**.
6. Be sure to keep the Create Directory For Solution check box selected.
7. Click OK to create your new project.
8. When the application is created, the default.js file is displayed with some template code.

   This code is covered later, and there is no need to alter it now.
9. Open the default.html file.

   It contains HTML from the template.
10. To see the default content, press F5 to start debugging the application.

    You should see a black screen and, in the upper-left corner, a message stating, "Content goes here."
11. Switch back to Visual Studio by pressing Alt+Tab.

    The title bar says (Running).
12. Press Shift+F5 to stop debugging.

    Shift+F5 works only when Visual Studio has the focus; it does not work when the running application has the focus.
13. In the default.html file, replace the "Content goes here" message with **Hello Visual Studio for Windows 8**.
14. Replace the reference to ui-dark.css with ui-light.css.

    Your HTML should look like the following.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>HelloWorldForWin8</title>
    <!-- WinJS references -->
    <link href="//Microsoft.WinJS.1.0/css/ui-light.css" rel="stylesheet" />
    <script src="//Microsoft.WinJS.1.0/js/base.js"></script>
    <script src="//Microsoft.WinJS.1.0/js/ui.js"></script>

    <!-- HelloWorld references -->
    <link href="/css/default.css" rel="stylesheet" />
    <script src="/js/default.js"></script>
</head>
<body>
    <p>Hello Visual Studio for Windows 8</p>
</body>
</html>
```

15. Press F5 to start debugging.

    The screen is white because you now reference the ui-light.css file instead of the ui-dark.css file. The screen also displays Hello Visual Studio For Windows 8. Congratulations—you have written your first Windows 8 application by using HTML5 technologies!

Exercise 2: Hello World with Visual Studio Express 2012 for Web

In this exercise, you create a simple HTML5 and JavaScript Hello World application by using Visual Studio Express 2012 for Web. This practice, like all Hello World applications, is intended to get you started by creating a minimal application. In later exercises, you get more exposure to Visual Studio. In this exercise, you create a new project in Visual Studio Express 2012 for Web and use HTML5 to display "Hello Visual Studio for Web" on the screen.

> ***NOTE*** **NO SERVER CODE IN THIS EXERCISE**
>
> **You will not be writing any server code in this exercise, so it doesn't matter whether you select Visual Basic or Visual C# when starting the new project.**

1. If you haven't installed Visual Studio Express 2012 for Web, do so now. You can download this from the Microsoft website.
2. Start Visual Studio. Click File and choose New Project to display the New Project dialog box. Navigate to Installed | Templates | Visual Basic | Web. Select the ASP.NET Web Form Application.
3. Set the name of your application to **HelloVisualStudioForWeb**.
4. Select a location for your application.
5. Set the solution name to **HelloVisualStudioForWeb Solution**.
6. Be sure to keep the Create Directory For Solution check box selected.
7. Click OK to create your new project.

8. When the application is created, the default.aspx page will be displayed with some template code.
9. In the Solution Explorer window, build the project by right-clicking the project node and choosing Build.
10. To see this template's default content, press F5 to start debugging the application.

    You should see a fancy screen with information on how to get started plus other useful information.
11. Switch back to Visual Studio by pressing Alt+Tab.

    The title bar says (Running).
12. Press Shift+F5 to stop debugging.

    Note that Shift+F5 works only when Visual Studio has the focus. Shift+F5 does not work when the running application has the focus.
13. Delete the default.aspx file by right-clicking this file in the Solution Explorer window, choosing Delete, and then clicking OK.
14. In the Solution Explorer window, add a default.html file by right-clicking the project node. Click Add and then choose HTML. Name the page **default.html**.
15. In the default.html file, place the text **Hello Visual Studio for Web** between the *<body>* and *</body>* tags.
16. In the default.html file, place the text **HelloVisualStudioForWeb** between the *<title>* and *</title>* tags.

    Your HTML should look like the following.

    ```
    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <title>HelloVisualStudioForWeb</title>
    </head>
    <body>
       Hello Visual Studio for Web
    </body>
    </html>
    </html>
    ```
17. In the Solution Explorer window, set the default.html file as the startup file by right-clicking the default.html file and choosing Set As Start Page.
18. Press F5 to start debugging.

    The screen now displays Hello Visual Studio For Web. Congratulations—you have written your first web application using HTML5 technologies!

Exercise 3: Hello World with Blend

In this exercise, you create a simple HTML5 and JavaScript Hello World application by using Blend. This practice, like all Hello World applications, is intended to get you started by creating a minimal application. In later exercises, you get more exposure to Blend. In this exercise, you create a new project in Blend and use HTML5 to display "Hello World" on the screen.

1. If you haven't installed Blend, do so now. Remember that Blend is installed automatically when you install Visual Studio Express 2012 for Windows 8. You can download this from the Microsoft website.
2. To start Blend, click New Project to display the New Project dialog box. Select the HTML (Windows Store) category in the left pane and select Blank App in the right pane.
3. Set the name of your application to **HelloBlend**.
4. Select a location for your application.
5. Click OK to create your new project.

   When the application is created, the default.html file is displayed.
6. To see the default content, press F5 to start debugging the application.

   You should see a black screen and, in the upper-left corner, a message stating, "Content goes here."
7. Close the running application by pressing Alt+F4.
8. If Blend is not displayed, return to Blend by pressing Alt+Tab.
9. In the default.html file, double-click the "Content Goes Here" message and replace the text with **Hello from Blend**.

   You see the change in the default.html source view window at the bottom of the screen.
10. Replace the reference to ui-dark.css with ui-light.css.

    Your HTML should look like the following.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>HelloBlend</title>

    <!-- WinJS references -->
    <link href="//Microsoft.WinJS.1.0/css/ui-light.css" rel="stylesheet" />
    <script src="//Microsoft.WinJS.1.0/js/base.js"></script>
    <script src="//Microsoft.WinJS.1.0/js/ui.js"></script>

    <!-- HelloBlend references -->
    <link href="/css/default.css" rel="stylesheet" />
    <script src="/js/default.js"></script>
</head>
<body>
```

```
    <p>Hello from Blend</p>
</body>
</html>
```

11. Press F5 to start the application.

    Notice that the screen is white because you now reference the ui-light.css file instead of the ui-dark.css file. The screen now displays Hello From Blend. Congratulations—you have written a Windows 8 application by using HTML5 technologies with Blend!

Suggested practice exercises

The following additional exercises are designed to give you more opportunities to practice what you've learned and to help you successfully master the lessons presented in this chapter.

- **Exercise 1** Learn more about Visual Studio Express 2012 for Web by creating new projects from each of the included project templates. After creating each project, try adding Hello World and run the application to see how the project looks and behaves.
- **Exercise 2** Learn more about Visual Studio Express 2012 for Windows 8 by creating new projects from each of the included project templates. After creating each project, try adding Hello World and run the application to see how the project looks and behaves.
- **Exercise 3** Learn more about Blend by creating new projects from each of the included project templates. After creating each project, try adding Hello World and run the application to see how the project looks and behaves.

Lesson review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. You want to embed a Flash file called myFlash.swf in your HTML document. Which is the most appropriate code?

   A. *<a href*="myFlash.swf" ></a>

   B. *<iframe src*="myFlash.swf" *></iframe>*

   C. *<img src*="myFlash.swf" />

   [✓ D.] *<embed src*="myFlash.swf" *></embed>*

2. You want to create a drawing of a machine that Contoso, Ltd., will be selling on its website. The drawing will be embedded in your HTML document, and you want it to maintain its quality when resized. Which is the most appropriate file type to use?

   A. SVG

   [✓ B.] GIF

   C. JPG

   D. PNG

Practice exercises

You've learned a bit about HTML elements and attributes, and it's time to create a website. In Exercise 1, you create a website for a fictitious company, Contoso, Ltd., and add a home page. In Exercise 2, you add the expense reports, human resources, and main content pages.

If you encounter a problem completing an exercise, the completed projects can be installed from the Practice Exercises folder that is provided with the companion content.

Exercise 1: Create a simple website by using Visual Studio Express for Web

In this practice, you create a simple website by using Visual Studio Express 2012 for Web. The quality of the webpages produced will be less than desirable because CSS hasn't been discussed yet. The goal of this practice is to use many of the tags that have been described in this lesson.

You start by creating an ASP.NET website by using Visual Studio Express 2012 for Web, and then you add to and modify the home page.

1. If you haven't installed Visual Studio Express 2012 for Web, do so now. You can download this from the Microsoft website.

2. Start Visual Studio Express 2012 for Web. Navigate to file and choose New Project. Navigate to Installed | Templates | Visual Basic | Web and select ASP.NET Empty Web Application.
3. Set the name of your application to **ContosoWebSite**.
4. Select a location for your application.
5. Set the solution name to **ContosoWebSiteSolution**.
6. Be sure to keep the Create Directory For Solution check box selected.
7. Click OK to create your new project.
8. When the application is created, click Debug and select Start Debugging. (The shortcut key is usually F5 but can vary based on your installation settings.)

   The ASP.NET Empty Web Application doesn't include a home page, so an error page is displayed, showing an HTTP Error 403.14 - Forbidden error. The error page indicates (in a rather indirect way) that you don't have a default page on your website, so the web server tries to display a list of all files in the directory. However, the security settings on the website will not permit directory browsing to display the directory contents.
9. Close the error page, which should automatically stop debugging. If you need to, you can stop debugging by clicking Debug and choosing Stop Debugging (or pressing Shift+F5).
10. Add a home page. In the Solution Explorer window, right-click the ContosoWebSite project, choose Add, and select HTML Page. Set the name to **default.html** and click OK.

    The home page is added to your website and contains the following HTML.

    ```
    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title></title>
    </head>
    <body>

    </body>
    </html>
    ```
11. In the *<title>* element, set the title to **Contoso Ltd. Home Page**.
12. In the Solution Explorer window, right-click the project, choose Add, select New Folder, and name the folder **Images**.
13. In the Solution Explorer window, right-click the Images folder that you just added, choose Add, select Existing Item, and select the ContosoLogo.jpg file that is located in the Chapter02 Resources folder.
14. In the *<body>* element, add a comment and set the text to **Add <img> with Contoso logo**.

    The <img> element is covered in Lesson 2.

15. Using the *<h1>* element to create a heading, add the **Welcome to Contoso Ltd.** heading to the body after the comment.

16. After the *<h1>* element, add a comment. Set the comment text to **Add <iframe> here**.

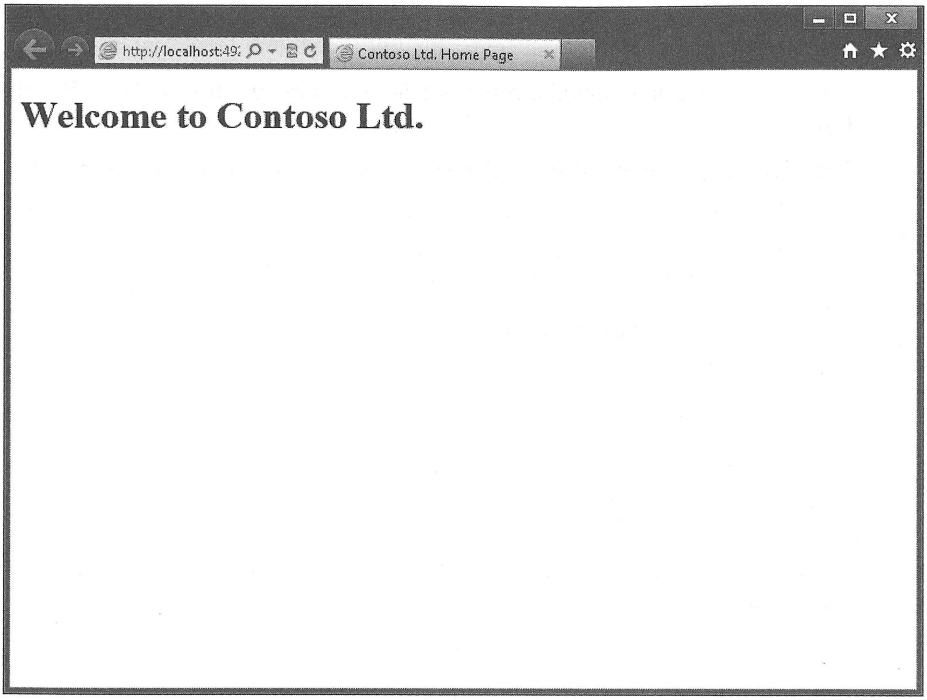
    The *<iframe>* element is discussed in Lesson 2. Your default.html page should look like the following.

    ```
    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Contoso Ltd. Home Page</title>
    </head>
    <body>
        <!--Add <img> with Contoso logo-->
        <h1>Welcome to Contoso Ltd.</h1>
        <!--Add <iframe> here-->
    </body>
    </html>
    ```

17. In the Solution Explorer window, right-click the default.html page and choose Set As Start Page.

18. Click Debug and choose Start Debugging (F5).

    You should see the rendered screen, as shown in Figure 2-1. Using the *<h1>* element produced a heading with a large font. Notice that comments are not displayed, but if you right-click the page and choose View Source, you see the HTML source, which has the comments. The text in the browser tab contains Contoso Ltd. Home Page, which is the text that you entered in the <title> element.



http://localhost:49:

Contoso Ltd. Home Page

Welcome to Contoso Ltd.

FIGURE 2-1 The Contoso Ltd. website

Exercise 2: Create additional pages

In this exercise, you add pages to the website you created in Exercise 1. If you didn't perform Exercise 1, you can use the files in the Exercise2\LabStart folder to start. You won't be able to get to the new pages from the home page yet because you add that functionality in Exercise 3.

1. In the Solution Explorer window, right-click the project, choose Add, and select HTML Page. Set the name to **MainContent.html** and click OK.

   The content will be kept simple.

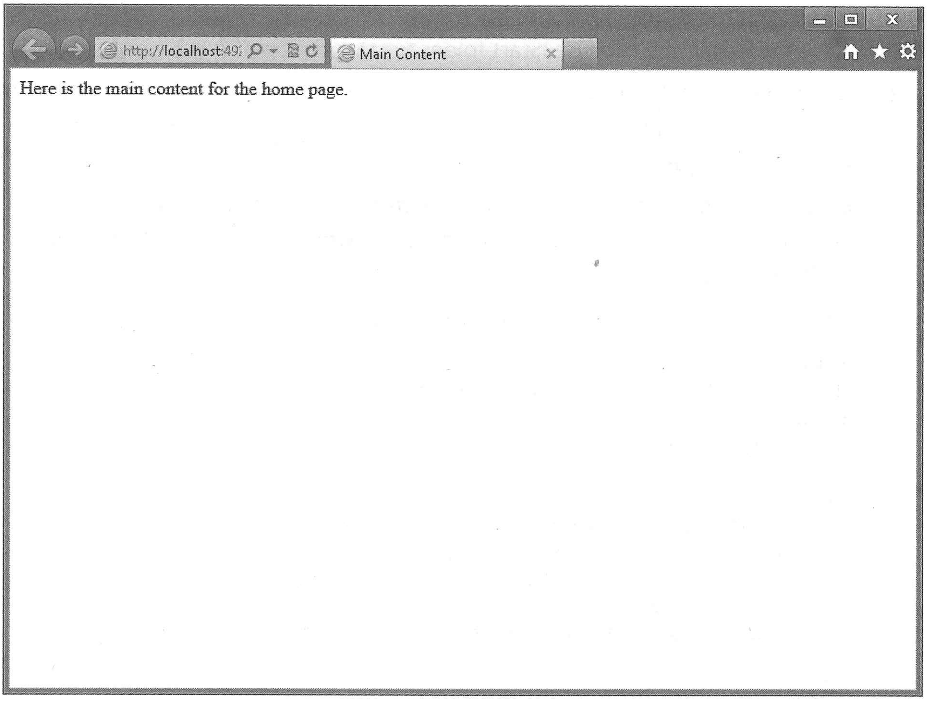
2. In the *<body>* element of the MainContent.html page, add the following text: **Here is the main content for the home page**. Set the title to **Main Content**.

   Your HTML should look like the following.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Main Content</title>
</head>
<body>
    Here is the main content for the home page.
</body>
</html>
```

3. To view this page, right-click the middle of the source and choose View In Browser (Internet Explorer).

   You should see the rendered screen, as shown in Figure 2-2. Note that almost nothing is on this page. It doesn't have a heading or logo. In Exercise 3 you combine this content with the home page to obtain the header and logo.



Here is the main content for the home page.

FIGURE 2-2 The rendered main content page

4. Close the browser window.
5. Add a new HTML file and name it **HumanResources.html**. In the *<body>* element, add the following text: **Human Resource content here**.
6. Right-click the middle of the source and choose View In Browser (Internet Explorer).

   You should see the rendered screen with the text you entered.
7. Add another HTML file and name it **ExpenseReports.html**. In the *<body>* element, add the following text: **Expense Report content here**.
8. Right-click the middle of the source and choose View In Browser (Internet Explorer).

   You should see the rendered screen with the text you entered.

Exercise 3: Embedding Content

You learned about embedding content in Lesson 2; this exercise uses this information to connect the pages of the Contoso, Ltd., website by embedding the pages in the home page, using an inline frame.

In this practice, you extend the Contoso, Ltd., website that you created in Exercise 1 by linking the pages and displaying pages in an inline frame. If you didn't perform Exercise 1, you can use the files in the Exercise3Start folder to start.

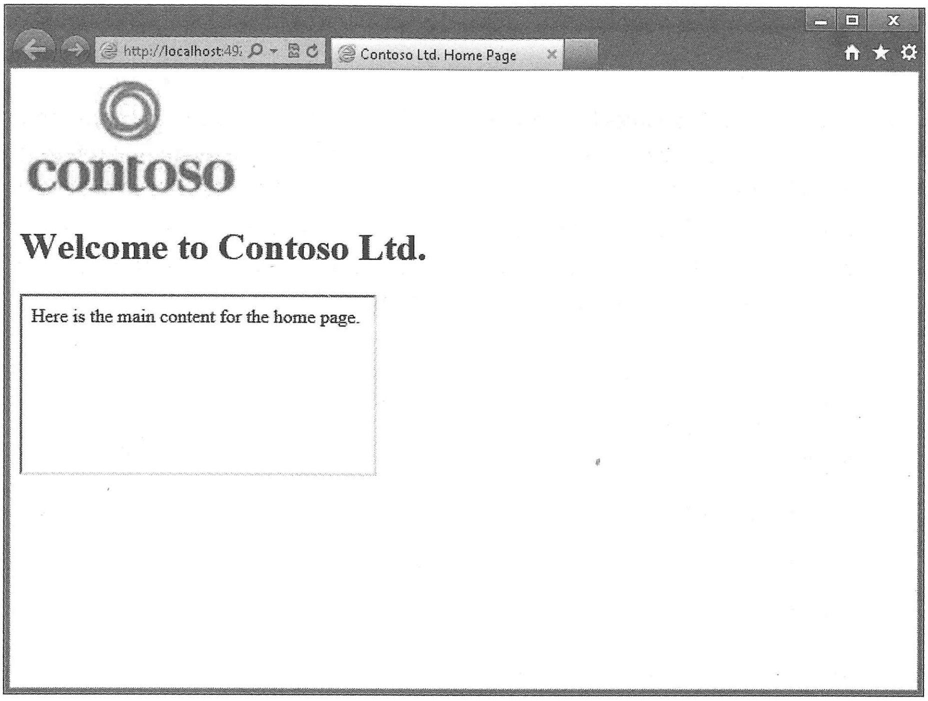
1. Open the default.html file and locate the comment that states that an *<img>* tag is to be added. Add an *<img>* tag after the comment. Set the src attribute to **Images/ContosoLogo.jpg**.
2. Locate the comment that states that an *<iframe>* tag is to be added. Add an *<iframe>* element with a name attribute set to **iframeContent** and an src attribute set to **MainContent.html**.

   Your default.html page should look like the following.

   ```
   <!DOCTYPE html>
   <html xmlns="http://www.w3.org/1999/xhtml">
   <head>
       <title>Contoso Ltd. Home Page</title>
   </head>
   <body>
       <!--Add <img> with Contoso logo-->
       <img src="Images/ContosoLogo.jpg" />
       <h1>Welcome to Contoso Ltd.</h1>
       <!--Add <iframe> here-->
       <iframe name="iframeContent" src="MainContent.html"></iframe>
   </body>
   </html>
   ```

3. Click Debug and choose Start Debugging (F5).

   You should see the Contoso logo, and the MainContent.html file is now included on the page. The rendered screen is shown in Figure 2-3.



Welcome to Contoso Ltd.

Here is the main content for the home page.

FIGURE 2-3 The rendered home page with the MainContent.html page embedded

4. Stop debugging.
5. On the default.html page, after the *<h1>* element, add a hyperlink for Human Resources that references the HumanResources.html file and targets the *<iframe>* element called iframeContent.
6. After the Human Resources hyperlink, add a hyperlink for Expense Reports that references the ExpenseReports.html file and targets the *<iframe>* element called iframeContent.
7. Add a *<br />* element after the Human Resources hyperlink.

   The completed default.html file should look like the following.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Contoso Ltd. Home Page</title>
</head>
<body>
    <!--Add <img> with Contoso logo-->
    <img src="Images/ContosoLogo.jpg" />
    <h1>Welcome to Contoso Ltd.</h1>
    <a href="HumanResources.html" target="iframeContent">Human Resources</a>
    <a href="ExpenseReports.html" target="iframeContent">Expense Reports</a>
    <br />
    <!--Add <iframe> here-->
    <iframe name="iframeContent" src="MainContent.html"></iframe>
</body>
</html>
```

8. In the HumanResources.html and ExpenseReports.html files, add a <br /> tag to the end of the body text, and then add a hyperlink to the home page that references MainContent.html.

   Note that you don't need to target the current frame because the default behavior of the hyperlink is to replace the current page with the new page. The following is an example of the completed ExpenseReports.html file.

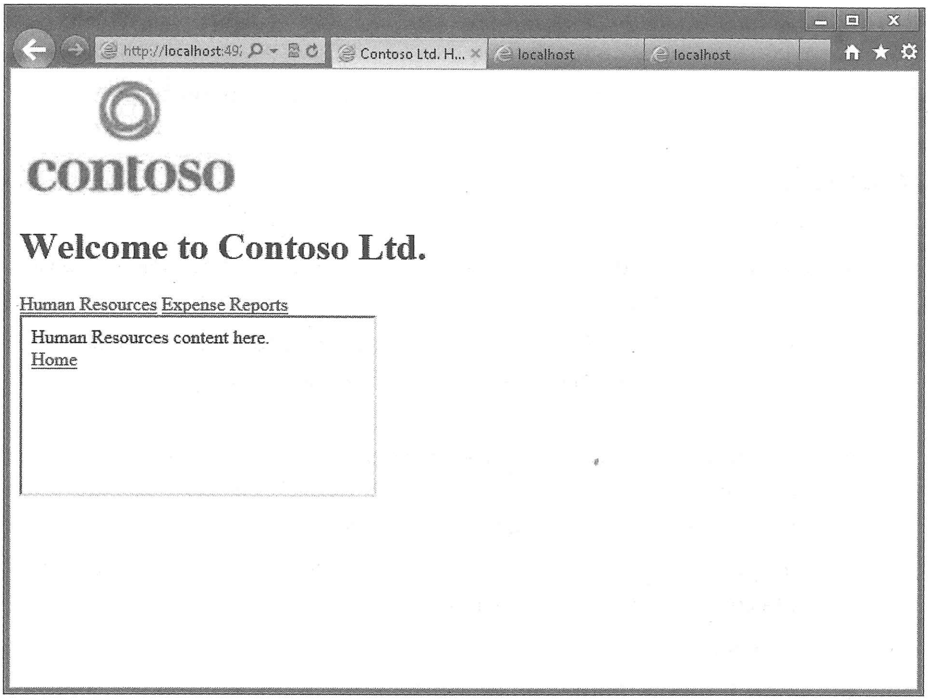
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    Expense Reports content here.
    <br />
    <a href="MainContent.html" >Home</a>
</body>
</html>
```

9. Click Debug and choose Start Debugging (F5).

   You should see the new home page with hyperlinks. If you click the Human Resources hyperlink, you should see the HumanResources.html content load into the *<iframe>* element, as shown in Figure 2-4. In the *<iframe>* content, you should see a hyperlink to the home page that you can click to reload the MainContent.html file into the *<iframe>* element.

NOTE **CLICK REFRESH TO SEE THE HOME HYPERLINK**

You might not see the Home hyperlink because the original page might be cached. Right-click in the *<iframe>* and click Refresh.



Welcome to Contoso Ltd.

Human Resources Expense Reports

Human Resources content here.
Home

FIGURE 2-4 The default.html page shown after clicking the Human Resources hyperlink

Suggested practice exercises

The following additional practices are designed to give you more opportunities to practice what you've learned and to help you successfully master the lessons presented in this chapter.

- **Exercise 1** Learn more about each of the HTML tags that are part of the HTML5 specification by using each in an HTML document.
- **Exercise 2** Learn more about embedding content by adding images and image maps to an HTML document.