

Microsoft®

OFFICIAL MICROSOFT LEARNING PRODUCT

20480B

Programming in HTML5 with JavaScript and
CSS3

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2012 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at

<http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Product Number: 20480B

Part Number (if applicable):

Released: xx/20xx

MICROSOFT LICENSE TERMS

OFFICIAL MICROSOFT LEARNING PRODUCTS

MICROSOFT OFFICIAL COURSE Pre-Release and Final Release Versions

These license terms are an agreement between Microsoft Corporation and you. Please read them. They apply to the Licensed Content named above, which includes the media on which you received it, if any. These license terms also apply to any updates, supplements, internet based services and support services for the Licensed Content, unless other terms accompany those items. If so, those terms apply.

BY DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below.

1. DEFINITIONS.

- a. “Authorized Learning Center” means a Microsoft Learning Competency Member, Microsoft IT Academy Program Member, or such other entity as Microsoft may designate from time to time.
- b. “Authorized Training Session” means the Microsoft-authorized instructor-led training class using only MOC Courses that are conducted by a MCT at or through an Authorized Learning Center.
- c. “Classroom Device” means one (1) dedicated, secure computer that you own or control that meets or exceeds the hardware level specified for the particular MOC Course located at your training facilities or primary business location.
- d. “End User” means an individual who is (i) duly enrolled for an Authorized Training Session or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.
- e. “Licensed Content” means the MOC Course and any other content accompanying this agreement. Licensed Content may include (i) Trainer Content, (ii) software, and (iii) associated media.
- f. “Microsoft Certified Trainer” or “MCT” means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program, and (iii) holds a Microsoft Certification in the technology that is the subject of the training session.
- g. “Microsoft IT Academy Member” means a current, active member of the Microsoft IT Academy Program.
- h. “Microsoft Learning Competency Member” means a Microsoft Partner Network Program Member in good standing that currently holds the Learning Competency status.
- i. “Microsoft Official Course” or “MOC Course” means the Official Microsoft Learning Product instructor-led courseware that educates IT professionals or developers on Microsoft technologies.

- j. "Microsoft Partner Network Member" or "MPN Member" means a silver or gold-level Microsoft Partner Network program member in good standing.
 - k. "Personal Device" means one (1) device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular MOC Course.
 - l. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
 - m. "Trainer Content" means the trainer version of the MOC Course and additional content designated solely for trainers to use to teach a training session using a MOC Course. Trainer Content may include Microsoft PowerPoint presentations, instructor notes, lab setup guide, demonstration guides, beta feedback form and trainer preparation guide for the MOC Course. To clarify, Trainer Content does not include virtual hard disks or virtual machines.
2. **INSTALLATION AND USE RIGHTS.** The Licensed Content is licensed not sold. The Licensed Content is licensed on a one copy per user basis, such that you must acquire a license for each individual that accesses or uses the Licensed Content.
- 2.1 Below are four separate sets of installation and use rights. Only one set of rights apply to you.
- a. **If you are a Authorized Learning Center:**
- i. If the Licensed Content is in digital format for each license you acquire you may either:
 - 1. install one (1) copy of the Licensed Content in the form provided to you on a dedicated, secure server located on your premises where the Authorized Training Session is held for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session, **or**
 - 2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session.
 - ii. You agree that:
 - 1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
 - 2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
 - 3. for all Authorized Training Sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
 - 4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,

5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of the Authorized Training Session,
 6. you will only provide access to the Licensed Content to End Users and MCTs,
 7. you will only provide access to the Trainer Content to MCTs, and
 8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.
- b. **If you are a MPN Member.**
- i. If the Licensed Content is in digital format for each license you acquire you may either:
 1. install one (1) copy of the Licensed Content in the form provided to you on (A) one (1) Classroom Device, or (B) one (1) dedicated, secure server located at your premises where the training session is held for use by one (1) of your employees attending a training session provided by you, or by one (1) MCT that is teaching the training session, **or**
 2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for use by one (1) End User attending a Private Training Session, or one (1) MCT that is teaching the Private Training Session.
 - ii. You agree that:
 1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
 2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
 3. for all training sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
 4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,
 5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of each training session,
 6. you will only provide access to the Licensed Content to End Users and MCTs,
 7. you will only provide access to the Trainer Content to MCTs, and
 8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.

c. **If you are an End User:**

You may use the Licensed Content solely for your personal training use. If the Licensed Content is in digital format, for each license you acquire you may (i) install one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install another copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content; or (ii) print one (1) copy of the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.

d. **If you are a MCT.**

- i. For each license you acquire, you may use the Licensed Content solely to prepare and deliver an Authorized Training Session or Private Training Session. For each license you acquire, you may install and use one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.
- ii. **Use of Instructional Components in Trainer Content.** You may customize, in accordance with the most recent version of the MCT Agreement, those portions of the Trainer Content that are logically associated with instruction of a training session. If you elect to exercise the foregoing rights, you agree: (a) that any of these customizations will only be used for providing a training session, (b) any customizations will comply with the terms and conditions for Modified Training Sessions and Supplemental Materials in the most recent version of the MCT agreement and with this agreement. For clarity, any use of “*customize*” refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

2.2 **Separation of Components.** The Licensed Content components are licensed as a single unit and you may not separate the components and install them on different devices.

2.3 **Reproduction/Redistribution Licensed Content.** Except as expressly provided in the applicable installation and use rights above, you may not reproduce or distribute the Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.

2.4 **Third Party Programs.** The Licensed Content may contain third party programs or services. These license terms will apply to your use of those third party programs or services, unless other terms accompany those programs and services.

2.5 **Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to that respective component and supplements the terms described in this Agreement.

3. **PRE-RELEASE VERSIONS.** If the Licensed Content is a pre-release (“**beta**”) version, in addition to the other provisions in this agreement, then these terms also apply:

- a. **Pre-Release Licensed Content.** This Licensed Content is a pre-release version. It may not contain the same information and/or work the way a final version of the Licensed Content will. We may change it for the final version. We also may not release a final version. Microsoft is under no obligation to provide you with any further content, including the final release version of the Licensed Content.
- b. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software, technologies, or products to third parties because we include your feedback in them. These rights

survive this agreement.

- c. **Term.** If you are an Authorized Training Center, MCT or MPN, you agree to cease using all copies of the beta version of the Licensed Content upon (i) the date which Microsoft informs you is the end date for using the beta version, or (ii) sixty (60) days after the commercial release of the Licensed Content, whichever is earliest (“**beta term**”). Upon expiration or termination of the beta term, you will irretrievably delete and destroy all copies of same in the possession or under your control.
4. **INTERNET-BASED SERVICES.** Microsoft may provide Internet-based services with the Licensed Content, which may change or be canceled at any time.
 - a. **Consent for Internet-Based Services.** The Licensed Content may connect to computer systems over an Internet-based wireless network. In some cases, you will not receive a separate notice when they connect. Using the Licensed Content operates as your consent to the transmission of standard device information (including but not limited to technical information about your device, system and application software, and peripherals) for internet-based services.
 - b. **Misuse of Internet-based Services.** You may not use any Internet-based service in any way that could harm it or impair anyone else's use of it. You may not use the service to try to gain unauthorized access to any service, data, account or network by any means.
5. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
 - install more copies of the Licensed Content on devices than the number of licenses you acquired;
 - allow more individuals to access the Licensed Content than the number of licenses you acquired;
 - publicly display, or make the Licensed Content available for others to access or use;
 - install, sell, publish, transmit, encumber, pledge, lend, copy, adapt, link to, post, rent, lease or lend, make available or distribute the Licensed Content to any third party, except as expressly permitted by this Agreement.
 - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation;
 - access or use any Licensed Content for which you are not providing a training session to End Users using the Licensed Content;
 - access or use any Licensed Content that you have not been authorized by Microsoft to access and use; or
 - transfer the Licensed Content, in whole or in part, or assign this agreement to any third party.
6. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content. You may not remove or obscure any copyright, trademark or patent notices that appear on the Licensed Content or any components thereof, as delivered to you.

7. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, End Users and end use. For additional information, see www.microsoft.com/exporting.
8. **LIMITATIONS ON SALE, RENTAL, ETC. AND CERTAIN ASSIGNMENTS.** You may not sell, rent, lease, lend or sublicense the Licensed Content or any portion thereof, or transfer or assign this agreement.
9. **SUPPORT SERVICES.** Because the Licensed Content is "as is", we may not provide support services for it.
10. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon any termination of this agreement, you agree to immediately stop all use of and to irretrievable delete and destroy all copies of the Licensed Content in your possession or under your control.
11. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
12. **ENTIRE AGREEMENT.** This agreement, and the terms for supplements, updates and support services are the entire agreement for the Licensed Content.
13. **APPLICABLE LAW.**
 - a. **United States.** If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
 - b. **Outside the United States.** If you acquired the Licensed Content in any other country, the laws of that country apply.
14. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
15. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS," "WITH ALL FAULTS," AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES GIVE NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS UNDER OR IN RELATION TO THE LICENSED CONTENT. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES EXCLUDE ANY IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**

16. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. TO THE EXTENT NOT PROHIBITED BY LAW, YOU CAN RECOVER FROM MICROSOFT CORPORATION AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO USD\$5.00. YOU AGREE NOT TO SEEK TO RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES FROM MICROSOFT CORPORATION AND ITS RESPECTIVE SUPPLIERS.

This limitation applies to

- anything related to the Licensed Content, services made available through the Licensed Content, or content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence , aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Welcome!

Thank you for taking our training! We've worked together with our Microsoft Certified Partners for Learning Solutions and our Microsoft IT Academies to bring you a world-class learning experience—whether you're a professional looking to advance your skills or a student preparing for a career in IT.

- **Microsoft Certified Trainers and Instructors**—Your instructor is a technical and instructional expert who meets ongoing certification requirements. And, if instructors are delivering training at one of our Certified Partners for Learning Solutions, they are also evaluated throughout the year by students and by Microsoft.
- **Certification Exam Benefits**—After training, consider taking a Microsoft Certification exam. Microsoft Certifications validate your skills on Microsoft technologies and can help differentiate you when finding a job or boosting your career. In fact, independent research by IDC concluded that 75% of managers believe certifications are important to team performance¹. Ask your instructor about Microsoft Certification exam promotions and discounts that may be available to you.
- **Customer Satisfaction Guarantee**—Our Certified Partners for Learning Solutions offer a satisfaction guarantee and we hold them accountable for it. At the end of class, please complete an evaluation of today's experience. We value your feedback!

We wish you a great learning experience and ongoing success in your career!

Sincerely,

Microsoft Learning
www.microsoft.com/learning



¹ IDC, Value of Certification: Team Certification and Organizational Performance, November 2006

Module 1

Overview of HTML and CSS

Contents:

| | |
|------------------------------------------------------------------|---|
| Lesson 1: Overview of HTML | 2 |
| Lesson 3: Creating a Web Application by Using Visual Studio 2012 | 6 |
| Module Review and Takeaways | 8 |

Lesson 1

Overview of HTML

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Creating a Simple Contact Form

Demonstration Steps

Create an HTML Page

1. On the Windows 8 **Start** screen, right-click outside of any tile, and in the task bar click **All apps**.
2. On the Windows 8 **Start** screen, in the **Windows Accessories** section, click **Notepad**.
3. Add the following basic HTML structure to the blank text file.

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Contact Us</title>
  </head>
  <body>
  </body>
</html>
```

Add Content to the Page

1. Add a level 1 page heading to the **<body>** element.

```
<h1>Contact Contoso Conferencing</h1>
```

2. Add the company address by using two paragraphs and the line break element as follows:

```
<p>Contoso Conferencing Ltd.</p>
<p>123 South Street<br />
Somewhere<br />
Over There<br />
<em>USA</em></p>
```

3. Add a hyperlink for the company's contact email address. Note the use of the **mailto:** protocol rather than the more common **http://**.

```
<p>
<a href="mailto:contact@contoso.com">
  contact@contoso.com</a>
</p>
```

4. Add the following text to invite users to use the contact form.

```
<p>
If you would like to contact Contoso Conferencing, whether you're interested in our
services or in a conference we're currently organising, don't hesitate to contact us
using our enquiry form (<strong>Bold fields</strong> are required).
</p>
```

Add a Form with Input Controls

1. Add the following **<form>** element to the page beneath the text created previously.

```
<form method="Post" action="support.aspx">
</form>
```

2. Add the following **<fieldset>** element and submit button to the form (between the **<form ...>** and **</form>** tags).

```
<fieldset>
  <legend>
    Your Details and Enquiry
  </legend>
</fieldset>
<input type="submit" value="Send" />
```

3. Add the unordered list shown in the following code sample to the **<fieldset>** element below the **</legend>** tag. This list contains input elements for the user's name, telephone number, email address, and a message.

```
<ol>
  <li>
    <label>
      <strong>Name</strong><br />
      <input type="text"
             name="UserName" />
    </label>
  </li>
  <li>
    <label>
      Telephone<br />
      <input type="text"
             name="Phone" />
    </label>
  </li>
  <li>
    <label>
      Email Address<br />
      <input type="text"
             name="Email" />
    </label>
  </li>
  <li>
    <label>
      <strong>Message</strong><br />
      <textarea name="Message"
                cols="30" rows="10">Add your message here
      </textarea>
    </label>
  </li>
</ol>
```

4. Save the file and close Notepad.

View the Page

1. In the Windows taskbar, click the File Explorer icon.
2. Browse to the folder **E:\Mod01\Democode**.
3. Double-click **ContactUs.html** to display the page in Internet Explorer.
4. In the **How do you want to open this type of file (.html)?** dialog box, click **Internet Explorer**.

The page should look like this:

A screenshot of an Internet Explorer window displaying a contact form. The title bar reads "E:\Mod01\Democode>ContactUs. Contact Us". The main content is titled "Contact Contoso Conferencing". It contains the company address: "Contoso Conferencing Ltd.", "123 South Street", "Somewhere", "Over There", "USA", and an email link "contact@contoso.com". A message encourages users to contact them via their enquiry form, noting that bold fields are required. Below this is a section titled "Your Details and Enquiry" with four fields: "Name" (text input), "Telephone" (text input), "Email Address" (text input), and "Message" (text area with placeholder "Add your message here"). A "Send" button is at the bottom.

Figure 1.1:The ContactUs page

5. Enter some sample data, but do not click **Send**.

 **Note:** If you do click Send, Internet Explorer will display the message **This page can't be displayed**. This occurs because the URL that is specified as the action attribute for the form (support.aspx) does not exist.

Lesson 3

Creating a Web Application by Using Visual Studio 2012

Contents:

| | |
|---------------|---|
| Demonstration | 7 |
|---------------|---|

Demonstration

Demonstration: Creating a Web Site by Using Visual Studio 2012

Demonstration Steps

Create a Web Site Project

1. On the Windows 8 **Start** screen, click the **Visual Studio 2012** tile.
2. In Visual Studio, on the **File** menu, point to **New**, and then click **Web Site**.
3. In the **New Web Site** dialog box, click **ASP.NET Empty Web Site**.

 **Note:** It does not matter whether you select the **Visual Basic** or **Visual C#** templates in the left pane; the templates for both languages enable you to create HTML5 web pages and implement functionality by using JavaScript.

4. From the web location drop-down list, click **File System**, set the file path to **E:\Mod01\Democode\DemoWebSite**, and then click **OK**.
5. On the **File** menu, click **Save DemoWebSite**.

Module Review and Takeaways

Question: What are the four elements that define the basic structure of an HTML page?

Answer: The DOCTYPE declaration, and the <html>, <head>, and <body> elements.

Question: What is the best way to apply CSS rules to HTML elements that occur in several different pages?

- () Include all rules for each element in the <style> attribute of the element.
- () Include the rules for each page in a <style> element in the <head> element.
- () Write the rules for the whole site in one or more style sheets and reference them by using a <style> element in the <head> element of each page.
- () Write the rules for the whole site in one or more style sheets and reference them by using a <link> element in the <head> element of each page.
- () Write the rules for the whole site in one or more style sheets and reference them by using a <stylesheet> element in the <head> element of each page.

Answer:

- () Include all rules for each element in the <style> attribute of the element.
- () Include the rules for each page in a <style> element in the <head> element.
- () Write the rules for the whole site in one or more style sheets and reference them by using a <style> element in the <head> element of each page.
- (✓) Write the rules for the whole site in one or more style sheets and reference them by using a <link> element in the <head> element of each page.
- () Write the rules for the whole site in one or more style sheets and reference them by using a <stylesheet> element in the <head> element of each page.

Module 2

Creating and Styling HTML Pages

Contents:

| | |
|-----------------------------------------|---|
| Lesson 1: Creating an HTML5 Page | 2 |
| Lesson 2: Styling an HTML5 Page | 6 |
| Module Review and Takeaways | 9 |

Lesson 1

Creating an HTML5 Page

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Using HTML5 Features in a Simple Contact Form

Demonstration Steps

Divide the Content for a Page into an Article with Sections

1. On the Windows 8 **Start** screen, click the **Visual Studio 2012** tile.
2. In Visual Studio, on the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, browse to the **E:\Mod02\Democode\Starter** folder, click **DemoWebSite.sln**, and then click **Open**.
4. In Solution Explorer, expand the **E:\...\DemoWebSite** web application, and then double-click **ContactUs.html**.
5. In the **ContactUs.html** file, enclose the entire contents of the **<body>** element in an **<article>** element as shown in bold in the following code example:

```
<!DOCTYPE HTML>
<html lang="en">
...
</head>
<body>
<article>
...
</article>
</body>
</html>
```

6. Within the **<article>** element, enclose the first three **<p>** elements containing the company name, address, and contact email in a **<section>** element, as shown in bold in the following code example:

```
...
<h1>Contact Contoso Conferencing</h1>
<section>
  <p>Contoso Conferencing Ltd.</p>
  <p>123 South Street<br />
  Somewhere<br />
  Over There<br />
  <em>USA</em></p>
  <p>
    <a href="mailto:contact@contoso.com">contact@contoso.com</a>
  </p>
</section>
<p>
  If you would like to contact Contoso Conferencing ...
</p>
...
```

7. Wrap the HTML form and **<p>** element immediately above it in a second **<section>** element, as shown in bold in the following code example:

```
...
<section>
  <p>
    If you would like to contact Contoso Conferencing ...
  </p>
  <form method="POST" action="support.aspx">
    ...
  </form>
</section>
```

8. On the **File** menu, click **Save All**.

Add a Header and a Footer to the Page

1. Enclose the **<h1>** element near the top of the **ContactUs.html** file in a **<header>** element, as shown in bold in the following code example:

```
...  
<article>  
  <b><header></header></b>  
    <h1>Contact Contoso Conferencing</h1>  
  </header>  
  ...  
</article>  
...
```

2. Add the following **** element shown in bold to the **<header>** element above the **<h1>** element.

```
<header>  
  <b></b>  
  <h1>Contact Contoso Conferencing</h1>  
</header>
```

3. Add the following HTML markup shown in bold immediately after the **</article>** tag near the end of the document.

```
...  
</article>  
<b><footer></footer></b>  
<p>  
  <small>  
    Last updated  
    <time datetime="2012-08">  
      August 2012  
    </time>  
  </small>  
</p>  
</b></body>  
</html>
```

4. On the **File** menu, click **Save All**.

View the Structure of the Page by Using the F12 Developer Tools

1. On the **Debug** menu, click **Start Without Debugging**.
2. In Internet Explorer, if the message **Intranet settings are turned off by default** appears, **click Don't show this message again**.
3. Press F12.
4. In the **F12** window, click the **HTML** tab.
5. Expand the **<html>** element.
6. Expand the **<body>** element.
7. Expand the **<article>** element and verify that it contains a **<header>** element and two **<section>** elements.

8. Expand the **<header>** element.
9. Expand the **<h1>** element.
10. Click each element, and verify that Internet Explorer surrounds each element on the page with a box as it is selected in the **F12** window.

Make a Temporary Change to the Page by Using the F12 Developer Tools

1. In the **<h1>** element, click **Contact Contoso Conferencing**.
2. Change this text to **We'd love to hear from you...**, and then press ENTER.
3. Verify that Internet Explorer displays the modified text.
4. Press F12 to close the **F12** window.
5. Close Internet Explorer, and then close Visual Studio 2012.

Lesson 2

Styling an HTML5 Page

Contents:

Demonstration

7

Demonstration

Demonstration: Adding CSS Styles to an HTML Page

Demonstration Steps

Create New Styles by Using Visual Studio

1. On the Windows 8 **Start** screen, click the **Visual Studio 2012** tile.
2. In Visual Studio, on the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, browse to the **E:\Mod02\Democode\Starter** folder, click **DemoWebSite.sln**, and then click **Open**.
4. In Solution Explorer, expand the **E:\...\DemoWebSite** web application, and then expand the **styles** folder.
5. Double-click **ContactUsStyles.css**
6. Review the existing rules for the **body** and **h1** elements.
7. Modify the **body** rule, remove the color rule, and change the font used on the whole page as shown in bold in the following code example.

```
body
{
    font-family: "Segoe UI", Helvetica, Arial, sans-serif;
}
```

8. Remove the following rule from the stylesheet:

```
h1 {
    font-family: 'Copperplate Gothic';
    color: red;
}
```

9. Add the following rules that make the header appear separately from the rest of the content.

```
header {
    padding-bottom: 10px;
    border-bottom: 2px dotted blue;
    margin-bottom: 10px;
}
header h1 {
    margin-left: 20px;
    display: inline-block;
}
```

10. Add the following empty rule:

```
section { }
```

11. Click after the opening curly brace for the section rule, and in the toolbar click the **Build Style** button.

 **Note:** If the toolbar is not visible, right-click in the body of the section rule and then click **Build Style**.

12. In the **Modify Style** dialog box, in the **Category** list, click **Box**.

13. Clear the **padding: Same for all** check box, and in the **bottom** box, type **5**.
14. In the **Category** list, click **Border**.
15. Under **border-style**, clear the **Same for all** check box, and in the **bottom** list box, click **dotted**.
16. Under **border-width**, clear the **Same for all** check box, and in the **bottom** box, type **1**.
17. Under **border-color**, clear the **Same for all** check box, and in the **bottom** box, type **grey**.
18. Click **OK**.
19. Verify that the section rule now looks like this:

```
section {  
    padding-bottom: 5px;  
    border-bottom-style: dotted;  
    border-bottom-width: 1px;  
    border-bottom-color: grey;  
}
```

20. Add the following rules to style the form and its elements.

```
fieldset {  
    background-color: pink;  
    margin-bottom: 10px;  
}  
legend {  
    font-size: 1.2em;  
    font-style: italic;  
}  
fieldset li {  
    list-style: none;  
    margin-bottom: 10px;  
}  
input[type="submit"] {  
    background-color: pink;  
    opacity: 0.6;  
    width: 200px;  
}
```

21. On the **File** menu, click **Save All**.

Module Review and Takeaways

Question: What are the new elements that HTML5 provides for specifying the semantic meaning of content in a web page?

Answer: The `<section>`, `<header>`, `<footer>`, `<nav>`, `<article>`, and `<aside>` elements.

Question: Which of the following items is **NOT** a property of the CSS box model?

- () Margin
- () Content
- () Border
- () Style
- () Padding

Answer:

- () Margin
- () Content
- () Border
- (✓) Style
- () Padding

Module 3

Introduction to JavaScript

Contents:

| | |
|----------------------------------|---|
| Lesson 3: Introduction to jQuery | 2 |
| Module Review and Takeaways | 5 |

Lesson 3

Introduction to jQuery

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Adding jQuery to a Web Project

Demonstration Steps

Add jQuery to a project by using nuGet

1. On the Windows 8 **Start** screen, click the **Visual Studio 2012** tile.
2. In Visual Studio, on the **File** menu, point to **New**, and then click **Web Site**.
3. In the **New Web Site** dialog box, in the middle pane click **ASP.NET Empty Web Site**.

 **Note:** It does not matter whether you select the **Visual Basic** or **Visual C#** templates in the left pane; the templates for both languages enable you to create HTML5 web pages and implement functionality by using JavaScript.

4. From the **Web location** list, click **File System**, set the file path to **E:\Mod03\Democode\jQueryDemoTest**, and then click **OK**.
5. On the **Tools** menu, point to **Library Package Manager**, and then click **Manage NuGet Packages for Solution**.
6. In the **Manage NuGet Packages** dialog box, click **Online**. A list of available packages for download appears in order of popularity.
7. In the **Search Online** box, type **jQuery**.
8. In the list of packages, click **jQuery**, and click **Install**.
9. In the **Select Projects** dialog box, click **OK**.
10. In the **Manage NuGet Packages** dialog box, click **Close**.
11. In Solution Explorer, notice that a **Scripts** folder has been added to the project and that it contains three files.

 **Note:** The three files are the uncompressed jQuery library, a minified version for use on production servers, and an IntelliSense file for Visual Studio to use.

12. On the **File** menu, click **Save jQueryDemoTest**.

Enable jQuery IntelliSense

1. In the Solution Explorer window, right-click the **Scripts** folder, point to **Add**, and then click **JavaScript File**.
2. In the **Specify Name for Item** dialog box, type **test**, and then click **OK**.
3. Type the following code and notice that IntelliSense is unable to offer suggestions for statement completion for jQuery functions such as **ready**.

```
$(document).ready(function () {
    // your code here
});
```

4. In the Solution Explorer window, right-click the **Scripts** folder, point to **Add**, and then click **JavaScript File**.
5. In the **Specify Name for Item** dialog box, type **_references.js**, and then click **OK**.

6. Add the following code to _references.js and then save it.

```
/// <reference path="jquery-1.8.2.js" />
```

-  **Note:** If necessary, replace the jQuery version number (1.8.2) with that of the files downloaded by the package manager.

7. In the _references.js file, type the following code and notice that IntelliSense is now able to offer suggestions for statement completion for jQuery functions, including the **ready** function.

```
$(document).ready(function () {  
    // your code here  
});
```

8. On the **File** menu, click **Save All**.
9. Close Visual Studio 2012.

Module Review and Takeaways

Question: All variables in JavaScript are strongly typed, and you must specify the type of a variable when you create it. True or False?

- () True
- () False

Answer:

- () True
- (✓) False

Question: What is the purpose of the DOM?

Answer: The DOM represents the structure of a web page. You use it to add dynamic functionality to the page.

Question: Which jQuery function indicates that the contents for a page have been loaded into the browser?

- () loaded
- () available
- () \$
- () ready
- () bind

Answer:

- () loaded
- () available
- () \$
- (✓) ready
- () bind

Module 4

Creating Forms to Collect and Validate User Input

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: If you define a field with an HTML5 input type that is not supported by the user's browser, the field does not appear on the form when the browser displays it. True or False?

- () True
() False

Answer:

- () True
(✓) False

Question: If you perform validation in the browser, is it necessary to perform the same validation checks in the server code that processes the data, or is this additional processing redundant?

Answer: You should always validate data at the server regardless of whether it has already been validated by the browser.

Question: You should always use the **input** event to validate data that a user enters into a field, in preference to the **submit** event of the form. True or False?

- () True
() False

Answer:

- () True
(✓) False

Module 5

Communicating with a Remote Server

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: In the **onreadystatechange** event handler for the **XMLHttpRequest** object, which property should you examine to ensure that data has been returned, and what value should this property contain?

- () The readyState property should be set to 0.
- () The responseText property should be set to a non-null value.
- () The readyState property should be set to 4.
- () The status property should be set to 200 (HTTP OK).
- () The HTTPResponse property should be set to 0.

Answer:

- () The readyState property should be set to 0.
- () The responseText property should be set to a non-null value.
- (v) The readyState property should be set to 4.
- () The status property should be set to 200 (HTTP OK).
- () The HTTPResponse property should be set to 0.

Question: If you use the jQuery **get()** function to retrieve data and you do not specify an error handling function, any failures while retrieving the data will cause your JavaScript code to stop with an unhandled exception. True or False?

- () True
- () False

Answer:

- () True
- (v) False

Module 6

Styling HTML5 by Using CSS3

Contents:

| | |
|----------------------------------|---|
| Lesson 2: Styling Block Elements | 2 |
| Module Review and Takeaways | 7 |

Lesson 2

Styling Block Elements

Contents:

Demonstration

3

Demonstration

Demonstration: Switching Between CSS Layout Models

Demonstration Steps

Switch between layout modes in a web page

1. On the Windows 8 **Start** screen, click the **Visual Studio 2012** tile.
2. In Visual Studio, on the **File** menu, point to **Open**, and then click **File**.
3. In the **Open File** dialog, browse to the **E:\Mod06\Democode** folder, click **positioning.html**, and then click **Open**.
4. Review the code with the students. This file contains an HTML **article** with four **div** elements. The file also contains styles for the **article** and **div** elements.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Positioning Demo</title>
    <style type="text/css">
        body {
            text-align: center;
        }
        article {
            padding: 10px;
            border: 2px solid red;
        }
        div {
            margin: 10px;
            padding: 5px;
            border: 2px solid black;
            width: 150px;
            height : 150px;
        }
        div:nth-child(odd) {
            font-size: 4rem;
        }
    </style>
</head>
<body>
    <article>
        <div id="one">One</div>
        <div id="two">Two</div>
        <div id="three">Three</div>
        <div id="four">Four</div>
    </article>
</body>
</html>
```

5. On the **File** menu, click **View in Brower (Internet Explorer)**.
6. In Internet Explorer, if the message **Intranet settings are turned off by default** appears, click **Don't show this message** again.
7. Observe the four **div** elements laid out underneath each other, in order, within the **article** element. The **article** element has a red border to highlight its boundaries. This is block layout mode.

The web page looks like this:

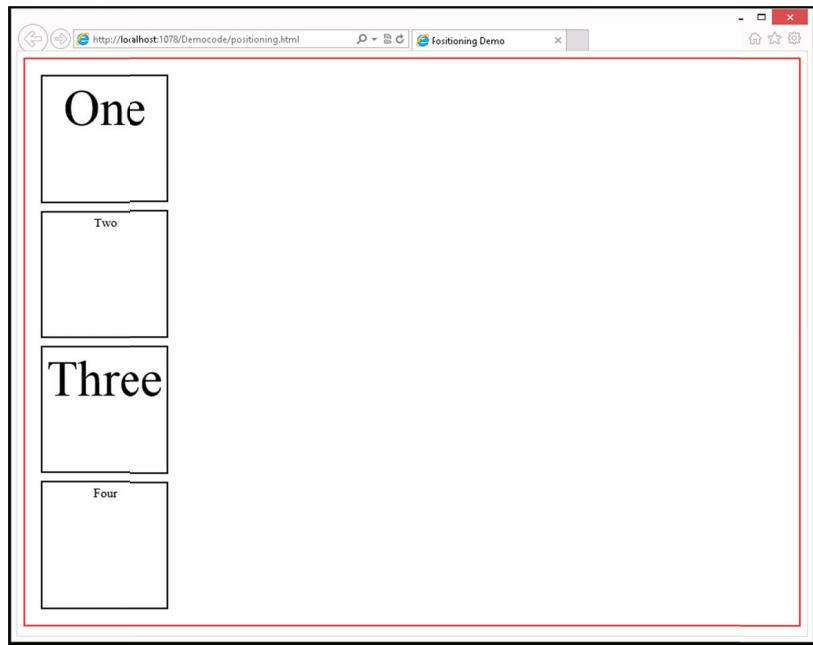


Figure 6.1:The div elements in block layout mode

8. Press **F12**.
9. In the F12 Developer Tools pane, press **Ctrl+P** to unpin the window. Position the F12 Developer Tools window so that you can see the Internet Explorer window at the same time.
10. Click the **CSS** tab to display the fully expanded version of the layout rules applied to the HTML content.
11. Right-click the **div** entry, and then click **Add attribute**.
12. Type **display: inline**, and then press ENTER.
13. In Internet Explorer, notice that the four **div** elements are now laid out side-by-side aligned by text baseline with height and width properties ignored. This is inline layout mode.

The web page now looks like this:

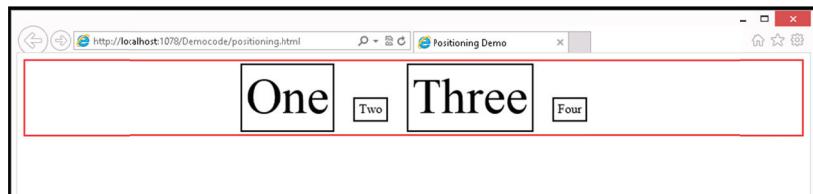


Figure 6.2:The div elements in inline layout mode

14. Resize the browser window to make it narrower, so you can see how blocks are wrapped onto the next line in inline layout mode.
15. In the F12 Developer Tools window, on the **CSS** tab, click the **display: inline** rule, change it to read **display:inline-block**, and then press ENTER.
16. Notice the layout is the same but the **height** and **width** properties are now preserved. This is inline-block mode.



Note: If necessary, make the browser window wider so that blocks **One** and **Three** are on the same line.

The web page now looks like this:

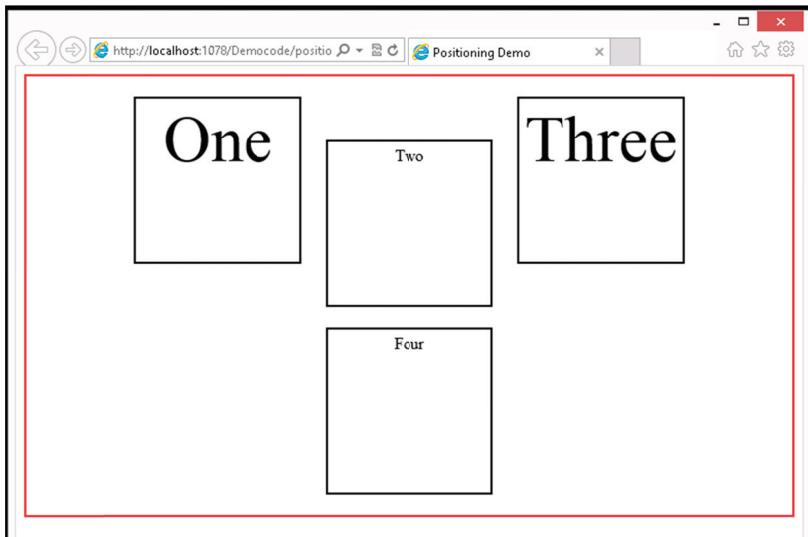


Figure 6.3:The div elements in inline-block layout mode

17. In Internet Explorer, resize the browser window so you can see how blocks are wrapped onto the next line in inline layout mode.
18. In the F12 Developer Tools window, on the **CSS** tab, click the **display: inline-block** rule. Change this rule to **display:-ms-flexbox**, and then press ENTER
19. Switch to Internet Explorer to view the new layout. The **div** elements are displayed in a vertical column.
20. In the F12 Developer Tools window, on the **CSS** tab, click the **display: -ms-flexbox** rule, change it to **display:table-cell**, and then press ENTER.
21. Switch to Internet Explorer to view the new layout. The **div** elements are displayed in a horizontal table.

Switch between positioning modes in a web page

1. In the F12 Developer Tools window, on the **CSS** tab, clear the three checkboxes next to the display attributes for **body**, **article**, and **div**.
2. Right-click the **div** entry, and then click **Add rule after**.
3. Type **#three** and then press **Tab**.



Note: This action creates a new rule for the **<div>** element with the **id** property set to **three**. This is the **<div>** containing the text **Three**.

4. Type **position: relative** and then press ENTER.
5. Right-click the **#three** entry, and then click **Add attribute**.
6. Type **top: 2em**, and then press ENTER.
7. Right-click the **#three** entry, and then click **Add attribute**.

8. Type **left: 2em**, and then press ENTER.
9. In Internet Explorer, notice how the **three** box is positioned relative to its normal position.
10. In the F12 Developer Tools window, on the **CSS** tab, click the **position:relative** rule for the **#three** selector, change it to **position:absolute**, and then press ENTER.
11. In Internet Explorer, notice how the **three** box is now positioned relative to its containing **article** block.
12. In the F12 Developer Tools window, on the **CSS** tab, on the **CSS** tab, click the **position:absolute** rule for the **#three** selector, change it to **position:fixed**, and then press ENTER.
13. In Internet Explorer, notice how the **three** box is positioned relative to the browser window. Make the window small enough to require scrolling and see how the **three** box remains stationary when you scroll (it does not scroll into view).
14. Close Internet Explorer, and then close Visual Studio 2012.

Module Review and Takeaways

Question: Which CSS rule can you use to download a font required by a web page?

- () @font-family
- () @font-style
- () @font-face
- () @font
- () You cannot download fonts by using CSS.

Answer:

- () @font-family
- () @font-style
- (✓) @font-face
- () @font
- () You cannot download fonts by using CSS.

Question: What are the main differences between the CSS box model, flex-box, and multi-column layout?

Answer: Flex-box is a layout module for the box model.

Multi-column layout sets out the number of content columns within a box.

Question: How do you select the first item in a list so that you can apply styling to it?

Answer: Use the **li:first-child** selector in CSS3.

Module 7

Creating Objects and Methods by Using JavaScript

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: How can you guard against name clashes in JavaScript?

Answer: You can use immediately invoked functions, define namespaces, and use strict mode.

Question: If you modify the prototype object for a constructor function, the changes are only visible to new objects that you create by using that constructor function; existing objects created by using the constructor function will be unaffected. True or False?

() True

() False

Answer:

() True

(✓) False

Question: Which of the following statements is true?

() JavaScript uses the public, private, and protected keywords to implement encapsulation.

() JavaScript does not support encapsulation.

() JavaScript uses closures to achieve encapsulation.

() JavaScript uses prototype chaining to achieve encapsulation.

() JavaScript uses the Object.create() function to implement encapsulation.

Answer:

() JavaScript uses the public, private, and protected keywords to implement encapsulation.

() JavaScript does not support encapsulation.

(✓) JavaScript uses closures to achieve encapsulation.

() JavaScript uses prototype chaining to achieve encapsulation.

() JavaScript uses the Object.create() function to implement encapsulation.

Module 8

Creating Interactive Pages by Using HTML5 APIs

Contents:

| | |
|-----------------------------------------------------|---|
| Lesson 4: Debugging and Profiling a Web Application | 2 |
| Module Review and Takeaways | 6 |

Lesson 4

Debugging and Profiling a Web Application

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Using the F12 Developer Tools to Debug JavaScript Code

Demonstration Steps

Set a Breakpoint in JavaScript code

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, browse to the file **E:\Mod08\Democode\Document.html**.
4. If the message **Internet Explorer restricted this webpage from running scripts or ActiveX controls** appears, click **Allow blocked content**.
5. Press F12 to display the Developer Tools window.
6. If the F12 Developer Tools window appears as a pane in Internet Explorer, in the F12 Developer Tools pane, click **Unpin**.



Note: This action causes the F12 Developer Tools to appear in a standalone window.

7. In the F12 Developer Tools window, click **Script**.
8. Click in the margin next to line 12, in order to create a breakpoint on the first statement inside the **onload()** function.

The **Script** window should look like this:

The screenshot shows the F12 Developer Tools window titled "Hello HTML5 - F12". The "Script" tab is selected. The code editor displays a script with several lines of code. A red dot, indicating a breakpoint, is positioned in the margin next to line 12, which contains the opening brace of the `onload()` function. The rest of the code includes HTML headers, a main container div, and a section with a navigation menu and an article header.

```

Hello HTML5 - F12
File Find Disable View Images Cache Tools Validate | Browser Mode: IE10 Document Mode: Standards
HTML CSS Console Script Profiler Network
file:///E:/Mod08/Democode/Document.html
7 <link rel="stylesheet" href="MyStylesheet.css">
8
9 <script type="text/javascript">
10
11     function onload() {
12         var hyperlinks = document.querySelectorAll("a");
13
14         for (var i = 0; i < hyperlinks.length; i++) {
15             hyperlinks[i].style.backgroundColor = "yellow";
16         }
17     }
18
19     window.addEventListener("load", onload, false);
20
21 </script>
22
23 </head>
24
25 <body>
26
27     <header>
28         <h1>Hello HTML5</h1>
29         <h2>Enjoy the ride!</h2>
30         <h3>Fancy eh?</h3>
31     </header>
32
33     <div id="MyMainContainer">
34
35         <nav>
36             <h3>Links</h3>
37             <a href="http://www.microsoft.com">Microsoft</a>
38             <a href="http://www.beautyoftheweb.com">Beauty of the Web</a>
39         </nav>
40
41         <section>
42
43             <article>
44                 <header>
45                     <h1>Welcome!</h1>
46                 </header>

```

Figure 8.1:The Script window with a breakpoint set

Step Through JavaScript Code and Examine Variables

1. On the F12 Developer Tools toolbar, click **Start debugging**.
2. In the F12 Developer Tools window, verify that the debugger pauses execution at the breakpoint.
3. In the right pane, click the **Locals** tab.

4. On the F12 Developer Tools toolbar, click the **Step over** button several times to step through the code. In the **Locals** tab, verify that the value of the **i** variable changes as execution progresses.
5. Close the F12 Developer Tools window.
6. Close Internet Explorer.

Demonstration: Using the F12 Developer Tools to Profile a Web Application

Demonstration Steps

Examine the Network Traffic for a Web Application

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, browse to the website <http://www.beautyoftheweb.com/>.
4. Press F12 to display F12 Developer Tools window.
5. In the F12 Developer Tools window, click **Network**.
6. On the F12 Developer Tools toolbar, click **Start capturing**.
7. In Internet Explorer, click the **touchgallery** icon in the navigation bar.
8. Return to the F12 Developer Tools window.

The network traffic should look like this (the actual values displayed may vary):

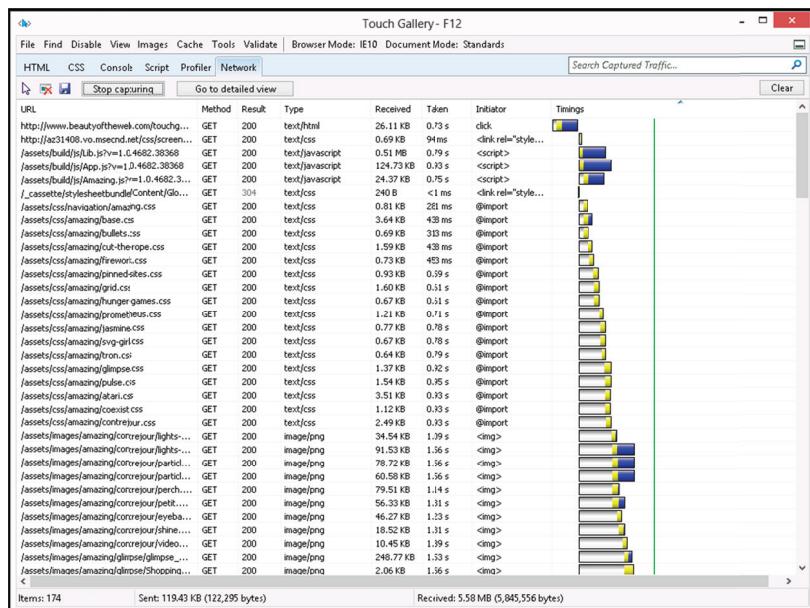


Figure 8.2:The network traffic captured for a web page

9. Click the first line of the network capture, and then on the toolbar click **Go to detailed view**.
10. Click each of the following tabs and show the data that they contain:
 - o **Request headers**.
 - o **Request body**.
 - o **Response headers**.
 - o **Response body**.

- **Cookies.**
 - **Initiator.**
 - **Timings.**
11. On the F12 Developer Tools toolbar, click **Stop capturing**.

Capture Profile Data for a Web Application

1. In the F12 Developer Tools window, click **Profiler**.
2. On the F12 Developer Tools toolbar, click **Start profiling**.
3. In Internet Explorer, click the **videos** icon in the navigation bar.
4. Return to the F12 Developer Tools window.
5. On the F12 Developer Tools toolbar, click **Stop profiling**.

The captured profile data should look like this:

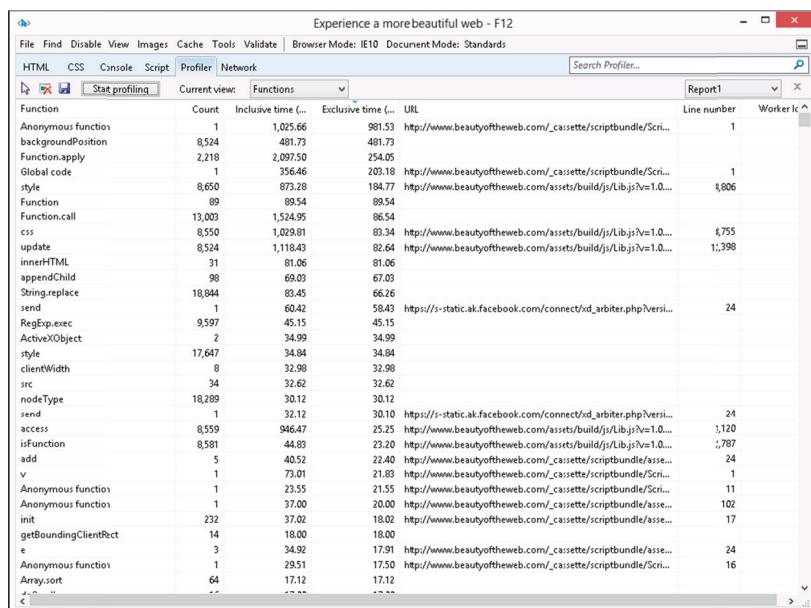


Figure 8.3:Profile data for a web page

6. On the F12 Developer Tools toolbar, in the **Current view** list, click **Call tree**.

7. In the **Search Profiler** box, type **onWindowEvent** and then press **ENTER**.

In the captured data, examine the work performed by the **onWindowEvent** event handler.

8. Double-click **onWindowEvent** to display the code for this event handler in the **Script** window
9. Press **Ctrl+ Alt+ O** to display the tools menu, and then select **Format JavaScript** to display the code in an easier to read format.
10. Close the F12 Developer Tools window.
11. Close Internet Explorer.

Demonstration: Creating Interactive Pages with HTML5 APIs

Module Review and Takeaways

Question: What methods are provided by the **FileReader** interface for reading files on the local file system?

Answer: **readAsText()**, **readAsDataURL()**, and **readAsArrayBuffer()**.

Question: HTML5 browsers are guaranteed to support the .mp4 video format. True or false?

- () True
- () False

Answer:

- () True
- (✓) False

Question: What methods are provided by the **navigator.geolocation** object for obtaining geolocation information?

Answer: **getCurrentPosition()** and **watchPosition()**.

Question: The F12 Developer Tools require that you have Visual Studio installed on your computer before you can use them to debug JavaScript code. True or False?

- () True
- () False

Answer:

- () True
- (✓) False

Module 9

Adding Offline Support to Web Applications

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: What are the primary differences between data retained on a user's device by using session storage and by using local storage?

Answer: Data stored by using session storage has a different scope and lifetime than data retained by using local storage.

Question: You configure a web page to use the application cache to cache a resource locally. If the resource is modified at the web server, then the browser automatically downloads the latest version. True or false?

() True

() False

Answer:

() True

(v) False

Module10

Implementing an Adaptive User Interface

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: What are the main device characteristics used by media queries to detect whether a client device is a hand-held tablet?

Answer: **device-width**, **device-height**, and **orientation**

Question: How can you style content to adapt to the type and form factor of the device on which a user is viewing your web site?

Answer: Use a media query.

Module 11

Creating Advanced Graphics

Contents:

| | |
|-------------------------------------------------------------|---|
| Lesson 1: Creating Interactive Graphics by Using SVG | 2 |
| Lesson 2: Drawing Graphics by Using the Canvas API | 4 |
| Module Review and Takeaways | 6 |

Lesson 1

Creating Interactive Graphics by Using SVG

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Using SVG Transformations and Events

Demonstration Steps

Transform SVG Elements

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod11\Democode\SvgDocument.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. In Internet Explorer, click the **Transformations** button.
6. Right-click the web page in Internet Explorer, and then click **View source**.
7. In the source window, locate the `<!-- Demonstrate transformations -->` comment and review the **<svg>** element:
 - o The **<transform>** elements translate the square by 200 units in the X and Y axes, scale it by a factor of 0.5, and rotate it.
8. Close the source window.

Handle Events on SVG Elements

1. In Internet Explorer, click the **Events** button.
2. Hover the mouse over the red shape on the left hand of the window. Verify that the shape changes to a yellow fill color and a dotted green border.
3. Move the mouse off the shape. Verify that it reverts to a red fill color with no outline.
4. Hover the mouse over the blue shape on the right side of the window. Verify that the shape changes to a yellow fill color and a dotted green border.
5. Move the mouse off the shape. Verify that it reverts to a blue fill color with no outline.
6. Click the red shape. Verify that a message box appears, indicating that the shape represents Alaska. Close the message box.
7. Click the blue shape. Verify that a message box appears, indicating that the shape represents Hawaii. Close the message box.
8. Right-click in the browser window, and then click **View source**.
9. In the source window, locate the `<!-- Demonstrate events -->` comment and review the **<svg>** element:
 - o The **<path>** elements contain the data that defines the two maps.
 - o Each **<path>** element responds to the **onmousedown** event and uses JavaScript code to display the appropriate message.
10. In the source window, locate the **path:hover** CSS rule near the top of the document. This CSS rule defines the style for all **<path>** elements when the user hovers over them with the mouse.
11. Close the source window.
12. Close Internet Explorer.

Lesson 2

Drawing Graphics by Using the Canvas API

Contents:

| | |
|---------------|---|
| Demonstration | 5 |
|---------------|---|

Demonstration

Demonstration: Performing Transformations by Using the Canvas API

Demonstration Steps

Perform Simple Transformations

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod11\Democode\CanvasDocument.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. In Internet Explorer, click the **Separate Transformations** button.
6. Right-click the web page in Internet Explorer, and then click **View source**.
7. In the source window, locate the **demoSeparateTransformations** function and review the code:
 - o The **demoSeparateTransformations** function uses the **drawShape** function to draw a triangle filled with an image of the Welsh flag.
 - o Before calling the **drawShape** function, the code transforms the context; it moves the canvas to the right and down by half the width and height of the canvas, then it scales the context by a different value in the X and Y dimensions, and then rotates the context by PI/4 radians.
 - o When the **drawShape** function is called, the image is transformed according to the context settings.



Note: If time allows, comment out each of the transformations and run the code again. Then uncomment each transformation one at a time, so that students can see the effects of each one.

Perform Matrix Transformation

1. In Internet Explorer, click the **Matrix Transformations** button.
2. In the source window, locate the **demoMatrixTransformations** function and review the code:
 - o This function is similar to the previous one in that it transforms the context and then calls the **drawShape** function to display the image.
 - o The difference is that this function uses the **transform** function to perform a matrix transformation, scaling, skewing, and translating the context in a single function call.
3. Close the source window.
4. Close Internet Explorer.

Module Review and Takeaways

Question: Which of the following statements about SVG is false?

- () You can use SVG to draw complex shapes, and fill them with gradients and patterns.
- () SVG elements are parsed by the browser when the page is first loaded, and they are then discarded from memory.
- () You can create SVG elements dynamically by using DOM functions such as `document.createElement()`.
- () You can handle events on SVG elements.
- () SVG elements must be enclosed in an `<svg>` container element on a web page.

Answer:

- () You can use SVG to draw complex shapes, and fill them with gradients and patterns.
- (v) SVG elements are parsed by the browser when the page is first loaded, and they are then discarded from memory.
- () You can create SVG elements dynamically by using DOM functions such as `document.createElement()`.
- () You can handle events on SVG elements.
- () SVG elements must be enclosed in an `<svg>` container element on a web page.

Question: When might you consider using the Canvas API instead of using SVG?

Answer: <Add Answer here>

Module 12

Animating the User Interface

Contents:

| | |
|---------------------------------------------------|----|
| Lesson 1: Applying CSS Transitions | 2 |
| Lesson 2: Transforming Elements | 4 |
| Lesson 3: Applying CSS Keyframe Animations | 8 |
| Module Review and Takeaways | 11 |

Lesson 1

Applying CSS Transitions

Contents:

| | |
|---------------|---|
| Demonstration | 3 |
|---------------|---|

Demonstration

Demonstration: Using CSS Transitions

Demonstration Steps

Apply CSS Transitions to HTML Elements

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod12\Democode\Transitions.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. Hover over the upper rectangle on the web page. Verify that the following transitions are applied simultaneously to the rectangle:
 - o The width, height, and font size increase over a period of two seconds.
 - o The background color transitions to red over a period of 3.75 seconds.
6. Move the mouse away from the upper rectangle. Verify that the rectangle reverts to its original appearance, over the same period of time.
7. Repeat the previous two steps for the second rectangle. Verify that the same animations apply.
8. Right-click the page in Internet Explorer and then click **View source**. Note that:
 - o The first **div** rule defines default CSS properties for all **<div>** elements.
 - o The **div.simple** rule defines a **transition** property that applies to transitions on the **width**, **height**, **font-size**, and **background-color** CSS properties.
 - o The **div.complex** rule defines similar transitions, but it uses separate **transition-property**, **transition-duration**, **transition-timing-function**, and **transition-delay** properties.
 - o The **div:hover** rule defines the final values for the **width**, **height**, **font-size**, and **background-color** CSS properties when the user hovers over a **<div>**.

Handle the **transitionend** Event

1. In Internet Explorer, expand the drop-down list box. Verify that it displays messages for all the transitions that have ended. The list includes events for the original transitions when you hover over a rectangle, as well as events for the reverse transitions when you move the mouse away from a rectangle.
2. Switch to the source window.
3. In the JavaScript code, note that:
 - o The **onLoad()** function sets up event-handlers that call the **onTransitionend()** function when the **transitionend** event is raised on all the **<div>** elements.
 - o The **onTransitionend()** function displays information about the **transitionend** event, by using the **propertyName** and **elapsedTime** properties of the event argument.
4. Close the source window.
5. Close Internet Explorer.

Lesson 2

Transforming Elements

Contents:

| | |
|---------------|---|
| Demonstration | 5 |
|---------------|---|

Demonstration

Demonstration: Performing 2D Transformations

Demonstration Steps

Perform 2D Translations

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod12\Democode\2DTranslations.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. Verify that the browser displays a series of rectangles. Each rectangle demonstrates how to perform a 2D translation by using the **translate()**, **translateX()**, or **translateY()** functions. The text message inside each rectangle describes the transformation for that rectangle.
6. Right-click in the browser window, and then click **View source**.
7. In the source window, scroll down to the bottom of the document. Note that the body of the document has a series of **<div>** elements, and each element has a distinct CSS class.
8. Scroll back up to the top of the code and locate the **<style>** element. The CSS rules implement translations for all the **<div>** elements. The CSS rules set the standard **transform** property, as well as the following vendor-specific properties:
 - **-ms-transform**: Perform a transformation on Internet Explorer 9.
 - **-webkit-transform**: Perform a transformation on Webkit-based browsers such as Chrome and Safari.
 - **-moz-transform**: Perform a transformation on Mozilla browsers.
 - **-o-transform**: Perform a transformation on Opera browsers.
9. Close the source window.

Perform 2D Scaling Transformations

1. In Internet Explorer, open the file **E:\Mod12\Democode\2DScaling.html**.
2. Verify that the browser displays a series of rectangles. Each rectangle demonstrates how to perform a 2D scaling transformation by using the **scale()**, **scaleX()**, or **scaleY()** functions. The text message inside each rectangle describes the transformation for that rectangle.
3. Right-click in the browser window, and then click **View source**.
4. In the source window, scroll down to the bottom of the document. Note that the body of the document has a series of **<div>** elements, and each element has a distinct CSS class.
5. Scroll back up to the top of the code and locate the **<style>** element. The CSS rules implement scaling transformations for all the **<div>** elements. The CSS rules set the standard **transform** property, as well as vendor-specific properties.
6. Close the source window.

Perform 2D Rotations

1. In Internet Explorer, open the file **E:\Mod12\Democode\2DRotations.html**.

2. Verify that the browser displays a series of rectangles. Each rectangle demonstrates how to perform a 2D rotation by using the **rotate()** function. The fourth rectangle also shows how to perform multiple transformations, and how to change the origin of the transformation to the top left of the target element.
3. Right-click in the browser window, and then click **View source**.
4. In the source window, scroll down to the bottom of the document. Note that the body of the document has a series of **<div>** elements, and each element has a distinct CSS class.
5. Scroll back up to the top of the code and locate the **<style>** element. The CSS rules implement rotations for all the **<div>** elements. The CSS rules set the standard **transform** property, as well as vendor-specific properties. The final CSS rule shows how to apply multiple transformations, specifically a translation followed by a rotation. The final CSS rule also shows how to set change the origin of the transformation by setting the **transform-origin** property and its vendor-specific equivalent properties.
6. Close the source window.

Perform 2D Skewing Transformations

1. In Internet Explorer, open the file **E:\Mod12\Democode\2DSkewing.html**.
2. Verify that the browser displays a series of rectangles. Each rectangle demonstrates how to perform a 2D skewing operation by using the **skew()**, **skewX()**, or **skewY()** functions.
3. Right-click in the browser window, and then click **View source**.
4. In the source window, scroll down to the bottom of the document. Note that the body of the document has a series of **<div>** elements, and each element has a distinct CSS class.
5. Scroll back up to the top of the code and locate the **<style>** element. The CSS rules implement skewing transformations for all the **<div>** elements.
6. Close the source window.
7. Close Internet Explorer.

Demonstration: Performing 3D Transformations

Demonstration Steps

Perform 3D Transformations that Include Transitions

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod12\Democode\3DTransformations.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. Verify that the browser displays a cube. There are six faces to the cube, each of which displays text and has a different background color. The front face is partially transparent, so that it does not completely obscure the other faces.
6. Hover over the cube. Verify that it rotates by 90 degrees over a period of five seconds. Then move the mouse off the cube, and verify that the cube rotates smoothly back to its original position.
7. Right-click in the browser window, and then click **View source**.

8. In the source window, scroll down to the bottom of the document. Note that the body of the document has six `<div>` elements that represent the six faces of the cube. These `<div>` elements are contained in a parent `<div>` element named **container**.
9. Scroll back up to the top of the code and locate the `<style>` element. Note the following CSS rules:
 - **#container**: Specifies a perspective for all the child elements of the **container** element, and a transition of five seconds for transformations.
 - **#container:hover**: Specifies a rotation of 90 degrees when the user hovers over the **container** element.
 - **#rightFace**: Transforms the **rightFace** element in 3D space, so that it appears on the right side of the cube.
 - **#leftFace**: Transforms the **leftFace** element in 3D space, so that it appears on the left side of the cube.
 - **#topFace**: Transforms the **topFace** element in 3D space, so that it appears on the top of the cube.
 - **#bottomFace**: Transforms the **bottomFace** element in 3D space, so that it appears on the bottom of the cube.
 - **#backFace**: Transforms the **backFace** element in 3D space, so that it appears at the back of the cube.
 - **#frontFace**: Transforms the **frontFace** element in 3D space, so that it appears at the front of the cube. The background color is partially transparent.
10. Close the source window.
11. Close Internet Explorer.

Lesson 3

Applying CSS Keyframe Animations

Contents:

| | |
|---------------|---|
| Demonstration | 8 |
|---------------|---|

Demonstration

Demonstration: Implementing KeyFrame Animations

Demonstration Steps

Define and Run a Keyframe Animation

1. On the Windows 8 **Start** screen, click the **Desktop** tile.
2. On the Windows taskbar, click **Internet Explorer**.
3. In Internet Explorer, open the file **E:\Mod12\Democode\KeyframeAnimations.html**.
4. If a message box appears asking if you want to allow blocked content, click the **Allow blocked content** button.
5. Verify that a green rectangle appears on the page, with a small white circle in the top left corner. The green rectangle represents a pool table and the white circle represents a ball. There is also a button that enables you to start the animation.
6. Click **Start Animation**.
 - o After three seconds, the ball starts moving diagonally on the pool table. The color of the pool table also changes to blue, and a message appears at the bottom of the page to indicate the start time of the animation.
 - o As the animation proceeds, the ball appears to bounce off the sides of the pool table and the color of the ball varies from white to yellow, then to orange, then to red, and then finally to purple.
 - o When the ball reaches the bottom right corner of the pool table, a message appears to indicate that the first iteration of the animation has completed. The next iteration of the animation begins; this iteration plays the animation in reverse, so that the ball ends up in its original position and with its original color.
 - o At the end of the animation, the pool table reverts to green and messages appear at the bottom of the page to indicate the elapsed time of the animation.
7. Right-click in the browser window, and then click **View source**.
8. In the source window, scroll down to the bottom of the document. Note that the body of the document has a **<div>** element named **pooltable** that represents the pool table, and a nested **<div>** named **ball** that represents the ball on the pool table. There is also a **<button>** element to start the animation, and a **<div>** named **messageLabel** where messages are displayed.
9. Scroll back up to the top of the file and locate the **<style>** element. Note the following CSS rules:
 - **#pooltable**: Specifies the initial appearance of the pool table.
 - **#pooltable.animate**: Specifies a different color for the pool table during an animation. There is JavaScript code elsewhere in the document that programmatically adds the **animate** class to the **pooltable** element when an animation starts, to cause the pool table to turn blue during an animation.
 - **#ball**: Specifies the initial appearance of the ball.
 - **@-ms-keyframes ballmovement**: Defines a keyframe animation named **ballmovement**. The first rule-set specifies the original color and location of the ball. Each subsequent rule-set simulates the ball hitting one of the sides of the pool table, and causes the ball to change color during each part of its journey. The final rule-set specifies the final color and location of the ball.

- **#ball.animate**: Applies the **ballmovement** keyframe animation to a ball when the ball has the **animate** class. There is JavaScript code elsewhere in the document that programmatically adds the **animate** class to the **ball** element when the user clicks the **Start Animation** button, to trigger the animation.

10. Locate the `<script>` element. Note that:

- The **init()** function, invoked as soon as the page has loaded, establishes event-handler functions for the **MSAnimationStart**, **MSAnimationIteration**, and **MSAnimationEnd** events on the **ball** element:
- The **MSAnimationStart** event-handler function is called when an animation starts on the **ball** element. The function adds the **animate** class to the pool table, so that the pool table becomes blue. The function also displays a message to indicate the time when the animation started.
- The **MSAnimationIteration** event-handler function is called when each iteration of the animation has completed. The function displays a message to indicate the elapsed time of the animation.
- The **MSAnimationEnd** event-handler function is called when an animation ends on the **ball** element. The function enables the button, removes the **animate** class from the ball and the pool table, and displays a message to indicate the elapsed time of the animation.
- The **startAnimation()** function is Invoked when the user clicks the **Start Animation** button. The function disables the button, and adds the **animate** class to the ball to trigger the animation. The animation starts three seconds later, due to the **-ms-animation-delay: 3s;** property in the **#ball.animate** CSS rule.

11. Close the source window.

12. Close Internet Explorer.

Module Review and Takeaways

Question: What happens if you do not set the **transition-duration** property of a CSS transition?

Answer: The transition occurs immediately.

Question: Which of the following operations can you **NOT** perform by using a CSS transformation?

- () Rotate
- () Translate
- () Animate
- () Scale
- () Skew

Answer:

- () Rotate
- () Translate
- (✓) Animate
- () Scale
- () Skew

Question: What are the steps for implementing a keyframe animation?

1. **Answer:** Define a **@keyframe** rule that specifies how property values change at each step of the animation.
2. Reference the **@keyframe** rule from the **animation-name** property of a style rule for the elements to be animated.

Module 13

Implementing Real-time Communication by Using Web Sockets

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: Web socket clients send and receive data over an HTTP connection. True or False?

() True

() False

Answer:

() True

(v) False

Question: How does a client create a connection to a web socket server?

() The client uses the open() function of the WebSocket object and specifies the URL of the server.

() The client uses the connect() function of the WebSocket object and specifies the URL of the server.

() Web sockets use a stateless protocol similar to HTTP. A client application simply specifies the address of the server as a parameter of the send() function of the WebSocket object. A connection is automatically established while the message is sent and then closed.

() The client creates a new WebSocket object and specifies the URL of the server.

() The client has to wait until the server responds to send a message that grants it permission to connect.

Answer:

() The client uses the open() function of the WebSocket object and specifies the URL of the server.

() The client uses the connect() function of the WebSocket object and specifies the URL of the server.

() Web sockets use a stateless protocol similar to HTTP. A client application simply specifies the address of the server as a parameter of the send() function of the WebSocket object. A connection is automatically established while the message is sent and then closed.

(v) The client creates a new WebSocket object and specifies the URL of the server.

() The client has to wait until the server responds to send a message that grants it permission to connect.

Module 14

Performing Background Processing by Using Web Workers

Contents:

| | |
|-----------------------------|---|
| Module Review and Takeaways | 2 |
|-----------------------------|---|

Module Review and Takeaways

Question: Shared web workers can share data in the DOM of a web page, but dedicated web workers cannot. True or false?

() True

() False

Answer:

() True

() False

Question: How does a web page communicate with a web worker?

Answer: A web page and a web worker communicate by sending messages.