

Trainer Preparation Guide for Course 20480B: Programming in HTML5 with JavaScript and CSS3

Design of the Course

The purpose of this course is to teach students how to build web applications by using HTML5, CSS3, and JavaScript. It is assumed that students are already familiar with the essentials of HTML, and so the course only covers the new features in HTML5. Additionally, students should be experienced programmers and they must be familiar with a programming language such as Java, C++, C#, or Visual Basic. JavaScript is explained in terms that should be familiar to developers who already know how to write software.

The course is structured around a collection of labs that take the student through the process of designing and implementing a web application. The application itself supports a conference that describes web technologies. It includes functionality that enables an attendee to register for the conference, and it provides an overview of the conference. The web application also presents detailed scheduling information about each session, a mechanism to enable attendees to ask questions of presenters (and see the responses), and the opportunity to provide feedback about the conference.

Module 1

Module 1 provides an overview of HTML and CSS. Much of the material on HTML should be review, but use it to ensure that all students are up to speed and understand the terminology that you will be using throughout the course. Students may be a little less familiar with CSS than they are with HTML, so be prepared to spend a bit more time on the second lesson. The important thing is that students should understand the basic syntax of a CSS rule and how CSS selectors work. The third lesson introduces Visual Studio 2012 as a tool for building web applications, and the F12 Developer Tools in Internet Explorer for examining and debugging web pages. The bulk of this lesson consists of two demonstrations; one for Visual Studio 2012 and the other showing the F12 Developer Tools. Be prepared to give any useful hints and tips based on your experience with these tools. The lab at the end of this module is a walkthrough of the complete web application, showing the purpose of each page and how it is structured. Starting in the lab for module 2, students will build this application from scratch, so this exercise gives students a thorough understanding of the final solution. The second exercise gives students some practice using Visual Studio 2012 to modify the content of the web application and to change the appearance of the navigation bar by editing the CSS style sheet for this element.

Module 2

In module 2, students learn about the new features in HTML5 and CSS3. In particular, lesson 1 describes the new HTML5 elements, such as using the **<nav>**, **<article>**, **<section>**, **<aside>**, and **<footer>** tags, to add semantic structure to an HTML document. It also covers markup for images and for text elements, such as **<hgroup>**, **<time>**, **<mark>**, **<small>**, **<figure>**, and **<figcaption>**. A demonstration provides you with a means to show how to use these elements in a document, and the effects that they have on the content. Note that this lesson does not cover the Canvas API, SVG, video, or audio; these items are described in later modules. Lesson 2 provides more information on how to apply CSS styles to the new HTML5 text elements. Make sure that students understand how the CSS Box Model works for styling content, as this information is used in many of the lab exercises throughout the course. The lab at the end of this module starts to build the conference web application, starting with the Home and About pages. The content for these pages is structured and styled by using the features described in this module.

Module 3

Module 3 describes how to use JavaScript to add dynamic functionality to a web page. Lesson 1 provides a quick tour through the core syntax of JavaScript. It is important that students understand the specific features of JavaScript syntax and semantics, so be prepared to take a little time explaining how JavaScript works. For example, you may need to carefully explain how variables are untyped when they are declared, and how the JavaScript interpreter uses type-inference at runtime to determine how to handle the contents of a variable. It is also important that students understand the syntax of JavaScript Object Notation (JSON), as this data format is used in many labs. The second lesson introduces the Document Object Model (DOM), and how to use the DOM to locate and modify elements at runtime by using JavaScript code. This lesson also shows how to catch and handle events; this is another key concept that students need to thoroughly understand before moving on. The final lesson is an overview of jQuery, showing how to access the DOM and handle events by using the jQuery library. In the lab for this module, students write JavaScript code to display the details of each session on the Session page in the web application. Students write code that populates the Session page from an array of JSON objects, and they implement a feature that enables the user to filter session information by conference track.

Module 4

In module 4, students learn about the new forms input features provided by HTML5. The first lesson contains an overview of the HTML5 input types and elements, together with the attributes that enable a browser to perform some limited types of input validation. Lesson 2 goes into more detail, with examples that show how to ensure that mandatory fields are not left empty, how to validate text and numeric data, and how to style fields that contain invalid data to provide feedback to the user. Lesson 3 shows how to implement extended and cross-field validation by using JavaScript code. In the lab, students create the Registration page for the web application, enabling the student to provide their details. All the information entered by a user is validated by using a combination of HTML5 input attributes and JavaScript code.

Module 5

Module 5 shows how to create web pages that can communicate with web services. The first lesson describes how to use the **XMLHttpRequest** object to send a request, and how to handle any response that is returned. This lesson illustrates how to serialize data as JSON, how to format a message, and how to handle the events that occur when a response message is received. This lesson also provides coverage of techniques that web applications can use to detect and handle common network exceptions. The second lesson concentrates on using the jQuery **ajax()** function to send and receive data, and how to serialize data by using jQuery. In the lab, students modify the JavaScript code for the Schedule page to receive information about sessions from a web service rather than by using an array of hard-coded JSON objects. Students add functionality to the page, enabling a user to indicate that they will attend specific sessions. This information is sent to the web service, which may respond with a message advising attendees to arrive early if a specific session is very popular.

Module 6

Module 6 goes into more detail on using CSS to style text and graphics. This module contains four short lessons. The first lesson describes how to apply fonts to text, and the different typographic measurements that CSS can use to specify the size of fonts and text elements. The second lesson summarizes how to use CSS to style block elements, covering the new block properties available in CSS3 and the block layout models that CSS implements. The third lesson describes how to use pseudo-classes and pseudo-elements to select text elements and modify the styling of these elements dynamically. The final lesson shows how to implement simple graphical effects and create shapes by using CSS styles. In the lab, students create

styles to modify the typography of the text on the About page. Students also use styles to implement graphical effects for the navigation bar.

Module 7

Module 7 provides more information on JavaScript, concentrating on techniques and best practices for writing well-structured, maintainable code. Lesson 1 covers variable scope and hoisting in JavaScript; these are concepts that may be unfamiliar to developers coming from a C++ or C# background, so take time to ensure that students fully understand them. Additionally, lesson 1 describes global objects and the Global namespace, and how it acts as the repository for global functions and variables. Lesson 2 describes the model that JavaScript implements for creating custom objects. Again, C++ and C# programmers may be familiar with the idea of constructors, but the way in which they work in JavaScript is different, so take time to explain them fully, along with prototypes. The final lesson describes how to extend objects by using an inheritance mechanism. Once more, the concepts may be familiar to C++ and C# programmers, but the details are different in JavaScript. The lab shows how to use JavaScript objects and inheritance to refine the code used by many of the web pages to make it more maintainable.

Module 8

Module 8 adds a little light relief to the proceedings, showing some of the new features that have been added to HTML5 to enable web pages to interact with the user and the operating system running the browser. The first lesson describes how to use the HTML5 File APIs to read data held in files on the computer running the browser, and how to implement drag-and-drop to enable the user to copy the contents of a file to a web page. The second lesson shows how to play and control video and audio content by using the new **<video>** and **<audio>** tags in HTML5. The third lesson covers the Geolocation API, to obtain information about the location of the user and customize the data displayed on the web page accordingly. The lab comprises three independent exercises, and these can be performed together at the end of the module or individually after each lesson. The first exercise uses drag-and-drop to enable the user to drag an image file onto a web page and to generate a badge containing this image. The second exercise adds a video player to the Home page, together with controls to enable the user to start, stop, pause, and resume the video. The final exercise uses the Geolocation API to obtain the user's geographic location and display the distance to the conference site on the Location page in the web application.

Module 9

In Module 9, students learn how to enable a web application to support offline operations. In the first lesson, students learn how to save and access data used by the web application on the computer running the browser. In particular, this lesson covers the Session Storage API and the Local Storage API. This lesson also provides an overview of the Indexed Database API. The second lesson describes how to use web application caching. The lesson shows how to configure the web application cache, and how to trigger updates to cached resources. In the lab, students add offline capabilities to the conference web application; they configure the web application cache to cache data for selected pages, and they use local storage to persist user data locally so that it can be used in different browser sessions.

Module 10

Module 10 describes how to use CSS styles to implement an adaptive user interface that can detect the form factor of the device running the browser, and to modify the layout of content accordingly. This module is reasonably straightforward; it focusses on the use of media types and media queries to determine device capabilities. In the lab, students add a print-friendly stylesheet that formats the contents of web pages when the user sends them to a printer; it removes elements such as the navigation header

that are not useful on printed matter, and organizes columnar data to make it easier to read on the printed page. In the second exercise, students add styles to cater for different form factors, such as those used by hand-held devices and smartphones.

Module 11

Module 11 concerns implementing high-resolution interactive graphics. This module features two technologies, Scalable Vector Graphics (SVG), and the Canvas API. In lesson 1, students learn how to use SVG to draw scalable graphics and text, and apply transformations to graphics to rotate, translate, scale, and skew SVG images. Students also learn how to use events to enable a user to interact with an SVG image. Lesson 2 focusses on the Canvas API, showing how to use this API to draw and transform shapes on a web page. Use the demonstrations for both of these lessons to illustrate the different effects that can be achieved by using SVG and the Canvas API. The lab contains two exercises. In the first exercise, students add an interactive venue map to the Location page of the conference web application. In the second exercise, students modify the Speaker-Badge page and use the Canvas API to add details to the badge that this page generates.

Module 12

In module 12, students see how to add interest to a web page by using CSS animations. Students learn how to implement transitions to modify the appearance of an element over time, as well as how to transform elements in 2D and 3D. The final lesson in this module describes how to implement key-frame animations by using CSS, enabling elements to morph, scale, and translate their appearance dynamically. In the lab, students add code and CSS styling to the Feedback page to animate the feedback form when the user submits it.

Module 13

Module 13 describes how to implement peer-to-peer communication between a web application and a web server by using web sockets. Students learn how to format the data retrieved from an HTML page and send it to a server by using a web socket. Students also learn how to receive messages from a web server over a web socket, and parse the messages to extract the data that they contain. In the lab, students implement functionality for the Live page of the web application. This page enables a user to submit questions to a presenter, and displays the response from the presenter when it arrives. The page also displays questions posted by other students, together with any answers to these questions.

Module 14

In module 14, students learn how to offload long-running operations to a web worker. This strategy enables a web page to remain responsive while the processing is performed by a separate task. The module describes how to create and control a web worker, and how to pass messages between a web worker and a web page. In the lab, students use a web worker to implement further processing for the speaker badge page. The images on badges are rendered as grayscale, but the image file that the user drops on the page might be color. Therefore, the JavaScript code for the page may need to convert the color image into a grayscale representation. This process requires examining the image pixel by pixel, and may take a considerable time if the image is large. To ensure that the web page remains responsive while this is happening, the grayscale processing is performed asynchronously by using a web worker.

IMPORTANT

Students will use 20480B-SEA-DEV11 virtual machine to perform all of the labs, and you will use this same virtual machine to do the demonstrations that occur in many of the modules. Internet Explorer has been configured on this virtual machine to automatically delete browsing history on exit. Additionally, all


caching has been disabled by default. The reason for this is to prevent Internet Explorer from caching old versions of a web page and JavaScript code when students perform the labs, which can lead to confusion: If caching is enabled, students won't always see the most recent version of a web page when they run the application code. Lab 9 requires that students enable caching, and the lab includes instructions for configuring Internet Explorer. However, make sure that students follow the instructions at the end of this lab and disable caching again, or lab 10 may not work correctly.

Additionally, if students manage to misconfigure the virtual machine, inform them that they can revert to the snapshot that was created when the classroom environment was set up. However, students will lose their code from previous labs if they do this, so you could also show students how to take a snapshot of the virtual machine at the start of each lab, enabling them to roll back only the changes made during that lab.

Required Materials to Teach This Course

You need the following materials to teach this course:

- Course Handbook.
- Course Companion Content at <http://www.microsoft.com/learning/companionmoc/>.
- Microsoft PowerPoint® files.
- OneNote® Trainer Pack (OTP) on [MCT Download Center](#).
- Microsoft Virtual Server Classroom Setup Guide.
- Course virtual machines.

 **Important** It is recommended that you use PowerPoint 2003 or later to display the slides for this course. If you use PowerPoint Viewer or an earlier version of PowerPoint, all the features of the slides might not display correctly.

Prerequisite Knowledge to Teach This Course

To present this course, you must have the following knowledge and skills:

- At least 12 months of experience building web applications by using HTML, CSS, and JavaScript by using Visual Studio 2010 or 2012.
- Practical experience with the following HTML5 and CSS APIs and features:
 - Web sockets
 - Web workers
 - Canvas API
 - Web application cache
 - File API
 - Local Storage API
 - Session Storage API
 - Geolocation API
 - SVG

- Video and audio
- CSS transforms, transitions, and key-frame animations

Preparation Tasks

Complete the following tasks to prepare for this course.

Exams

To identify your technical proficiency with the content of this course, it is highly recommended that you pass the following exam:

- Exam 70-480: Programming in HTML5 with JavaScript and CSS3.

Technical Preparation Activities

It is highly recommended that you complete the following technical preparation activities:

- Read the additional readings and references that are included in the Course Companion Content at <http://www.microsoft.com/learning/companionmoc>.
- Use the OneNote Trainer Pack (OTP) on [MCT Download Center](#) to prepare for delivering the course.
 - The OneNote Trainer Packs include the following content for each course on each page in the OTP (and in this order):
 - Slides
 - Instructor Notes
 - Student Handbook Content
 - Also included for each module are the Lab Answer Keys (LAKs).
- Practice setting up the classroom by following the instructions in the Microsoft Hyper-V™ Classroom Setup Guide.
- Review the learning product error log, which is available on the MCT Download Center.
- Practice using the Microsoft product(s) and tools that are associated with this learning product:
 - Visual Studio 2012
 - Internet Explorer 10, including the F12 Developer Tools
- Review the Microsoft product error log, which is available in the Microsoft Download Center.

Instructional Preparation Activities

It is highly recommended that you complete the following instructional preparation activities:

- Read the Portfolio Design Guide for the broader context of the portfolio of which this course is a part. The Portfolio Design Guide is located on the MCT Member Site at <https://mcp.microsoft.com/mct/resources/ResourceCenter.mspix>.
- Read the About This Course at the beginning of the Course handbook for the learning product.
- Walk through the Introduction slide deck for the learning product.
- Walk through each module presentation slide deck and read the corresponding Instructor Notes (located in the notes view of the presentation slide deck) for the module. Note that additional hidden

slides are used in each slide deck to accommodate the amount of Instructor Notes information for a given topic.

- Familiarize yourself with the Course Handbook and the Course Companion Content on the <http://www.microsoft.com/learning/companionmoc> site. Make note of when to direct students' attention to the Course Companion Content for further learning support. More information pertaining to the course components is present in the Introduction slide deck.
- Practice presenting each module:
- Identify the key points and must-know information for each topic.
- Perform each demonstration and hands-on lab.
- Anticipate the questions that students might have.
- Identify examples, analogies, impromptu demonstrations, and additional delivery tips that will help to clarify module content and provide a more meaningful learning experience for your specific audience.
- Note any problems that you might encounter during a demonstration or a lab exercise, and determine a course of action for how you will resolve the problems in the classroom. To access the lab answer keys, refer to the appendix in the Course Handbook.
- Work through the Module Review and Takeaways section at the end of each module and determine how you will use this section to reinforce student learning and promote knowledge transfer to on-the-job performance.
- Customize and enhance your instructor notes.
- Consult the MCT newsgroup for additional tips and strategies posted by your fellow MCTs for teaching the learning product.
- Review the updated information about the Microsoft Certification Program on the Microsoft Learning Certifications website.

Instructor Computer Setup

Set up the instructor computer by following the setup instructions in the "Microsoft Virtual Server Classroom Setup Guide" document. This document provides hardware requirements for the instructor computer, in addition to detailed setup instructions.

Course Timing

The following schedule is an estimate of the course timing. Your timing might vary. Every student might not finish every lab. Use your judgment to set a reasonable time to move on to the next module.

Day 1

Start	End	Module
9:00	9:30	Introduction
9:30	10:35	Module 1: Overview of HTML and CSS
10:35	10:50	Break
10:50	11:20	Lab: Exploring the Contoso Conference Application
11:20	12:25	Module 2: Creating and Styling HTML Pages
12:25	13:25	Lunch
13:25	14:10	Lab: Creating and Styling HTML5 Pages
14:10	14:30	Module 3: Introduction to JavaScript
14:30	14:45	Break
14:45	15:45	Module 3: Introduction to JavaScript (<i>continued</i>)
15:45	16:45	Lab: Displaying Data and Handling Events by Using JavaScript

Day 2

Start	End	Module
9:00	9:30	Day 1 review
9:30	10:30	Module 4: Creating Forms to Collect Data and Validate User Input
10:30	10:45	Break
10:45	11:15	Lab: Creating a Form and Validating User Input
11:15	12:15	Module 5: Communicating with a Remote Server
12:15	13:15	Lunch
13:15	14:15	Lab: Communicating with a Remote Server
14:15	14:45	Module 6: Styling HTML5 by Using CSS3
14:45	15:00	Break
15:00	15:45	Module 6: Styling HTML5 by Using CSS3 (<i>continued</i>)
15:45	16:45	Lab: Styling Text and Block Elements by Using CSS3

Day 3

Start	End	Module
9:00	9:30	Day 2 review
9:30	10:45	Module 7: Creating Objects and Methods by Using JavaScript
10:45	11:00	Break
11:00	12:00	Lab: Refining Code for Maintainability and Extensibility
12:00	12:30	Module 8: Creating Interactive Pages by Using HTML5 APIs
12:30	13:30	Lunch
13:30	14:10	Module 8: Creating Interactive Pages by Using HTML5 APIs (<i>continued</i>)
14:10	14:40	Lab: Creating Interactive Pages with HTML5 APIs
14:40	14:55	Break
14:55	15:25	Lab: Creating Interactive Pages with HTML5 APIs (<i>continued</i>)
15:25	16:25	Module 9: Adding Offline Support to Web Applications

Day 4

Start	End	Module
9:00	9:30	Day 3 review
9:30	10:30	Lab: Adding Offline Support to Web Applications
10:30	10:45	Break
10:45	11:45	Module 10: Implementing an Adaptive User Interface
11:45	12:15	Lab: Implementing an Adaptive User Interface
12:15	13:15	Lunch
13:15	13:45	Lab: Implementing an Adaptive User Interface (<i>continued</i>)
13:45	14:30	Module 11: Creating Advanced Graphics
14:30	14:45	Break
14:45	15:20	Module 11: Creating Advanced Graphics (<i>continued</i>)
15:20	16:20	Lab: Creating Advanced Graphics

Day 5

Start	End	Module
9:00	9:15	Day 4 review
9:15	10:15	Module 12: Animating the User Interface
19:15	10:30	Break
10:30	10:50	Module 12: Animating the User Interface (<i>continued</i>)
10:50	11:50	Lab: Animating the User Interface
11:50	12:35	Module 13: Implementing Real-Time Communication by Using Web Sockets
12:35	13:35	Lunch
13:35	15:05	Lab: Performing Real-Time Communication by Using Web Sockets
15:05	16:05	Module 14: Performing Background Processing by Using Web Workers
16:05	17:05	Lab: Creating a Web Worker Process