Netec Digital
Certificaciones Profesionales enTI

CURSO
OD20480C Programming in HTML5 with JavaScript and CSS3

Microsoft Partner

Gold Learning

Course Outline

Module 1: Overview of HTML and CSS

Most modern web applications are built upon a foundation of HTML pages that describe the content that users read and interact with, style sheets to make that content visually pleasing, and JavaScript code to provide a level of interactivity between user and page, and page and server. The web browser uses the HTML markup and the style sheets to render this content, and runs the JavaScript code to implement the behavior of the application. This module reviews the basics of HTML and CSS, and introduces the tools that this course uses to create HTML pages and style sheets.

Lessons
- Overview of HTML
- Overview of CSS
- Creating a Web Application by Using Visual Studio 2017

Lab : Exploring the Contoso Conference Application
- Exploring the Contoso Conference Application
- Examining and Modifying the Contoso Conference Application

After completing this module, students will be able to:
- Explain how to use HTML elements and attributes to lay out a web page.
- Explain how to use CSS to apply basic styling to a web page.
- Describe the tools that Microsoft Visual Studio provides for building web applications.

Module 2: Creating and Styling HTML Pages

The technologies forming the basis of all web applications—HTML, CSS, and JavaScript—have been available for many years, but the purpose and sophistication of web applications have changed significantly. HTML5 is the first major revision of HTML in 10 years, and it provides a highly suitable means of presenting content for traditional web applications, applications running on handheld mobile devices, and also on the Windows 10 platform. This module introduces HTML5, describes its new features, demonstrates how to present content by using the new features in HTML5, and how to style this content by using CSS.

Lessons
- Creating an HTML5 Page
- Styling an HTML5 Page

Lab : Creating and Styling HTML5 Pages
- Creating HTML5 Pages
- Styling HTML pages

After completing this module, students will be able to:
- Describe the purpose of and new features in HTML5, and explain how to use new HTML5 elements to lay out a web page.
- Explain how to use CSS to style the layout, text, and background of a web page.

Module 3: Introduction to JavaScript

HTML and CSS provide the structural, semantic, and presentation information for a web page. However, these technologies do not describe how the user interacts with a page by using a browser. To implement this functionality, all modern browsers include a JavaScript engine to support the use of scripts in a page. They also implement Document Object Model (DOM), a W3C standard that defines how a browser should reflect a page in memory to enable scripting engines to access and alter the contents of that page. This module introduces JavaScript programming and DOM.

Lessons
- Overview of JavaScript
- Introduction to the Document Object Model

Lab : Displaying Data and Handling Events by Using JavaScript.
- Displaying Data Programmatically
- Handling Events

After completing this module, students will be able to:
- Describe basic JavaScript syntax.
- Write JavaScript code that uses the DOM to alter and retrieve info from a web page.

Module 4: Creating Forms to Collect and Validate User Input

Web applications frequently need to gather user input in order to perform their tasks. A web page needs to be clear and concise about the input expected from a user in order to minimize frustrating misunderstandings about the information that the user should provide. Additionally, all input must be validated to ensure that it conforms to the requirements of the application. In this module, you will learn how to define input forms by using the new input types available in HTML5. You will also see how to validate data by using HTML5 attributes. Finally, you will learn how to perform extended input validation by using JavaScript code, and how to provide feedback to users when their input is not valid or does not match the application's expectations.

Lessons
- Creating HTML5 Forms
- Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

Lab : Creating a Form and Validating User Input
- Creating a Form and Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

After completing this module, students will be able to:
- Create input forms by using HTML5.
- Use HTML5 form attributes to validate data.
- Write JavaScript code to perform validation tasks that cannot easily be implemented by using HTML5 attributes.

Module 5: Communicating with a Remote Server

Many web applications require the use of data held by a remote site. In some cases, you can access this data simply by downloading it from a specified URL, but in other cases the data is encapsulated by the remote site and made accessible through a web service. In this module, you will learn how to access a web service by using JavaScript code and to incorporate remote data into your web applications. You will look at two technologies for achieving this: the XMLHttpRequest object, which acts as a programmatic wrapper around HTTP requests to remote web sites, and Fetch API, which simplifies many of the tasks involved in sending requests and receiving data. Because the Fetch API and the XMLHttpRequest object are asynchronous api You will first learn about how to handle asynchronous tasks with the Promise object, arrow functions and the new async/await syntax that lets you handle asynchronous request as if they were synchronous.

Lessons
- Async programming in JavaScript
- Sending and Receiving Data by Using the XMLHttpRequest Object
- Sending and Receiving Data by Using the Fetch API

Lab : Communicating with a Remote Data Source
- Retrieving Data
- Serializing and Transmitting Data
- Refactoring the Code by Using the jQuery ajax Method

After completing this module, students will be able to:
- Handle asynchronous JavaScript tasks using the new async programing technologies.
- Send data to a web service and receive data from a web service by using an XMLHttpRequest object.
- Send data to a web service and receive data from a web service by using the Fetch API.

Module 6: Styling HTML5 by Using CSS3

Styling the content displayed by a web page is an important aspect of making an application attractive and easy to use. CSS is the principal mechanism that web applications use to implement styling, and the features added to CSS3 support many of the new capabilities found in modern browsers. Where CSS1 and CSS2.1 were single documents, the World Wide Web Consortium has chosen to write CSS3 as a set of modules, each focusing on a single aspect of presentation such as color, text, box model, and animations. This allows the specifications to develop incrementally, along with their implementations. Each specification defines properties and values that already exist in CSS1 and CSS2, and also new properties and values. In this module, you will examine the properties and values defined in several of these modules, the new selectors defined in CSS3, and the use of pseudo-classes and pseudo-elements to refine those selections.

Lessons
- Styling Text by Using CSS3
- Styling Block Elements
- Pseudo-Classes and Pseudo-Elements
- Enhancing Graphical Effects by Using CSS3

Lab : Styling Text and Block Elements by Using CSS3
- Styling the Navigation Bar
- Styling the Register Link
- Styling the About Page

After completing this module, students will be able to:
- Use the new features of CSS3 to style text elements.
- Use the new features of CSS3 to style block elements.
- Use CSS3 selectors, pseudo-classes, and pseudo-elements to refine the styling of elements.
- Enhance pages by using CSS3 graphical effects.

Module 7: Creating Objects and Methods by Using JavaScript

Code reuse and ease of maintenance are key objectives of writing well-structured applications. If you can meet these objectives, you will reduce the costs associated with writing and maintaining your code. This module describes how to write well-structured JavaScript code by using language features such as namespaces, objects, encapsulation, and inheritance. These concepts might seem familiar if you have experience in a language such as Java or C#, but the JavaScript approach is quite different and there are many subtleties that you must understand if you want to write maintainable code.

Lessons
- Writing Well-Structured JavaScript Code
- Creating Custom Objects
- Extending Objects

Lab : Refining Code for Maintainability and Extensibility
- Object Inheritance
- Refactoring JavaScript Code to Use Objects

After this module, students will be able to:
- Write well-structured JavaScript code.
- Use JavaScript code to create custom objects.
- Implement object-oriented techniques by using JavaScript idioms.

Module 8: Creating Interactive Pages by Using HTML5 APIs

Interactivity is a key aspect of modern web applications, enabling you to build compelling web sites that can quickly respond to the actions of the user, and also adapt themselves to the user's location. This module describes how to create interactive HTML5 web applications that can access the local file system, enable the user to drag-and-drop data onto elements in a web page, play multimedia files, and obtain geolocation information.

Lessons
- Interacting with Files
- Incorporating Multimedia
- Reacting to Browser Location and Context
- Debugging and Profiling a Web Application

Lab : Creating Interactive Pages with HTML5 APIs
- Dragging and Dropping Images
- Incorporating Video
- Using the Geolocation API to Report the User's Current Location

After completing this module, students will be able to:
- Access the local file system, and add drag-and-drop support to web pages.
- Play video and audio files in a web page, without the need for plugins.
- Obtain information about the current location of the user.
- Use the F12 Developer Tools in Microsoft Edge to debug and profile a web application.

Module 9: Adding Offline Support to Web Applications

Web applications have a dependency on being able to connect to a network to fetch web pages and data. However, in some environments a network connection may be intermittent. In these situations, it might be useful to enable the application to continue functioning by using data cached on the user's device. HTML5 provides a choice of new client-side storage options, including session storage and local storage, and a resource caching mechanism called the Application Cache. In this module, you will learn how to use these technologies to create robust web applications that can continue running even when a network connection is unavailable.

Lessons
- Reading and Writing Data Locally
- Adding Offline Support by Using the Application Cache

Lab : Adding Offline Support to Web Applications
- Caching Offline Data by Using the Application Cache API
- Persisting User Data by Using the Local Storage API

After completing this module, students will be able to:
- Save data locally on the user's device, and access this data from a web application.
- Configure a web application to support offline operations by using the Application Cache.

Module 10: Implementing an Adaptive User Interface

One of the most enduring features of the web is its temporary nature. For the first time, the monopoly of the keyboard and mouse is coming under challenge, and that means questioning how user interfaces are designed. You may develop a web application on a computer with a large, high-resolution monitor, a mouse, and a keyboard, but other users might view and interact with your application on a smartphone or a tablet without a mouse, or have a monitor with a different resolution. Users may also want to print pages of your application. In this module, you will learn how to build a website that adapts the layout and functionality of its pages to the capabilities and form factor of the device on which it is being viewed. You will see how to detect the type of device being used to view a page, and learn strategies for laying out content that effectively targets particular devices.

Lessons
- Supporting Multiple Form Factors
- Creating an Adaptive User Interface

Lab : Implementing an Adaptive User Interface
- Creating a Print-Friendly Style Sheet
- Adapting Page Layout to Fit Different Form Factors

After completing this module, students will be able to:
- Describe the requirements in a website for responding to different form factors.
- Create web pages that can adapt their layout to match the form factor of the device on which they are displayed.

Module 11: Creating Advanced Graphics

High-resolution, interactive graphics are a key part of most modern applications. Graphics can help to enhance the user's experience by providing a visual aspect to the content, making a website more attractive and easier to use. Interactivity enables the graphical elements in a website to adapt and respond to user input or changes to the environment, and is another important element in retaining the attention of the user and their interest in the content. This module describes how to create advanced graphics in HTML5 by using Scalable Vector Graphics (SVG) and the Microsoft Canvas API. You will learn how to use SVG-related elements such as , , and to display graphical content on a web page. You will also learn how to enable the user to interact with SVG elements through the use of events such as keyboard events and mouse events. The Canvas API is somewhat different than SVG. The Canvas API provides a

Lessons
- Creating Interactive Graphics by Using SVG
- Drawing Graphics by Using the Canvas API

Lab : Creating Advanced Graphics
- Creating an Interactive Venue Map by Using SVG
- Creating a Speaker Badge by Using the Canvas API

After completing this module, students will be able to:
- Use SVG to create interactive graphical content.
- Use the Canvas API to generate graphical content programmatically.

Module 12: Animating the User Interface

Animations are a key element in maintaining the interest of a user in a website. Implemented carefully, animations improve the usability of a web page and provide useful visual feedback on user actions. This module describes how to enhance web pages by using CSS animations. You will learn how to apply transitions to property values. Transitions enable you to specify the timing of property changes. For example, you can specify that an element should change its width and height over a five-second period when the mouse pointer hovers over it. Next, you will learn how to apply 2D and 3D transformations to elements. Transformations enable you to scale, translate, rotate, and skew elements. You can also apply transitions to transformations, so that the transformation is applied gradually over a specified animation period. At the end of this module, you will learn how to apply keyframe animations to elements. Keyframe animations enable you to define a set of property values at specific moments during an animation. For example, you can specify the color and position of an element at 0 percent, 33 percent, 66 percent, and 100 percent of the animation period.

Lessons
- Applying CSS Transitions
- Transforming Elements
- Applying CSS Keyframe Animations

Lab : Animating the User Interface
- Applying CSS Transitions
- Applying Keyframe Animations

After completing this module, students will be able to:
- Apply transitions to animate property values to HTML elements.
- Apply 2D and 3D transformations to HTML elements.
- Apply keyframe animations to HTML elements.

Module 13: Implementing Real-time Communication by Using Web Sockets

Web pages request data on demand from a web server by submitting HTTP requests. This model is ideal for building interactive applications, where the functionality is driven by the actions of a user. However, in an application that needs to display constantly changing information, this mechanism is less suitable. For example, a financial stocks page is worthless if it shows prices that are even a few minutes old, and you cannot expect a user to constantly refresh the page displayed in the browser. This is where web sockets are useful. The Web Sockets API provides a mechanism for implementing real-time, two-way communication between web server and browser. This module introduces web sockets, describes how they work, and explains how to create a web socket connection that can be used to transmit data in real time between a web page and a web server.

Lessons
- Introduction to Web Sockets
- Using the WebSocket API

Lab : Performing Real-time Communication by Using Web Sockets
- Receiving Messages from a Web Socket
- Sending Messages to a Web Socket
- Handling Different Web Socket Message Types

After completing this module, students will be able to:
- Describe how using web sockets helps to enable real-time communications between a web page and a web server.
- Use the Web Sockets API to connect to a web server from a web page, and exchange messages between the web page and the web server.

Module 14: Performing Background Processing by Using Web Workers

JavaScript code is a powerful tool for implementing functionality in a web page, but you need to remember that this code runs either when a web page loads or in response to user actions while the web page is being displayed. The code is run by the browser, and if the code performs operations that take a significant time to complete, the browser can become unresponsive and degrade the user's experience. HTML5 introduces web workers, which enable you to offload processing to separate background threads and thus enable the browser to remain responsive. This module describes how web workers operate and how you can use them in your web applications.

Lessons
- Understanding Web Workers
- Performing Asynchronous Processing by Using Web Workers

Lab : Creating a Web Worker Process
- Improving Responsiveness by Using a Web Worker

After completing this module, students will be able to:
- Explain how web workers can be used to implement multithreading and improve the responsiveness of a web application.
- Perform processing by using a web worker, communicate with a web worker, and control a web worker.

Module 15: Packaging JavaScript for Production Deployment

Using models allows you to build large, complex applications. The progress of the language in the version of ECMAScript6 allows the build app with to simplify the application construction process. However, the use of ECMAScript6 modules and other features is not yet supported in all browsers. Tools such as Node.js, Webpack, and Babel enable the use of new language features along with support for different browsers in order to avoid harming the user experience. In this module we will introduce the theory behind these tools, when we need to used them, and the different options for use. At the end of the module we will see how to use these tools to write ECMAScript6 code supported in all browsers.

Lessons
- Understanding Transpilers And Module bundling
- Creating Separate Packages for Cross Browser Support

Lab : Setting Up Webpack Bundle for Production
- Creating and Deploying Packages using WebPack

Para más información, contáctenos al correo:
informes@netecdigital.com

www.netecdigital.com