



Microsoft 70-480 Programming in HTML5 with JavaScript and CSS3

Introduction

The 70-480 certification focuses on using HTML5/CSS3/JavaScript to implement programming logic, define and use variables, perform looping and branching, develop user interfaces, capture and validate user input, store data, and create well-structured application.


After completing this students will be able to:

- Describe the new features of HTML5, and create and style HTML5 pages.
- Add interactivity to an HTML5 page by using JavaScript.
- Create HTML5 forms by using different input types, and validate user input by using HTML5 attributes and JavaScript code.
- Send and receive data to and from a remote data source using XMLHttpRequest objects and jQuery AJAX
- Style HTML5 pages by using CSS3.
- Create well-structured and easily-maintainable JavaScript code.
- Use common HTML5 APIs in interactive Web applications.
- Create Web applications that support offline operations.
- Create HTML5 Web pages that can adapt to different devices and form factors.
- Add advanced graphics to an HTML5 page by using Canvas elements, and by using and Scalable Vector Graphics.
- Enhance the user experience by adding animations to an HTML5 page.
- Use Web Sockets to send and receive data between a Web application and a server.
- Improve the responsiveness of a Web application that performs long-running operations by using Web Worker processes.



Module 1: Overview of HTML and CSS
Module 2: Creating and Styling HTML5 Pages
Module 3: Introduction to JavaScript
Module 4: Creating Forms to Collect Data and Validate User Input
Module 5: Communicating with a Remote Data Source
Module 6: Styling HTML5 by Using CSS3
Module 7: Creating Objects and Methods by Using JavaScript
Module 8: Creating Interactive Pages using HTML5 APIs
Module 9: Adding Offline Support to Web Applications
Module 10: Implementing an Adaptive User Interface
Module 11: Creating Advanced Graphics
Module 12: Animating the User Interface
Module 13: Implementing Real-Time Communications by Using Web Sockets
Module 14: Creating a Web Worker Process

Exam Questions and Preparation

	<p>A free and as an official Microsoft learning course, it's safe to assume the information it gives is very reliable and accurate to the exam expectation</p> <p>https://mva.microsoft.com/en-US/training-courses/developing-in-html5-with-javascript-and-css3-jump-start-8223?l=MyzyHlly_504984382</p>
<p>Briefmenow</p>	<p>Navigate the 197 questions on the right menu.</p> <p>http://www.briefmenow.org/microsoft/which-expression-should-you-insert-into-each-function-3/</p>
<p>Pluralsight</p>	<p>https://www.pluralsight.com/blog/software-development/learning-path-programming-in-html5-with-javascript-and-css3-microsoft-exam-70-480</p>
<p>Microsoft</p>	<p>https://www.microsoft.com/en-us/learning/exam-70-480.aspx</p> <p>http://www.microsoft.com/learning/en/us/exam.aspx?ID=70-480#tab2</p>
<p>Exam Labs</p>	<p>https://www.exam-labs.com/exam/70-480</p> <p>https://www.exam-labs.com/exam/70-480</p>
	<p>http://geekswithblogs.net/WTFNext/archive/2012/10/08/exam-70-480-study-material-programming-in-html5-with-javascript-and.aspx</p>

Module 1: Overview of HTML and CSS

This module provides an overview of HTML and CSS, and describes how to use Visual Studio 2012 to build a Web application. **Lessons**

- Overview of HTML
- Overview of CSS
- Creating a Web Application by Using Visual Studio 2012

HTML5 introduces new semantic elements to clearly define sections within an HTML document. Semantic elements provide mechanisms to improve accessibility via screen readers. Search engines take advantage of HTML5 by leveraging the <article> element.

Pg3-21

HTML5
<article>
<aside>
<figcaption>
<figure>
<footer>
<header>
<hgroup>
<mark>
<nav>
<progress>
<section>

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

Sample Questions	
Questions	
Which of the following elements are not HTML5? A <article> B <footer> C <hgroup> D <input>	D
Which element(s) does the <hgroup> element contain? A <h1> to <h6> B <header> C <nav> D all of the above	A
Which HTML5 element would you use to organise content so that the page maximises a search engines algorithm? A <div id="Company News"> B <header>Company News</header> C<article> Company News</article> D All of the above	C Header element is within the section element The article element tells the search engine the content within it is what users are searching for. It is the best element for SEO
Which HTML5 elements should be used to create a structured layout A <div> B <p> C <table> D <form>	C Table element

Module 2: Creating and Styling HTML5 Pages

This module describes the new features of HTML5, and explains how to create and style HTML5 pages. **Lessons**

- Creating an HTML5 Page
- Styling an HTML5 Page

Use CSS3 in Applications 25%

Style HTML text properties

This objective may include but is not limited to: apply styles to text appearance (color, bold, italics); apply styles to text font (WOOF and @font-face, size, understudy fonts); apply styles to text alignment, spacing, and indentation; apply styles to text hyphenation; apply styles for a text drop shadow

- [Styling Text](#) (W3Schools)
Covers “apply styles to text appearance (color, bold, italics)” and “apply styles to text alignment”
- [Styling Fonts](#) (W3Schools) and [CSS3 Fonts](#) (W3Schools)
Covers “apply styles to text font (WOOF and @font-face, size, understudy fonts)”
- [Styling Links](#) (W3Schools)
Covers “apply styles to text hyphenation”
- [CSS3 Text Effects](#) (W3Schools)
Covers “apply styles to text appearance (color, bold, italics)” and “apply styles for a text drop shadow”
- [CSS Selectors](#) (W3Schools)
Covers “applies to all topics indirectly”
- [CSS3 Text Effects](#) (W3Schools)
Covers “apply styles to alter graphic effects (shadow)”
- [CSS3 Gradient Backgrounds](#) (W3Schools)

Covers “*apply styles to alter graphic effects (gradients)*”

Style HTML box properties

This objective may include but is not limited to: apply styles to alter appearance attributes (size, border and rounding border corners, outline, padding, margin); apply styles to alter graphic effects (transparency, opacity, background image, gradients, shadow, clipping); apply styles to establish and change an element’s position (static, relative, absolute, fixed)

- [CSS3 Borders](#) (W3Schools) & [CSS Border](#) (W3Schools)
Covers “*apply styles to alter appearance attributes (border and rounding border corners)*”
- [CSS Box Model](#) (W3Schools)
Covers “*apply styles to alter appearance attributes (size, border and rounding border corners, outline, padding, margin)*”
- [CSS Dimension](#) (W3Schools)
Covers “*apply styles to alter appearance attributes (size ...)*”
- [CSS Outline](#) (W3Schools), [CSS Margin](#) (W3Schools), & [CSS Padding](#) (W3Schools)
Covers “*apply styles to alter appearance attributes (outline, padding, margin)*”
- [CSS3 Backgrounds](#) (W3Schools)
Covers “*apply styles to alter graphic effects (background image)*”
- [background-clip](#) (W3Schools)
Covers “*apply styles to alter graphic effects (clipping)*”
- [CSS Image Opacity](#) (W3Schools)
Covers “*apply styles to alter graphic effects (transparency, opacity)*”

Module 3: Introduction to JavaScript

This module provides an introduction to the JavaScript language, and shows how to use JavaScript to add interactivity to HTML5 pages. **Lessons**

- Overview of JavaScript Syntax
- Programming the HTML DOM with JavaScript
- Introduction to jQuery

Implement Program Flow 25%

This objective may include but is not limited to: iterate across collections and array items; manage program decisions by using switch statements, if/then, and operators; evaluate expressions

- [JavaScript Array Object](#) (W3Schools) & [JavaScript For...In Statement](#) (W3Schools)
Covers “iterate across collections and array items”
- [JS Switch](#) (W3Schools)
Covers “manage program decisions by using switch statements”
- [JS If...Else](#) (W3Schools)
Covers “if/then, and operators”
- [JavaScript search\(\) Method](#) (W3Schools)
Covers “evaluate expressions”

Raise and handle an event

This objective may include but is not limited to: handle common events exposed by DOM (OnBlur, OnFocus, OnClick); declare and handle

bubbled events; handle an event by using an anonymous function

- [HTML DOM Events](#) (W3Schools), [onclick](#) (W3Schools), [onblur](#) (W3Schools), [onfocus](#) (W3Schools), etc
Covers “handle common events exposed by DOM (OnBlur, OnFocus, OnClick)”
- [bubbles Event Attribute](#) (W3Schools) & [Event Bubbling and Event Propagation: Demo](#) (Other: javascripter.net)
Covers “declare and handle bubbled events”
- [Advanced event registration models](#) (Other: quirksmode.org)
Covers “handle an event by using an anonymous function”

Implement exception handling

This objective may include but is not limited to: set and respond to error codes; throw an exception; request for null checks; implement try-catch-finally blocks

- [JS Try...Catch](#) (W3Schools), & [JS Throw](#) (W3Schools)
Covers sections: “throw an exception” and “implement try-catch-finally blocks”

[JavaScript undefined vs. null](#) (Other: saladwithsteve.com)

Covers “request for null checks”

Implement and Manipulate Document Structures and Objects 24%

Create the document structure.

This objective may include but is not limited to: structure the UI by using semantic markup, including for search engines and screen readers (Section, Article, Nav, Header, Footer, and Aside); create a layout container in HTML

- [HTML5 Tag Reference](#) (W3Schools), [<section>](#) (W3Schools), [<area>](#) (W3Schools), [<article>](#) (W3Schools), [<header>](#) (W3Schools), [<aside>](#) (W3Schools)

The entirety of these links will help you with the section: “structure the UI by using semantic markup, including for search engines and screen readers (Section, Article, Nav, Header, Footer, and Aside)”

[HTML Layouts](#) (W3Schools), and [Document Outlines](#) (Other: HTML Doctors)

Covers “create a layout container in HTML”

Find elements by using CSS selectors and JQuery

This objective may include but is not limited to: choose the correct selector to reference an element; define element, style, and attribute selectors; find elements by using pseudo-elements and pseudo-classes (for example, :before, :first-line, :first-letter, :target, :lang, :checked, :first-child)

- [CSS Selector Reference](#) (W3Schools) and [Selectors – jQuery API](#) (jQuery)
Covers “choose the correct selector to reference an element”
- [Attribute selectors](#) (W3C), [ID selectors](#) (W3C), [Type selectors](#) (W3C), and [Selectors – jQuery API](#) (jQuery)
Covers “define element, style, and attribute selectors”
- [Pseudo-elements and pseudo-classes](#) (W3C) and [Selectors – jQuery API](#) (jQuery)
Covers “find elements by using pseudo-elements and pseudo-classes (for example, :before, :first-line, :first-letter, :target, :lang, :checked, :first-child)”

Manipulating the DOM

Pg 22

The DOM is a representation of the structure of the HTML page that you can interact with programmatically. Behind the scenes the browser constructs the DOM.

Using Javascript you can access HTML elements to manipulate the DOM in the browser

Selecting Items in the DOM

getElementById	Get SINGLE element with unique ID
getElementbyClassName	Get ALL elements with specified CLASS
getElementByTagName	Get ALL elements with TAG or ELEMENT name
querySelector	Get FIRST child element that matches
querySelectorAll	Get ALL child elements that matches



Having access to the DOM and retrieving elements is so that you can then programmatically

Altering the DOM

Create a new element (P29)

```
var element = document.createElement("article");  
element.innerText = "My new article element";
```

Add an element

Append element

```
outerDiv.appendChild(element);
```

```
var element = document.getElementById("outerDiv").appendChild(document.createElement("article"));  
element.innerText = "My new article element";
```

Remove or Replace an element

```
var innerDiv = document.getElementById("InnerDiv");  
var p = innerDiv.removeChild(document.getElementById("P1");
```

```
var innerDiv = document.getElementById("InnerDiv");  
innerDiv.removeNode(true);
```

*note - all child nodes also removed

```
replaceNode()
```

```
replaceChild()
```

Properties available on a DOM element

childNodes	Collection of ALL child nodes of the parent element
firstChild	Reference to the first child node in a list of child nodes
lastChild	Reference to the last child node in a list of child nodes
hasChildNodes	Useful property to determine if a parent element has child nodes



Objective: Insert the article element as the first child element in the innerDiv element
<pre>var inner = document.getElementById("innerDiv") var element = inner.insertBefore(document.createElement("article"),inner.firstChild); element.innerText = "My new article element";</pre>
*Some elements do not support child nodes eg for example and the interpreter will throw a run time error.

Questions	
<p>Which of the following Javascript methods can't be used to select an element in the DOM?</p> <p>A getElementById B querySelector C getElementByClassName D queryAll</p>	<p>D queryAll isn't available to search the DOM –</p> <p>It is NOT a valid method ! querySelectorAll is</p>
<p>Which line of Javascript successfully retrieves only the image element with the ID myDog from the HTM? Choose all that apply.</p> <pre><form> <div id="main" class="mainStyle"> <p id="dogs">This is a page about dogs </p> </div> </form></pre> <p>A document.getElementById("myDog"); B <p>getChildNodes("img"); C document.getElementById("dogs").querySelector("thumb");</p>	<p>A</p>

D document.querySelectorAll("thumb");	
<p>To hide an element in the DOM and still be able to add it back later what should you do?</p> <p>A Nothing because the DOM is always available in static form</p> <p>B Keep a reference to the removed node to add it back later</p> <p>C Call the document.restoreNodes method</p> <p>D You can't add an element back after it is removed</p>	<p>B</p> <p>You need to keep a reference to the removed node to add it back later</p> <p>No such method document.restoreNodes</p>

Scope and Lifetime of variables (Pg87)	
<p>Variables begin life with a declaration. You can declare and initialise (set a value) at the same time.</p> <p>var x=0, y=1, z=2;</p> <p>Until initialised a variable is not really "ALIVE" – it has a value of "UNDEFINED"</p> <p>Global : available throughout the webpage</p> <p>Local scope : available in a given context</p>	
<p>In Javascript how do you determine the scope of a variable?</p> <p>A The scope of a variable is global within the context of the page</p> <p>B The scope of the variable depends on where inside the script it is declared</p> <p>C The scope of a variable changes depending on the type it represents</p>	B
Why is it important to avoid creating custom JS objects in the global namespace?	B

<p>A The global namespace is reserved for the browser</p> <p>B The global namespace is available to all so naming conflicts could result</p> <p>C The global namespace creates a security risk to users systems</p>	
<p>What Javascript keyword in an event handler can be easily used to reference the object that raised the event?</p> <p>A The it keyword</p> <p>B The document.current property</p> <p>C The this keyword</p> <p>D None of the above</p>	C

Program Flow (Pg111-156)
<p>Implement program flow</p> <p>SELECTION IF</p> <pre>If(condition){ //true logic }else{ //false logic }</pre> <p>== ignores underlying datatype === considers datatype (stricter and more accurate)</p> <p>TERNARY : <expression?<true part>: <false part></p> <p>ITERATION</p> <pre>for(<counter>;<expression>;<counter increment>) { //code }</pre>

ARRAYS

Arrays are dynamically sized

`indexOf(what to search for, index no. where to begin searching)` `var index = sports.indexOf('football',0);`

`lastIndexOf()` to be used if searching from the end of the array

`.join()` joins elements in an array into a single string

`.concat()` combines 2 or more arrays

`.sort()`

`.slice()`

`.splice()`

`.push()` //add item to end of the array

`.pop()` //remove last item in array

`.shift()` //removes first item in array

`.unshift()` //adds item to start of array

Raise and handle an event

Implement exception handling

Implement a callback

Create a webworker process

Which of the following keywords provide iterative control flow?

A if statement

B switch statement

C for

D break

C

Which of the following array methods combines 2 arrays?

D

A join B combine C split D concat	
Which keyword is used to exit a loop? A continue B break C stop D next	B

Raising and Handling Events (pg 131)

Events

Focus : focus, blur, focusin, focusout

Keyboard : keydown, keyup, keypress,

Mouse : click, dblclick, mousedown, mouseup, mouseover, mouseenter, mouseleave, mousemove

Drag and Drop (143)

Event Listeners

`addEventListener(eventname, function, optional cascade)`

`removeEventListener`

anonymous functions Pg135 – a function that does not have a name

Event Bubbling (Pg 137)

Which of the following isn't a supported way to add an event handler to a DOM?

A Declaring within the HTML element by assigning the event attribute to a JS function

B Setting the attribute in CSS to a valid JS function

C Dynamically through JS by assigning a JS function to the objects event property

D Dynamically through JS via the assign/remove event listener methods

B

Cannot assign event handlers
through CSS

All other options are valid

Which code would successfully cancel an event?

A `window.event.returnValue = false;`

B `return false;`

C `window.event.Return();`

D `window.Stop();`

A

Which event occurs when a DOM element receives the cursor?

A focus

B change

C keydown

D mouseleave

A

Which option provides the correct sequence of events in a drag and drop operation?

A

- A dragstart, drag, dragenter, drop
- B dragstart, drag, dragenter, drop
- C drag, dragstart, drop, dragenter
- D drag, dragstart, dragenter, dragstop

Exception Handling (pg149)

Try—catch—finally blocks

Checking for null values

```
try{
//code to try
} catch (e) {
//code to execute if error/exception eg. Alert
console.log(e.message)
}
finally{
//complete any final logic
}

var b= 10;
var c; //declared but NOT initialised
function multiply(b,c){
if (b == null || c == null){
throw new Error(5, "forgot to initialise number");
}
return b * c;
```

}	
<p>Which statement correctly describes proper error handling?</p> <p>A Proper error handling allows code execution to continue and provide suitable user feedback B Proper error handling allows users to fix problems with the webpage C Proper error handling allows you to debug the application at runtime D Proper error handling allows you to suppress all the bugs in your scripts</p>	A
<p>Which of the following is not a property of the exception object?</p> <p>A message B description C number D name</p>	B All the others are
<p>Why is checking for null a good practice?</p> <p>A Checking for null prevents the use of an object before it is initialised B Checking for null prevents errors resulting in NaN C Checking for null prevents the need to throw custom errors</p>	A

Implement a Callback (Pg156)

Anytime a function is called that expects a function as a parameter is called a callback. Knowing what parameters the callback function needs is important.

Module 4: Creating Forms to Collect Data and Validate User Input

This module describes the new input types available with HTML5, and explains how to create forms to collect and validate user input by using the new HTML5 attributes and JavaScript code. **Lessons**

- Overview of Forms and Input Types
- Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

Validate user input by using JavaScript

This objective may include but is not limited to: evaluate a regular expression to validate the input format; validate that you are getting the right kind of data type by using built-in functions; prevent code injection

- [KB: Validate an Email Address using Regular Expressions](#) (Other: consultsarath.com) & [jQuery Plugins/Validation](#) (jQuery)
Covers “evaluate a regular expression to validate the input format”
- [Checking Your Work: Validating Input — The Cool Stuff](#) (Other: htmlgoodies.com)
Covers sections: “evaluate a regular expression to validate the input format”, “validate that you are getting the right kind of data type by using built-in functions”, and “prevent code injection”
- [DOM Form](#) (W3Schools) and [Form submit\(\) Method](#) (W3Schools)
Covers “Form.Submit”

[decodeURI\(\)](#) (W3Schools), [decodeURIComponent\(\)](#) (W3Schools), [encodeURIComponent\(\)](#) (W3Schools), and [encodeURIComponent\(\)](#) (W3Schools)

Covers “sanitize input by using URI/form encoding”

Validate user input by using HTML5 elements

This objective may include but is not limited to: choose the appropriate controls based on requirements; implement HTML input types (for example, , ,) and content attributes (for example, required) to collect user input

- [HTML5 Input Type](#) (W3Schools)

Covers sections: “choose the appropriate controls based on requirements”, “implement HTML input types and content attributes to collect user input”

[HTML5 New Form Attributes](#) (W3Schools)

Covers “implement HTML ... content attributes to collect user input”

Input (Pg189)

Several HTML input controls have built in validation

<Input **type** = text, password, url, number, checkbox, radio, range, email

<Input type=“submit”, “reset”, “button”

To make controls required add required eg <input type=“email” placeholder=“me@mydomain.com” required/>

Which input control is better suited for allowing users to make multiple selections?

A radio button

B text area

C check box

D radio or checkbox

C

Which input control is designed to allow users to enter secure information in a way that keeps others from seeing what is typed?

D

<p>A text</p> <p>B textarea</p> <p>C url</p> <p>D password</p>	
<p>Which input control posts form data to a server?</p> <p>A button</p> <p>B Submit</p> <p>C Reset</p> <p>D Radio</p>	<p>B</p> <p>Button is generic and requires an event handler. Submit is the correct answer</p>
<p>Which of the following declarations are valid ways to make a text control non editable?</p> <p>A <input type="text" edit="false"/></p> <p>B <input type="text" editable="false"/></p> <p>C <input type="text" readonly="yes"/></p> <p>D <input type="text" readonly /></p>	<p>C</p>

Validate input (Pg211)

Ensuring data is validated

Evaluate regular expressions

Validate with built in functions

Prevent code injection

Regular Expressions (Pg211)

A regular expression is a mix of special characters and literal characters that make up a pattern that someone would want to match against. Regular expressions are a useful way of validating input of a specific pattern. eg. Postal codes, licence plates etc..

- ^ caret denotes beginning of a string
- \$ dollar denotes end of a string
- .
- period denotes to match on any character
- [A-Z] match any alphabet character (case sensitive) [a-z]
- \d match any numerical character
- + the preceding character or character set must match at least once
- * may or may not match
- ? preceding character is optional
- \w match a word character including underscore
- \ escape character
- \s match on a space

Regular expressions are objects in JS.

JS validation (Pg216)

```
If(isNaN(value)){  
  //handle the non number  
} else {  
  // proceed with the number  
}
```

<p>Which of the following regular expression characters denote the end of the string?</p> <p>A \$ B % C ^ D &</p>	<p>A</p>
<p>Which of the following sandbox attributes allows the iFrame to load content from the containing HTML document?</p> <p>A allow-script-execution B allow-same-origin C allow-forms D allow-top-navigation</p>	<p>D</p>

Consuming XML and JSON Data (Pg 220)	
<p>Consume data Serialise, Deserialise and transmit data</p> <p>The 2 data formats commonly used in data transmission are JSON and XML. JSON is unstructured, XML is structured.</p> <p>XMLHttpRequest object Methods: Send() Open() Properties: Status, Response</p>	
<p>Which of the following is a valid JSON string?</p> <p>A { fName, Rick, lName, Smith, hairColour, brown} B { fName: Rick; lName: Smith; hairColour: brown} C { fName, "Rick"; lName, "Smith"; hairColour: "brown"} D { fName: "Rick", lName: "Smith", hairColour: "brown"}</p>	D
<p>With the XMLHttpRequest object which of the following properties provides a response in a human readable format?</p> <p>A Response B responseBody C responseText D responseXML</p>	C
<p>At which stage during an XMLHttpRequest are user credentials sent to the server?</p> <p>A When the connection is opened B When the request is sent</p>	D

C When the ready state is complete D When the server sends a security response requesting credentials	
--	--

Serialise, Deserialise and Transmit Data (Pg 220)	
<p>Data can be received and sent in many forms</p> <p>Serialisation : Data is put into a format for transmission</p> <p>Deserialisation : Data is converted into something that can be worked with such as a custom object</p>	
<p>Which of the following creates an object from a JSON string stored in a variable called jsonString?</p> <p>A var o = JSON.split(jsonString)</p> <p>B var o = JSON.stringify(jsonString)</p> <p>C var o = JSON.parse(jsonString)</p> <p>D var o = JSON.join(jsonString)</p>	<p>C</p> <p>Stringify = serialise</p> <p>Parse = deserialise</p>
<p>How do you control what is sent to the server when submitting a form?</p> <p>A Add a submit button to the form</p> <p>B Handle the submit event of the form</p> <p>C Specify the action attribute of the form element</p> <p>D Ensure all elements on the form have a name</p>	<p>B</p>

Module 5: Communicating with a Remote Data Source

This module describes how to send and receive data to and from a remote data source by using an XMLHttpRequest object and by performing jQuery AJAX operations. **Lessons**

- Sending and Receiving Data by Using XMLHttpRequest
- Sending and Receiving Data by Using jQuery AJAX operations

Consume data

This objective may include but is not limited to: consume JSON and XML data; retrieve data by using web services; load data or get data from other sources by using XMLHttpRequest

- [XMLHttpRequest Level 2](#) (W3C)
Covers all topics
- [XML HTTP Request](#) (W3Schools) and [XML Parser](#) (W3Schools)
Covers all topics

Serialize, deserialize, and transmit data

This objective may include but is not limited to: binary data; text data (JSON, XML); implement the JQuery serialize method; Form.Submit; parse data; send data by using XMLHttpRequest; sanitize input by using URI/form encoding

- [3D models and parsing binary data with JavaScript](#) (Other: fhtr.blogspot.com)
The best I could find that covers “binary data”
- [JSON HOME](#) (W3Schools) (and all introductory sub-topics) and [XML HOME](#) (W3Schools) (and all introductory sub-topics)
Covers “text data (JSON, XML)”
- [XML HTTP Request](#) (W3Schools) and [XML Parser](#) (W3Schools)
Covers “send data by using XMLHttpRequest”

[.serialize\(\) – jQuery API](#) (jQuery)

Covers “implement the JQuery serialize method”

- [jQuery.ajax\(\)](#) (jQuery), [jQuery.get\(\)](#) (jQuery), and [jQuery.post\(\)](#) (jQuery)
Covers “use JQuery to make an AJAX call”, and “implement a callback by using anonymous functions”

AJAX and jQuery (pg161)

jQuery is a Javascript library that specialises in working with the DOM to make webpages dynamic

Use jQuery and AJAX to make **server requests** to updated content for sections of the webpage and not require a full page refresh. Because of its ease of use and popularity jQuery is a good option to choose.

Which list identifies the properties that need to be set up to make an AJAX call?

- A cache, datatype, success
- B url, cache, datatype, success
- C url, datatype, onsuccess
- D url, datatype, oncomplete

C

Cache isn't required
oncomplete is not a property

Why is wiring up events with jQuery easier?

- A It allows you to assign the event listener to many elements at once via the selector syntax
- B There is no difference when compared to the addEventListener method
- C jQuery works more efficiently in a loop
- D jQuery allows both named and anonymous functions to be used as event listeners

A

Module 6: Styling HTML5 by Using CSS3

This module describes how to style HTML5 pages and elements by using the new features available in CSS3. **Lessons**

- Styling Text
- Styling Block Elements
- CSS3 Selectors
- Enhancing Graphical Effects by Using CSS3

Apply Styling to HTML Elements Programmatically

This objective may include but is not limited to: change the location of an element; apply a transform; show and hide elements

- [CSS Positioning](#) (W3Schools) and [HTML DOM Style object](#) (W3Schools)
Covers “change the location of an element”
- [CSS Display](#) (W3Schools) and [HTML DOM Style object](#) (W3Schools)
Covers “show and hide elements”
- [jQuery Effect hide\(\) Method](#) (jQuery)
Covers “alternative show and hide elements”

[CSS3 2D Transforms](#) (W3Schools), [CSS3 3D Transforms](#) (W3Schools), [CSS3 Transitions](#) (W3Schools), [CSS3 Animations](#) (W3Schools), [Skewing Web Page Elements Using The CSS3 Skew Transform](#) (developerdrive.com)

Covers “apply a transform”

Structure a CSS file by using CSS selectors

This objective may include but is not limited to: reference elements correctly; implement inheritance; override inheritance by using !important; style an element based on pseudo-elements and pseudo-classes (for example, :before, :first-line, :first-letter, :target, :lang, :checked, :first-child)

- [CSS Structure and Rules](#) (htmlhelp.com)

Covers “reference elements correctly”

- [6 Assigning property values, Cascading, and Inheritance](#) (W3C)

Covers “implement inheritance” and “override inheritance by using !important”

- [Pseudo-elements and pseudo-classes](#) (W3C)

Covers “style an element based on pseudo-elements and pseudo-classes (for example, :before, :first-line, :first-letter, :target, :lang, :checked, :first-child)”

Change the location of an element (Pg 61)

Apply a transform

Show and Hide elements

HTML elements by default flow statically from left to right in the same order they are declared in the HTML page. The CSS provides a mechanism to specify advanced options

Positioning elements

Absolute positioning : element is placed in the exact location

Relative position : element is placed relative to the left siblings coordinates

You can apply Top, Left, Right, Bottom position properties

Transform elements

Transforms enable you to change an elements appearance larger, smaller, rotate etc..

Show and Hide elements

HTML Declaratively show/hide an element

```
document.getElementById("test").style.display = 'none';
```

```
document.getElementById("test").style.display = 'inline';
```

CSS

visible : sets the property to visible to show the element

hidden : hides the element

visibility: hidden; 'preserves the layout of the page – only hides the element

display: none; 'hides the element AND collapses the layout

Questions	
<p>Absolute positioning positions an object relative to what?</p> <p>A the top left corner of the browser window B the top left corner of its parent element C centred inside the window D centered inside its parent element</p>	<p>B</p>
<p>Which transformation enable you to change the size of an element?</p> <p>A rotate B skew C translate D scale</p>	<p>D – Scale = size</p>
<p>Which syntax preserves the layout of the page when hiding an element in the DOM?</p> <p>A display='hidden' B display='inline' C visibility='none' D visibility='hidden'</p>	<p>D</p> <p>A and C are NOT valid options B shows an element</p>

Using CSS in applications (Pg 235)

Styling HTML Text, box

Each HTML element is a box and each box begins its own new coordinate system. If you place a div element on the page at (50px, 50px) any elements placed inside it (child elements) start at coordinate (0,0) which is the top left corner of the parent element.

All child elements are positioned relative to the container in which they are placed.

When overlapping elements using absolute positioning, CSS provides a z-index property which allows you to specify the order elements should stack on the page along the z-axis (the third dimension)

<style>

declaration

H1{

property:value;

color:#00FF00;

}

</style>

margin: 5px;

padding: 5px

border: 1px solid black;

float: left;

float:right;

opacity: 0.4;

box-shadow: 10px 10px 10px;

box-shadow: h-shadow, v-shadow, blur, spread;

<p>Which of the following CSS would not change the appearance of text?</p> <p>A font-style:italic; B font-weight:heavy; C font: bolder 12px arial; D color:green;</p>	<p>B</p> <p>Heavy is not a valid option</p>
<p>Which of the following aligns text to the full width of the available box?</p> <p>A right B full C center D justify</p>	<p>D</p> <p>Full is not a valid option</p>
<p>Which of the following is a way to configure the amount of space between words?</p> <p>A word-margin B letter-margin C word-spacing D word-padding</p>	<p>C</p> <p>Other 3 are not valid options</p>
<p>Which of the following is not a valid way to alter the size of an element?</p> <p>A div {height:50px; width:50%;} B div {height:50px; width:50px;} C div {height:50cm; width:50px} D div {height:50ft; width:50ft;}</p>	<p>D</p> <p>ft is not a valid unit of measure</p>



<p>Which of the following will successfully style the border of a div element?</p> <p>A border: 1px solid black; B border-sides: 5px solid green; C border-all: 1px solid black; D border: full red;</p>	<p>A</p> <p>A has correct syntax. All others incorrect syntax</p>
<p>When looking from the outside edge of an HTML element and moving to the inside edge what order does the padding, margin and border occur in?</p> <p>A padding, border, margin B margin, border, padding C border, padding, margin D margin, padding, border</p>	<p>B</p>
<p>Which of the following will apply a box shadow to the right and bottom edge of a div element?</p> <p>A box-shadow: gray 5px 5px; B box-shadow: gray -5px 5px; C box-shadow: gray 5px -5px; D box-shadow: gray -5px -5px;</p>	<p>A</p> <p>B shadow left, bottom C shadow right, top D shadow left, top</p> <p>Box-shadow color h, v</p>
<p>Which of the following will place an element relative to the browser window?</p> <p>A absolute B fixed C relative</p>	<p>B</p> <p>Absolute and relative will position an element within the container it sits in.</p>

Module 7: Creating Objects and Methods by Using JavaScript

This module explains how to write well-structured and easily-maintainable JavaScript code, and how to apply object-oriented principles to JavaScript code in a Web application. **Lessons**

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

Establish the scope of objects and variables

This objective may include but is not limited to: define the lifetime of variables; keep objects out of the global namespace; use the “this” keyword to reference an object that fired an event; scope variables locally and globally

- [Global Variables in JavaScript](#) (Other: snook.ca)
Converts “scope variables locally and globally”
- [JavaScript Closures](#) (Other: jabbering.com)
Covers “define the lifetime of variables; keep objects out of the global namespace; use the “this” keyword to reference an object that fired an event; scope variables locally and globally”
- [JavaScript Closures \(Better Link\)](#) (Other: jabbering.com)
Covers “define the lifetime of variables; keep objects out of the global namespace; use the “this” keyword to reference an object that fired an event; scope variables locally and globally”

Create and implement objects and methods

This objective may include but is not limited to: implement native objects; create custom objects and custom properties for native objects using prototypes and functions; inherit from an object; implement native methods and create custom methods

- [JavaScript Native Objects](#) (Other: yaldex.com)



Covers *"implement native objects"*

- [Extending JavaScript Native](#) (Other: javascriptweblog.wordpress.com)
Covers sections *"create custom objects and custom properties for native objects using prototypes and functions"* & *"implement native methods and create custom methods"*
- [Classical Inheritance in JavaScript](#) (Other: crockford.com)
Covers sections: *"inherit from an object"* & *"implement native methods and create custom methods"*

Create and implement objects and methods (Pg 93-102)

JS has NATIVE objects and CUSTOM objects which developers create

Implement native objects

Create custom objects

Implement inheritance

Implement native methods and create custom methods

Using Native objects and methods

Var squareValue = Math.sqrt(144);

Create custom object and method (pg95)



Create constructors

Create a prototype object (class) and create instances from the prototype

https://www.w3schools.com/js/js_object_prototypes.asp

You can build inheritance trees from prototypes in JS

Questions	
<p>In JS which of the following is not a native object?</p> <p>A Function B Array C Integer D Person</p>	<p>D People aren't native objects</p>
<p>Which of the following snippets shows the correct way to create a custom book object?</p> <p>A var book = Title: "my book" , Author:"Jane",Pages400; B var book = {Title: "my book" , Author:"Jane",Pages400}; C var book = (Title: "my book" , Author:"Jane",Pages400); D var book = new {Title: "my book" , Author:"Jane",Pages400};</p>	<p>B Requires {} to construct D isn't as new is keyword to create an instance of the object</p>
<p>Inheritance is accomplished in JS through the use of which construct?</p> <p>A inherits keyword B implements keyword C this keyword D Prototypes</p>	<p>D Prototypes are used to create inheritance trees in JS</p>

Module 8: Creating Interactive Pages using HTML5 APIs

This module describes how to use some common HTML5 APIs to add interactive features to a Web application. This module also explains how to debug and profile a Web application.

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

Implement HTML5 APIs

This objective may include but is not limited to: implement storage APIs, AppCache API, and Geolocation API

- HTML5 Geolocation (W3Schools)
Covers “Geolocation API”
- HTML5 Web Storage (W3Schools)
Covers “storage APIs”
- HTML5 App Cache (W3Schools)
Covers “AppCache API”

HTML5 APIS

Storage API (Pg73)

Local storage : persistent – data available after browser closed and reopened localStorage object

Session storage : available for the duration of the session. Automatically cleaned if browser is closed sessionStorage object

Methods: setItem, getItem, clear, key, removeItem

AppCache API (Pg 77)

Create an application that works when disconnected from the Internet. Makes content and pages available offline.

Requires a **manifest file** and **Javascript API** to support it.

```
<html manifest="webApp.appcache">
```

```
...
```

```
</html>
```

The manifest attribute tells the browser that the webpage needs to be available offline. The value of the manifest attribute points to a manifest file. The file name can be anything but convention is usually **.appcache**

Ensure the webserver can support the chosen file extension

Status

Uncached : application is not associated with application manifest

Idle : caching activity idle – latest

Checking : manifest being checked for updates

Downloading : resources in manifest being downloaded

UpdateReady : resources in manifest successfully downloaded

Obsolete : manifest no longer be downloaded

swapCache() : indicates cache can be replaced to a newer version

update() : tells the browser if an update is available

Geolocation API (Pg81)

```
var geolocator = window.navigator.geolocation;
```

The code above saves a reference to the Geolocation API in a variable to provide shorthand access for use.

```
getCurrentPosition(positionCallback, [positionErrorCallback], [positionOptions])
```

Questions

When using the webStorage API where should you store data to ensure it is cleared when the user closes the browser?

C

<p>A localStorage</p> <p>B cookieStorage</p> <p>C sessionStorage</p> <p>D A hidden input element</p>	<p>B and C are not valid options</p> <p>A is persisent</p>
<p>What do you need to do to designate a page is available offline?</p> <p>A Specify in JS as document.offLine=True</p> <p>B Specify the manifest attribute on the form element</p> <p>C Specify the manifest attribute on the HTML element</p> <p>D Tell users to switch to offline mode</p>	<p>C</p> <p>A and D are not valid options</p>
<p>Which of the following are not valid sections of the AppCache manifest?</p> <p>A Cache manifest</p> <p>B Session manifest</p> <p>C Network manifest</p> <p>D Fallback manifest</p>	<p>B</p> <p>Session manifest is not a valid option</p> <p>Cache manifest lists all resources</p> <p>Network manifest specifies resources from Internet</p> <p>Fallback manifest specifies fallback option</p>
<p>Which event is fired by the AppCache object when the cache download in complete?</p> <p>A oncached</p> <p>B onupdateready</p> <p>C ondownloading</p> <p>D onchecking</p>	<p>A</p> <p>The oncached event is fired when the download is complete</p>
<p>When using the Geolocation API how do you configure the ability to use cached data?</p> <p>A Set the enableCache property to true on PositionOptions object</p> <p>B Set maximumAge property to a non-zero value on PositionOptions object</p> <p>C Set the timeout property of the PositionOptions object</p> <p>D Using the cache is always on to save bandwidth so nothing required</p>	<p>B</p> <p>A enableCache doesn't exist</p>

Pg 34-58 Implementing Media Controls	
Questions	
Which of the following elements are not HTML5? A <article> B <footer> C <hgroup> D <input>	D
Which element(s) does the <hgroup> element contain? A <h1> to <h6> B <header> C <nav> D all of the above	A
Which HTML5 element would you use to organise content so that the page maximises a search engines algorithm? A <div id="Company News"> B <header>Company News</header> C <article> Company News</article> D All of the above	C Header element is within the section element The article element tells the search engine the content within it is what users are searching for. It is the best element for SEO
Which HTML5 elements should be used to create a structured layout A <div> B <p> C <table> D <form>	C Table element
When implementing HTML5 video element how do you ensure that the	C

<p>rendering of the element can function in different browsers?</p> <p>A Do nothing as HTML5 is now a standard specification B Specify all the source video types in the src attribute C Include the <source> element for each video type so the browser can play the version it supports D Include the <object> element for each video type so the browser can play the version it supports</p>	<p>The source elements specifies multiple video formats so the browser can choose the correct one.</p> <p>The object element is a fallback in the event the browser doesn't support the HTML5 video element</p>
<p>When drawing on the HTML5 <canvas> element what method is used on the context to begin drawing at a new point?</p> <p>A moveTo B lineAt C beginPath D stroke</p>	<p>C</p> <p>The beginPath method tells the context to start a new drawing from its current point.</p> <p>The moveTo method moves the current context to a new point but doesn't begin to draw The lineAt method draws a line within the current positional context The stroke method tells the context to draw the graphics</p>
<p>When performance is critical for an HTML5 graphics application what should you use?</p> <p>A <canvas> using a declarative syntax B <svg> using a declarative syntax C <canvas> using JS to create the graphics D <svg> and <canvas> in combination</p>	<p>C</p> <p>Canvas provides better performance than svg when the graphics require a lot of refreshing</p>

Module 9: Adding Offline Support to Web Applications

This module describes how to add offline support to a Web application, to enable the application to continue functioning in a user's browser even if the browser is disconnected from the network. **Lessons**

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

Module 10: Implementing an Adaptive User Interface

This module describes how to create HTML5 pages that can dynamically detect and adapt to different devices and form factors. **Lessons**

- Supporting Multiple Form Factors
- Creating an Adaptive User Interface

Create a flexible content layout

This objective may include but is not limited to: implement a layout using a flexible box model; implement a layout using multi-column; implement a layout using position floating and exclusions; implement a layout using grid alignment; implement a layout using regions, grouping, and nesting

- [box-flex](#) (W3Schools), [box-flex-group](#) (W3Schools), [CSS3 Flexible Box Layout Explained](#) (coding.smashingmagazine.com)
Covers “implement a layout using a flexible box model”
- [CSS3 Multiple Columns](#) (W3Schools), [CSS3: Multiple Column](#) (zenelements.com)
Covers “implement a layout using multi-column”
- [CSS Exclusions](#) (Adobe), [CSS Exclusions W3C Specification](#) (W3C)
Covers “implement a layout using position floating and exclusions”
- [CSS Regions Module](#) (W3C)
Covers “implement a layout using regions, grouping, and nesting”

Flexible Box Model (Pg 266)

Flexbox is a CSS3 construct that provides a way to layout elements that **flow**.

Module 11: Creating Advanced Graphics

This module describes how to create advanced graphics for an HTML5 Web application by using a Canvas element, and by using Scalable Vector Graphics. **Lessons**

- Creating Interactive Graphics by Using Scalable Vector Graphics
- Programmatically Drawing Graphics by Using a Canvas

Write Code that Interacts with UI controls

This objective may include but is not limited to: programmatically add and modify HTML elements; implement media controls; implement HTML5 canvas and SVG graphics

- [DOM Element Object](#) & [XML DOM – The Element Object](#) (W3Schools)
Covers “programmatically add and modify HTML elements”
- [HTML5 Canvas](#) (W3Schools) & [HTML5 Canvas getContext\(“2d”\) Reference](#) (W3Schools)
Covers “implement HTML5 canvas”
- [SVG Tutorial](#) (W3Schools)
Covers “implement ... SVG graphics”

[HTML5 Audio](#) (W3Schools) & [HTML5 Video](#) (W3Schools)

Covers “implement media controls”

Module 12: Animating the User Interface

This module describes how to enhance the user experience in an HTML5 Web application by adding animations. **Lessons**

- Applying CSS Transitions
- Transforming Elements
- Applying CSS Key-frame Animations

Create an animated and adaptive UI

This objective may include but is not limited to: animate objects by applying CSS transitions; apply 3-D and 2-D transformations; adjust UI based on media queries (device adaptations for output formats, displays, and representations); hide or disable controls

- CSS3 Transitions (W3Schools)
Covers “animate objects by applying CSS transitions”
- CSS3 2D Transforms (W3Schools)
Covers “apply [...] 2-D transformations”
- CSS3 3D Transforms (W3Schools)
Covers “apply 3-D [...] transformations”
- Media Queries (W3C)
Covers “adjust UI based on media queries (device adaptations for output formats, displays, and representations)”
- CSS Display and Visibility (W3Schools)
Covers “hide or disable controls”

Module 13: Implementing Real-Time Communications by Using Web Sockets

This module explains how to use Web Sockets to transmit and receive data between an HTML5 Web application and a server.

- Introduction to Web Sockets
- Sending and Receiving Data by Using Web Sockets

Bidirectional communication with WebSocket API (Pg 157)

Websocket API is ideal for real-time applications such as messenger/chat, server based games, video conferencing etc..

Allows connection to the server over a socket

Primary object is called the **WebSocket object**

Websocket object connects to the socket when its constructor is invoked

The constructor accepts 2 parameters: URL of the server -side socket to connect to, an optional list of subprotocols

WebSocket constructor is called

WebSocket API establishes a connection to the server (it either successfully connects or fails)

TEST for feedback **onopen** and **onerror** events

OPEN

CONNECTING

CLOSING

CLOSED

Which of the following is a valid WebSocket instantiation?

A wsConnection = new WebSocket('http://studygroup.70480.com');

B wsConnection = new WebSocket('tcp://studygroup.70480.com', ['soap', 'xmpp']);

C wsConnection = new WebSocket('wss://studygroup.70480.com', ['soap', 'xmpp']);

D wsConnection = new WebSocket('ftp://studygroup.70480.com', ['soap', 'xmpp']);

C

ws OR wss are valid protocols for a websocket connection

Which of the following statements properly handles the reception of data from a WebSocket?	C
A wsConnection.onpost = function(msg){...}; B wsConnection.onreceive = function(msg){...}; C wsConnection.onmessage = function(msg){...}; D wsConnection.ongetdata = function(msg){...};	A, B, C are not valid methods in the WebSocket API

Module 14: Creating a Web Worker Process

This module describes how to use Web Worker Processes to perform long-running operations asynchronously and improve the responsiveness of an HTML5 Web application.

- Introduction to Web Workers
- Performing Asynchronous Processing by Using a Web Worker

[HTML5 Web Workers](#) (W3Schools), and [Using web workers](#) (Mozilla)



Web Worker API provide a way of developing multi threaded JS applications. JS is a single threaded environment. Everything run in JS is queued up synchronously. In more complex applications this means scripts can take a long time to execute giving a poor user experience.

The Web Worker API enables you to specify that pieces of work should be processed on their own thread.

There are advantages and pitfalls which you need to be aware of and respect

`postMessage(param1)` – starts the worker process. Expects a single parameter containing the data for the thread

`terminate()` – stops the worker process from continuing

`onmessage(param1)` – specifies the function for the worker thread to call back to when complete

`onerror()` - specifies a function to call when an error occurs in the worker thread

Limitations

Creating workers is a heavy operation so the number of workers needs careful consideration.

Which of the following isn't a valid web worker operation?

- A `postMessage`
- B `onmessage`
- C `close`
- D `terminate`

C

Only option that ISN'T a valid webworker process

Which method cancels a web worker?

- A `close`
- B `terminate`
- C `suspend`
- D `sleep`

B



<p>Where must you place the JS code to run in the context of a web worker?</p> <p>A Between the <head></head> elements B In any <script> block in the page C In its own JS file D As a dynamic function assigned to the self worker</p>	<p>C</p> <p>The code must be in its own file</p>
<p>How many web workers/sub workers can run concurrently?</p> <p>A A multiple of 4 web workers including sub workers per process B 16 workers by default but you can change via self configuration C A limitless number of workers D A limit of 8 each with a maximum of 8 subworkers</p>	<p>C</p>
<p>To have a script run continually every 30 seconds which line of code should be used?</p> <p>A wsConnection.repeatWork("workerFile.js",30000); B setTimeout(function(){worker.postMessage("");},30000); C setTimeout(worker.postMessage(""),30000); D setInterval(function(){ worker.postMessage("");},30000);</p>	<p>C</p>