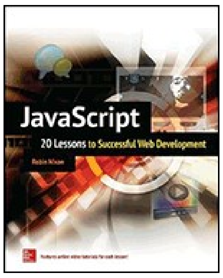

JavaScript: 20 Lessons to Successful Web Development

by Robin Nixon

McGraw-Hill/Osborne. (c) 2015. Copying Prohibited.

Appendix A: Answers to the Self-Test Questions

This appendix contains the answers to all the questions posed at the end of the lessons in this book. To ensure you have understood everything, try to refrain from checking these answers until you have attempted to answer all the questions in a lesson.

If you don't know an answer, try to find it in the book before you look here if you can, as this will help you remember it next time.

Lesson 1 Answers

1. ECMAScript is the name of the official language of which JavaScript and Jscript are dialects.
2. JavaScript is so named due to a marketing tie-in with the Java programming language name; JavaScript is actually not very similar to Java.
3. The trademark for the name JavaScript is owned by Oracle Corporation due to its purchase of Sun Microsystems, which developed Java.
4. It was necessary to create a new programming language (rather than simply embed an existing one such as C) in order to support tight integration with (and manipulation of) HTML elements.
5. HTML stands for HyperText Markup Language—this is the language used to separate HTML documents into their various elements and to describe the basic form and content of a web page.
6. HTTP stands for HyperText Transfer Protocol—this is the method used for communicating between a web browser and server to transfer HTML (and other) documents.
7. CSS stands for Cascading Style Sheets—a system for styling HTML elements without altering a document's contents.
8. DOM stands for Document Object Model—the system devised for separating all parts of an HTML document into separate elements, or objects.
9. JavaScript's Math and Date objects are based on those of the Java language—this is about as close as the two languages get.
10. To view the source of a web page in all major browsers, right-click the document and then select View Source or View Page Source. In some browsers, a new window will open, while others may display the source in a tab.

Lesson 2 Answers

1. To create a single-line comment, start it with the sequence //. For a multiline comment, start it with the sequence /* and end with */.
2. You do not need to end lines of code with a semicolon unless you intend to include another instruction on the same line, in which case the semicolon will act as a new line to separate them. However, if you begin a new line with either a left parenthesis or left bracket (either (or [), you should place a semicolon either at the start of that line or at the end of the preceding one; otherwise, JavaScript will run the two lines together.
3. JavaScript can be included in the <head> of a document or the <body>, or it can be saved in an external file and loaded in where needed, using the src attribute of the <script> tag.
4. JavaScript is case sensitive; the variables YourName, yourname, and YOURNAME are all different and contain separate values.
5. Variable names must begin with either an upper- (A–Z) or lowercase (a–z) letter, or the $ or _ symbols, and may include any of these characters afterward, as well as the digits 0–9.
6. To add values together, use the + operator, for example, result = 23 + 7.
7. To concatenate a string, use the + operator, for example, greeting = ″Hello ″ + YourName. JavaScript works out that the + is being used for string concatenation rather than addition.

8. To incorporate a quotation mark within a string that is enclosed with the same character, escape it with a \ character, for example: ″He said, \″Hello\″.″.
9. To change a string to a number, you can pass it to the Number() function, for example, result = Number(″12345″).
10. In JavaScript, NaN stands for Not a Number, a result you may encounter when a failed attempt is made to convert a value to a number.

Lesson 3 Answers

1. The ++ operator increments a variable by 1 (inversely, -- decrements by 1).
2. The ++ operator can be used in either pre- or post-incrementing mode. The expression if (++a) uses pre-incrementing; it increments a and then performs the if evaluation. On the other hand, the expression if (a++) is evaluated using the current value in a, and only then is it post-incremented.
3. The % operator is used to calculate the modulus of a division operation (the part left over, or the remainder), for example, 20 % 7 returns a modulus of 6 because 7 goes into 20 twice (making 14), leaving a remainder of 6.
4. To assign a value to a variable, you use the = operator, for example, area = Math.PI * radius * radius (Math.PI being a constant containing the value π).
5. To increment a variable by a specified value, use the += operator, for example, a += 27.
6. To turn a value from negative to positive, use the Math.abs() function, for example, Math.abs(myvar). If myvar is negative (such as –92), the value returned is positive (for example, 92); if it is zero or positive, the value returned is simply myvar.
7. To create a random number with a value between 1 and 60 inclusive, call the Math.random() function and turn its result into an integer with Math.floor(), like this: Math.floor(Math.random() * 60) + 1. The + 1 is required at the end because the preceding expression returns a random number between 0 and 59.
8. There are three functions that turn floating point numbers into integers: Math .floor()rounds down to the next lowest integer, Math.ceil() rounds up to the next highest integer, and Math.round() rounds down or up to whichever is the nearest integer value.
9. In JavaScript, the addition operator is also used to concatenate strings, for example, happy_singer = ′Pharrell′ + ′ ′ + ′Williams′.
10. The expression a = a / 20 can be written more succinctly as a/= 20.

Lesson 4 Answers

1. To check whether two values are equal, you use the == operator, for example, if (score == 21).
2. The == operator tests for two expressions having the same value, whereas the === operator is the same, but tests whether they are also of the same type. For example, 23 == ′23′ is true because both sides evaluate to the number 23, but 23 === ′23′ is false because the former is a number but the latter is a string.
3. The values returned by JavaScript to indicate whether an expression evaluates or not are true for success or false for failure.
4. To test whether two expressions both evaluate to true, use the && operator, for example, if (score == 21 && level == 3).
5. To test whether at least one of two expressions is true, use the | | operator, for example, if (city == ′Dallas′ || state == ′Texas′).
6. To test whether an expression is not true, use the ! operator, for example, if (!(input == ′quit′)).
7. The * operator has a higher precedence than + because multiplication and division occur before addition and subtraction.
8. The operator with the lowest precedence is the , (comma) operator, which is used to separate expressions or arguments.
9. The *, /, +, and - operators all evaluate from left to right (except where overridden by operator precedence).

10. You can shorten code that refers to an object by using a with statement, for example, with (string) { alert(length) }. Care must be taken when using this statement as its use can be ambiguous (i.e., to which element does the length property refer?) potentially leading to obscure bugs.

Lesson 5 Answers

1. The first character of an array name must be either an upper- (A–Z) or lowercase (a–z) letter, or the $ or _ character.
2. After the first character, an array name may include any of the characters that can begin an array or variable name, and any of the digits 0–9.
3. To create a new array called mydata, use code such as mydata = new Array().
4. To specify an array's initial length, you can supply a single value within the parentheses representing the length, like this: mydata = new Array(20).
5. To reference item 11 (the element at index 11) in the array mydata, refer to it like this: mydata[11].
6. The first item in an array is always at index 0.
7. You may populate an array with data when you create it by providing a list of values, like this: mydata = new Array('peas', 'corn', 'nuts'). Beware if you supply only a single numeric value, as this will instead set the initial length of the array to that value.
8. An associative array is one in which its elements are accessed in key/value pairs rather than numeric indexes.
9. To add the key/value pair of Name / Alice as a new element in the associative array mydata, use code such as this: mydata['Name'] = 'Alice', or create the array like this: mydata = { 'Name' : 'Alice' }.
10. To retrieve the value for the key Name in the associative array mydata, use code such as this: variable = mydata['Name'].

Lesson 6 Answers

1. To create a multidimensional array, you place additional arrays within elements of a parent array.
2. To access a two-dimensional numeric array, follow the array name with two pairs of square brackets, with numeric index values in each, for example, myarray[23][17].
3. To access a two-dimensional associative array, follow the array name with two pairs of square brackets, with key values in each, for example, myarray['cheese']['mature'].
4. You may nest as many levels of arrays as there is room for in memory. There is no hard and fast rule, but each deeper level requires an order of magnitude more memory. For example, a 10×10 array would use 100 memory locations, 10×10×10 would use 1000, and 10×10×10×10 would use 10,000, and so on.
5. One way to construct a multidimensional array for a class of 30 history students to hold their grades for a year's two semesters would be to create a master array populated with sub-arrays for the semesters, for example, students[0-29][0-1] for 0–29 students, with 0–1 semesters each.
6. To extend this array to handle four years' worth of semesters, you could add a third level between that of the students and the semesters to represent the years, like this: students[0-29][0-3][0-1] for 0–29 students, with 0–3 years each, within which there are 0–1 semesters for each.
7. In the chessboard example, code to represent the black player responding by moving *pawn to queen 4* might be Board[1][3] = '-'; Board[3][3] = 'p'.
8. Code to represent the white player's pawn at king 4 taking the black player's pawn might be Board[4][4] = '-'; Board[3][3] = 'P'.
9. To turn a three-dimensional chessboard array into a four-dimensional array, you would need to add a further level after the third level, like this: 1 master array → 8 sub-arrays → 64 sub–sub-arrays → 512 sub–sub-sub-arrays.
10. To increment the stock of toddlers' bricks by 12, you could use code such as this: Categories['Toddlers']['Wooden Bricks']['Stock'] += 12.

Lesson 7 Answers

1. To iterate through an array one element at a time with for(… in …), use it like this: for (i in myarray) document.write(myarray[i]).
2. The forEach() function is similar to using for(… in …) to iterate through an array, but is simpler to implement. For example, you could use this to iterate through myarray: myarray.forEach(myfunc); however, you must have already written the function myfunc to process the array.
3. You can join two arrays together with the concat() function, for example, programs = utilities.concat(applications).
4. To join together two arrays called tennis and golf into a third called sports, you could use either sports = tennis.concat(golf) or sports = golf .concat(tennis).
5. To quickly pass an entire array to a function, simply pass the array name without the [] brackets, for example, document.write(sports). Here, the resulting output is a comma-separated string.
6. To invoke a function on each element of an array, you could use the map() function; for example, integers = floats.map(Math.round) applies the Math.round function (which converts floating point values to integers) to every element in the array floats (a set of floating point numbers). The resulting values are then saved in the array integers.
7. Contrary to what you might assume, the join() function doesn't join arrays together (that is achieved with the concat() function); instead, it joins all elements of an array together into a single string.
8. To display all the elements in the array sports as a string, with each separated from the next by the string ′ plus ′, you would use the join() function, for example, document.write(sports.join(′ plus ′)).
9. The command document.write(hobbies) displays all the elements in the array hobbies, separated by commas.
10. The command activities = sports.concat(hobbies) creates the new array activities, populating it with all the elements from the sports and hobbies arrays.

Lesson 8 Answers

1. The push() function adds a new element to the bottom of an array, for example, myarray.push(value).
2. The pop() function removes an element from the bottom of an array, for example, value = myarray.pop().
3. The unshift() function adds a new element to the top of an array, for example, myarray.unshift(value).
4. The shift() function removes an element from the top of an array, for example, value = myarray.shift().
5. A First In/Last Out (FILO) array is also known as a *stack*.
6. A First In/First Out (FIFO) array is also known as a *buffer*.
7. You invert the order of elements in an array using the reverse() function; for example, myarray.reverse() will reverse the array in place.
8. You can determine the number of elements in an array by accessing its length property, for example, count = myarray.length.
9. The name given to the process of a section of code repeatedly calling itself is recursion. A common programmer's joke goes, "Question: *What is the dictionary definition for recursion?* Answer: *See recursion!*".
10. A stack structure (being FILO) is best suited for recursive programming, as described in Question 9.

Lesson 9 Answers

1. To sort an array alphabetically in ascending order, call the sort() function on it, for example, myarray.sort().
2. To sort an array alphabetically in descending order, call the sort() and reverse() functions on it, for example, myarray.sort().reverse().

3. To sort an array numerically in ascending order, create an external function that returns the first argument minus the second one, like this: function SortNumeric(a, b) { return a - b }. Then call that function in the sort() function, for example, myarray.sort(SortNumeric).

4. To sort an array numerically in descending order, create an external function that returns the second argument minus the first one, like this: function SortDescend(a, b) { return b - a }. Then call that function in the sort() function, for example, myarray.sort(SortDescend). Or you can perform an ascending numeric search and use the reverse() function.

5. To sort an array numerically in ascending order with an inline function, use code such as myarray.sort(function(a, b) { return a - b }).

6. To sort an array numerically in descending order with an inline function, use code such as myarray.sort(function(a, b) { return b - a }).

7. You can insert and remove elements from an array in the same command by using the splice() function.

8. The command fruits.splice(4, 2) removes two elements from the array fruits starting at index 4 (the fifth element—*another great movie*).

9. The command fruits.splice(5, 0, 'Apples', 'Pears') inserts the values Apples and Pears at index location 5 (the sixth element—*will that be the sequel?*) in the array fruits.

10. By issuing the command fruits.splice(6, 2, 'Apples', 'Pears', 'Grapes'), two elements are removed from the array fruits at index 6 (the seventh element—*which… oh, never mind!*), and then replaced with the values Apples, Pears, and Grapes.

Lesson 10 Answers

1. You can have JavaScript do something if an expression is true using the if() statement, for example, if (score > high_score) { … }.

2. In an if() statement, you do not need braces unless there would be more than one statement within them.

3. To provide a second option to an if()statement for when an expression is false, use the else statement, for example, if (this == that) { … } else { … }.

4. You can extend if() statements to make further tests by placing another if() statement following an else, for example, if (this == that) { … } else if { … }.

5. The best statement to use when you wish to test an expression or variable for a range of values and act differently on each is the switch() statement, for example, switch(result) { … }.

6. To test a value in a switch() statement, use the case keyword, for example, case 42.

7. After the value following a case keyword, you must place a colon before the instructions to execute, for example, case 42: dothis().

8. In a switch() statement, the default keyword processes all remaining values not specifically handled.

9. In a switch() statement, the break keyword is used to jump out of the switch() to the following statement.

10. Braces are not required to enclose the instructions for each case of a switch() statement (but may be used if desired). The statements are listed following the colon after a case keyword and are normally (but not always) terminated with a break keyword.

Lesson 11 Answers

1. The condition of a while() loop is tested before each iteration.

2. The condition of a do … while() loop is tested after each iteration.

3. It is preferable to use do … while() in place of while() when at least one iteration of a loop is needed.

4. One way to display the 8 times table with a while() loop would be: j = 1; while (j <= 12) document.write(j + ' times 8 = ' +

j++ * 8 + '<br>').

5. A for() loop requires three arguments (or sets of arguments): one or more initializers, a test statement, and one or more statements to run after each iteration.
6. To include additional initialization statements in a for() loop, separate them with commas.
7. To include additional statements to the third argument of a for() loop, separate them with commas.
8. A for(… in …) loop iterates through an array one item at a time, for example, for (element in myarray) { … }.
9. To break out of a loop, use a break statement.
10. To skip the current iteration of a loop and move on to the next one, use a continue statement.

Lesson 12 Answers

1. The main purpose of a function is to combine a sequence of one or more instructions into a single unit that can be called by name and that (optionally) can be passed and/or can return values.
2. An anonymous function is one that has not been given a name and that is attached directly to the code that references it.
3. Being in line with code, you should avoid using anonymous functions when you find you need to use the same function in more than one place. In which case, rather than creating redundant extra instances, it's better to make each a named function and simply refer to the single function by name wherever it is called.
4. The main way a value is returned by a function is via the use of a return keyword.
5. To pass values to a function, place them in the parentheses following the function name, separated with commas, for example, myfunc(arg1, arg2, arg3).
6. To access the values passed to a function, its declaration should list a series of variable names in parentheses, each of which will have the value associated with its position assigned to it.
7. You can also access the arguments passed to a function using its arguments object, for example, for (i in arguments) document.write(arguments[i]).
8. When you wish a function to accept an unknown number of different arguments, it is best to use the arguments object rather than named arguments.
9. When a function is called by attaching it to an object using a period (for example, myarray.join()), the function is passed to an object called this that refers to the attached object (in this instance, myarray).
10. To tell a function that a variable is to be used only locally, the first time it is referenced, you can preface it with a var keyword, for example, for (var j = 0 ; j < 10 ; ++j) { … }. Variables not prefaced in this manner are treated as having global scope.

Lesson 13 Answers

1. To declare a class, you use the same syntax as for a function.
2. To create a new instance of a class, you use the new keyword, for example, User = new userclass(fname, lname). Or, if no arguments are required to be passed, simply User = new userclass().
3. To declare properties in a class declaration, attach them to the this keyword, for example, this.firstname = fname.
4. To declare methods in a class declaration, either attach an anonymous or a named function to the this keyword and method name, for example, this.getName = function() { … }. Or, if you already have a named function (called, for example, myfunc), you can attach it like this: this.getName = myfunc.
5. Once you have an object created from a class using the new keyword, you can access its properties by name, for example, User.firstname = 'Alice', or document.write(User.firstname).
6. Once you have an object created from a class using the new keyword, you can access its methods by name, for example,

document.write(User.getName()).

7. The prototype keyword reduces memory usage by allowing reference to a single property or method when a new object is created. Rather than copying a property or method and embedding a copy of it in each new object created, only the single instance of that property or method will be used.

8. A static property or method is one of which there is only a single instance accessible from any object created from a class.

9. The prototype keyword is attached to the class name, and then the property or method that is being prototyped is attached to that, for example, UserClass .prototype.getName = function() { … }, or UserClass. prototype .appName = ′My App′. These are static methods and properties.

10. You can add new functions to extend JavaScript by applying the prototype keyword to an existing class (such as Array or String), for example, String .prototype.Repeat = function(r) { return new Array(++r) .join(this) }.

Lesson 14 Answers

1. You can trap JavaScript errors using the onerror event.

2. To trap errors in a whole document, attach the onerror event to the window object, or attach it to any individual element to trap errors only in that element, for example, window.onerror = function(msg, url, line) { … } or element.onerror = fixError.

3. You can mark a section of code to be tried by a browser without it issuing errors by placing it in a try statement, for example, try { … }. A catch() function or finally section must always be supplied with each try statement.

4. To handle an error caught using a try statement, use a matching catch() function, for example, try { … } catch(e) { … }.

5. A regular expression is a sequence of characters that forms a pattern for making searches and/or replacements.

6. You start and end a regular expression in JavaScript with the / character, for example, /<.+>/.

7. You can (a) check an object using a regular expression with the test() function (for example, document.write(/sat/.test(′The cat sat… ′))), and (b) modify an object with a regular expression using the replace() function (for example, document.write(′The cat sat… ′.replace(/cat/, ′dog′))).

8. To represent a whitespace character in a regular expression, use the \s metacharacter.

9. A shorter way to express the set of characters abcdefghijk in a regular expression is a-k.

10. To ensure an expression is applied both case insensitively and globally, place the i and g characters immediately following the expression, for example, document.write(′The cat sat… ′.replace(/cat/ig, ′dog′)).

Lesson 15 Answers

1. To return an object for an element based on its ID, call the getElementById() function, for example, MyObject = document .getElementById(′MyDiv′).

2. To modify a style property of an object, access it like this: obj.style .background = ′yellow′.

3. To return an array of objects for all elements of a specified type in a document, call the getElementsByTagName() function, for example, images = document.getElementsByTagName(′img′).

4. To set the font size of the object MyObject to 12 point using the setAttribute() function, use code such as MyObject.setAttribute(′style′,′font-size:12pt′).

5. To set the font size of the object MyObject to 12 point without using the setAttribute() function, use code such as MyObject.style. fontSize = ′12pt′.

6. To determine how much space there is available in the current window of the web browser, check the innerHeight and innerWidth properties of the window object, for example, w = window.innerWidth; h = window .innerHeight.

7. To determine the width and height of the screen of the user’s device, check the availWidth and availHeight properties of the screen object, for example, w = screen.availWidth; h = screen.availHeight.

8. To change the title of the current document from JavaScript, access the title property of document, for example, document.title = ′New title′.

9. You can change the image displayed by an <img> tag by changing its src property, for example, image.src = ′newimage.jpg′.

10. You can change the width and height of an image (or other element) with the width and height properties of its style object, for example, image.style .width = ′100px′; image.style.height = ′60px′.

Lesson 16 Answers

1. To change the file displayed by an <img> tag to *newimage.jpg* when the mouse passes over it, change its src attribute value, for example, <img src=′photo .jpg′ onmouseover=″this.src=′newimage.jpg′″>.

2. When an element is clicked, its onclick event is triggered. You could attach to it in the following manner: onclick=″this.src= ′newimage.jpg′″.

3. You can create a new element using the createElement() function, passing it the type of element to create, for example, newspan = document .createElement(′span′).

4. You can attach a new element to the DOM using the appendChild() function, for example, document.body.appendChild(newspan).

5. You can remove an element from the DOM by calling the removeChild() function on the parentNode object of the element, for example, element .parentNode.removeChild(element).

6. You can change the visibility of an object by manipulating its visibility property, supplying values of hidden or visible, for example, MyObject .style.visibility = ′hidden′.

7. To prevent an object from displaying at all, you can change its display property, for example, MyObject.style.display = ′none′.

8. You can stop and restart playback of HTML5 audio or video using the play() and pause() functions.

9. The play() and pause() functions should be attached to the <audio> or <video> element to which they apply.

10. You can specify whether or not the default play and other buttons display on an audio or video player by omitting or including the controls attribute, for example, <audio controls>.

Lesson 17 Answers

1. The cookie property of the current document holds its cookies in one long string; for example, alert(document.cookie) will display all the current cookies and their values.

2. To create a new cookie, assign it to document.cookie, for example, document.cookie = ′cookiename=′ + cookievalue. Either a new cookie will be added or an existing one will be updated.

3. To read a cookie’s value, you must process the document.cookie property string, searching for the cookie’s name. When found, the value following the = sign is the cookie’s value (in escaped format) and is terminated with a semicolon. The substring() function is a good way of pulling out just that part of the string.

4. To delete a cookie, you do the same as creating a cookie but provide an expiry time in the past, for example, document.cookie = ′name=; expires=Thu, 01 Jan 1970 00:00:01 GMT′. No value is supplied following the = because the cookie is to be deleted, and passing a value would be pointless.

5. The domain argument lets you limit the scope of a cookie to only a part of your website such as *blog.mydomain.com*.

6. To test whether a browser supports local storage, use an if() statement to check whether the localStorage object is not undefined, for example, if (typeof localStorage != ′undefined′) { /* Supported */ }.

7. To save an item of data to local storage, call the setItem() function, for example, localStorage.setItem(′username’, ’Mary′).

8. To read an item of data from local storage, call the getItem() function, for example, username = localStorage.getItem(′username′).

9. To remove an item of data from local storage, call the removeItem() function, for example, localStorage.removeItem('username').
10. To empty all local storage for the current domain, call the clear() function, for example, localStorage.clear().

Lesson 18 Answers

1. The best way to test whether a browser is Internet Explorer is to check whether the document.all object exists, for example, if (document.all) agent = 'IE'. If not, then check to see whether the substring Trident exists in the user agent string.
2. The best way to test whether a browser is Opera is to check whether the window.opera object exists, for example, if (window.opera) agent = 'Opera'. If not, then check to see whether the substring OPR exists in the user agent string.
3. You can test for all other browser types by interrogating the userAgent property of the navigator object, for example, if (navigator .userAgent.indexOf('Chrome') != -1) agent = 'Chrome'.
4. A query string is the tail part of a URL that contains data to be passed to a web page.
5. The character that immediately precedes a query string is the question mark (?).
6. The character that separates key/value pairs in a query string is the ampersand (&).
7. Space characters are represented by + signs in query strings (or by %20).
8. The search property of the location object of the current window contains the current query string, for example, query = window. location.search .substr(1). The substr() in this example skips over the initial ? character that precedes the query string.
9. You can split key=value substrings from a query string into an array using the split() function with the & separator, for example, parts = window .location.search.substr(1).split('&').
10. Given an array of key=value substrings called parts, you can turn each string element into a sub-array containing the key in its first element and value in its second using the split() function with the = separator, like this: for (i in parts) parts[i] = parts[i].split('=').

Lesson 19 Answers

1. To set an interrupt to occur at a specific time in the future, call the setTimeout() function, for example, handle = setTimeout(DoThis, 5000). The object handle is saved for future control of the timeout.
2. To set interrupts to occur at repeating intervals, call the setInterval() function, for example, handle = setInterval(DoThat, 1000).
3. To cancel a timeout from occurring, call the clearTimeout() function, passing the handle or (ID) of the timeout to cancel, for example, clearTimeout(handle).
4. To cancel repeating interrupts from occurring, call the clearInterval() function, passing the handle (or ID) of the interval to cancel, for example, clearInterval(handle).
5. The delay used for timeouts and intervals is milliseconds (thousandths of a second) so, for example, use a value of 1000 for 1 second.
6. To test whether a browser supports web workers, test the worker object of the current window, for example, if (!!window.Worker) { /* Supported */ }.
7. You can create a new web worker like this: worker = new Worker('filename.js'), where *filename.js* is the name of a file containing JavaScript to run in the background.
8. Web workers communicate with the calling process using the onmessage event of the worker object, for example, worker.onmessage = function(event) { … }.
9. To send a message to the calling process, a web worker should call the postMessage() function, for example,

postMessage('This is a message').

10. The data property of the worker event object contains the posted message, for example, worker.onmessage = function(event) { document.write(event.data) }.

Lesson 20 Answers

1. To let JavaScript try out the three methods of creating an Ajax object in turn, the try keyword is used in conjunction with catch().
2. Whenever an Ajax object's ready state changes, its onreadystatechange event is triggered.
3. The ready state of the onreadystatechange event is in its readyState property.
4. The status of the onreadystatechange event is in its status property.
5. The response text of the onreadystatechange event is in its responseText property.
6. The purpose of a callback function is to be called when a background process (such as an Ajax call) completes or has something to report.
7. You call a callback function using the call() function by attaching it to the callback function and passing any arguments necessary, for example, callback.call(this.responseText).
8. It can be a good idea to add a random string to Ajax Get requests in order to prevent any caching proxy or other cache mechanism from serving up an older cached version of the requested page. Post requests are never cached and so this "trick" is not necessary for them.
9. To overwrite the HTML contents of an element, you write to its innerHTML property, for example, mydiv.innerHTML = 'New DIV contents'.
10. The supplied OnDOMReady() function is superior to using the built-in onload() function because it triggers earlier, meaning your JavaScript is called sooner, leading to a better user experience. However, be aware that although the DOM will be complete, not all document resources, such as images, may be loaded. If you need to access a particular image, for example, attach a function to the image's onload event.