

Lesson 1 Review

1. What advantage do scripting languages such as JavaScript offer over X/HTML?
Scripting languages offer increased user interaction.
2. What advantage do scripting languages such as JavaScript offer over programming languages such as Java?
Scripting languages offer extended functionality, but are easier and quicker to learn than full programming languages.
3. Name at least three key characteristics of JavaScript.
JavaScript is a scripting language; JavaScript is object-based; JavaScript is event-driven; JavaScript is platform-independent; JavaScript enables quick development; JavaScript is easy to learn.
4. Where does client-side JavaScript code reside, and how is it activated?
Client-side JavaScript code resides within an X/HTML document, and in some cases is embedded directly within certain X/HTML tags (which is called inline scripting). JavaScript is activated by an event, to which the script is instructed to respond with a designated action.
5. In programming, what is an object?
An object is a programming function that models the characteristics of abstract or real "objects." An object encapsulates predesignated attributes and behaviors, and is often grouped with similar objects into classes.
6. In programming, developers refer to attributes and behaviors by what other three terms?
Properties, values and methods.
7. How do JavaScript, JScript, VBScript and the ECMA scripting standard differ?
JScript, ECMAScript and VBScript are similar but different implementations, or "flavors," of JavaScript. JScript is the Microsoft version of Netscape's JavaScript. Minor differences exist in the implementations of JavaScript and JScript, with some differences causing compatibility problems. The ECMA flavor intends to diminish the differences between the JavaScript and JScript flavors. ECMA attempted to promote this standard as "ECMAScript," but the name did not gain widespread popularity. VBScript is similar in purpose to JavaScript, but is rarely used anymore in Web pages. It was created by Microsoft as a subset of the Visual Basic programming language.

8. What is the term for JavaScript's server-side technology?

Server-Side JavaScript (SSJS) is JavaScript's server-side scripting environment. Originally created for Netscape Enterprise Server (and called LiveWire), SSJS is the next generation of the server-side JavaScript language

9. What two types of comment indicators can you use to annotate your JavaScript code?

Single-line comments (//) and multiple-line comments (/*...*/).

Lesson 2 Review

1. How does JavaScript allow you to communicate with Web users?
JavaScript allows you to script actions that occur in response to user events. You can display messages to users, request input from users, and incorporate user input in response messages. JavaScript enables you to communicate in these ways by manipulating objects, and by naming and using variables.
2. Which simple JavaScript methods can you use to communicate with users?
The alert(), prompt(), confirm() and document.write() methods.
3. What event can trigger JavaScript functions as an X/HTML page loads?
The onload event.
4. What is concatenation?
Synthesis of code to simplify it and reduce duplication. In JavaScript, concatenation is often used to combine text strings for further manipulation.
5. When more than one method is defined in a single line of JavaScript code, which method will execute first?
A method defined inside another method will always be executed before the outer method.
6. The alert(), prompt() and confirm() methods are methods of which object?
The window object.
7. In programming, which term refers to the syntax used to associate an object's name with its properties or methods?
Dot notation.
8. What is the purpose of the document.write() method?
To write text into the document dynamically as the page loads in the browser.
9. Name the five data types that can be stored in a variable in JavaScript.
Object, Boolean, number, string and null. An additional type, undefined, occurs when no value is specified; this differs from null, which is actually a value.
10. In JavaScript, what is an expression? Name the six basic JavaScript expressions types.
An expression is a part of a JavaScript statement that is evaluated as a value or assigns a value to a variable. The four JavaScript expression types are assignment, arithmetic, string, logical, conditional and comparison.

11. In JavaScript, what is an operator? List at least three operators and describe their functions or types.

An operator is a character or characters used in an expression to store or return a value, usually manipulating operands (data) in the process. Examples of operators that can be listed for this question include = (assignment), + (arithmetic or string), && (logical), == (comparison), ++ (unary), and so forth. (A complete list of operators is provided in this lesson.)

12. What is an event handler? Name two simple event handlers.

An event handler is a JavaScript mechanism for intercepting an event and associating executable code with that event. Two simple event handlers are onload and onunload.

13. Consider the following JavaScript statements:

```
var myValue = 20;  
myValue += 20;  
myValue = myValue * 10;  
myValue += "20";  
myValue = "New value";
```

For each of these statements, what does the myValue variable evaluate to after each statement executes in JavaScript?

The result would appear as follows:

20

40

400

40020

New value

Lesson 3 Review

1. In programming, what is a function? In JavaScript, how does a function differ from a method?
A function is a named block of code that can be called when needed. In JavaScript, functions and methods are interchangeable. Any method that returns a value can be called a function.
2. What is a built-in function? Name at least one built-in JavaScript function.
A built-in function is a value-returning method provided by the JavaScript language with a predefined purpose. JavaScript also allows users to define their own functions. Built-in JavaScript functions include toUpperCase(), isNaN(), parseInt(), parseFloat(), and others.
3. How can you define a function in JavaScript?
You can define your own JavaScript function by encompassing a group of statements in a statement block beginning with the keyword function. All statements associated with the function are enclosed within curly braces ({ }).
4. What is a floating-point calculation?
A calculation in which the decimal point is allowed to move as needed to account for significant digits.
5. Explain good coding practice and when it is required in JavaScript.
Good coding practice refers to coding standards that are not required but have been created by developers to make code more easily readable and effective. Examples include hierarchical code indentation, consistent capitalization schemes, and comments that state a function's intended purpose.
6. In what ways can you call a function in JavaScript?
You can call a JavaScript function from another function, from a user or browser event, or from a separate <script> block on the X/HTML page.
7. When passing arguments in JavaScript, what must you include with a string literal that is not required for numeric literals?
Quotation marks surrounding the string.
8. In JavaScript, how many arguments can you pass, and how long does each argument exist?
You can pass as many arguments as needed into a function. An argument exists only for the life of the function.

9. What is operator precedence? Give at least one example that relates to JavaScript.
Operator precedence specifies which expressions will be evaluated before others in JavaScript statements. By default, JavaScript follows mathematical precedence. For example, multiplication takes precedence over addition, but parenthetical operators can be used to change operator precedence.
10. How do global variables differ from local variables?
Local variables are declared within a function and can be accessed only from that function. Global variables are declared outside of any function and can be accessed from any function or <script> block you declare.
11. What is pass by value? What is pass by reference? Which does JavaScript use?
Pass by value means that when values are passed to a function, the function's parameters receive a copy of its argument's value (the value of the original data is not changed). Pass by reference means that when values are passed to a function's parameters, the parameters receive the argument's actual value. JavaScript uses pass by value, with the exception that it uses pass by reference for object arguments.
12. What is the conditional operator, and what is its purpose?
The conditional operator is indicated as (condition) ? x : y (where (condition) is a Boolean expression and x and y are values). The conditional operator is used to decide which of these values will be returned. If the Boolean expression is true, x is returned. Otherwise, y is returned.
13. What is a user event? Name at least three examples of JavaScript user events.
A user event is an action performed by a user that can be intercepted by JavaScript and used to call an action. JavaScript user events include click, blur, focus, load, mouseOver, submit, unload and others.
14. What is an event handler? Name at least three examples of JavaScript event handlers.
An event handler is a JavaScript mechanism for intercepting an event and associating executable code with that event. JavaScript event handlers include onclick, onblur, onfocus, onload, onmouseover, onsubmit, onunload and others.

15. Consider the following JavaScript statements:

```
<head>
<scri pt type="text/j avascr ipt">
<!--

functi on myFuncti on() {
  myVal ue = 10;
  return myVal ue * 10;
}

//-->
</scri pt>

</head>
<body>

<scri pt type="text/j avascr ipt">
<!--

document.wri te(myVal ue + "<br />");
document.wri te(myFuncti on());

//-->
</scri pt>
</body>
```

What is the result of these JavaScript statements? Why?

These statements result in the error message "myValue is not defined." The variable myVal ue is defined without the var keyword, making it global in scope. It would seem that the myVal ue variable could be accessed from any script block. However, because myFuncti on is not called until the next line, the myVal ue variable is not accessible to the preceding line. If the second <scri pt> block were written as follows, the myVal ue variable would be accessible:

```
<scri pt type="text/j avascr ipt">
<!--
document.wri te(myFuncti on() + "<br />");
document.wri te(myVal ue);

//-->
</scri pt>
```

The result of the preceding statements would be as follows:

100
10

Lesson 4 Review

1. What is the purpose of the `if . . . else` statement?
The `if . . . else` statement allows you to branch to one of two processes depending on the result of some test condition that you have scripted.
2. What syntax error is common when writing `if` statements?
Forgetting the curly braces when designating a block of code for execution.
3. Where should the `else` keyword be placed in an `if . . . else` statement?
The `else` keyword is placed after the closing curly brace for the `if` block of code associated with the test condition. The `else` block of code is enclosed in its own set of curly braces.
4. What is the purpose of the `else if` statement?
You can use additional `else if` statements to test for multiple conditions in an `if` statement.
5. What is the purpose of the `while` statement?
The `while` statement is used to execute a block of code for as long as a certain test condition is true.
6. What can happen if a `while` statement is improperly coded? How can you avoid this mistake?
An improperly coded `while` statement can set up an infinite loop that could cause the browser to wait indefinitely. To avoid this mistake, include some statement within the `while` condition to ensure that at some point the test condition will become false, thus terminating the loop.
7. What is the purpose of the `for` statement?
The `for` statement is used to repeat a group of statements for some particular range of values.
8. What is the `isNaN()` method?
The `isNaN()` method ("is not a number") determines whether a given value is a valid number.
9. What is nesting?
Nesting is a common programming practice of embedding a script block within another script block.

10. What is the purpose of the break statement?

The break statement is used to exit a loop that would otherwise continue to execute.

11. What is the purpose of the continue statement?

The continue statement is used to force the flow of control back to the top of a loop. It acts like a "skip" or bypass command.

12. What is the purpose of the switch statement? To what type of statement is it similar?

The switch statement is used to compare a value against other values, searching for a match. If a match is found, the code associated with the match is found. The switch statement functions the same as multiple else if clauses within an if statement.

13. What is the purpose of the do...while statement? How does it differ from the while statement?

Like the while statement, the do...while statement is used to execute a block of code for as long as a certain test condition is true. However, the do...while statement does not check the conditional expression until after the first time through the loop, guaranteeing that the code within the curly braces will execute at least once.

14. Write an if statement that checks a variable named grade for a range of values between 70 and 80. If the value is in this range, output the following text: *Your grade is a C.*

One way to write this statement is as follows:

```
if (grade >= 70) {  
  if (grade <= 80) {  
    document.write("Your grade is a C");  
  }  
}
```

A better way to write this statement is as follows:

```
if (grade >= 70 && grade <= 80) {  
  document.write("Your grade is a C");  
}
```

15. Write a while loop that outputs the even numbers between 10 and 20.

```
var x = 10;  
while (x <= 20) {  
  if ((x % 2) == 0) {  
    document.write(x + "<br />");  
  }  
  x++;  
}
```

16. Write a for loop that adds 10 numbers received from user input. Offer the user a chance to exit the program using the break statement.

One way to write this loop is as follows:

```
var userInput = 0;  
var total = 0;  
for (x = 1; x <= 10; x++){  
  userInput = prompt("Enter a number or s to stop.", "");  
  if (userInput == "s") break;  
  total += parseInt(userInput);  
}  
document.write("The total is " + total)
```

17. Write a JavaScript program that asks the user to guess a number between one and five, then uses a switch statement to evaluate the user's input. Use three as the number to be guessed. Use a default clause to catch input that is not in the required range.

```
var userInput = parseInt(prompt("Guess a number between 1 and 5.",  
""));  
switch (userInput) {  
  case 1 :  
    alert("Too low.");  
    break;  
  case 2 :  
    alert("Too low.");  
    break;  
  case 3 :  
    alert("Congratulations!");  
    break;  
  case 4 :  
    alert("Too high.");  
    break;  
  case 5 :  
    alert("Too high.");  
    break;  
  default :  
    alert("Input not in required range.");  
    break;  
}
```

Lesson 5 Review

1. What is the JavaScript object model?

The JavaScript language provides objects that help you manipulate information in useful ways. The JavaScript object model divides these objects into three general groups (browser objects, language objects and form field objects) and organizes them into a hierarchy.

2. What is the Document Object Model (DOM)?

The DOM is a W3C programming specification intended to render an X/HTML page or XML document as a programmable object, giving developers access to all elements within such a document.

3. Describe the principle of containership. How does it apply to the JavaScript object model?

Containership means that some objects are contained within a "parent" object, which is in turn contained within a higher-level object. This principle is illustrated in the hierarchical JavaScript object model. Some objects cannot be used or referenced without referring to the parent (container) object.

4. What is the default object? Must it be referenced by name?

The default object is the window object. It does not need to be referenced by name when calling methods or invoking a subordinate object's methods.

5. What is the purpose of the window object's status property? How can you use inline scripting to manipulate it?

The status property refers to the text string that appears in the status bar at the bottom of the browser window. You can change the status bar text anytime during a script. Inline scripting allows you to change status bar text from hyperlinks and form elements in response to events by using event handlers.

6. When is the document object defined, and for how long does it exist?

The document object is defined when the <body> tag is evaluated in an X/HTML page. The object exists as long as the page is loaded.

7. What is the purpose of the with statement?

The with statement allows you to use several properties and/or methods with a single object by communicating that, for the body of the statement, any references that follow refer to the same object. Using the with statement can save you time coding.

8. What is the purpose of the `image` object? What is one of its more popular uses?
The `image` object allows you to manipulate images in current browsers. (Older browsers, e.g., earlier than Internet Explorer 4.x, may not recognize these functions.) One popular use is to create buttons that animate when a mouse passes over them, by using the `image` object's `src` property to swap images.
9. In what two ways can you access an image when using the JavaScript `image` object?
By the image's name or by the image's array number.
10. What is the purpose of the `history` object?
The `history` object provides access to the list of recently visited URLs that is maintained by the browser.
11. What is the purpose of the `location` object?
The `location` object allows you to specify URLs in a script and provides access to all elements contained in a URL.
12. What is the purpose of the `navigator` object?
The `navigator` object reflects information about the browser being used. It can determine both the brand and version of the user's browser, as well as identify the user's operating system.
13. Write JavaScript code that sets the background and foreground colors of a document.
`document.bgColor = "#00FF00";`
`document.fgColor = "#FF0000";`
14. Write a JavaScript statement that outputs information about the most recent modification of a document.
`document.write("This document was last modified on " +`
`document.lastModified + ".");`
15. Write a JavaScript code block array that outputs the `src` property of the various images that an X/HTML page might contain.
One way to write these statements is as follows:
`var imageNum = document.images.length;`
`for (var i = 0; i < imageNum; i++) {`
`document.write("Image source: " +`
`document.images[i].src + "
");`
`}`

16. Write JavaScript code to preload an image named myImage.jpg that exists in the images directory of the root directory.

```
if (document.images) {  
var newImg = new Image();  
newImg.src = "/images/myImage.jpg"  
}
```

17. A document has an image named img1. Create a button object that uses the onclick event handler to dynamically change img1 to the image that was preloaded in the previous question.

```
<form>  
<input type="button" name="myButton" value="Change Image"  
onclick="document.img1.src = newImg.src" />  
</form>
```

18. Write JavaScript statements that output the protocol used to access the current X/HTML page.

One way write these statements is as follows:

```
document.write("The protocol is " + location.protocol + ".");
```

Another way to write these statements is as follows:

```
switch (location.protocol) {  
case "http:":  
document.write("This page accessed from the Web. <br />");  
break;  
case "file:":  
document.write("This page accessed from local computer. <br />");  
default:  
document.write("This page not accessed from Web or local ↴  
computer. <br />");  
break;  
}
```

19. Write JavaScript statements that use a confirm dialog box and the history object to ask users if they want to go back two pages.

```
if (confirm("Go back two pages?")) {  
history.go(-2);  
}
```

Lesson 6 Review

1. Name the JavaScript language objects discussed in this lesson.
String, Math, Array, Date and RegExp.
2. What can a string consist of? What are two ways strings can exist in JavaScript?
A string can consist of text, numbers, or any combination of characters that functions as text. Strings can exist as literals and as values of variables.
3. What is a constructor?
A mechanism that creates an object in JavaScript.
4. Name at least two uses for which the String object provides methods?
Some examples include formatting text; testing for the presence of certain characters; extracting a subset (called a substring) from a given string; finding the position of a given character or substring in a string; and retrieving the number of characters in a string.
5. When you evaluate strings with the indexOf() and lastIndexOf() methods, what return value indicates that the sought-after string was not found?
A return value of -1 means the string value specified in the method was not found. (
6. What are regular expressions? What JavaScript object allows you to work with them?
Regular expressions provide a mechanism whereby developers can search for specified patterns in text. The JavaScript RegExp object allows you to work with regular expressions.
7. With the RegExp object, how do you delimit the regular expression when using literal text? How do you delimit the regular expression when creating a pattern with the RegExp object?
When using literal text, delimit the regular expression with two forward slashes (/): one to mark the beginning of the regular expression and one to mark the end. When creating patterns with the RegExp object, use quotation marks inside parentheses after the new RegExp constructor.
8. What is an array? What indexing scheme does JavaScript use to make references to array elements?
An array is a single variable with multiple "slots" for different values. Each slot can be referenced by its index number. In JavaScript, array index numbers are zero-based, meaning the first slot in any array has a default index number of 0.

9. What information does the JavaScript Date object allow you to access and manipulate? How can you use this information in a Web page?

The Date object allows you to access and use time and date information on the client machine. You can use this information to provide or verify the current date or time, calculate projected dates for deliveries, or create an onscreen clock, for example.

10. What is the purpose of the JavaScript Math object? How does it differ from other JavaScript objects such as Date?

The Math object allows you create and perform advanced mathematical calculations. Unlike other objects, the Math object does not hold a value; it is a static object that contains various mathematical constants as properties. Also, the Math object does not need to be created before its properties can be used in a script.

11. Consider the following JavaScript statements:

```
var myStr = "joseB@company.com";
document.write(myStr.indexOf("@", 0) + "<br />");
document.write(myStr.lastIndexOf("com") + "<br />");
document.write(myStr.indexOf("b@", 0));
```

What is the output of these statements?

5
14
-1

12. Consider the following searchString() function:

```
function searchString(oneChar, testString) {
  var returnValue = false
  for (var i = 1; i <= testString.length; i++) {
    if (testString.substring(i, i + 1) == oneChar) {
      returnValue = true;
      break;
    }
  }
  return returnValue;
}
```

Describe the functionality of this searchString() function.

The searchString() function is designed to search a string for a particular character. The search character is received by the oneChar parameter, and the string to search is received by the testString parameter. The returnValue variable is declared and initialized to false. A for loop is defined, using the length property of the testString parameter as the middle condition for the loop. Inside the loop, an if statement is used to determine if a match has been found. The substring() method is used to perform this test. The substring() method's arguments are the loop counter variable i, and i + 1. This has the effect of examining one character of the testString parameter each time through the loop. If a match is found, the returnValue variable is set to true, and a break statement ends the loop. After the string is examined, the value of the returnValue variable is returned to the statement that called the searchString() function.

13. Consider the following array:

```
var myArray = new Array("red", "white", "blue");
```

Write code that outputs the values from this array on separate lines of an XHTML page.

One way to write this code is as follows:

```
document.write(myArray[0] + "<br />");  
document.write(myArray[1] + "<br />");  
document.write(myArray[2] + "<br />");
```

A better way to write this code is as follows:

```
for (var i = 0; i < myArray.length; i++) {  
    document.write(myArray[i] + "<br />");  
}
```

14. Consider the following JavaScript statements:

```
var myArray = new Array("red", 10, "white", 30, "blue", 5);  
document.write(myArray.sort());
```

What is the output of these statements?

10, 30, 5, blue, red, white

15. Write a JavaScript block of code that uses the Date object to send the user to an XHTML page titled specifically for the day of the week.

One way to write this code is as follows:

```
var myDate = new Date();  
var myURL = "day" + myDate.getDay() + ".htm";  
location.href = myURL;
```

16. Write a JavaScript block of code that returns the current hour and displays it to the user.

One way to write this code is as follows:

```
var myDate = new Date();  
var myHour = myDate.getHours();  
document.write(myHour + "<br />");
```

A better way to write this code is as follows:

```
var myDate = new Date();  
var myHour = myDate.getHours();  
if (myHour > 12) myHour = myHour - 12;  
if (myHour == 0) myHour = 12;  
document.write(myHour + "<br />");
```


17. What is the output of the following JavaScript statements? Why?

```
var myArray = new Array();
for(i = 0; i < 10; i++) {
  myArray[i] = "value " + i + 1;
  document.write(myArray[i] + "<br />");
}
```

The output is as follows:

value 01

value 11

value 21

value 31

value 41

value 51

value 61

value 71

value 81

value 91

A for loop is used to assign values to the elements of myArray. The loop counter variable is used as the subscript for the array's elements each time through the loop. The literal text "value" is assigned each time through the loop, with the value of the loop counter variable appended to this string. The integer 1 is then appended to the string. The code appears to be written to add 1 to the loop counter variable each time through the loop. If this was the intent, that line of code should be written as:
myArray[i] = "value " + (i + 1);

The output from the code would then be as follows:

value 1

value 2

value 3

value 4

value 5

value 6

value 7

value 8

value 9

value 10

Lesson 7 Review

1. Why are forms important to Web developers?
Forms are the primary user interface in Web pages because they allow a Web developer to retrieve and verify data from users.
2. Name at least three form objects.
form, button, checkbox, hidden, password, radio, reset, select, submit, text, and textarea.
3. When you use form elements, what important requirement makes the form appear on the Web page in some browsers?
To enclose all the form components within a pair of opening and closing <form> </form> tags.
4. In what two ways can you refer to a form object's element?
By its name, or by its index number in the form object's elements array.
5. What does the form object represent in JavaScript, and what is its purpose?
In JavaScript, the form object represents an X/HTML form. It is used to access all properties of a form, including the form's elements, and also provides methods to work with forms.
6. What is the main event handler of the button object? Which property specifies the text that will appear on the button?
The main event handler of the button object is onclick. The value property specifies the text that will display on the button.
7. What is the main property associated with a checkbox object? What value does this property return?
The main property of the checkbox object is checked. The checked property returns a Boolean value of true (if the box is submitted checked) or false (if the box is submitted unchecked).
8. How does the text object differ from the textarea object? How are they similar?
The text object displays a text field that can accept a single line of text, whereas a textarea object displays a larger, scrolling text box that can accept multiple lines of text. They use the same properties, methods and event handlers, which consequently return strings.

9. For what type of user input is the radio object best suited, and why?

The radio object is best suited for yes-or-no questions, or situations in which you want the user to choose only one response out of a group. This is because radio button choices are mutually exclusive; the user can only choose one response.

10. What event handler is commonly used to execute code as the result of the user choosing an option from a select object?

The onchange event handler is most commonly used because it avoids launching script every time a user accesses the list simply to look at it.

11. What is one of the most common and functional uses of client-side JavaScript? What are some of the benefits of this use?

To validate form field submissions. Some benefits of client-side form validation include increased validity of form submissions, increased end-user satisfaction, and conservation of bandwidth.

12. Name at least three tips for effective and appropriate form validation.


Make your messages clear. Be concise in your alerts. Do not test multiple fields in one validation. Only use validation when absolutely necessary. Verify that validations do not override each other. If you require something in your form, be sure to test for it in the validation. Make expectations clear before a user completes a form. Always test your form validations in several browsers.

13. Write a <form> tag that uses inline scripting to call a function named clear() when the form is reset.

<form method="method" action="action" onreset="clear()"; ">

14. Write a JavaScript function that writes true or false on a button object (named myButton in a form named myForm) depending on the result of a confirm dialog box.

One way to write this code is as follows:

```
function changeButtonValue() {  
    document.myForm.myButton.value = confirm("True or False on the   
button?");  
}
```

15. Write JavaScript code to create a function that checks a text object (named myText in a form named myForm) for an empty value. If there is no value, ask the user to enter a value and place the cursor back in the text object.

One way to write this code is as follows:

```
function checkText() {  
  if(document.myForm.myText.value == "") {  
    alert("Please enter a value.");  
    document.myForm.myText.focus();  
  }  
}
```

Another way to write this code is as follows (the form element's values are passed to the function when the function is called):

```
function checkText(form) {  
  if(form.myText.value == "") {  
    alert("Please enter a value.");  
    form.myText.focus();  
  }  
}
```

16. Consider the following code:

```
<form name="myForm">  
  val ue1<i nput type="radi o" name="myRadi o" val ue="test1" />  
  val ue2<i nput type="radi o" name="myRadi o" val ue="test2" />  
  val ue3<i nput type="radi o" name="myRadi o" val ue="test3" />  
</form>
```

Write a function that determines which radi o object was selected, with the selected radi o object's value reflected back to the user in an alert dialog box.

One way to write this code is as follows:

```
function checkRadi o () {  
  if(document.myForm.myRadi o[0].checked) {  
    alert(document.myForm.myRadi o[0].val ue);  
  }  
  if(document.myForm.myRadi o[1].checked) {  
    alert(document.myForm.myRadi o[1].val ue);  
  }  
  if(document.myForm.myRadi o[2].checked) {  
    alert(document.myForm.myRadi o[2].val ue);  
  }  
}
```

A better way to write this code is as follows (the form element's values are passed to the function when the function is called):

```
function checkRadi o(form) {  
  var num = form.myRadi o.length;  
  for(i = 0; i < num; i++) {  
    if(form.myRadi o[i].checked) {  
      alert(form.myRadi o[i].val ue);  
      break;  
    }  
  }  
}
```

17. Consider the following code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html >
<head>
<title></title>
</head>
<body>
<form name="myForm" id="myForm"><select name="mySelect">
<option value="value1">value1</option>
<option value="value2">value2</option>
<option value="value3">value3</option>
</select>
</form>
</body>
</html >
```

Write a JavaScript function that determines which select object's option was chosen, with the chosen select option's value reflected back to the user in an alert dialog box.

One way to write this code is as follows:

```
function checkSelect() {
  if(document.myForm.mySelect.selectedIndex == 0) {
    alert(document.myForm.mySelect.options[0].value);
  }
  else if (document.myForm.mySelect.selectedIndex == 1) {
    alert(document.myForm.mySelect.options[1].value);
  }
  else {
    alert(document.myForm.mySelect.options[2].value);
  }
}
```

A better way to write this code is as follows (the form element's values are passed to the function when the function is called):

```
function checkSelect(form) {
  var num = form.mySelect.length;
  for(i = 0; i < num; i++) {
    if(form.mySelect.options[i].selected) {
      alert(form.mySelect.options[i].value);
      break;
    }
  }
}
```

Lesson 8 Review

1. How does the browser differ from the operating system in relation to the elements responsible for security in a computer system?

The operating system provides an environment for various computer applications. A Web browser is one of the many applications that run on the operating system. The browser is one of a few applications that connect your operating system to the unprotected network that is the Internet. So the browser can be considered a doorway for security problems that can affect your operating system. The operating system can be secured with anti-virus software, patches and updates. Browsers can be also be secured by anti-virus programs, but should also use pop-up and malware-blocking tools. .

2. Name two ways that JavaScript can be used maliciously (or accidentally) on the Web.

Hackers can use JavaScript programs to launch malicious helper applications. Accidental or malicious coding can generate infinite loops that can lock the browser and cause it to malfunction. JavaScript is used in frame-to-frame URL redirects, which can be exploited by hackers. Signed scripts can bypass the usual browser security restrictions. JavaScript can be used for cross-site scripting to steal users' information without their knowledge. JavaScript can be used to create cookies, which can pose security problems.

3. What is a signed script?

Signed scripts can create programs that perform operations once considered to be security risks. A signed script is validated by a certificate to prove the identity of the author and the integrity of the script. Because it carries this validation, the script is allowed to request expanded privileges and abilities, which may override the usual security.

4. What is cross-site scripting? Name at least one associated security risk.

Cross-site scripting (XSS) is a security vulnerability in which an attacker embeds malicious script into a link that appears to be from a trusted site. Upon clicking the link, the embedded script is submitted in the client's HTTP request and can execute on the user's computer, enabling the attacker to steal information without the user's knowledge.

5. What is a cookie?

A cookie is a small piece of information sent from a server to a client's computer and stored in memory. It can be scripted to store its information more permanently on a user's hard drive.

6. What abilities does JavaScript offer the developer in relation to cookies?
JavaScript allows developers to assign cookies, retrieve cookies and store cookies permanently on the hard drive.
7. How are cookies sent? What two actions concerning cookies can occur when a user generates an HTTP request?
Cookies are sent from a server to a client's browser in the HTTP response header. When a user generates an HTTP request, any cookies already present that match the server's domain can be passed along in the request header, or a server can send a cookie back to the client in an HTTP response header.
8. How can cookies aid user authentication?
A cookie can be used to store password information on a user's hard drive so he or she does not have to authenticate to access the site each time. However, this method of security is not recommended for sensitive information; it is safer to store user names and passwords in a database and use server-side security features for authentication.
9. What types of information about a client can a cookie obtain?
Some examples include operating system, browser type and version, service provider, IP address, a history of sites visited, and any information the user enters into the site.
10. How can cookies aid state maintenance?
A cookie can remember information about a particular connection and indicate where a user terminated activity in that connection. When the user returns, he or she can resume activity at the point of termination. Another example would be a cookie used to retain registration information so a return user need not re-enter his or her shipping address for a new order.
11. How can you test for the presence of a cookie? What information is returned when you do this?
One way to test for the presence of a cookie is by using an `alert()` method with the `document.cookie` statement in your script as follows:

`alert(document.cookie);`

When a user accesses the page, he or she will see a list of all cookies associated with that browser session. If no cookies are present, the user will see an empty alert.
12. How does a cookie appear in an HTTP response header?
Set-Cookie: name=value; expires=date; path=path; domain=domain; secure

13. How would two cookies appear in a client's HTTP request header?

Cookie: name=value; name2=value2

14. Describe the best mechanism for deleting a JavaScript cookie.

The best way to delete a JavaScript cookie is to assign the cookie an expiration date that is in the past. This causes the cookie to expire and to be deleted from the user's computer.

15. Write JavaScript code to create a function that sets a cookie named USER that uses the user's input from the following form field as its value:

```
<form name="myForm" >
```

Enter Name:

```
<input type="text" name="name" />  
<input type="button" VALUE="Set Cookie"  
onclick="setCookie(this.form);" />  
</form>
```

Set an expiration date of Saturday, December 31, 2011 at one second before midnight. Also, set the path to root (/) and indicate that a secure protocol should be used for the cookie.

```
function setCookie(form) {  
var userName = form.name.value;  
document.cookie = "user=" + userName + "; path=/; expires=Saturday, 31-December-2011 23:59:59 GMT; secure";  
}
```

16. Consider the following malicious JavaScript code:

```
var myColors = new Array;  
myColors[0] = "#FF0000";  
myColors[1] = "#00FF00";  
myColors[2] = "#0000FF";  
  
for(var i = 0; i < myColors.length; i++){  
    document.bgColor = myColors[i];  
    if(i == 2){  
        i = 0;  
    }  
}
```

Describe this malicious JavaScript code.

This malicious JavaScript code creates an array with three elements. The values for the elements are colors. Inside a for loop, the array's elements are assigned to the bgColor property of the document object. An if statement checks the value of the loop counter variable i and resets it to 0 if it equals 2. Thus, the user's screen constantly changes background colors for the XHTML page, and the user must use the CTRL+ALT+DELETE mechanism to close the browser.

Lesson 9 Review

1. Name two advantages of creating user-defined objects in JavaScript.
You can create sophisticated solutions with a minimum of coding, and you can represent programming constructs as objects (which allows you to code efficient schemes for evaluating forms or creating client-side arrays.
2. What special function allows you to create a custom JavaScript object? What two elements does this function define?
A constructor. It defines the properties and methods of a custom object.
3. What is the term for the real-time objects that are generated from the empty constructor template?
Instances.
4. What must you do after defining a constructor to implement its functionality? What is this process called?
Once the constructor is defined, you need to create new instances of the object. This process is called instantiation.
5. How is the keyword `this` used with constructors?
After the constructor function is called, the keyword `this` returns the specific properties for the custom object to the calling instantiation statement.
6. What must you do to instantiate and then populate the properties of each new instance with actual data?
Declare variables that will become object references to the newly instantiated objects.
7. How can you create your object references to make them easier to use for repetitive tasks?
Create your object references as elements of an array rather than creating individual variables.
8. How many methods can you create for an object?
As many as you need, or as many as memory allows. Memory is always a limiting factor, whether on a server or a desktop.

9. When adding a method name to a custom object constructor, should you follow the method name with parentheses? Why or why not?

No, you should not follow a method name added to a constructor with parentheses, because if parentheses are placed after the method name, the code would attempt to call the function at that moment. You want the JavaScript interpreter to encounter this assignment statement, assign the function name to the identifier, and automatically establish the method for the object.

10. With what do you operate on a single instance of a custom object? What should you use to evaluate multiple custom objects?

A method operates on a single object rather than all objects. A function is used to evaluate multiple custom objects.

11. Does JavaScript offer a predefined method for determining when you should use custom objects instead of arrays for your programs? How else can you determine this?

No, there is no predetermined method for this. Consider whether the object will benefit from having a method or methods associated with it.

12. What is a simple custom object? What is a complex custom object?

A simple custom object has only one level of properties and methods associated with it. A complex custom object has a property or properties that are also objects complete with their own properties.

13. Consider the following showOne() function:

```
function showOne() {  
}
```

Complete the showOne() function so that an alert dialog box gives the user the appropriate information for each employee. Remember that showOne() is a method of the empObject() constructor and will be called for one specific instance of the custom object.

One way to write this code is as follows:

```
function showOne() {  
  var str = "Name: " + this.name + "\n";  
  str += "Age: " + this.age + "\n";  
  str += "Department: " + this.department;  
  alert(str);  
}
```

14. Consider the following form:

```
<form name="myForm">
<input type="text" name="name"/>
<br />
<input type="button" value="Get Department"
oncl ick="getDepartment(thi s. form);" />
</form>
```

```
functi on getDepartment(form) {
}
}
```

Using this form, complete the getDepartment() function so that users can enter a name in the text box and receive an alert dialog box informing them of the employee's department. Add functionality that informs users if a match is not found for the entered name.

One way to write this code is as follows:

```
functi on getDepartment(form) {
var name = form.name.value.toLowerCase();
var num = employees.length;
var i = 0;
while(i < num && employees[i].name.toLowerCase() != name){
  i++;
}
if(i < num) {
  alert(employees[i].name + " is in " + employees[i].
  department.");}
else {
  alert(name + " is not in one of our departments.");
}
}
```

15. Suppose you need a function named getAllEmployees() that returns all instances of employee objects. Using the empObj ect from a previous question, write this function. Use document.write() statements to output the data to an XHTML page.

One way to write this code is as follows:

```
functi on getAllEmployees() {
var num = employees.length;
for (var i = 0; i < num; i++) {
  document.write("Employee " + (i + 1) + ": <br />" +
  employees[i].name+ "<br />");
  document.write("Age: " + employees[i].age + "<br />");
  document.write("Department : " + employees[i].department);
}
}
```

Lesson 10 Review

1. List two advantages of changing X/HTML on the fly.
The developer can make X/HTML much more interactive from the client side. The developer can provide instant feedback to users when they click buttons on the site, making the site much more enjoyable to use.
2. Which JavaScript methods can you use to locate elements within the DOM?
GetElementByID, GetElementsByName and GetElementsByTagName are all ways to locate elements within X/HTML. Once you have located the elements, you can then begin manipulating them, typically using the innerHTML property.
3. Why should you test your code in multiple browsers? Name at least three browsers in which you should test.
JavaScript code does not behave the same in every browser, and often the commands that work in one browser will not work in others. JavaScript is gradually working its way toward standardization, but it is not there yet. You should always test your scripts in Internet Explorer, Firefox and Safari. Google Chrome is also an acceptable answer.
4. What does the innerHTML property do?
The innerHTML property sets and retrieves the contents of a specified element.
5. What is meant by the phrase "changing X/HTML 'on the fly'"?
"Changing X/HTML 'on the fly'" means that you are altering and manipulating DOM (Document Object Model) objects. Changing X/HTML requires the Web page to change state, meaning that an action (event) must occur before the change can happen. The event can be a button click, a timer, the page loading, or a number of other actions.
6. Which JavaScript methods allow you to modify attributes in X/HTML using DOM elements? Provide a brief description of each.
The getAttribute, setAttribute and removeAttribute methods allow you to modify attributes in X/HTML using DOM elements. Getting an attribute is identifying an attribute through JavaScript code. Setting the attribute is changing the attribute dynamically. Removing the attribute takes out the attribute and will return nothing if it the getAttribute looks for it.
7. How will using the document.write method affect the existing Web page?
If run after the page has rendered (i.e., finished loading), then the document.write method will remove the entire contents of the existing page and render only what the document.write contains. The point at which it runs depends on where in the file the document.write is placed and which events are associated with it.

8. Give a brief description of the `appendChild` method.

The `appendChild` method works by adding a node to the DOM, and appending, or placing, the new elements at the end of the DOM. This is useful when a developer does not want to have the page overwritten by the `document.write` method.

9. Why is it good practice to name and place unique IDs on your X/HTML tags?

When a developer needs to access his X/HTML tags, he can access them by name or unique ID. It is good practice to get in the habit of placing IDs and names on your X/HTML tags because if there is ever a need to access a tag, you have already begun the processes needed.

10. Describe elements and attributes in relation to X/HTML tags.

An X/HTML tag consists of an element, an attribute and a value for the attribute (`<element attribute="value">`). An attribute is a property assigned to an X/HTML tag to expand on the capabilities of the tag. Most X/HTML tags have attributes, and these attributes have properties. The more a developer knows about the elements and attributes of X/HTML, the better he or she will be at producing rich, interactive Web sites.

Lesson 11 Review

1. Name three advantages to using code libraries.

Libraries speed up overall development time. Libraries provide standardization. Libraries require a shorter learning curve than JavaScript. Libraries are plentiful and widely available. Libraries do not overwrite your existing X/HTML. Most libraries are mature versions, so the bugs have been worked out. Libraries make it easier to use scripting in your Web sites.

2. What is the term for the project-based pieces of code that are provided to developers by most code libraries?

A plug-in is a specialized extension of a JavaScript library that has been created to perform a single task or set of similar tasks. Plug-ins are the project-based pieces of code provided by most code libraries for developers to download and use in their own programs.

3. How do libraries differ from plug-ins?

A library is a collection of fully formed functions housed within one or more files and made available to developers to simplify the overall use of JavaScript. A plug-in is a single pre-made script from a specific library that performs a specific task, such as data validation. A library consists of many plug-ins, which are essentially the "books" of the library.

4. Why should you check periodically for updates to JavaScript libraries and plug-ins?

Any piece of software is updated from time to time. You should check the library sites you use periodically to ensure that you have the latest version and get any bug fixes for security holes, etc.

5. Can you manipulate the DOM using libraries? Why or why not?

You can manipulate the DOM using pre-made scripts from libraries. Library scripts use the same commands and syntax as JavaScript, but they employ functions and variables that make JavaScript development easier by simplifying the scripting process. Because JavaScript can change the DOM on the fly, so can libraries.

6. In jQuery, what is the purpose of the dollar sign (\$)?

The dollar sign (\$) variable is an alias for jQuery. Although you can use the \$ sign for other things in code, it is standard syntax for jQuery. Most jQuery scripts use \$ signs for the variables that they implement. Some other libraries also use \$ as their alias.

7. You can load a code library into your X/HTML file by adding some code to the <head></head> section. Which tag and attribute allow you to specify the name of the library you are loading?

To load a code library into your X/HTML file, you use the <script> tag and the src attribute. For example:

<script src="jquery-1.4.4.min.js" type="text/javascript" charset="utf-8"></script>

8. Your supervisor gives you an X/HTML file to work on. You open the file and find that it has some JavaScript and some CSS included. For purposes of good coding practice, what should be your first step?

Your first step should be to dissect the page into separate files for X/HTML, script and CSS.

Lesson 12 Review

1. What JavaScript object sends HTTP or HTTPS requests directly to a Web server, and then loads the server response data directly back into the script?

The XMLHttpRequest object, which is associated with AJAX.

2. What is the purpose of XML?

XML (Extensible Markup Language) is a meta-language that allows developers to create unique tags and specific grammars for documents. It is used to structure documents based on the context of content rather than the appearance of content.

3. What is the advantage of using AJAX on a Web site?

AJAX enables the page to update portions without refreshing the entire page, saving valuable bandwidth and time, and making a Web site more interactive.

4. What is a DTD and where is it used?

A DTD (Document Type Definition) is a file used with XML to define the XML tags used within a document. It provides the "grammar" for one or more XML documents. A reference to a DTD is required for a well-formed XML document to be a valid XML document.

5. What language is used to communicate with and extract data from a database?

SQL (Structured Query Language).

6. Name some alternatives to XML-based AJAX.

Alternatives to AJAX include Java applets, Adobe Flash and Microsoft Silverlight.

7. What do libraries do in relation to AJAX?

Libraries make AJAX simpler to use by providing pre-written code that developers can use in their programs.

Lesson 13 Review

1. What is a native debugger?

A native debugger is a debugging tool that is integrated, or built in, to a browser, and can be used to troubleshoot JavaScript.

2. What is a logic error in JavaScript?

Logic errors include mathematical errors, errors in proper command usage, and errors in the structure of the script, which result in the script running improperly. Logic errors do not cause error alerts. The script may return unexpected results or may fail to execute at all

3. Why would you test your JavaScript code in multiple browsers?

JavaScript runs differently in every browser and every version of every browser. Therefore, you need to code for as large an audience as possible.

4. What is a common method used to troubleshoot logic errors?

Set watchpoints (alerts) to check on the program at certain points, and to ensure the expected output is what the actual output is.

5. Name two browsers that use native debugging tools, and name one debugger add-on and the browser that uses it.

Microsoft Internet Explorer and Google Chrome both use native debugging tools. Mozilla Firefox browser offers the Firebug debugger add-on for download and installation.