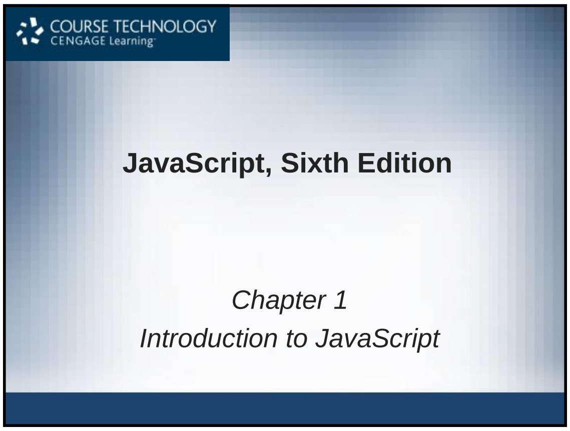
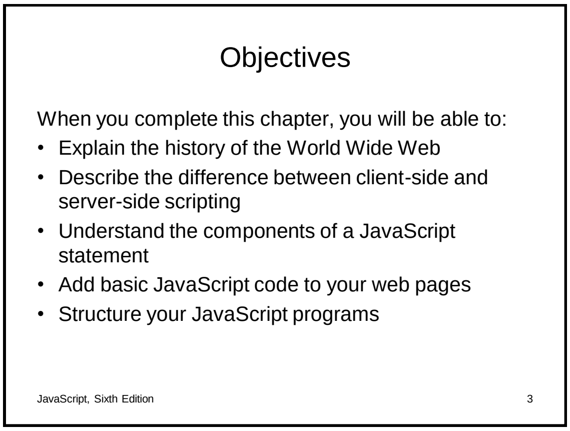


COURSE TECHNOLOGY
CENGAGE Learning

JavaScript, Sixth Edition

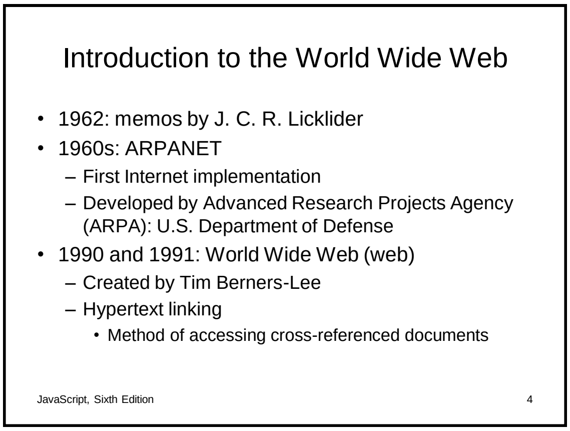
Chapter 1
Introduction to JavaScript



Objectives

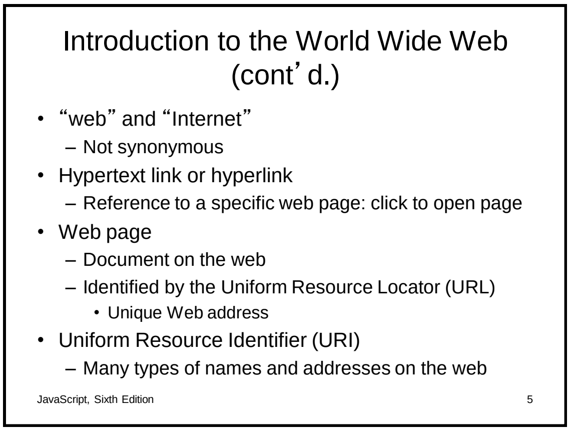
When you complete this chapter, you will be able to:

- Explain the history of the World Wide Web
- Describe the difference between client-side and server-side scripting
- Understand the components of a JavaScript statement
- Add basic JavaScript code to your web pages
- Structure your JavaScript programs



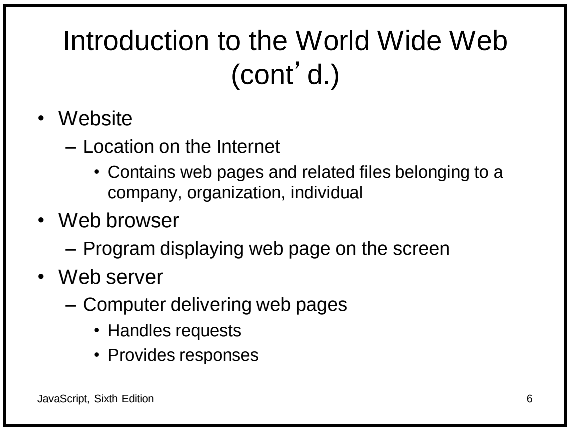
Introduction to the World Wide Web

- 1962: memos by J. C. R. Licklider
- 1960s: ARPANET
  - First Internet implementation
  - Developed by Advanced Research Projects Agency (ARPA): U.S. Department of Defense
- 1990 and 1991: World Wide Web (web)
  - Created by Tim Berners-Lee
  - Hypertext linking
    - Method of accessing cross-referenced documents



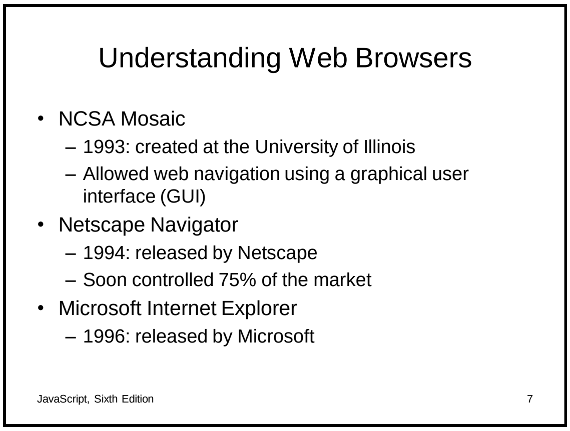
Introduction to the World Wide Web (cont’d.)

- “web” and “Internet”
  - Not synonymous
- Hypertext link or hyperlink
  - Reference to a specific web page: click to open page
- Web page
  - Document on the web
  - Identified by the Uniform Resource Locator (URL)
    - Unique Web address
- Uniform Resource Identifier (URI)
  - Many types of names and addresses on the web



Introduction to the World Wide Web (cont’d.)

- Website
  - Location on the Internet
    - Contains web pages and related files belonging to a company, organization, individual
- Web browser
  - Program displaying web page on the screen
- Web server
  - Computer delivering web pages
    - Handles requests
    - Provides responses



Understanding Web Browsers

- NCSA Mosaic
  - 1993: created at the University of Illinois
  - Allowed web navigation using a graphical user interface (GUI)
- Netscape Navigator
  - 1994: released by Netscape
  - Soon controlled 75% of the market
- Microsoft Internet Explorer
  - 1996: released by Microsoft

Understanding Web Browsers (cont’d.)

- Browser wars began over DHTML
  - Forced web industry to rapidly develop and adopt advanced web page standards
- 1994: World Wide Web Consortium (W3C) established
  - Oversee web technology standards development
  - Adopted Internet Explorer version four DHTML
    - Loyal Netscape followers defected to Microsoft
- Major modern browsers
  - Internet Explorer, Mozilla Firefox, Google Chrome

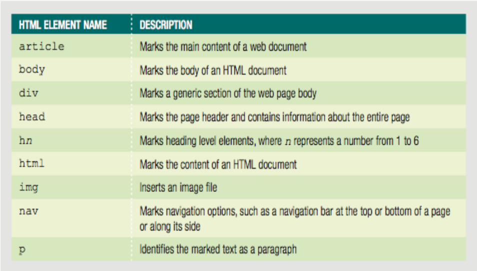
Creating Web Pages

- Hypertext Markup Language (HTML)
  - Markup language
    - Set of characters or symbols defining a document’s logical structure
  - Based on an older Standard Generalized Markup Language (SGML)

Basic HTML Syntax

- Tags: formatting instructions
  - Specify how data in document is treated by browser
- Element: tag pair and any data it contains
  - Element content: information contained between element’s opening and closing tags
  - Empty elements: do not require a closing tag
- Root element (`html`): contains all other elements in a document
- `<head>` element: information used by the browser
- Document body: `<body>` element and the text and elements it contains

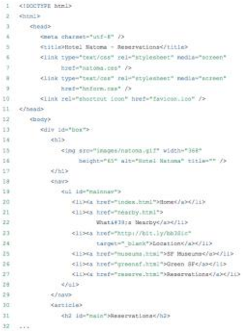
Basic HTML Syntax (cont’d.)



HTML ELEMENT NAME	DESCRIPTION
article	Marks the main content of a web document
body	Marks the body of an HTML document
div	Marks a generic section of the web page body
head	Marks the page header and contains information about the entire page
h*n*	Marks heading level elements, where *n* represents a number from 1 to 6
html	Marks the content of an HTML document
img	Inserts an image file
nav	Marks navigation options, such as a navigation bar at the top or bottom of a page or along its side
p	Identifies the marked text as a paragraph

Table 1-1 Common HTML elements

Basic HTML Syntax (cont’d.)



```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8" />
      <title>Hotel Natoma - Reservations</title>
      <link type="text/css" rel="stylesheet" media="screen"
         href="natoma.css" />
      <link type="text/css" rel="stylesheet" media="screen"
         href="hnform.css" />
      <link rel="shortcut icon" href="favicon.ico" />
</head>
   <body>
      <div id="box">
         <h1>
            <img src="images/natoma.gif" width="368"
                 height="65" alt="Hotel Natoma" title="" />
         </h1>
         <nav>
            <ul id="mainnav">
               <li><a href="index.html">Home</a></li>
               <li><a href="nearby.html">
                    What&#39;s Nearby</a></li>
               <li><a href="http://bit.ly/bb3Sic"
                    target="_blank">Location</a></li>
               <li><a href="museums.html">SF Museums</a></li>
               <li><a href="greensf.html">Green SF</a></li>
               <li><a href="reserve.html">Reservations</a></li>
            </ul>
         </nav>
         <article>
            <h2 id="main">Reservations</h2>
...
```

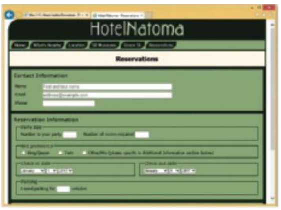


Figure 1-1 Web page in a browser

Creating an HTML Document

- Text editors: Notepad or TextEdit
  - Word-processing applications capable of creating simple text files
- Web development tools: Adobe Dreamweaver and Microsoft Visual Studio
  - Graphical interfaces allowing immediate view of the results
  - Automates process of applying elements
    - May add many unfamiliar elements and attributes to documents

Creating an HTML Document

- Text editors created for coding
  - Non-graphical
  - Number lines of code
  - Color code text based on meaning
- Many available for free:
  - Aptana Studio 3 (Windows and Mac)
  - Komodo Edit (Windows and Mac)
  - Notepad++ (Windows)
  - TextWrangler (Mac)

Working with HTML5

- HTML5
  - Most current version of HTML
- Extensible Hypertext Markup Language (XHTML)
  - Once seen as future language for web development
  - Aspects including inflexible syntax kept from being widely embraced

Introduction to Web Development

- Web page design (web design)
  - Visual design and creation of documents appearing on the World Wide Web
- Web page authoring (web authoring)
  - Creation and assembly of the tags, attributes, data making up a web page
- Web development or web programming
  - Design of software applications for a website
- Web browsers contain commands to view underlying HTML code
  - Only view to improve skills

Understanding Client/Server Architecture

- Two-tier system
  - Client and server
- Server or back end
  - Usually a database: client requests information
- Client or front end
  - Responsible for user interface
  - Gathers information from user
    - Submits information to server
    - Receives, formats, presents results returned from the server

Understanding Client/Server Architecture (cont’d.)

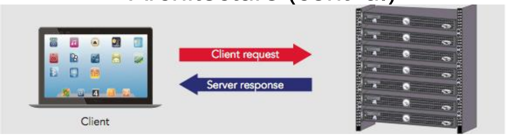


Figure 1-5 A two-tier client/server system

- Web built on a two-tier client/server system
  - Requests and responses through which a web browser and web server communicate happen with HTTP

Understanding Client/Server Architecture (cont’d.)

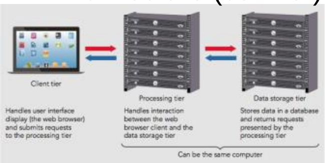

Client tier
Processing tier
Data storage tier
Handles user interface display (the web browser) and submits requests to the processing tier
Handles interaction between the web browser client and the data storage tier
Stores data in a database and returns requests presented by the processing tier
Can be the same computer

Figure 1-6 A three-tier client/server system

- Three-tier, multitier, client/server system
  - Client tier
  - Processing tier
  - Data storage tier

JavaScript and Client-Side Scripting

- Static web pages
  - Cannot change after browser renders them
- HTML produced static documents
- JavaScript
  - Allows web page authors to develop interactive web pages and sites
  - Client-side scripting language: runs on local browser
- Scripting engine executes scripting language code
- Scripting host
  - Web browser containing scripting engine

JavaScript and Client-Side Scripting (cont’d.)

- JavaScript history
  - First introduced in Navigator (LiveScript)
  - Navigator 2.0: name changed to JavaScript 1.0
  - Microsoft released Internet Explorer 4.0 version of JavaScript (Jscript)
- ECMAScript
  - International, standardized version of JavaScript
  - Most recent version: edition 5.1

JavaScript and Client-Side Scripting (cont’d.)

- Limitations of JavaScript
  - Cannot be used outside the web browser
  - Cannot run system commands or execute programs on a client

Understanding Server-Side Scripting

- Server-side scripting
  - Scripting language executed from a web server
  - Popular languages: PHP, ASP.NET, Python, Ruby
- Can develop interactive web sites to communicate with a database
- Server-side scripting language limitations
  - Cannot access or manipulate a web browser
  - Cannot run on a client tier

Understanding Server-Side Scripting (cont’d.)

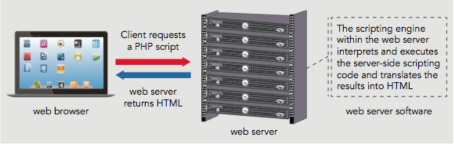

Client requests a PHP script
web server returns HTML
web browser
web server
The scripting engine within the web server interprets and executes the server-side scripting code and translates the results into HTML
web server software

Figure 1-7 How a web server processes a server-side script

Should You Use Client-Side or Server-Side Scripting?

- General rule of thumb
  - Allow client to handle user interface processing and light processing (data validation)
  - Have the web server perform intensive calculations and data storage
- Important to perform as much processing as possible on the client

Adding JavaScript to Your Web Pages

- Basic procedures
  - Used for adding JavaScript to web pages

Using the `script` Element

- Scripts
  - JavaScript programs contained within a web page
- `script` element
  - Tells the browser that the scripting engine must interpret the commands it contains

Understanding JavaScript Objects

- Object
  - Programming code and data
    - Treated as an individual unit or component
- Procedures
  - Individual statements used in a computer program grouped into logical units
  - Used to perform specific tasks
- Methods
  - Procedures associated with an object

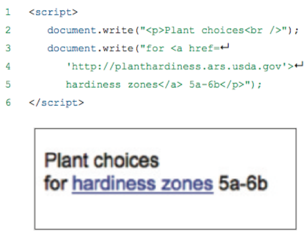
Understanding JavaScript Objects (cont'd.)

- Property
  - Piece of data associated with an object
  - Assign value to a property using an equal sign
- Argument
  - Information that must be provided to a method
- Passing arguments
  - Providing an argument for a method

Using the `write()` Method

- `Document` object
  - Represents content of a browser's window
- Create new web page text with the `write()` method of the `Document` object
  - Method requires a text string as an argument
  - Text string or literal string
    - Text contained within double or single quotation marks

Using the `write()` Method (cont'd.)



```
<script>
   document.write("<p>Plant choices<br />");
   document.write("for <a href=↵
      'http://planthardiness.ars.usda.gov'>↵
      hardiness zones</a> 5a-6b</p>");
</script>
```

Plant choices
for hardiness zones 5a-6b

Figure 1-8 Output of a script that uses the `write()` method of the `Document` object

Case Sensitivity in JavaScript

- JavaScript is case sensitive
- Within JavaScript code:
  - Object names must always be all lowercase

Adding Comments to a JavaScript Program

- Comments
  - Nonprinting lines placed in code containing various types of remarks
- Line comment
  - Hides a single line of code
  - Add two slashes `//` before the comment text
- Block comments
  - Hide multiple lines of code
  - Add `/*` before the first character included in the block and `*/` after the last character in the block

Writing Basic JavaScript Code

- Learn how to write basic JavaScript code
  - Start with variables

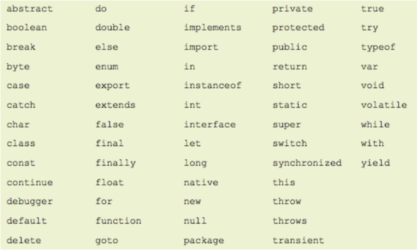
Using Variables

- Variables
  - Values a program stores in computer memory
- Assigning a value to a variable
  - Same as storing a value in a variable

Assigning Variable Names

- Identifier
  - Name assigned to a variable
  - Rules and conventions
    - Must begin with an uppercase or lowercase ASCII letter, dollar sign (`$`), or underscore (_)
    - Can use numbers in an identifier: not as the first character
    - Cannot include spaces in an identifier
    - Cannot use reserved words for identifiers
- Reserved words (keywords)
  - Special words: part of the JavaScript language syntax

Assigning Variable Names (cont’d.)



abstract	do	if	private	true
boolean	double	implements	protected	try
break	else	import	public	typeof
byte	enum	in	return	var
case	export	instanceof	short	void
catch	extends	int	static	volatile
char	false	interface	super	while
class	final	let	switch	with
const	finally	long	synchronized	yield
continue	float	native	this	
debugger	for	new	throw	
default	function	null	throws	
delete	goto	package	transient	

Figure 1-10 JavaScript reserved words

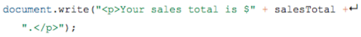
Assigning Variable Names (cont’d.)

- Declaring and initializing variables
  - Use reserved keyword `var` to create variables
    - Example: `var curTime;`
  - Initialize variable using the following syntax:
    - `var variable_name = value;`
  - Assignment operator: equal sign (=)
    - Assigns value on the right side of the expression to the variable on the left side of the expression
  - Assigning a literal string value to a variable
    - Enclose text in quotation marks
  - Can assign the value of one variable to another

Assigning Variable Names (cont’d.)

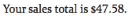
- Displaying variables: print a variable
  - Pass variable name to `document.write()` method
  - Do not enclose it in quotation marks

Code:



```
document.write("<p>Your sales total is $" + salesTotal +↵
   ".</p>");
```

Result in browser:

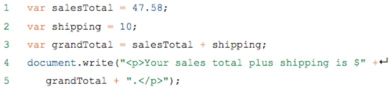


Your sales total is $47.58.

Assigning Variable Names (cont’d.)

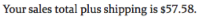
- Displaying variables (cont’d.)
  - Use a plus sign to perform arithmetic operations involving variables containing numeric values

Code:



```
var salesTotal = 47.58;
var shipping = 10;
var grandTotal = salesTotal + shipping;
document.write("<p>Your sales total plus shipping is $" +↵
   grandTotal + ".</p>");
```

Result in browser:



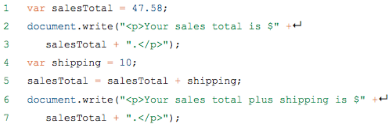
Your sales total plus shipping is $57.58.

Assigning Variable Names (cont’d.)

- Modifying variables
  - Change a variable’s value at any point in a script
    - Use a statement including the variable’s name
    - Followed by an equal sign
    - Followed by the value to assign to the variable

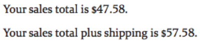
Assigning Variable Names (cont’d.)

Code:



```
var salesTotal = 47.58;
document.write("<p>Your sales total is $" +↵
   salesTotal + ".</p>");
var shipping = 10;
salesTotal = salesTotal + shipping;
document.write("<p>Your sales total plus shipping is $" +↵
   salesTotal + ".</p>");
```

Result in browser:



Your sales total is $47.58.

Your sales total plus shipping is $57.58.

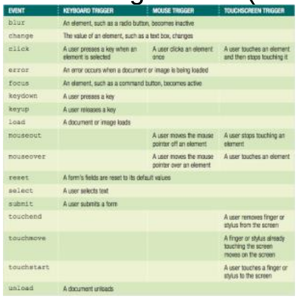
Building Expressions

- Expression
  - Literal value or variable or a combination of literal values, variables, operators, and other expressions
    - Evaluated by JavaScript interpreter to produce a result
- Operands
  - Variables and literals contained in an expression
- Literal
  - Value such as a literal string or a number
- Operators
  - Symbols used in expressions to manipulate operands

Understanding Events

- Event
  - Specific circumstance monitored by JavaScript
  - Script can respond to in some way
  - Allows users to interact with web pages
- Common events: actions users perform
- Can also monitor events not resulting from user actions

Understanding Events (cont' d.)



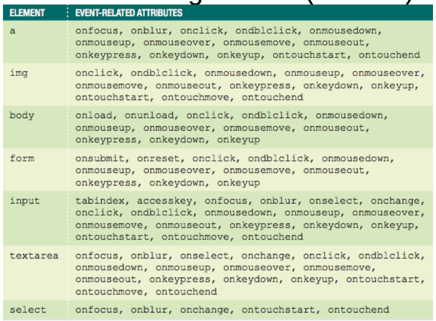
EVENT	KEYBOARD TRIGGER	MOUSE TRIGGER	TOUCHSCREEN TRIGGER
blur	An element, such as a radio button, becomes inactive		
change	The value of an element, such as a text box, changes		
click	A user presses a key when an element is selected	A user clicks an element once	A user touches an element and then stops touching it
error	An error occurs when a document or image is being loaded		
focus	An element, such as a command button, becomes active		
keydown	A user presses a key		
keyup	A user releases a key		
load	A document or image loads		
mouseout		A user moves the mouse pointer off an element	A user stops touching an element
mouseover		A user moves the mouse pointer over an element	A user touches an element
reset	A form's fields are reset to its default values		
select	A user selects text		
submit	A user submits a form		
touchend			A user removes finger or stylus from the screen
touchmove			A finger or stylus already touching the screen moves on the screen
touchstart			A user touches a finger or stylus to the screen
unload	A document unloads		

Table 1-2 JavaScript events

Understanding Events (cont' d.)

- Working with elements and events
  - Events: associated with HTML elements
  - Event handler
    - Code that executes in response to a specific event on a specific element
  - JavaScript code for an event handler
    - Can be specified as attribute of element that initiates event

Understanding Events (cont' d.)



ELEMENT	EVENT-RELATED ATTRIBUTES
a	onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, ontouchstart, ontouchend
img	onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, ontouchstart, ontouchmove, ontouchend
body	onload, onunload, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
form	onsubmit, onreset, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
input	tabindex, accesskey, onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, ontouchstart, ontouchmove, ontouchend
textarea	onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, ontouchstart, ontouchmove, ontouchend
select	onfocus, onblur, onchange, ontouchstart, ontouchend

Table 1-3 HTML elements and some of their associated events

Understanding Events (cont' d.)

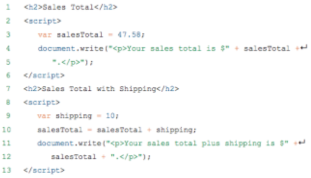
- Referencing web page elements
  - Use the `getElementById()` method
    - Method of the `Document` object
    - Uses element's `id` value
  - Specific element properties
    - Appended to the element reference
  - Allows for the retrieval of information about an element or the ability to change the values assigned to its attributes

Structuring JavaScript Code

- Adding JavaScript code to a document
  - Must follow certain rules regarding placement and organization of that code

Including a `script` Element for Each Code Section

- Can include as many script sections as desired
  - Must include a `script` element for each section
  - Example code below
    - See Figure 1-13 for results



```
<h2>Sales Total</h2>
<script>
   var salesTotal = 47.58;
   document.write("<p>Your sales total is $" + salesTotal +↵
      ".</p>");
</script>
<h2>Sales Total with Shipping</h2>
<script>
   var shipping = 10;
   salesTotal = salesTotal + shipping;
   document.write("<p>Your sales total plus shipping is $" +↵
      salesTotal + ".</p>");
</script>
```

Placing JavaScript in the Document Head or Document Body

- `script` element placement varies
  - Can place in the document head or document body
  - Usually placed at end of body section before `</body>`
    - Statements rendered in the order in which they appear in the document
    - Statements in head prevent rest of page from rendering

Creating a JavaScript Source File

- External file containing JavaScript code
  - Usually designated by the .js file extension
    - Can technically have any extension
  - Contains only JavaScript statements
    - No `script` element and no HTML elements
  - Use the `src` attribute of the `script` element
- Advantages
  - Neater code; code sharing; ability to hide JavaScript code from incompatible browsers
- Can use embedded JavaScript code and JavaScript source files combination

Working with Libraries

- Libraries: especially useful generic scripts used on different websites
  - Often developed by single programmer or team
  - Many available for free reuse
- Common libraries
  - Node.js
  - Backbone.js
  - Modernizr

Validating Web Pages

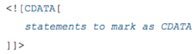
- Validating parser
  - Checks for a well formed web page
  - Verifies document conforms to a specific DTD
- Validation
  - Process of verifying a well-formed document and checking the elements in your document
- Web development tools offer validation capabilities
- Validating services found online
  - W3C Markup Validation Service:
    - *http://validator.w3.org*

Writing Valid XHTML Code with JavaScript

- JavaScript statements contain symbols
  - Prevents XHTML document from being well formed
- HTML handles successfully
  - `script` element statements interpreted as character data
  - Character data (CDATA)
    - Section of a document not interpreted as markup
- XHTML documents
  - `script` element statements interpreted as markup
    - Parsed character data (PCDATA)

Writing Valid JavaScript Code (cont’d.)

- JavaScript code in an XHTML document
  - Treated as PCDATA
    - Validation fails
- Two options to resolve validation issue
  - Move code into a source file
  - Keep JavaScript code within the document
    - Enclose code within a `<script>` element within a CDATA



```
<![CDATA[
   statements to mark as CDATA
]]>
```

Summary

- Hypertext linking: allows quick opening of web pages
- HTML5 is current version
- Web built on a two-tier client/server system
- JavaScript programming language allows for interactive web pages and sites
  - `script` element tells web browser to interpret the commands it contains
  - Can save JavaScript code in a source file
- Validating parser verifies a web page