

JavaScript

Chapter 1

Introduction to JavaScript

Lecturer: Thiều Quang Trung

Objectives

- Study the history of the World Wide Web
- Work with well-formed Web pages
- Learn about Web development
- Learn about the JavaScript programming language
- Add structure to your JavaScript programs
- Learn about logic and debugging

The World Wide Web

- ARPANET
 - First implementation of the Internet
 - Developed in the 1960s by the Advanced Research Projects Agency (or ARPA) of the U.S. Department of Defense
- **World Wide Web**, or the **Web**
 - Created in 1990 and 1991 by Tim Berners-Lee
 - **Hypertext linking**: method of accessing cross-referenced documents

The World Wide Web (continued)

- **Hypertext link, or hyperlink**
 - Contains a reference to a specific Web page that you can click to open that Web page
- **Web page**
 - Document on the Web
 - Identified by a unique address called the **Uniform Resource Locator, or URL (or Web address)**
- **Uniform Resource Identifier (URI)**
 - Generic term for many types of names and addresses on the World Wide Web

The World Wide Web (continued)

- **Web site**
 - Location on the Internet of the Web pages and related files that belong to a company, organization, or individual
- **Web browser**
 - Program that displays a Web page on your screen
- **Web server**
 - Computer that delivers Web pages
 - Request
 - Response

Web Browsers

- NCSA Mosaic
 - Created in 1993 at the University of Illinois
 - First program to allow users to navigate the Web using a graphical user interface (GUI)
- Netscape Navigator
 - Released in 1994 by Netscape
 - Soon controlled 75% of the market
- Microsoft Internet Explorer
 - Released in 1996 by Microsoft
 - Most popular browser on the market today

Web Browsers (continued)

- Browser wars
 - Microsoft and Netscape fought for control of the browser market
- Mozilla Firefox
 - Newcomer that has caused Internet Explorer to lose a significant part of the market
 - **Open source** software (can be freely used and modified)

HTML Documents

- **Hypertext Markup Language, or HTML**
 - Markup language used to create Web pages (commonly referred to as **HTML pages** or **documents**)
 - **Markup language** is a set of characters or symbols that define a document's logical structure
 - Not intended as a method of designing the actual appearance of Web pages
 - Evolved into a language for defining how elements should appear in a Web browser

Basic HTML Syntax

- HTML documents are text documents that contain tags
 - **Tags**: formatting instructions that determine how data is displayed on a Web page
- **Element**
 - Tag pair and any data it contains
- Element's **content**
 - Information contained in element's opening and closing tags

Basic HTML Syntax (continued)

- `<html>` element
 - **Root element** by all HTML documents
- `<head>` element
 - Contains information used by the Web browser
 - **Document head:** `<head>` element and the elements it contains
- `<body>` element
 - **Document body:** `<body>` element and the text and elements it contains

Basic HTML Syntax (continued)

HTML element	Description
<code></code>	Formats enclosed text in a bold typeface
<code><body></body></code>	Encloses the body of the HTML document
<code>
</code>	Inserts a line break
<code><center></code>	Centers a paragraph in the middle of a Web page
<code><head></head></code>	Encloses the page header and contains information about the entire page
<code><hn></hn></code>	Indicates heading level elements, where n represents a number from 1 to 6
<code><hr></code>	Inserts a horizontal rule
<code><html></html></code>	Begins and ends an HTML document; these are required elements
<code><i></i></code>	Formats enclosed text in an italic typeface
<code></code>	Inserts an image file
<code><p></p></code>	Identifies enclosed text as a paragraph
<code><u></u></code>	Formats enclosed text as underlined

Table 1-1: Common HTML elements

Creating an HTML Document

- HTML documents can be created in:
 - Text editors, such as Notepad or WordPad
 - Word-processing applications capable of creating simple text files
 - HTML editors, such as Macromedia Dreamweaver and Microsoft FrontPage
 - Have graphical interfaces that allow you to create Web pages and immediately view the results

Creating an HTML Document (continued)

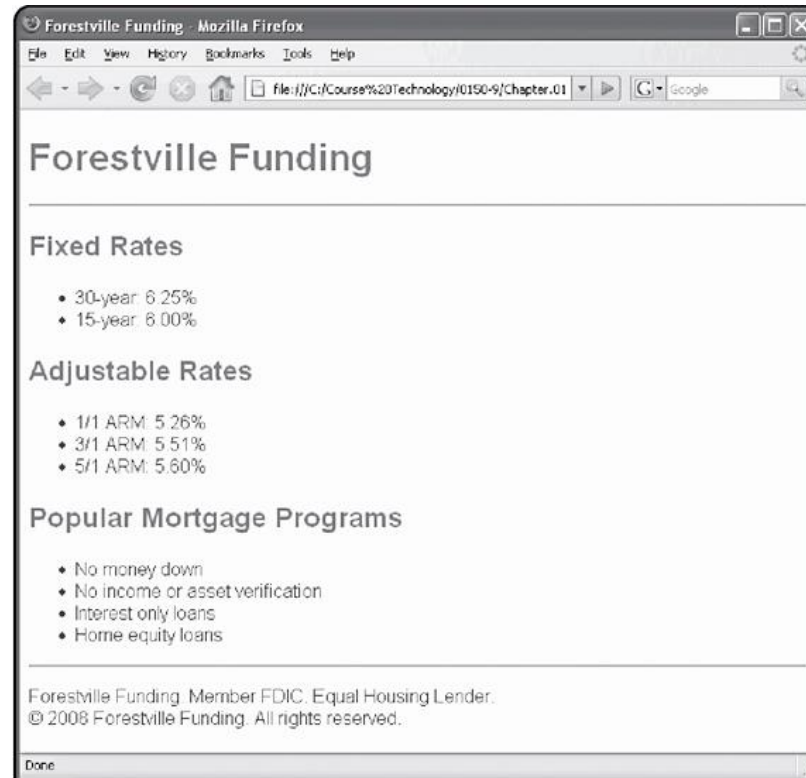


Figure 1-3: ForestvilleFunding.html in Firefox

Working with Well-Formed Web Pages

- **Extensible Hypertext Markup Language (XHTML)**
 - Next generation markup language for creating Web pages
 - Replaces HTML
- **User agent**
 - Application that can retrieve and process HTML and XHTML documents
 - Many user agents (such as mobile phones and PDAs) can't process HTML elements that handle the display and formatting of data

XHTML Document Type Definitions (DTDs)

- **Well-formed** document
 - Conforms to the rules and requirements of XHTML
- **<!DOCTYPE> declaration**
 - Determines the Document Type Definition with which the document complies
- **Document Type Definition, or DTD**
 - Defines elements and attributes that can be used in a document, and rules that a document must follow

XHTML Document Type Definitions (DTDs) (continued)

- Elements and attributes that are obsolete and will be eliminated are said to be **deprecated**
- **Transitional DTD**
 - Allows you to use deprecated style elements
- **Frameset DTD**
 - Identical to transitional DTD, but includes the `<frameset>` and `<frame>` elements

XHTML Document Type Definitions (DTDs) (continued)

- **Strict DTD**
 - Eliminates the elements that were deprecated in the transitional DTD and frameset DTD
- Always try to use strict DTD

Writing Well-Formed Documents

- A well-formed document must include:
 - `<!DOCTYPE>` declaration
 - `<html>`, `<head>`, and `<body>` elements
- All XHTML documents must use `<html>` as the root element
 - The `xmlns` attribute is required in the `<html>` element and must be assigned the URI:
`http://www.w3.org/1999/xhtml`
- XHTML is case sensitive
- All XHTML elements must have a closing tag

Writing Well-Formed Documents (continued)

- Attribute values must appear in quotation marks
- Empty elements must be closed
- XHTML elements must be properly nested
 - **Nesting** refers to how elements are placed inside other elements

Using Phrase Elements

- **Formatting elements**
 - Provide specific instructions about how their contents should be displayed
- **Phrase elements**
 - Primarily identify or describe their contents
 - Help ensure that your Web pages are compatible with user agents that may not be capable of handling formatting elements

Using Phrase Elements (continued)

Element	Description	Renders as
<abbr>	Specifies abbreviated text	Default text
<acronym>	Identifies an acronym	Default text
<cite>	Defines a citation	Italics
<code>	Identifies computer code	Monospace font
<dfn>	Marks a definition	Italics
	Defines emphasized text	Italics
<kbd>	Indicates text that is to be entered by a visitor to a Web site	Monospace font
<q>	Defines a quotation	Italics
<samp>	Identifies sample computer code	Monospace font
	Defines strongly emphasized text	Bold
<var>	Defines a variable	Italics

Table 1-3: Phrase elements

Cascading Style Sheets (CSS)

- CSS
 - W3C standard for managing the design and formatting of Web pages in a Web browser
- **Style**
 - Single piece of CSS formatting information
- CSS information can be:
 - Added directly to documents
 - Stored in separate documents and shared among multiple Web pages

Cascading Style Sheets (CSS) (continued)

- CSS styles have two parts separated by a colon:
 - **Property**: a specific CSS style
 - Value assigned to it, which determines the style's visual characteristics
- **Declaration** or **style declaration**
 - CSS property and the value assigned to it
- **Inline styles**
 - Add style information to a single element
 - Assigned to an element using the **style attribute**

Cascading Style Sheets (CSS) (continued)

The diagram shows the CSS declaration `color: blue`. The word `color` is underlined with a horizontal line, and a diagonal line points from this underline to the word `Property` below it. The word `blue` is also underlined with a horizontal line, and a diagonal line points from this underline to the word `Value` below it.

```
color: blue
  _____
   /
Property

  _____
   \
Value
```

Figure 1-5: Style declaration

Cascading Style Sheets (CSS)

(continued)

- **Internal style sheet**
 - Create styles that apply to an entire document
 - Created in a `<style>` element in document head
- **External style sheets**
 - Separate text documents containing style declarations used by multiple documents on a Web site
 - The empty `<link>` element links a document to a style sheet

The Content-Type `<meta>` Element

- Content-type `<meta>` element is used to specify:
 - Content type that the document uses
 - Document's character encoding
- **Metadata** means information about information
- Can use three primary attributes with the `<meta>` element: `name`, `content`, and `http-equiv`
- `<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />`

Validating Web Pages

- **Validating parser**
 - Checks whether a Web page is well formed and whether the document conforms to a specific DTD
- **Validation**
 - Verifying that your document is well formed
 - Checking that the elements are written according to the element definitions in a specific DTD

Validating Web Pages (continued)



Figure 1-10: W3C Markup Validation Service

Understanding Web Development

- **Web page design or Web design**
 - Visual design and creation of the documents that appear on the World Wide Web
- **Web page authoring or Web authoring**
 - Creation and assembly of the tags, attributes, and data that make up a Web page
- **Web development or Web programming**
 - Design of software applications for a Web site

Client/Server Architecture

- **Two-tier system**
 - Consists of a client and a server
- **Server or back end**
 - Usually a database from which a client requests information
- **Client or front end**
 - Responsible for user interface
 - Gathers information from the user, submits it to a server, and then receives, formats, and presents the results returned from the server

Client/Server Architecture (continued)

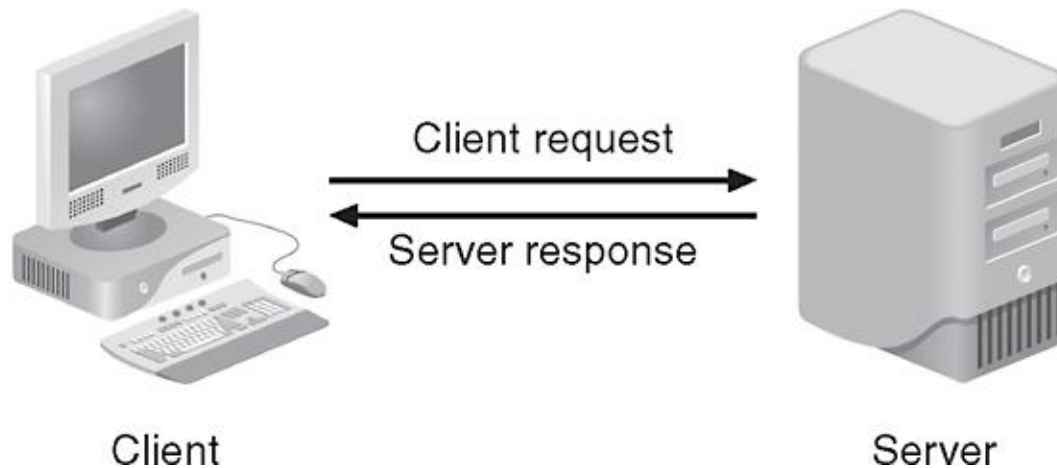


Figure 1-12: The design of a two-tier client/server system

Client/Server Architecture (continued)

- The Web is built on a two-tier client/server system
 - Requests and responses through which a Web browser and Web server communicate happen with HTTP
- **Three-tier, or multitier, client/server system**
 - Client tier
 - Processing tier
 - Data storage tier

Client/Server Architecture (continued)

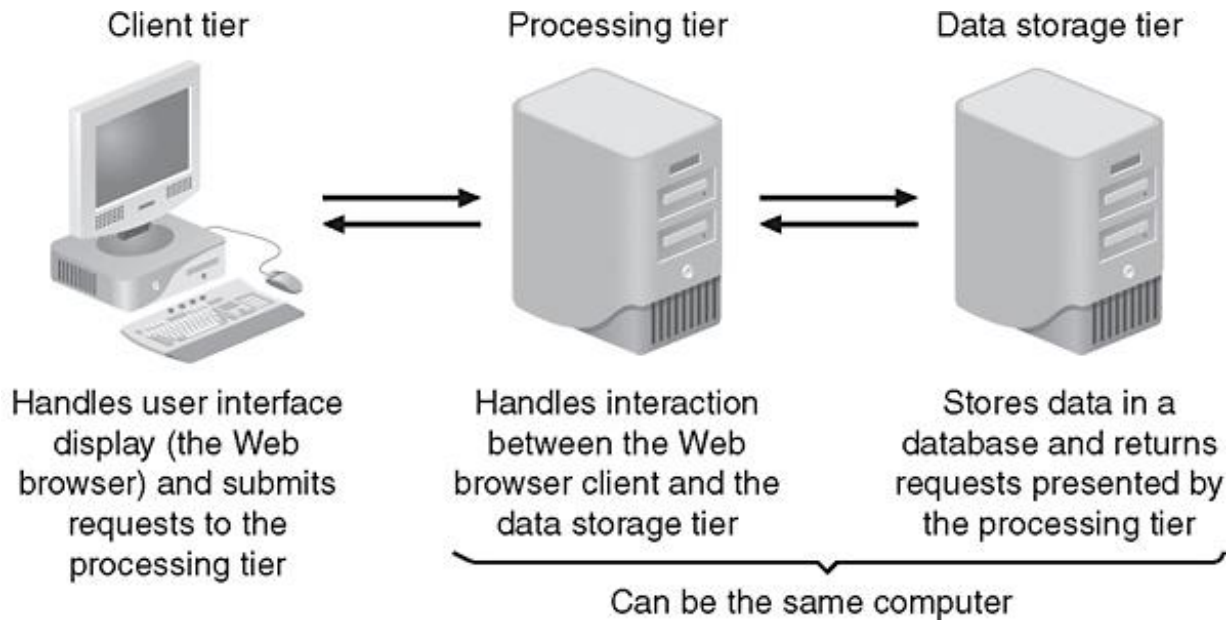


Figure 1-13: The design of a three-tier client/server system

JavaScript and Client-Side Scripting

- When HTML was first developed, Web pages were static
 - **Static** Web pages cannot change after the browser renders them
- HTML and XHTML could only be used to produce static documents
- **JavaScript**
 - Client-side scripting language that allows Web page authors to develop interactive Web pages and sites

JavaScript and Client-Side Scripting (continued)

- **Client-side scripting**
 - Scripting language that runs on a local browser (on the client tier)
- JavaScript gives you the ability to:
 - Turn static Web pages into applications such as games or calculators
 - Change the contents of a Web page after a browser has rendered it
 - Create visual effects such as animation
 - Control the Web browser window itself

Server-Side Scripting and PHP

- Server-side scripting
 - Scripting language executed from a Web server
 - Popular languages: PHP, ASP, and JSP
- One of the primary reasons for using a server-side scripting language is to develop interactive Web sites that communicate with a database
- Server-side scripting language cannot:
 - Access or manipulate a Web browser
 - Run on a client tier

Server-Side Scripting and PHP (continued)

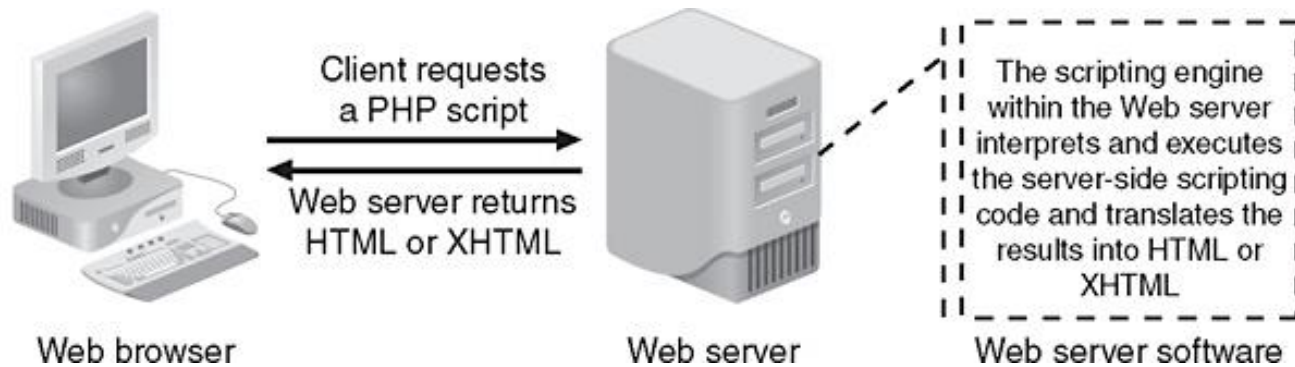


Figure 1-14: How a Web server processes a server-side script

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

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

The JavaScript Programming Language

- The following sections introduce basic procedures for adding JavaScript to your Web pages

The `<script>` Element

- **Scripts**
 - JavaScript programs contained within a Web page
- **`<script>` element**
 - Tells the Web browser that the scripting engine must interpret the commands it contains
 - The **`type`** attribute tells the browser which scripting language and which version of the scripting language is being used

Understanding JavaScript Objects

- **Object**
 - Programming code and data that can be 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
 - For example: `loan.calcPayments()` ;

Understanding JavaScript Objects (continued)

- **Property**

- Piece of data associated with an object
- Assign a value to a property using an equal sign

```
loan.interest = .08;
```

- **Argument**

- Information that must be provided to a method
- Providing an argument for a method is called **passing arguments**

```
loan.calcPayments(800);
```

Using the `write()` and `writeln()` Methods

- `Document` object represents the content of a browser's window
- You create new text on a Web page with the **`write()` method** or the **`writeln()` method** of the `Document` object
 - Both methods require a text string as an argument
 - **Text string or literal string:** text that is contained within double or single quotation marks

```
document.write("Bienvenue au Canada!");
```

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 that you place in your code to contain 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 you want included in the block and */ after the last character in the block

Structuring JavaScript Code

- When you add JavaScript code to a document, you need to follow certain rules regarding the placement and organization of that code
- The following sections describe some important rules to follow when structuring JavaScript code

Including a `<script>` Element for Each Code Section

- Include as many script sections as you like within a document
- When you include multiple script sections in a document, you must include a `<script>` element for each section

Placing JavaScript in the Document Head or Document Body

- You can place `<script>` elements in either the document head or document body
- Good idea to place as much of your JavaScript code as possible in the document head
- Important to put JavaScript code in document head
 - When code performs behind-the-scenes tasks required by script sections in the document body

Creating a JavaScript Source File

- JavaScript source file
 - Usually designated by the file extension `.js`
 - Does not contain a `<script>` element
 - Cannot include XHTML elements
- To access JavaScript code saved in an external file, assign to the `src` attribute of the `<script>` element the URL of the JavaScript source file
- Use a combination of embedded JavaScript code and JavaScript source files in your documents

Writing Valid JavaScript Code

- In HTML documents, statements in a `<script>` element are interpreted as character data instead of as markup
 - **Character data** or **CDATA**: a section of a document that is not interpreted as markup
- In XHTML documents, the statements in a `<script>` element are treated as parsed character data, or PCDATA
 - **Parsed character data** or **PCDATA**: a section of a document that is interpreted as markup

Writing Valid JavaScript Code (continued)

- If you attempt to validate an XHTML document that contains a script section, it will fail the validation
- To avoid this problem:
 - Move your code into a source file
 - Enclose the code within a `<script>` element within a CDATA section
- Syntax for including a CDATA section

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

Logic and Debugging

- All programming languages have a **syntax** (rules)
- **Logic**
 - Order in which various parts of a program run, or execute
- **Bug**
 - Any error that causes program to function incorrectly
- **Debugging**
 - Act of tracing and resolving program errors

Summary

- HTML documents are text documents that contain formatting instructions, called tags, which determine how data is displayed
- XHTML is the next generation markup language for creating Web pages
- To design and format the display of Web pages for traditional Web browsers, you use CSS
- Web development or Web programming is the design of software applications for a Web site
- The Web is built on a two-tier client/server system

Summary (continued)

- Three-tier, or multitier, client/server system consists of client tier, processing tier, and data storage tier
- JavaScript is a client-side scripting language that allows you to develop interactive Web pages
- An object is programming code and data that can be treated as an individual unit or component
- “Logic” is the order in which various parts of a program run, or execute
- “Debugging” is the act of tracing and resolving errors in a program