

AJAX

AJAX (Asynchronous JavaScript and XML) is a Web development technique for creating interactive and responsive Web applications by exchanging small amounts of client data with the Web server behind the scenes. The entire Web page does not have to be reloaded each time the user makes a change, thereby increasing the Web page's interactivity, speed, and usability. Using a project-based approach, this course explains how to distribute a Web application's functionality between the Web server and the browser. Topics include AJAX components XHTML, CSS, DOM and the XMLHttpRequest object; AJAX design principles and patterns; AJAX Frameworks (Prototype, X, and Rico); and other important considerations such as usability, performance, and security.

Who should take this course?

This course is for students who wish to learn how to integrate the Web technologies of XHTML, JavaScript, CSS, and the Document Object Model (DOM) to create AJAX-enabled Web applications.

Course Objectives

- Format elements using the inline style attribute, internal style sheets, and external style sheets.
- Create an XMLHttpRequest object.
- Write event handling code that will work in (nearly) all browsers.
- Delete, insert and sort rows of an HTML table.
- Use Prototype, Rico, Script.aculo.us, and the X frameworks in Web pages.
- Identify Web site vulnerability to security exploits.
- Minimize the memory footprint used by an AJAX application.
- List the components of an AJAX application for viewing and editing a live database of items.

Course Details

- Length: 18 hours
- Format: Classroom
- Prerequisites: XHTML – Level 1 and 2, Cascading Style Sheets – Level 1 and JavaScript or equivalent. Recommended: XML Introduction and familiarity with Web server programming. *The above prerequisites are considered to be the basic skills and knowledge needed prior to taking this class. Instructors will assume your readiness for the class materials and will NOT use class time to discuss prerequisite materials.*



Course Contents

AJAX Background

- The definition of AJAX
- The Web technologies that comprise AJAX
- Four defining principles of AJAX
- Some Web sites that currently employ AJAX

CSS and DOM

- The four places from which an HTML element obtains its styling rules
- The significance of the id and class attributes of HTML elements
- Formatting elements using the inline style attribute, internal style sheets, and external style sheets
- The Cascading (inheritance) aspect of CSS
- How CSS styles may be manipulated by JavaScript
- DOM objects, including document, Node, and Element objects
- Using key DOM methods, including `document.getElementById`, `document.createElement`, and `node.appendChild`

XMLHttpRequest Object

- The purpose of the XMLHttpRequest object
- Creating an XMLHttpRequest object in a “browser-agnostic” way
- How to use the Ready States of an XMLHttpRequest object
- Using methods of the XMLHttpRequest object to send requests to a server
- Authoring JavaScript functions that handle the responses from requests initiated by the XMLHttpRequest object



Course Contents, continued

Event Handling

- The JavaScript event object
- Event handling methods among browsers
- Writing event handling code that works in (nearly) all browsers
- The significance of the “this” keyword in event handler functions
- Utility functions that make cross-browser event handler relatively simple

DOM and HTML Tables

- Deleting rows of an HTML table
- Inserting new rows of an HTML table
- How to iterate over the rows collection of an HTMLTableElement
- How to iterate over the cells collection of an HTMLTableRow
- Sorting rows of an HTML table

AJAX Framework

- Useful AJAX client-side frameworks
- Using Prototype, Rico, Script.aculo.us, and the X frameworks in Web pages
- Server-side frameworks (including Dynamic Web Remoting [DWR], Ajax4JSE, and Icefaces)

Security

- The ramifications of the “Server of Origin” policy
- Security mechanisms among browser implementations
- Several advantages and disadvantages of the HTTPS protocol
- Web site vulnerability to security exploits
- Techniques for improving the security of a Web server



Course Contents, continued

Performance

- The two primary facets of performance
- Techniques for measuring JavaScript's execution speed
- Using the profiling feature of Venkman
- Minimizing the memory footprint used by an AJAX application

Model View Controller

- Several characteristics of a classic Model View Controller design
- A "nested" Model View Controller design
- The components of an AJAX application for viewing and editing a live database of items