

## Developing Ajax Web 2.0 Applications: Hands-On - 4 Days

### *Course 986 Overview*

- You Will Learn How To**
- Develop powerful, easy-to-use Web 2.0 sites using Ajax techniques
  - Construct robust user interfaces that are compelling, intuitive and accessible
  - Create effective, dynamic content for integration with your web applications
  - Enhance application functionality with the JQuery, Prototype and Dojo libraries
  - Apply best practices to create standards-compliant, robust web applications
  - Enhance the security of Ajax-based web applications

**Course Benefits** Modern websites must be intuitive to use, accessible to all users, and have responsive applications that promote a fluid, uninterrupted workflow. Ajax is a powerful tool for creating interactive web applications that meet these requirements. Enhancing your organization's current website with Ajax increases usability and customer satisfaction while maintaining a competitive edge. In this course, you gain the practical skills required to effectively design, create and implement Ajax-enabled Web 2.0 sites.

**Who Should Attend** Those involved in developing and managing web applications. Experience with JavaScript at the level of Course 489, "JavaScript for Web Development," is required.

**Hands-On Training** You are immersed in an evolving case study creating an Ajax-enabled website. Exercises include:

- Adding Ajax functionality to an existing website
- Creating intuitive user interfaces with drag and drop functionality
- Plotting and mapping with third-party geolocation APIs
- Expanding your site search with autosuggest
- Controlling browser back button behavior with the RSH framework
- Preventing the theft of JSON data

## Developing Ajax Web 2.0 Applications: Hands-On - 4 Days

### Course 986 Outline

#### Exploring Ajax Fundamentals

##### Identifying core Ajax/Web 2.0 components

- XHTML
- XML
- JavaScript
- CSS
- DOM
- JSON
- Exchanging information using the XMLHttpRequest object

#### Building rich, interactive web applications

- Measuring the business benefits of Ajax
- Improving data exchange efficiency
- Streamlining data entry and presentation

#### Applying Client-Side Ajax Techniques

##### Implementing Ajax communication approaches

- Exchanging data with the server using XHR
- Selecting GET or POST methods
- Processing the server response
- Handling communication errors

#### Development and debugging methods

- Monitoring communications
- Utilizing browser tools

#### Developing Dynamic Content

##### Exploiting the power of DOM

- Accessing page elements
- Modifying HTML content
- Avoiding memory leaks
- Searching and manipulating XML with DOM

#### Enhancing the user experience

- Latency and feedback
- Ensuring up-to-date data with cache defeat
- Compressing for optimal download speed

#### Leveraging Third-Party Libraries

##### Optimizing functionality

- Tapping into the power of client libraries
- Extending standard objects
- Decreasing time-to-market with code-reuse
- Defining and configuring JavaScript objects with JSON

#### Exploiting third-party libraries

- Assessing the benefits of JavaScript library code
- Utilizing the library selection criteria

- Introducing JQuery, Prototype and Dojo

#### Simplifying page logic

- Streamlining page content manipulation
- Reacting to the user with event handling
- Cleaning up HTML with unobtrusive JavaScript
- Encapsulating Ajax requests

#### Making the User Interface Accessible

##### Meeting accessibility standards

- Architecting for progressive enhancement and graceful degradation
- Complying with W3C and statutory guidelines

#### Improving ease of use

- Resolving back button limitations
- Adding bookmarking capabilities

#### Managing Security and Validation

##### Reducing security threats

- Analyzing the XHR security model
- Preventing theft of JSON data

#### Validating user input

- Checking form data
- Addressing the limitations of client-side validation

#### On-Demand JavaScript Capabilities

##### Hiding complexity

- Dynamically injecting script blocks
- Creating self-contained JavaScript libraries

#### Constructing cross-domain requests

- Making use of the <script> tag to access JSON data from a second site
- Integrating cross-domain RSS with a server-side proxy

#### Maximizing Toolkits for Rich User

##### Interfaces

##### Taking advantage of code libraries

- Adding a graphical calendar control
- Enabling drag and drop
- Marrying data and presentation through in-line editing
- Exploiting special effects for the "wow" factor

#### Developing a rich user interface

- Populating forms with dynamic requests

- Assisting the user with autosuggest
- Supercharging data forms with rich text editors