Advanced JavaScript Patterns and Techniques

Course 42583 - 24 Hours

Overview

JavaScript, the world's most misunderstood programming language, and the language of the web, is now more important than ever. JavaScript is the only real cross platform programming language, providing us the tools to write interactive web applications and use them on the desktop, on our TVs, in our cars and on the mobile phones.
Building reusable, scalable and performance oriented JavaScript code requires a deep understanding of the language and as our applications grow more complex, this is no longer something we can live without.
In this course we will learn the tools and patterns to build scalable JavaScript web sites that will serve visitors from many platforms.
This course is for developers experienced in OOP with working knowledge of JavaScript, who wish to focus on advanced JavaScript techniques and patterns.
The course will consist of 70% lectures and 30% exercise.

On Completion, Delegates will be able to

- Develop scalable, reusable and testable JavaScript code as the infrastructure for large scale web applications

Who Should Attend

This course is intended for experience developers who wish to focus on rich client-side techniques and frameworks.

Prerequisites

Before attending this course, students must have:

- Previous experience in Object Oriented Programming (like: Java, C++ or C#)
- Basic knowledge in web concepts and HTML/CSS
- Basic JavaScript knowledge is highly recommended

JOHN BRYCE
Leading in IT Education
a matrix company

matrix

Course Contents

Module 1: JavaScript Pitfalls

- Typeof operator
- Undeclared vs. Uninitialized variable
- Implicit Variable declaration
- No integral data type
- String is immutable
- Undefined value
- Strict vs. Abstract comparison
- Logical Operator
- Array is dynamic
- Don't mix object with array
- Where to declare variables inside function?
- Function overloading
- Function is an object
- Function.apply vs. Function.call
- Function creates scope
- Closure
- Self-executing function

Module 2: Object Oriented JavaScript

- Module Pattern
- From Module to Class
- Function as Constructor
- Prototype
- Inheritance
- Namespace
- Objects and DOM
- me, self and that

Module 3: ECMA Script 5.0 & 6.0

- Strict Mode
- Object.create
- Getters and Setters
- Reflection
- let
- Class
- Module
- Iterator
- Generator
- Arrow function
- Binary data
- Collections
- Proxy
- Promise

Module 4: High Performance JavaScript

- Loading and Execution Optimizations
- Data Access Optimizations
- DOM Optimizations
- Flow Control Optimizations
- Strings Optimizations
- Regular Expression Optimizations
- Timers Optimizations
- UI Thread Optimizations
- Web Worker Optimizations
- AJAX Optimizations