



# JavaScript Advanced

Course ID #: 7000-367-ZZ-Z

Hours: 21

## Course Content

### Course Description:

This course teaches the advanced aspects of JavaScript. The material of the course covers elements that will enable students to create powerful programs using JavaScript. One of the main focuses of this class is the AngularJS JavaScript library that makes front-end web development consistent and web developers productive. This intensive training course covers both theoretical and technical aspects of front-end development using JavaScript. At the end of the course, attendees will have solid understanding of how to create and test JavaScript programs and how to write programs using AngularJS. The course is supplemented by numerous hands-on labs that help attendees reinforce their theoretical knowledge of the learned material.

### At Course Completion:

After completing this course, student will have learnt:

- Testing JavaScript programs
- JavaScript best practices
- AngularJS

### Prerequisites:

Participants should have the basic knowledge of the front-end web and JavaScript development.

### Target Student:

Web Designers and Developers.

### Topics:

#### Chapter 1. Testing JavaScript with Jasmine

- What is Jasmine
- Supported Integrations
- Jasmine in Standalone Mode
- Jasmine Folder Structure
- The Spec Runner Page
- Test Suites
- Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- More on Matchers
- Examples of Using Matchers
- Using the not Property
- Test Failures
- Setup and Teardown in Unit Test Suites
- Example of beforeEach and afterEach Functions
- Disabling Test Suites and Specs
- Method Stubbing and Call Tracking with Spies
- A Spy Example
- spyOn Setup Variations



# JavaScript Advanced

Course ID #: 7000-367-ZZ-Z

Hours: 21

- Simulating Exceptions
- Asynchronous Calls
- Summary

## Chapter 2. JavaScript Performance and Best Practices

- Leverage Modern JavaScript Libraries
- Loading Scripts
- JavaScript Code Minification
- Use JavaScript Style Guides
- Examples of JavaScript Rules and Style Guides Suggestions
- Avoid Global Variables, Use Proper Scopes
- Some Subtle Problems with Var
- The Block Scope with ECMAScript 6 Let
- Use Proper Scoping for Better Performance
- Example of Scoping
- Speeding up Programs with Proper Variable Scoping
- Cache Properties Accessed More Than Once
- Go Asynchronous
- Leverage HTML5 Web Workers
- Prefer Literals to Constructors
- Avoid Using eval()
- Performance Optimization Considerations
- Performance Profiling
- JavaScript Program Profilers in Browsers
- The IE Profiler
- IE Code Execution Profiler
- IE Memory Allocation Profiler
- Chrome Profiler
- Firebug with Firefox
- Clean-up Your Code
- Things a JavaScript Lint Normally Checks
- Use Strict Mode
- Use JOIN for Concatenation
- Equal or Not Equal
- Summary

## Chapter 3. Test Automation and Continuous Integration Systems for JavaScript

- Continuous Integration, Build and Test Automation
- What is Node.js
- Node Package Manager
- What is Grunt
- Excerpt from Gruntfile.js
- Installing Grunt
- What is Bower
- Installing Packages in Bower
- What is Karma
- How Karma Works
- Working with Browser Launchers
- Setting Up Path to Browser Binaries
- TeamCity Build Management System
- TeamCity Overview
- Summary

## Chapter 4. Introduction to AngularJS

- What is AngularJS?
- Why AngularJS?
- Scope and Goal of AngularJS
- Using AngularJS
- A Very Simple AngularJS Application
- Building Blocks of an AngularJS Application
- Use of Model View Controller (MVC) Pattern
- A Simple MVC Application
- The View
- The Controller
- Data Binding
- Basics of Dependency Injection (DI)
- Other Client Side MVC Frameworks
- Summary



# JavaScript Advanced

Course ID #: 7000-367-ZZ-Z

Hours: 21

## Chapter 5. AngularJS Module

- What is a Module?
- Benefits of Having Modules
- Life Cycle of a Module
- The Configuration Phase
- The Run Phase
- Module Wide Data Using Value
- Module Wide Data Using Constant
- Module Dependency
- Using Multiple Modules in a Page
- Summary

## Chapter 6. AngularJS Controllers

- Controller Main Responsibilities
- About Constructor and Factory Functions
- Defining a Controller
- Using the Controller
- Controller Constructor Function
- More About Scope
- Example Scope Hierarchy
- Using Scope Hierarchy
- Modifying Objects in Parent Scope
- Modified Parent Scope in DOM
- Handling Events
- Another Example for Event Handling
- Storing Model in Instance Property
- Summary

## Chapter 7. AngularJS Expressions

- Expressions
- Operations Supported in Expressions
- AngularJS Expressions vs JavaScript Expressions
- AngularJS Expressions are Safe to Use!
- What is Missing in Expressions
- Considerations for Using src and href Attributes in Angular
- Examples of ng-src and ng-href Directives
- Summary

## Chapter 8. Basic View Directives

- Introduction to AngularJS Directives
- Controlling Element Visibility
- Adding and Removing an Element
- Dynamically Changing Style Class
- The ng-class Directive
- Example Use of ng-class
- Setting Image Source
- Setting Hyperlink Dynamically
- Preventing Initial Flash
- Summary

## Chapter 9. Advanced View Directives

- The ng-repeat Directive
- Example Use of ng-repeat
- Dynamically Adding Items
- Special Properties
- Example: Using the \$index Property
- Scope and Iteration
- Event Handling in Iterated Elements
- The ng-switch Directive
- Example Use of ng-switch
- Inserting External Template using ng-include
- Summary

## Chapter 10. Working with Forms

- Forms and AngularJS
- Scope and Data Binding
- Role of a Form
- Using Input Text Box
- Using Radio Buttons
- Using Checkbox
- Using Checkbox - Advanced
- Using Select
- Using Select - Advanced
- Disabling an Input
- Reacting to Model Changes in a Declarative Way
- Example of Using the ng-change Directive
- Summary



## Chapter 11. Formatting Data with Filters in AngularJS

- What are AngularJS Filters?
- The Filter Syntax
- Angular Filters
- More Angular Filters
- Using Filters in JavaScript
- Using Filters
- A More Complex Example
- The date Filter
- The date's format Parameter
- Examples of Using the date Filter
- The limitTo Filter
- Using limitTo Filter
- Filter Performance Considerations
- Summary

## Chapter 12. AngularJS \$watch Scope Function

- The \$watch Function
- The \$watch Function Signature
- The \$watch Function Details
- Canceling the Watch Action
- Example of Using \$watch
- Things to be Aware Of
- More Things to Be Aware Of
- Performance Considerations
- Speeding Things Up
- Summary

## Chapter 13. Communicating with Web Servers

- The \$http AngularJS Service
- The Promise Interface
- The \$http Service
- \$http Function Invocation
- Callback Parameters
- Request Configuration Properties
- Shortcut Methods
- Complete List of Shortcut Methods
- Using \$http.get()
- Working with JSON Response
- Using \$http.post()

- Combining \$http POST Request Data with URL Parameters
- The then() Method of the Promise Object
- The Response Object
- Making Parallel Web Service Calls
- Combining Multiple Promises into One
- Wait for the Combined Promise
- Setting Up HTTP Request Headers
- Caching Responses
- Disabling Caching in IE9
- Setting the Request Timeout
- Unit Testing with ngMock
- Writing Unit Tests
- Summary

## Chapter 14. Custom Directives

- What are Directives?
- Directive Usage Types
- Directive Naming Convention
- Defining a Custom Directive
- Using the Directive
- Scope of a Directive
- Isolating Scope
- Example Scope Isolation
- Using External Template File
- Manipulating a DOM Element
- The Link Function
- Event Handling from a Link Function
- Wrapping Other Elements
- Accepting a Callback Function
- Supplying Callback Function
- Summary

## Chapter 15. AngularJS Services

- Introduction to Services
- Defining a Service
- The factory() Method Approach
- The service() Method Approach
- The provider() Method Approach
- Using a Service
- Configuring a Service using its Provider
- Summary



# JavaScript Advanced

Course ID #: 7000-367-ZZ-Z

Hours: 21

## Chapter 16. Unit Test using Jasmine

- Introduction to Node.js
- What is Jasmine?
- Running Jasmine
- Jasmine Folder Structure
- Running Tests
- Example Explained
- End-to-End Testing with Protractor
- Writing E2E Test with Protractor
- Summary