



Course Content

Course Description:

This course teaches the fundamentals of JavaScript, the language of the Web. The material of the course covers everything you need to know to become proficient using JavaScript. You will learn JavaScript's core elements, as well as some popular JavaScript libraries, such as jQuery, Underscore and others. The course also covers such topics as JavaScript best practices, program debugging and testing.

At Course Completion:

After completing this course, student will have learnt:

- JavaScript fundamentals
- Working with various popular JavaScript libraries (Underscore, jQuery, etc.)
- AJAX
- Testing JavaScript
- JavaScript performance
- JavaScript best practices

Prerequisites:

Participants should have the general knowledge of the front-end web development.

Target Student:

Web Designers and Developers.

Topics:

Chapter 1. Introduction to JavaScript

- What JavaScript Is
- What JavaScript Is Not
- Not All JavaScripts are Created Equal ...
- ECMAScript Language and Specification
- What JavaScript Can Do
- What JavaScript Can't Do
- JavaScript on the Server-side
- Elements of JavaScript
- Values, Variables and Functions
- Embedded Scripts
- External Scripts
- Browser Dialog Boxes
- Network Communication
- Summary



JavaScript Introduction

Course ID #: 7000-366-ZZ-Z

Hours: 21

Chapter 2. JavaScript Fundamentals

- Variables
- JavaScript Reserved Words
- Dynamic Types
- JavaScript Strings
- Escaping Control Characters
- What is False in JavaScript?
- Numbers
- The Number Object
- Not A Number (NaN) Reserved Keyword
- JavaScript Objects
- Operators
- Primitive Values vs Objects
- Flow Control
- 'if' Statement
- 'if...else' Statement
- 'switch' Statement
- 'for' Loop
- 'for / in' Loop
- 'while' Loop
- 'do...while' Loop
- Break and Continue
- Labeled Statements
- The undefined and null Keywords
- Checking for undefined and null
- Checking Types with typeof Operator
- Date Object
- Document Object
- Other Useful Objects
- Browser Object Detection
- The eval Function
- Enforcing Strict Mode
- Summary

Chapter 3. JavaScript DOM API

- What is DOM?
- Element Hierarchy
- DOM Standardization
- The Document Object
- Nodes and Elements
- The Element Object
- Element Event Handlers
- The window Object
- The Frame Object
- The History Object
- Summary

Chapter 4. JavaScript Functions

- Functions Defined
- Declaring Functions
- Function Arguments
- More on Function Arguments
- Return Values
- Multiple Return Values
- Default Parameter Values
- Anonymous Function Expressions
- Functions to Create Private Scope
- Linking Functions to Page Elements
- Common DOM Events
- Local and Global Variables
- Declaring Object Methods
- The arguments Parameter
- Example of Using arguments Parameter
- Summary

Chapter 5. JavaScript Arrays

- Arrays Defined
- Creating an Array
- The length Array Member
- Traversing an Array
- Appending to an Array
- Deleting Elements
- Inserting Elements
- Other Array Methods
- Accessing Objects as Arrays
- Summary



JavaScript Introduction

Course ID #: 7000-366-ZZ-Z

Hours: 21

Chapter 6. Using Lodash JavaScript Library

- What is Lodash
- Why use Lodash?
- Lodash Functional Categories
- Imperative Programming
- Functional Programming
- The JavaScript Map Examples
- The Collections Category
- The filter Collections Function
- The countBy Collections Function
- The Arrays Category Functions
- The union Arrays Function
- The difference Arrays Function
- The Functions Category
- The delay Function
- The once Function
- The Object Category Functions
- The
- keys
- Objects Category Function
- The Util Category Functions
- The
- random
- Util Function
- Summary

Chapter 7. Advanced Objects and Functionality in JavaScript

- Basic Objects
- Constructor Function
- More on the Constructor Function
- Object Properties
- Deleting a Property
- The instanceof Operator
- Object Properties
- Constructor and Instance Objects
- Constructor Level Properties
- Namespace
- Functions Are First-Class Objects
- Closures
- Closure Examples

- Private Variables with Closures
- Immediately Invoked Function Expression (IIFE)
- The Module Pattern
- Module Pattern Example
- Prototype
- Inheritance in JavaScript
- The Prototype Chain
- Traversing Prototype Property Hierarchy
- Prototype Chain
- Inheritance Using Prototype
- Extending Inherited Behavior
- Enhancing Constructors
- Improving Constructor Performance
- Inheritance with Object.create
- The hasOwnProperty Method
- Summary

Chapter 8. jQuery Overview

- What Is jQuery?
- Benefits of Using a JavaScript Library
- jQuery Example
- CSS Selectors
- How to Use jQuery
- Practical Usage Notes
- Background – DOM
- Background - DOM Ready Events
- Background - JavaScript Functions
- The jQuery Function Object
- What Does the \$() Function Take as Argument?
- What Does the \$() Function do?
- The jQuery Wrapper
- The jQuery Wrapper as an Array-Like Object
- Note: innerHTML() vs. .html()
- jQuery Wrapper Chaining
- API Function Notation
- Handling DOM Ready Event
- References
- Summary



JavaScript Introduction

Course ID #: 7000-366-ZZ-Z

Hours: 21

Chapter 9. Selectors

- Background: The Sizzle Selector Engine
- Selecting Elements by Attribute
- Pseudo-Selectors
- Form Pseudo-Selectors
- Faster Selection
- Selecting Elements Using Relationships
- Selecting Elements Using Filters
- More on Chaining: .end()
- Testing Elements
- Is the Selection Empty?
- Saving Selections
- Iterating Through Selected Elements Using .each()
- JavaScript Methods
- JavaScript "this"
- Function Context
- The Function call() Method
- .each() Revisited
- Summary

Chapter 10. DOM Manipulation

- The \$ Function Revisited
- Getters and Setters
- The text() Element Method
- Appending DOM Elements
- Removing DOM Elements
- Performance
- Summary

Chapter 11. Events

- Event Overview
- Old School: Event Handling Using HTML Element Attributes
- Unobtrusive JavaScript
- Unobtrusive JavaScript Example
- Multiple Handlers
- Using jQuery Wrapper Event Registration Methods
- The .on() Method
- Event Propagation
- Handlers for Elements Before They Exist!

- The Event Object
- Triggering Events
- Summary

Chapter 12. Utility Functions

- The jQuery Object Revisited
- Functions May Have Methods
- A jQuery Utility Function: \$.trim()
- \$.each()
- Example jQuery Utility Functions
- Summary

Chapter 13. Network Requests

- Network Requests
- HTTP Servers
- The HTTP Request
- The HTTP Response
- Using the XMLHttpRequest Object
- Requests Using jQuery .get()
- jQuery .get() - Details
- Response Data Types
- Making Requests with jQuery .load()
- Using JavaScript fetch() to make HTTP calls
- The jQuery .data() method
- Summary

Chapter 14. The JavaScript Console and the console Object API

- JavaScript Console in Browsers
- Dev Tooling Support in Browsers
- The Developer Tools Console
- The Console API
- Coloring Your Output
- Using the time () / timeEnd () Console Commands
- Using the console.trace() Command
- Using the console.count() Console Command
- Using the console.assert() Console Command
- The console.assert() Console Command Example
- Summary



JavaScript Introduction

Course ID #: 7000-366-ZZ-Z

Hours: 21

Chapter 15. Debugging JavaScript Code

- Why to Debug JavaScript Code?
- Debugging Tools in Browsers
- Chrome's DevTools
- The Debugging Controls on the Sources Panel
- Debugging Actions
- Additional Debugging Actions
- Selecting a JavaScript File To Debug
- Debugging Scenarios for Breakpoints
- Setting and Removing Breakpoints
- Information Available on a Breakpoint
- Toggling Breakpoints
- Other Ways to Manage Breakpoints
- Programmatic Control During Debug Pauses
- Breaking on Exceptions
- Other Debugging Techniques
- Breakpoints on JavaScript Event Listeners
- Supported Events
- Summary

Chapter 16. Exception Handling in JavaScript

- Exception Handling
- Try Syntax
- The Finally Block
- The Nested Try Blocks
- Exceptions Types in JavaScript
- The Throw Statement
- Using the Error Object
- Summary

Chapter 17. Testing JavaScript with Jasmine

- What Is Jasmine?
- Supported Integrations
- Jasmine in Standalone Mode
- Jasmine Folder Structure
- The Spec Runner Page
- Viewing Test Results
- 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
- Simulating Exceptions
- Asynchronous Calls
- Summary

Chapter 18. 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



JavaScript Introduction

Course ID #: 7000-366-ZZ-Z

Hours: 21

- IE Code Execution Profiler
- IE Memory Allocation Profiler
- Chrome Profiler
- Clean-up Your Code
- Things a JavaScript Lint Normally Checks
- Use Strict Mode
- Use JOIN for Concatenation
- Equal or Not Equal
- Summary

Chapter 19. 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
- Continuous Integration/ Delivery Tools
- TeamCity Build Management System
- TeamCity Overview
- Summary

Lab Exercises

- Lab 1. Intro to JavaScript
- Lab 2. Basic JavaScript Syntax
- Lab 3. JavaScript Functions
- Lab 4. Arrays in JavaScript
- Lab 5. Using the Lodash JavaScript Library
- Lab 6. Objects in JavaScript
- Lab 7. Getting Started with jQuery
- Lab 8. More on Selectors
- Lab 9. Dynamic Style Class Assignment
- Lab 10. DOM Manipulation
- Lab 11. Form Event Handling
- Lab 12. Network Requests with jQuery
- Lab 13. The JavaScript Console and the console Object API
- Lab 14. Debugging JavaScript Programs
- Lab 15. Exception Handling in JavaScript
- Lab 16. Testing JavaScript with Jasmine
- Lab 17. JavaScript Performance Profiling