

Intermediate JavaScript

Duration: 2 Days Course Code: JSI Delivery Method: Virtual Learning

Overview:

English - Please note this course is only available in English.

Nederlands - Let op: deze training is alleen in het Engels beschikbaar.

Franais - Veuillez noter que ce cours est uniquement disponible en anglais. Intermediate JavaScript Course Overview

JavaScript is one of the most popular and commonly used programming languages today. Not only is it the language of the web and front-end frameworks like Angular and React, but it can now be used to build anything (front-end, back-end, or otherwise) thanks to runtimes like NodeJS. This course is designed to move delegates beyond the fundamentals. It serves to set delegates on course to dive into one of the many JavaScript frameworks. Delegates will learn about modern syntax, object-orientation, functional programming, asynchronous programming, managed apps, and testing.

Exercises and examples are used throughout the course to give practical hands-on experience with the techniques covered.

NB: This course assumes ES6 as the baseline specification for JavaScript. This is because ES6 is now widely supported.

Virtual Learning

This interactive training can be taken from any location, your office or home and is delivered by a trainer. This training does not have any delegates in the class with the instructor, since all delegates are virtually connected. Virtual delegates do not travel to this course, Global Knowledge will send you all the information needed before the start of the course and you can test the logins.

Target Audience:

Who will the Course Benefit?

The Intermediate JavaScript course is aimed at JavaScript developers who have a handle on the fundamentals and want to take the next step. It may also be of value to those who wish to refresh/renew their knowledge of JavaScript after some time away from it.

Objectives:

Course Objectives

- This course aims to provide delegates with the skills and knowledge necessary to build modern, managed JavaScript applications that exploit modern syntax. Delegates should be able to devise object-oriented and functional solutions to problems, and deal with asynchronous tasks such as IO.

Prerequisites:

- Delegates should be able to produce scripts that exploit the fundamental features of the language including variables, functions, conditions and loops, DOM manipulation and event handling. Delegates should also be familiar with HTML and CSS.

Follow-on-Courses:

Further Learning

- React
- Angular Development
- Developing Node.js Web Applications

Content:

Intermediate JavaScript Training Course Course Contents - DAY 1

Course Introduction

- Administration and Course Materials
- Course Structure and Agenda
- Delegate and Trainer Introductions

Session 1: MODERN SYNTAX, OPERATORS, AND DATA STRUCTURES

- Destructuring
- Spread
- Rest parameters
- Short circuiting
- Logical assignment operators
- Enhanced object literals
- Optional chaining
- Nullish coalescing
- Sets and Maps

Session 2: SCOPE

- What is scope?
- Types of scope
- Lexical scope
- The scope chain
- Closures
- IIFEs

Session 3: OBJECT-ORIENTED PROGRAMMING

- What is object-oriented programming?
- Object templates
- Constructor functions
- The this keyword (review)
- Object prototypes (briefly)
- Classes
- Fields, constructors, and methods
- Static members
- Encapsulation
- Inheritance
- Composition
- Mutating arrays, e.g. push, splice etc.

Session 4: FUNCTIONAL PROGRAMMING

- Functions (review)
 - What is functional programming?
 - Pure functions
 - Immutability
 - Spread (again)
 - Functions as first-class objects
 - Higher-order functions
 - Callbacks
 - Transforming arrays, e.g. filter, map etc.
- Intermediate JavaScript Training Course
Course Contents - DAY 2

Session 5: ASYNCHRONOUS PROGRAMMING

- What is asynchronous programming?
- The Event Loop
- Asynchronous callbacks
- setTimeout and setInterval
- Promises
- async and await
- Reactive programming and observables (briefly)
- Asynchronous programming and the fetching of data

Session 6: MODULES, MANAGED APPS, AND TOOLING

- What is a module?
- Exporting and importing module features
- Embedding a module in the HTML
- Default and named exports
- Aliases
- Module objects
- Modules, packages, libraries, and frameworks
- Categories of tools
- Package management
- Module bundling
- Transpilation
- TypeScript
- Creating a managed app

Session 7: TESTING

- The purpose of testing
- The types of testing
- Running tests
- Test tools
- Unit testing
- Integration testing
- E2E testing
- Testing async code
- Using mocks

Further Information:

For More information, or to book your course, please call us on 0800/84.009

info@globalknowledge.be

www.globalknowledge.com/en-be/