olsen
software ltd

☎ +44 7989 401397
✉ info@olsensoft.com

Modern JavaScript Development   (3 days)

Course overview

This course takes a detailed look at ECMAScript 10 (aka ES10 or ES2019), the current version of the ECMAScript (JavaScript) standard.

JavaScript has undergone a transformation in the last few years. JavaScript now has full support for object-oriented programming, functional programming (via arrow functions), asynchronous programming, and modular development (via CommonJS modules). JavaScript also now has some extremely handy syntactic features such as destructuring, the spread operator, and generators.

This course takes a thorough look at all the new features in JavaScript from ES6 to ES10, and also shows how to transpile into ES5 for compatibility with all browsers.

What you'll learn

- Arrow functions and other function enhancements
- Destructuring
- Array and object enhancements
- Object-oriented programming using classes and inheritance
- Iterables, iterators, and generators
- API enhancements
- Asynchronous working
- Modules

Prerequisites

- Familiarity with HTML and CSS
- Experience with earlier versions of JavaScript beneficial

Course details

- **Getting Started:** Overview of ES6 onwards; Goals of ES6 onwards; Tool support
- **Core Features in ES6 Onwards:** Variables, scope, and string literals; Function enhancements; Arrow functions and functional programming
- **Destructuring:** What is destructuring; Array destructuring; Object destructuring; Destructuring techniques
- **Miscellaneous New APIs:** Number and Math enhancements; String enhancements; Regular expression enhancements
- **Array Enhancements:** Core Array enhancements; Typed arrays
- **Object Enhancements:** Object literal enhancements; Symbols; New methods in Object

- Classes: Defining classes; Encapsulation; Statics; Inheritance

- Iterables, Iterators, and Generators: Introduction to iterables and iterators; Implementing iterable objects; Generators

- Modules: Introduction to modules; Importing and exporting syntax; Additional techniques

- Asynchronous Working: Introduction to Promises; Using Promise objects; Example of Promises; Async functions; Asynchronous iteration; Shared memory and atomics

- Classes: Defining classes; Encapsulation; Statics; Inheritance

- Iterables, Iterators, and Generators: Introduction to iterables and iterators; Implementing iterable objects; Generators