

## INT303: Cursus Advanced JavaScript Programmeren

Code: INT303

Duur: 3 dagen

Prijs: € 1650

### Doelgroep Cursus Advanced JavaScript Programmeren

De cursus Advanced JavaScript Programmeren is bedoeld voor ervaren Web developers die geavanceerde aspecten van JavaScript in Web Applications willen leren en toepassen.

### Voorkennis Cursus Advanced JavaScript Programmeren

Basis kennis van JavaScript en HTML is vereist. Voorafgaande kennis van Object Oriented Programming is bevordelijk voor een goede begripsvorming.

### Uitvoering Cursus Advanced JavaScript Programmeren

De cursus heeft een hands-on karakter. De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met demos en praktische oefeningen. De cursustijden zijn van 9.30 tot 16.30.

### Officieel Certificaat Advanced JavaScript Programmeren

De deelnemers krijgen na het doorlopen van de cursus een officieel certificaat Advanced JavaScript Programmeren.

### Inhoud Cursus Advanced JavaScript

De cursus Advanced JavaScript Programmeren start met een behandeling van scope in JavaScript, waarbij het verschil tussen var, let en const variabele declaraties en het begrip hoisting wordt besproken. De betekenis en werking van het context afhankelijke keyword this wordt hierbij ook aan de orde gesteld. Vervolgens wordt aandacht besteed aan de vele nieuwe features die hun intrede hebben gedaan met de ECMAScript 2015 (ES6) standaard, zoals arrow of lambda functies, rest operators, spread operators en destructuring assignments. De veranderingen met de ECMAScript 2016 (ES7) standaard komen aansluitend aan de orde. Dan is er aandacht voor Object Georiendeerd programmeren in JavaScript dat nog steeds gebaseerd is op prototypes, maar sinds ES6 geprogrammeerd wordt met classes en objects. De JavaScript object orientatie is daardoor meer in lijn met wat gebruikelijk is in andere talen. Vervolgens komen een aantal geavanceerde aspecten van functies zoals functie objecten, immediate invocation, callbacks, closures en recursie aan de orde en is er aandacht voor geavanceerde functionele operaties op arrays zoals filter, map en reduce. Na een systematische bespreking van hoe in JavaScript XML en JSON data kunnen worden benaderd, komen asynchrone Ajax calls met het XMLHttpRequest Object en de verwerking daarvan met Promises aan de orde. Aansluitend wordt aandacht besteed aan nieuwe Web Component specificaties, zoals shadow en shady DOM, HTML templates en HTML Imports. Ook Module Systemen zoals CommonJS en AMD zijn onderwerpen die op het programma staan. De cursus wordt afgesloten met een bespreking van het gebruik van Reguliere Expressies in JavaScript voor onder andere input validatie.



## Advanced JavaScript Programming



Module 1 : Scope and Context	Module 2 : ECMAScript 6 and 7	Module 3 : Classes and Objects
Scope in JavaScript var Keyword Scope in Nested Function Lexical Scope Hoisting Nested Anonymous Function let Keyword const Keyword Immutability Refresher on this this in Global Scope Lexical this Binding this apply and call Function bind Function	ECMAScript 6 Arrow Functions Object Literal Syntax Rest Parameters Spread Operator Template Strings Iteration with for..of Destructuring Assignments Generators Symbols Built-in Symbols ECMAScript 7 Array.prototype.includes Exponentiation Operator Shared Array Buffer	Object Oriented Programming Objects in JavaScript Constructor Functions Object Ancestor _proto_ Property Prototype Chains Prototype Keyword Composing Prototypes ES2015 Classes Constructors Static Keyword Static Methods Inheritance Overriding Methods super Keyword
Module 4 : Advanced Functions	Module 5 : Advanced Arrays	Module 6 : XML and JSON Access

<ul style="list-style-type: none"> <li>Function Declarations</li> <li>Function Expressions</li> <li>Default Parameters</li> <li>Variable Arguments</li> <li>Argument Object</li> <li>Array Conversion</li> <li>Closures</li> <li>Immediate Invocation and IIFE's</li> <li>Callbacks</li> <li>Recursion</li> <li>Runtime Errors</li> <li>Globally Handled Errors</li> <li>Structured Error Handling</li> <li>Try and Catch</li> </ul>	<ul style="list-style-type: none"> <li>Arrays Review</li> <li>Iterators</li> <li>Arrow Functions</li> <li>Array.isArray()</li> <li>Array.from()</li> <li>Array.of()</li> <li>Array.prototype.fill()</li> <li>Array.prototype.filter()</li> <li>Array.prototype.find()</li> <li>Array.prototype.forEach()</li> <li>Array.prototype.indexOf()</li> <li>Array.prototype.keys()</li> <li>Array.prototype.map()</li> <li>Array.prototype.reduce()</li> </ul>	<ul style="list-style-type: none"> <li>Document Object Model (DOM)</li> <li>Properties and Methods</li> <li>DOM Navigation</li> <li>DOM Manipulation</li> <li>DOM Node Types</li> <li>Querying the DOM</li> <li>What is JSON?</li> <li>JSON Data Structures</li> <li>JSON Object Representation</li> <li>JSON Array</li> <li>JSON with Array and Objects</li> <li>JSON Conversion</li> <li>JSON Parser</li> <li>JSON Object Creation</li> </ul>
<b>Module 7 : Ajax and Promises</b>	<b>Module 8 : Components and Modules</b>	<b>Module 9 : Regular Expressions</b>
<ul style="list-style-type: none"> <li>Ajax Asynchronous Model</li> <li>Creating XMLHttpRequest</li> <li>Sending Requests</li> <li>XMLHttpRequest Properties</li> <li>Processing the Response</li> <li>Accessing Remote Sites</li> <li>Same Origin Policy</li> <li>Cross Site Request Sharing</li> <li>JSON with Padding</li> <li>Promises</li> <li>Promises Guarantees</li> <li>Promise Chains</li> <li>Chaining after Catch</li> <li>ES7 Async Await</li> </ul>	<ul style="list-style-type: none"> <li>Web Component Specifications</li> <li>Custom Elements</li> <li>Creating and Extending</li> <li>HTML Templates</li> <li>Activating Templates</li> <li>HTMLImports</li> <li>Shadow DOM</li> <li>Shady DOM</li> <li>Module Systems</li> <li>CommonJS Module System</li> <li>Asynchronous Module Definition</li> <li>Internal Modules</li> <li>External Modules</li> <li>Imports and Exports</li> </ul>	<ul style="list-style-type: none"> <li>JavaScript's RegExp</li> <li>String Methods</li> <li>Regular Expression Syntax</li> <li>Start and End ( ^ \$ )</li> <li>Number of Occurrences</li> <li>Grouping ( [ ] )</li> <li>Negation ( ^ )</li> <li>Subpatterns ( ( ) )</li> <li>Alternatives (   )</li> <li>Escape Character ( \ )</li> <li>Backreferences</li> <li>Named Backreferences</li> <li>Form Validation</li> <li>Cleaning Up Form Entries</li> </ul>