



Agenda - Beispiel

Dies ist unsere empfohlene Agenda für das Thema JavaScript. Wir verfügen darüber hinaus über weiteres Kursmaterial, um Themen einfach auszutauschen.

Grundlagen (Tag 1)

Der JavaScript-Workshop beginnt mit den Grundlagen, vermittelt aber am ersten Tag auch fortgeschrittene Konzepte wie den Umgang mit Asynchronität in JavaScript.

JavaScript Crashkurs

- Syntax, Datentypen
- Typsystem, Typumwandlungen
- Vergleichsoperatoren
- Funktionen und Scopes
- Closures
- Funktionen höherer Ordnung
- Objekte und Arrays im Detail
- Prototypen und Objektorientierung
- Exception Handling

Asynchrones JavaScript

- Wie funktioniert Asynchronität in JavaScript?
- Promises
- Funktionale Reaktive Programmierung (FRP)
- Kommunizierende Sequenzielle Prozesse (CSP)
- Asynchrone Sequenzen mit ES6 Generatoren
- Tools und Hilfsmittel

Hilfreiche Bibliotheken

- Warum wir jQuery brauchen
- jQuery 101

- Underscore und Lo-Dash
- Weitere hilfreiche Bibliotheken

OOP und Ökosystem (Tag 2)

Am zweiten Tag geht es in erster Linie darum, zu vermitteln, wie objektorientierte Programmierung in JavaScript funktioniert und welche Möglichkeiten es gibt, um von einem Modulsystem zu profitieren.

OOP im Detail

- Prototypen im Detail
- Wie funktioniert this?
- Objektliterale
- Mixins und Mixin-Techniken
- Konstruktor-Funktionen
- Methoden von Objekt A auf Objekt B anwenden
- Konstruktor-Vererbung & Funktionale Mixins
- DIY-Klassen (z.B. klass.js)
- ES5-APIs (Object.create())
- ES6-Klassen

Modulsysteme

- CommonJS und NPM
- AMD und Bower
- ES6-Module
- Tools und Optimierung
- insb. Browserify und RequireJS

Fortgeschrittene Debugging-Techniken

- Browser-Dev-Tools
- Source Maps
- Mobile Debugging

JS der Zukunft (Tag 3)

Tag 3 wird bestimmt von dem Thema ECMAScript 6 und dem Ökosystem, das sich um die neue Sprachversion gebildet hat. Ein perfekter Einstieg in die zukünftigen Möglichkeiten von JavaScript.

JavaScript als wandelbare Sprache

- JS-Parser und Codegeneratoren
- JavaScript-Dialekte
- Makros mit SweetJS
- Typisiertes JavaScript
- insb. TypeScript und Flow

ECMAScript-Ausblick

- ECMAScript-Historie
- Neuer syntaktischer Zucker
- Block Scope und Konstanten
- Neue Datenstrukturen (Maps, Sets, Weak Maps)
- Generatoren
- ES6-Promises und APIs mit Promises
- Arrow Functions
- Modulsystem
- Template Strings
- Proxies
- Klassen
- ES7-Ausblick
- Tooling (Traceur und andere Transpiler)