

Program szkolenia:

Nowoczesne standardy JavaScriptu

Informacje:

Nazwa:	Nowoczesne standardy JavaScriptu
Kod:	JS-ESx
Kategoria:	JavaScript
Odbiorcy:	developerzy, architekci
Czas trwania:	3 dni
Forma:	33% teoria, 66% ćwiczenia

Szkolenie przeznaczone jest dla programistów mających doświadczenie z klasycznym JavaScript (ES5), chcących zacząć skutecznie używać nowych standardów języka, jak również dla osób rozpoczynających pracę w JavaScriptcie.

Każda z części szkolenia jest najpierw omawiana teoretycznie, a następnie uczestnicy rozwiązują liczne ćwiczenia w stylu Test-Driven Development. Ćwiczenia te badają zrozumienie danego zagadnienia oraz umiejętność samodzielnego rozwiązywania problemów. Kładą nacisk na dobre praktyki programistyczne. Są także „naszpikowane” klasycznymi pułapkami, w jakie programiści często wpadają – pod okiem trenera zdobywają „doświadczenie w boju”, mające na celu uchronienie przed popełnianiem błędów w projektach.

Zalety szkolenia:

- Praktyczne zastosowanie ES6 w codziennej pracy
- Porównanie mechanizmów ES6/7/8 z odpowiednikami ES5
- Nauka w oparciu o TDD

Szczegółowy program:

1. Basics and amp; Syntax

1.1. var, let and amp; const keywords

1.2. Symbols

1.3. Destructuring Objects and amp; Arrays

1.4. Spread and Rest Operators

1.5. Default Parameters

1.6. Template Literals

1.6.1. Tagged template literals

2. Arrow functions

2.1. Lexical Scope

2.2. ES5 functions vs ES6 arrow functions

2.2.1. Managing function context (this)

2.3. Advanced usage, common mistakes

2.4. Functional programming with Arrow Functions

3. Classes

3.1. Prototypal inheritance in ES6 classes

3.2. Class Properties

3.3. ES5 constructors, ES6 classes, TypeScript classes – comparison

3.4. OOP and FP – two approaches in JavaScript

4. Iterators

4.1. for..of loops

4.2. Iteration Protocol

4.3. Generators

5. Modules

5.1. Imports, exports

5.2. ESM, CommonJS, AMD - comparison

6. Promises

6.1. Asynchronicity in JS (event loop, message queue, ...)

6.2. Basic promise usage

6.3. Parallel and Sequential processing

6.4. Aggregate operations

6.5. Deferred anti-pattern (deferred vs ES6 promise)

6.6. Best practices and anti-patterns

6.7. ES6 promises vs jQuery, bluebird

7. Async Functions

7.1. Coroutines

7.1.1. coa, bluebird.coroutine

7.2. Async Await

8. Automation

8.1. Transpiling with Babel

8.2. Unit Testing with Karma preprocessors