

Modern JavaScript

Ducin IT
Consulting

ES2015 - ES2023

Ducin IT Consulting - Program szkolenia

Czas trwania: **2-3 dni**

Formuła: **25% teoria, 75% praktyka**

Szkolenie przeznaczone jest dla programistów mających pewne doświadczenie z JavaScript, 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.

Kluczowe Aspekty:

- Praktyczne zastosowanie nowoczesnego JavaScript w codziennej pracy
- Dogłębne zrozumienie działania obiektów (OOP) i funkcji (FP)
- Nauka w oparciu o Test-Driven Development

Program szkolenia:

1. JS Evolution: past, present, future

- 1.1. Browsers, node.js: standards & compatibility
- 1.2. Polyfills, transpiling
- 1.3. Runtime environments
- 1.4. tc39, proposals system

2. Syntax

- 2.1. var, let & const keywords
- 2.2. (Tagged) Template Literals
- 2.3. Nullish Coalescing
- 2.4. Optional Chaining
- 2.5. Logical Assignments

3. Objects

- 3.1. Properties and Descriptors
- 3.2. Symbols
- 3.3. Type operators
- 3.4. Destructuring Objects & Arrays
- 3.5. Spread and Rest Operators
- 3.6. Object: keys, values, entries, fromEntries
- 3.7. Object: freeze, seal
- 3.8. JSON format
- 3.9. Maps, Sets
- 3.10. WeakMaps, WeakSets, WeakRefs
- 3.11. Proxies

4. Arrow functions

- 4.1. Function Scope vs Lexical Scope
- 4.2. Function (keyword) vs ES6 arrow functions
- 4.3. Function Context (this)
- 4.4. Advanced usage, common mistakes
- 4.5. Functional programming

5. Classes

- 5.1. Prototypes and Prototypal Inheritance
- 5.2. Class members and Static members
- 5.3. Private properties, methods and accessors
- 5.4. OOP and FP – two approaches in JavaScript

6. Iteration Protocol

- 6.1. Generators
- 6.2. Iterators
- 6.3. for..of loops
- 6.4. Custom Iterators
- 6.5. Async Iterators

7. Modules

- 7.1. Imports, exports
- 7.2. ESM, CommonJS, AMD - comparison
- 7.3. Dynamic Imports

8. Asynchronous JavaScript

- 8.1. Asynchronicity in JS (event loop, message queue, ...)
- 8.2. Promises: Parallel and Sequential processing
- 8.3. Promise combinators: all, race, any, allSettled

- 8.4. Best practices and anti-patterns
- 8.5. native promises vs jQuery, bluebird
- 8.6. Async Await
- 8.7. Top-level await
- 8.8. Async 'Iteration

9. (Optional) Design Patterns in JavaScript

- 9.1. Strategy
- 9.2. Command + Command Handler
- 9.3. Builder
- 9.4. Mediator
- 9.5. Memento
- 9.6. Observer + Publisher-Subscriber
- 9.7. Bridge
- 9.8. Decorator
- 9.9. Iterator
- 9.10. Template Method

10. Automation

- 10.1. Transpiling with Babel
- 10.2. Unit Testing with Karma/Jasmine and Jest