

## Program szkolenia:

# AngularJS

### Informacje:

<b>Nazwa:</b>	<b>AngularJS</b>
<b>Kod:</b>	<b>JS-AngularJS</b>
<b>Kategoria:</b>	JavaScript
<b>Grupa docelowa:</b>	developerzy
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

---

Szkolenie przeznaczone jest dla programistów i projektantów JavaScript pragnących poszerzyć swoje praktyczne umiejętności o AngularJS.

Wszystkie zagadnienia omawiane są podczas wspólnego tworzenia aplikacji.

### Zalety szkolenia:

- Zawiera wzorce i najlepsze praktyki
- Aspekty architektury

## Szczegółowy program:

### 1. Fundamenty JavaScript

#### 1.1. Functional Programming

##### 1.1.1. Function as first-class citizen

##### 1.1.2. Scopes

##### 1.1.3. Closures

##### 1.1.4. Contexts

##### 1.1.5. Event Loop, Run to completion

#### 1.2. Dynamic, Weak typing

#### 1.3. Object Oriented Programming

##### 1.3.1. Prototypes vs Classes

##### 1.3.2. Prototype inheritance and extending

##### 1.3.3. Encapsulation

### 2. Architektura Aplikacji Webowych

#### 2.1. HTTP, AJAX, Client-Server model

#### 2.2. API Contracts

#### 2.3. Modele programowania

##### 2.3.1. Synchroniczny

##### 2.3.2. Asynchroniczny

##### 2.3.3. Współbieżny

##### 2.3.4. Rozproszony

#### 2.4. Asynchroniczność: Callbacks, Events, Promises, RxJS

### 3. Wstęp do AngularJS

#### 3.1. Idea, zastosowanie, mocne i słabe strony frameworka

3.2. Wzorce Projektowe i Najlepsze Praktyki

3.3. Kluczowe elementy

## 4. Fundamenty AngularJS

4.1. Dependency Injection

4.2. Digest Cycle

4.3. Data Binding

4.4. Module Structure

## 5. Konfiguracja

5.1. Application, Modules

5.2. Values, Constants

## 6. Widok

6.1. Wyrażenia

6.2. Szablony

6.2.1. Template Cache

6.3. Filtry

6.3.1. Filtry wbudowane

6.3.2. Definiowanie własnych filtrów

## 7. Logika

7.1. Services, Factories, Providers

7.2. Wbudowane serwisy

7.3. Definiowanie własnych serwisów

## 8. Dyrektywy

8.1. Idea, Design

8.2. Rodzaje dyrektyw, restrictions

8.3. Controllers

8.4. Scopes

8.5. Transclusion

8.6. Isolated Scope

8.7. Wbudowane dyrektywy

8.8. Definiowanie własnych dyrektyw

## 9. Komunikacja między komponentami

9.1. Service-based

9.2. Event-based

9.3. Root scope

9.4. Controller hierarchy and inheritance

## 10. Automatyzacja

10.1. Cele, Rodzaje tasków oraz narzędzi

10.2. NPM, Bower

10.3. Grunt, Gulp, NPM scripts

10.4. Yeoman

## 11. Warstwy aplikacji

11.1. Struktura aplikacji

11.2. Bazowy setup (boilerplate)

11.3. Warstwa prezentacji (ui-bootstrap)

11.4. Warstwa modelu (\$http, \$resource, restangular)

11.5. Routing (ui-router)

## 12. Typowe problemy i błędy

12.1. Performance

12.2. Dependency injection, minifikacja

12.3. Konfiguracja i ładowanie modułów

## 13. Zaawansowane koncepty frontendowe

13.1. Backend-less development

13.2. Contract-First Design