

## Program szkolenia:

# Angular i TypeScript

### Informacje:

<b>Nazwa:</b>	<b>Angular i TypeScript</b>
<b>Kod:</b>	<b>ang-AngularTS</b>
<b>Kategoria:</b>	Angular
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	3-5 dni
<b>Forma:</b>	25% wykłady / 75% warsztaty

---

Szkolenie przeznaczone jest dla programistów i projektantów JavaScript pragnących poszerzyć swoje praktyczne umiejętności o Angular w wersji 2+ i TypeScript.

**Program szkolenia jest ogólną ramą - konkretne szkolenie poprzedzamy analizą przed-szkoleniową.**

### Zalety szkolenia:

- Zawiera wzorce i najlepsze praktyki
- Aspekty architektury aplikacji i systemu
- Popularne i sprawdzone biblioteki

## Szczegółowy program:

### 1. Angular 2

1.1. Architecture

1.2. Building blocks

1.2.1. Modules

1.2.2. Components

1.2.3. Templates

1.2.4. Metadata

1.2.5. Data Binding

1.2.6. Services

1.2.7. Directives

1.2.8. Dependency Injection

1.3. Best and worst practices

1.3.1. Architecture

1.3.2. Design

1.3.3. Style guides (project structure, conventions and more)

1.3.4. Common mistakes developers make

1.4. Real Apps

1.4.1. Real projects challenges and problems

1.4.2. Non trivial examples

1.5. Best additional external resources

### 2. Modern JS Tools

2.1. Introduction to TypeScript (TS)

2.2. Introduction to ECMAScript 6(ES6/ES2015)

### 3. Modules

3.1. Library modules

3.2. Custom modules

3.3. ng2 Barrels and Bundles

### 4. Components

4.1. Lifecycle

4.2. Lifecycle hooks

### 5. Templates

5.1. Syntax

5.2. Binding targets

5.3. Local template variable

5.4. Inputs, outputs

### 6. Pipes

6.1. Built in

6.2. Custom

6.3. Stateful pipes

### 7. Views

7.1. DOM rendering

7.2. Server-side rendering

7.3. Best practices

### 8. Metadata

8.1. Syntax

8.2. TS decorators (@Component, @Directive, @Injectable, @Input, @Output, @RouterConfig, and more)

### 9. Data Binding

9.1. Interpolation

9.2. Data binding directions

9.3. Property binding

9.4. Event binding

## 10. Services

10.1. Responsibilities

10.2. Built in services

10.3. Custom services

10.4. Best practices

10.5. Common mistakes

## 11. Directives

11.1. Built in directives

11.2. Custom directives

11.3. Common pitfalls

11.4. Best and worst practices

## 12. Dependency Injection

12.1. Injectors

12.2. Providers

12.3. Hierarchical injectors

## 13. Routing and Navigation

13.1. Router config

13.2. Router navigation

13.3. Navigating under program control

13.4. Router lifecycle hooks

## 14. Testing

14.1. Introduction to Jasmine.js

14.2. Introduction to Protractor

14.3. Unit testing

14.4. Integration testing

14.5. E2E testing