

Course code: **ANGULAR**

Course title: **AngularJS Framework**

Days: 4

## Description:

### Course intended for:

The course is intended for programmers having basic knowledge of JavaScript, who would like to develop their skills in the field of development of interactive Web applications.

### Course objective:

The training objective is to get the participants familiar with the best practices in JavaScript programming, as well as the practical issues, associated with development of Web applications in the "Single-page application" infrastructure using the AngularJS framework.

### Requirements:

The participants are required to have at least the basic JavaScript programming skills. JavaScript.

### Course parameters:

4\*8 hours (4\*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

## Course curriculum:

### 1. JavaScript – advanced issues

- jQuery vs native implementation
  - Selectors
  - DOM manipulation
  - Event management



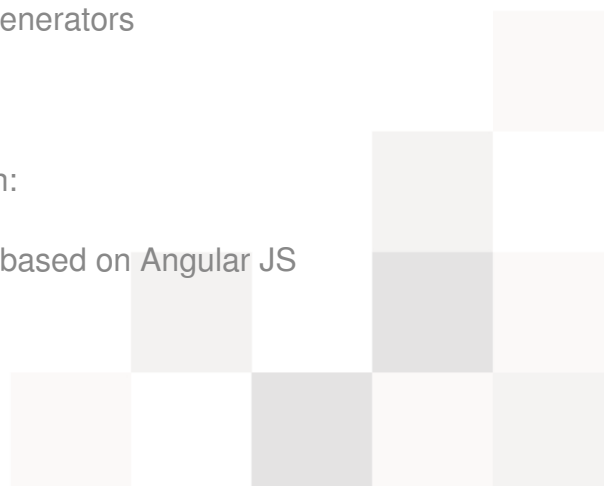
- Ajax
- Best practices
  - Debugging
  - Static code analysis
  - Strict mode
  - Prototypal inheritance
  - Hoisting
  - IIFE
  - Promises
  - Module pattern
  - ECMAScript 5 and lodash – function programming
  - ECMAScript 6 – where to start
- Unit tests
  - Karma
  - The most popular test frameworks

## 2. Front-end workflow

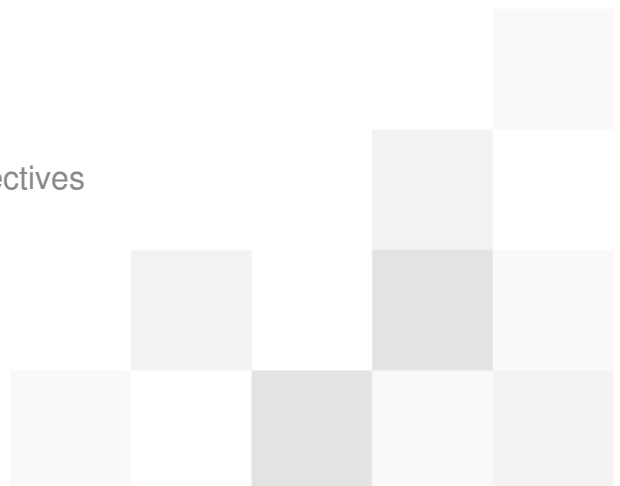
- Package management for server and browser
- Task automation
- Extensions enhancing work with HTML, CSS and JS
- Configuration and project structure generators

## 3. Angular JS

- What is the MVVW and MVW pattern:
  - Development of applications based on Angular JS



- Models in Angular JS:
  - Explaining of functioning of models in Angular JS
  - Linking of models in two and one direction
- Views and standard directives:
  - The principle of view functioning in Angular JS
  - Exemplary in-built directives and their practical use
  - Use of filters in views and outside
  - Building of own filters
  - Filter unit tests
- Services
  - What are services and what they are used for
  - Service injection: opportunities and threats
  - Use of services for data sharing between controllers
  - Various possibilities of service declaration
- Directives:
  - Discussing of directive types: components, decorators and structural
  - Inherited and isolated scope, data transmission to directive
  - Transclusion
  - preLink, postLink and compile functions
  - ngModelController
  - Embedded directives
  - Communication between directives
  - Unit tests of directives



- Forms:
  - Form development, use of in-built validators
  - Development of own validators
  - Asynchronous validators
  - Displaying information on data correctness
- Routing:
  - Template loading after URL change
  - Using many views for one and many controllers
  - Comparison of capabilities of ngRoute and UI Router
- Best practices:
  - Code structuring
  - Component configuration modes

