



Atos

Global Knowledge.®

# Angular Module 1 - Introduction

Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

WORLDWIDE LOCATIONS

BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR  
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

# Peter Kassenaar

On Peter Kassenaar:

- Trainer, author, developer – since 1996
- Specialty: "*Everything JavaScript*"
- JavaScript, ES6, Angular, NodeJS, TypeScript,

[www.kassenaar.com/blog](http://www.kassenaar.com/blog)

[info@kassenaar.com](mailto:info@kassenaar.com)

Twitter: [@PeterKassenaar](https://twitter.com/PeterKassenaar)





Angulartraining.nl

Home Training Dates Information Contact

2018 dates now available!

```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomeModule' },
  { path: 'training', loadChildren: './training/training.module#TrainingModule' },
];

const config: ExtraOptions = {
  enableTracing: false,
  preloadingStrategy: PreloadAllModules
};

@NgModule({
  imports: [RouterModule.forRoot(routes, config)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

# World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

LEARN MORE SIGN UP!

[www.angulartraining.nl](http://www.angulartraining.nl)

**About you...**



- How many of you have Front-end knowledge (HTML, CSS, JavaScript)?
- How many of you have (some) Angular experience?
- What are your expectations of this weeks course?

# **Agenda - Fundamentals, 3 days**

- Introduction & short history – Why Angular?
- Key features of Angular 2 to 9
- Hello World in Angular – Looking at the boilerplate-code - CLI
- Angular in depth (modules):
  - Components
  - ECMAScript 2015 + TypeScript
  - Data binding
  - Dependency Injection (DI) – more components
  - Services and Http, Observables (RxJS)
  - Routing, Forms, Unit testing
- BEST PRACTICES / STYLE GUIDE

# **Agenda - Advanced, 2 days**

- Working with multiple Modules
- Advanced Routing & Lazy loading
- Advanced components, content projection
- Unit Testing & End-2-end testing
- Introduction State Management
- ...

# Schedule, roughly...

- 9:00 ~ 12:00 Morning session
  - Coffee/tea break
- 12:00 – 13:30 Lunch
- 13:30 ~ 16:00-16:30 Afternoon session
  - Coffee/tea break
- Intense! Talks & hands-on workshops

# Materials

Software (NodeJS & NPM, Editor, browser)

*Océ specific batch script*

Handouts (Github, Océ Repo)

Workshops (Github)

Websites (online)



[angular.io/](http://angular.io/)

## 2 Guidelines

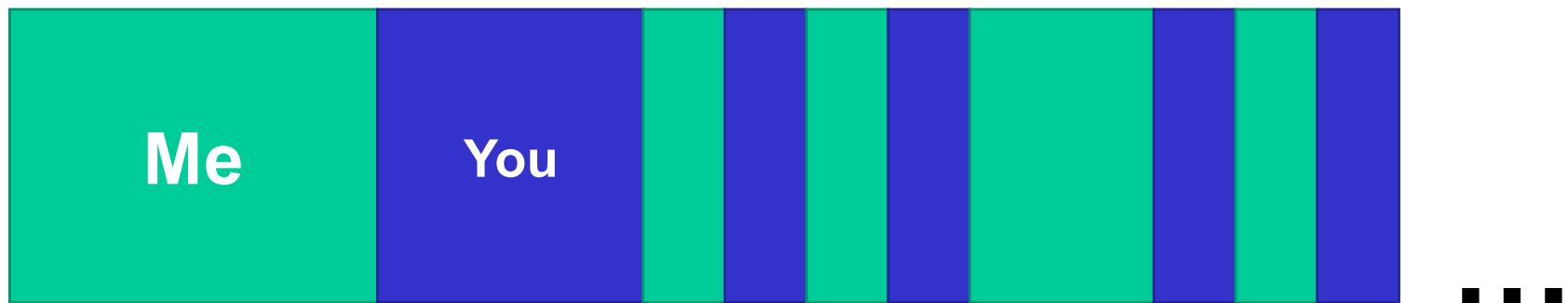
### 1. Workshops / Exercises

- But: get off the beaten path! Create your own project, app, website...

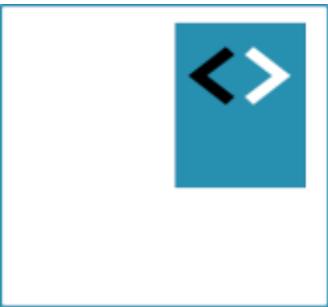
### 2. Example code – Updated to Angular V8.

- To support the exercises – ready made examples
- Work in progress – check Angular-site!
- <https://github.com/PeterKassenaar/voorbeeldenAngular2> (Dutch)

# How I work...



# Questions?



# AngularJS vs. Angular 2-8

Key features, differences  
And similarities



# A look at front-end frameworks

What is a good choice, what is popular?

# Addressing the “WHY” question!

WHY, would we want to use a frontend framework.

It is all HTML, CSS and JavaScript right?

Rethorical question:

*“Do we want to go back  
to the jQuery days?”*

speed,

SPA,

consistency,,

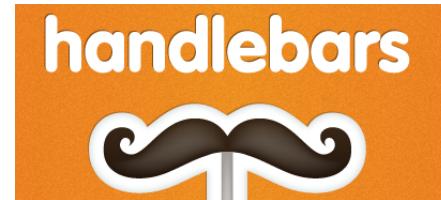
community,

performance,

testing....

# Old school web apps

HTML + templates



Data Binding



Routing



DOM-manipulation

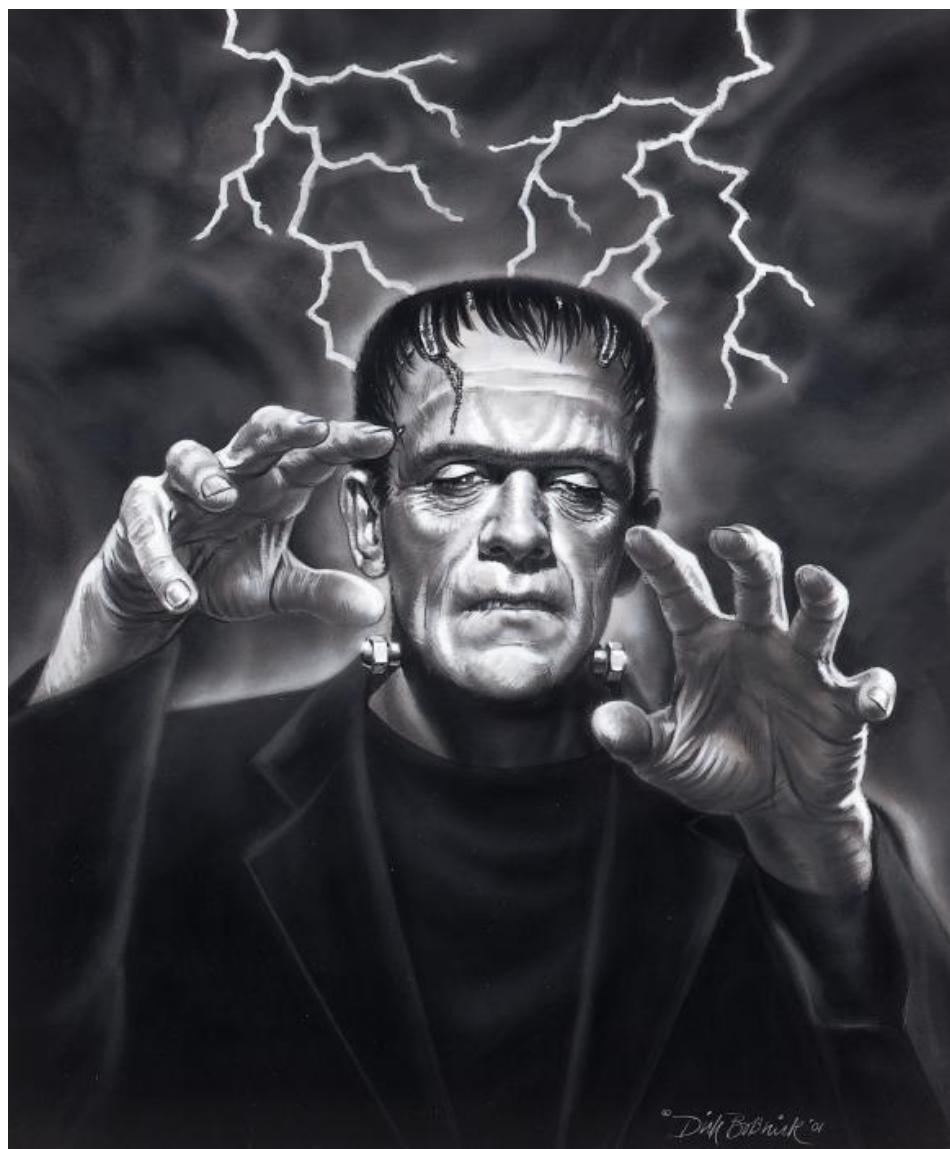


Mobile development

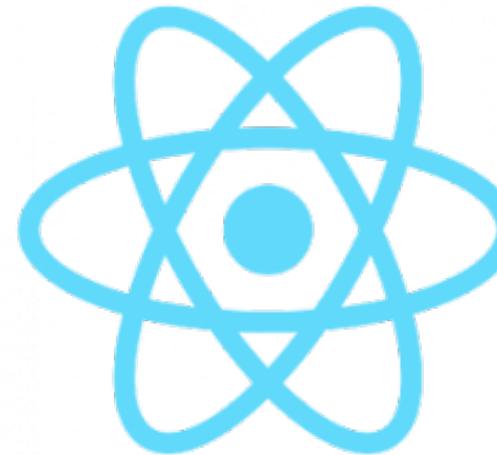


...

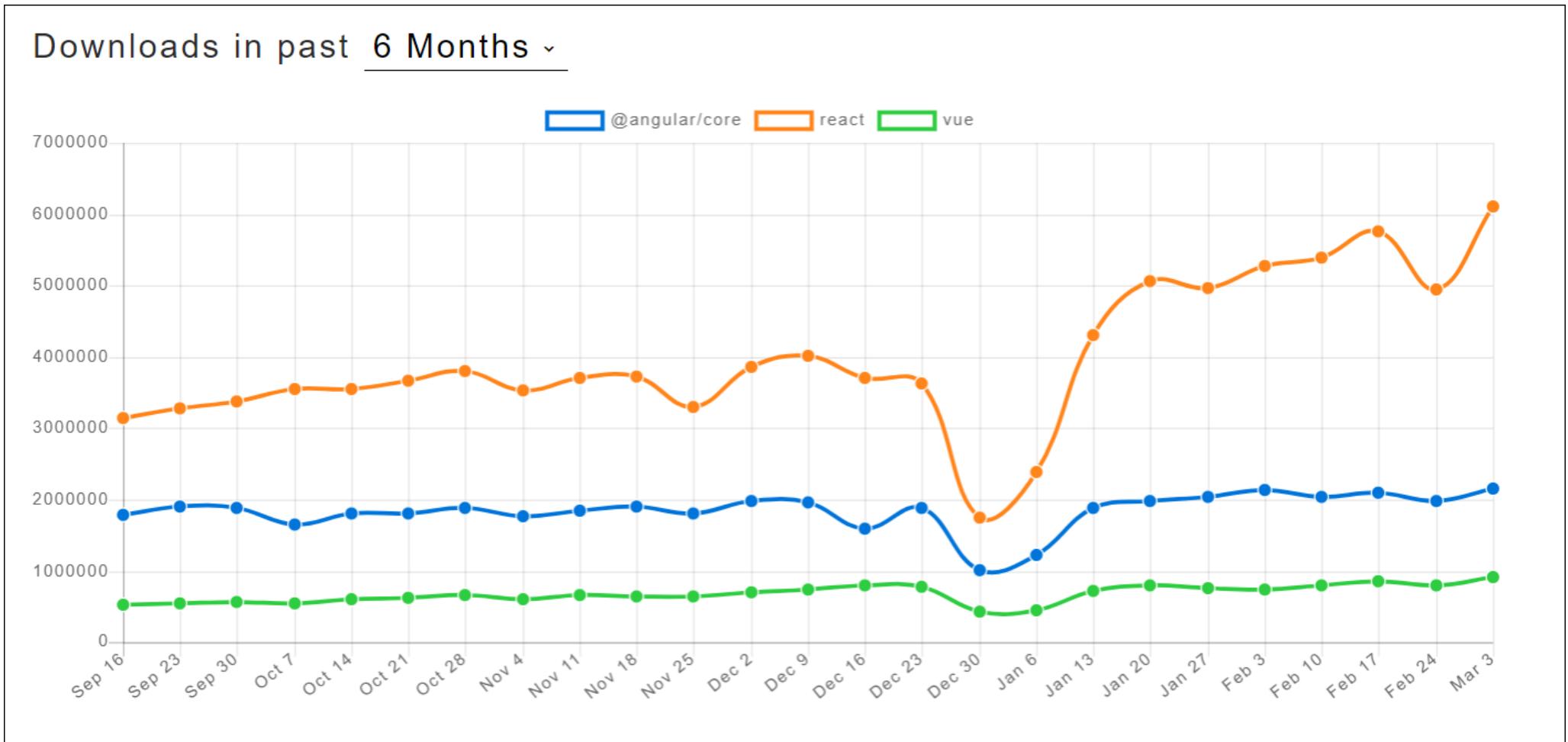
# **“The Frankenstein Framework”**



# Front-end Frameworks – the big four

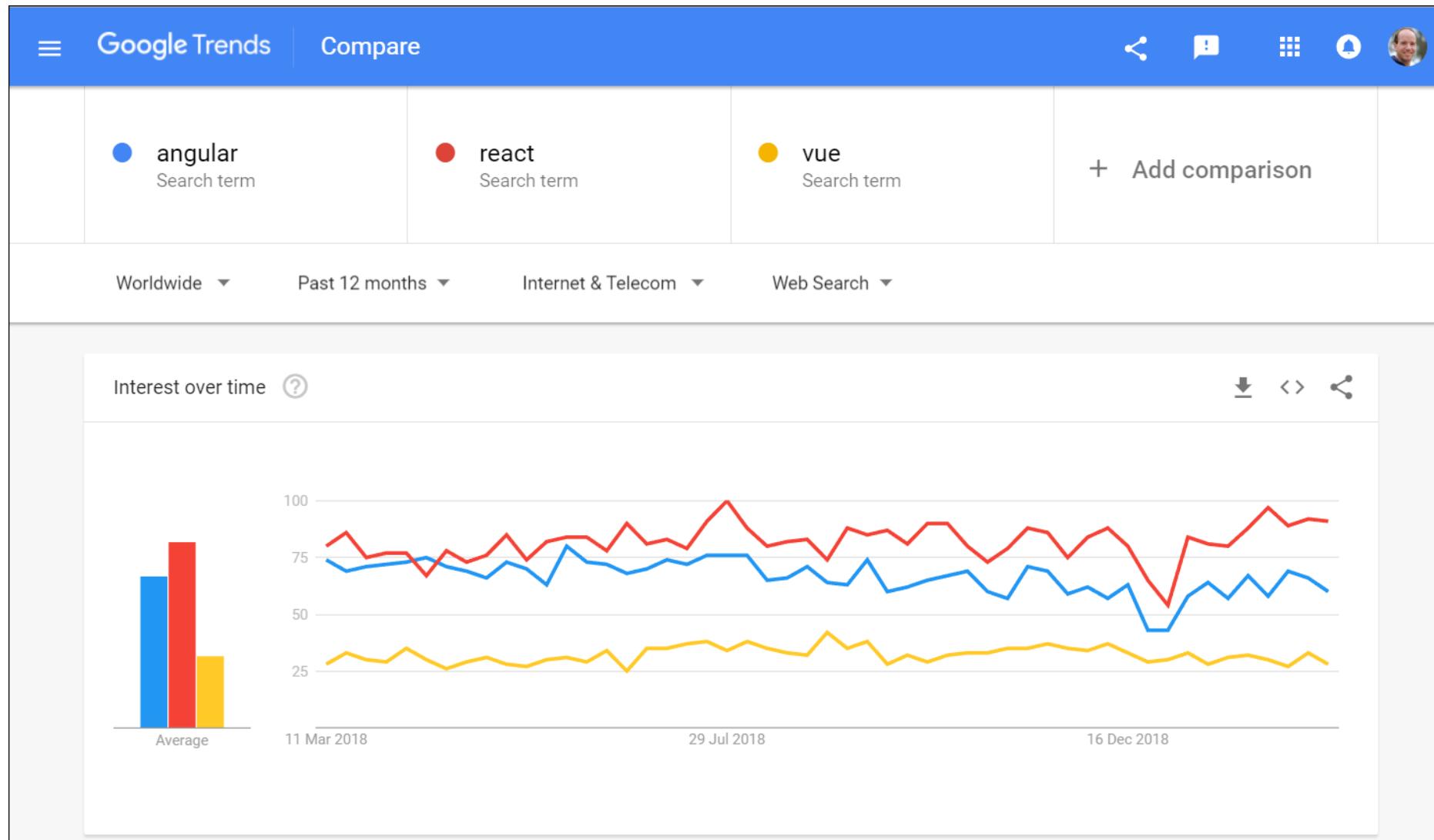


# A 2019 comparison



<https://www.npmtrtrends.com/@angular/core-vs-react-vs-vue>

# Google Trends



<https://trends.google.com/trends/explore?cat=13&q=angular,react,vue>

M | unicorn.supplies [Follow](#)

 Jens Neuhaus [Follow](#)  
CTO to rent, Full-stack developer with Python/Django, supporting startups & mid-size companies, loves go...  
Aug 28 · 25 min read

## Angular vs. React vs. Vue: A 2017 comparison

Deciding on a JavaScript framework for your web application can be overwhelming. [Angular](#) and [React](#) are very popular these days, and there is an upstart which has been getting a lot of traction lately: [VueJS](#). What's more, these are just a few of the [new kids on the block](#).



DO I REALLY HAVE TO PICK

 4.5K  38    [Next story](#) [Moving From Angular to Vue : A ...](#)

<https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>



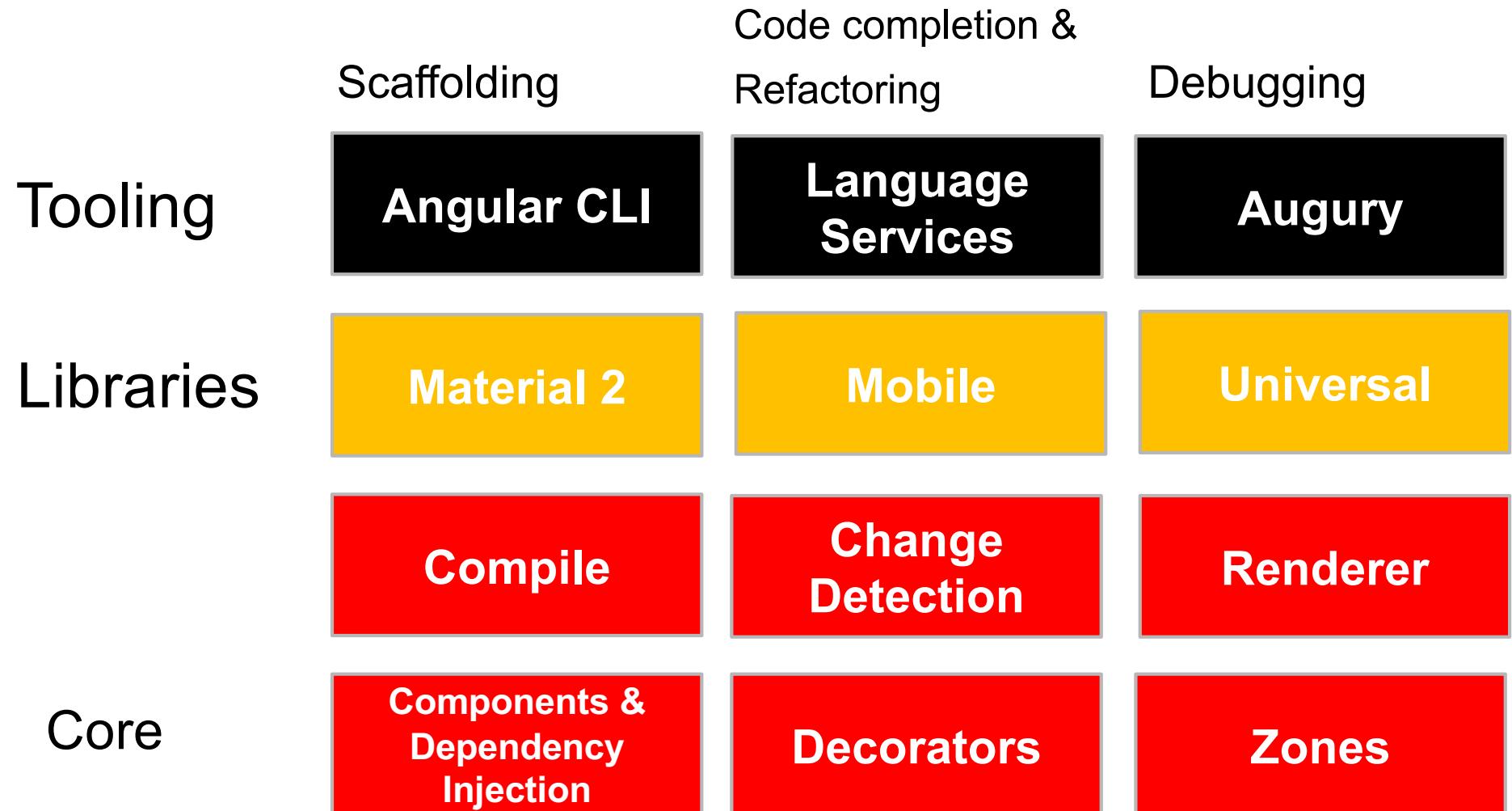
MV\*

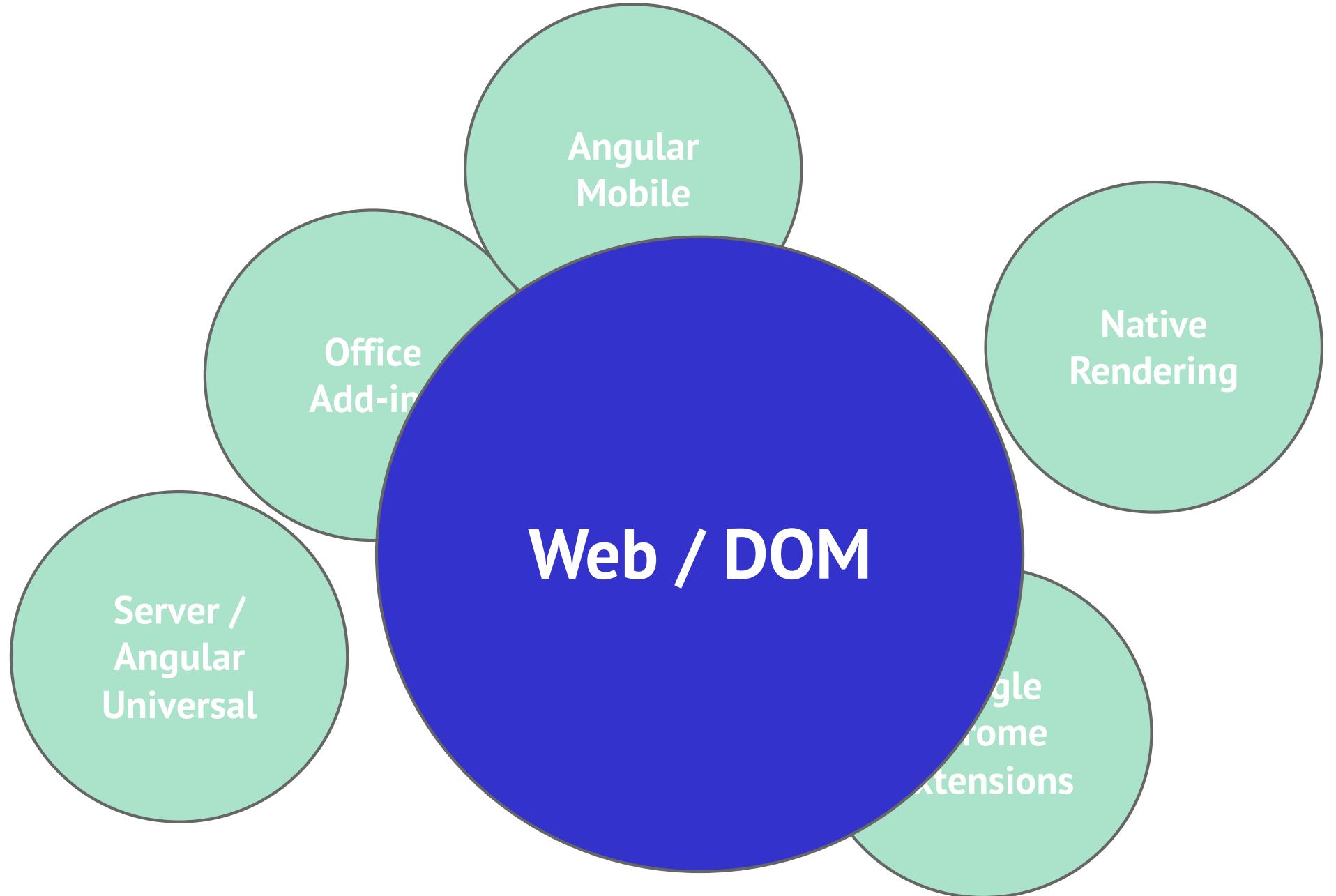
Framework



# Platform

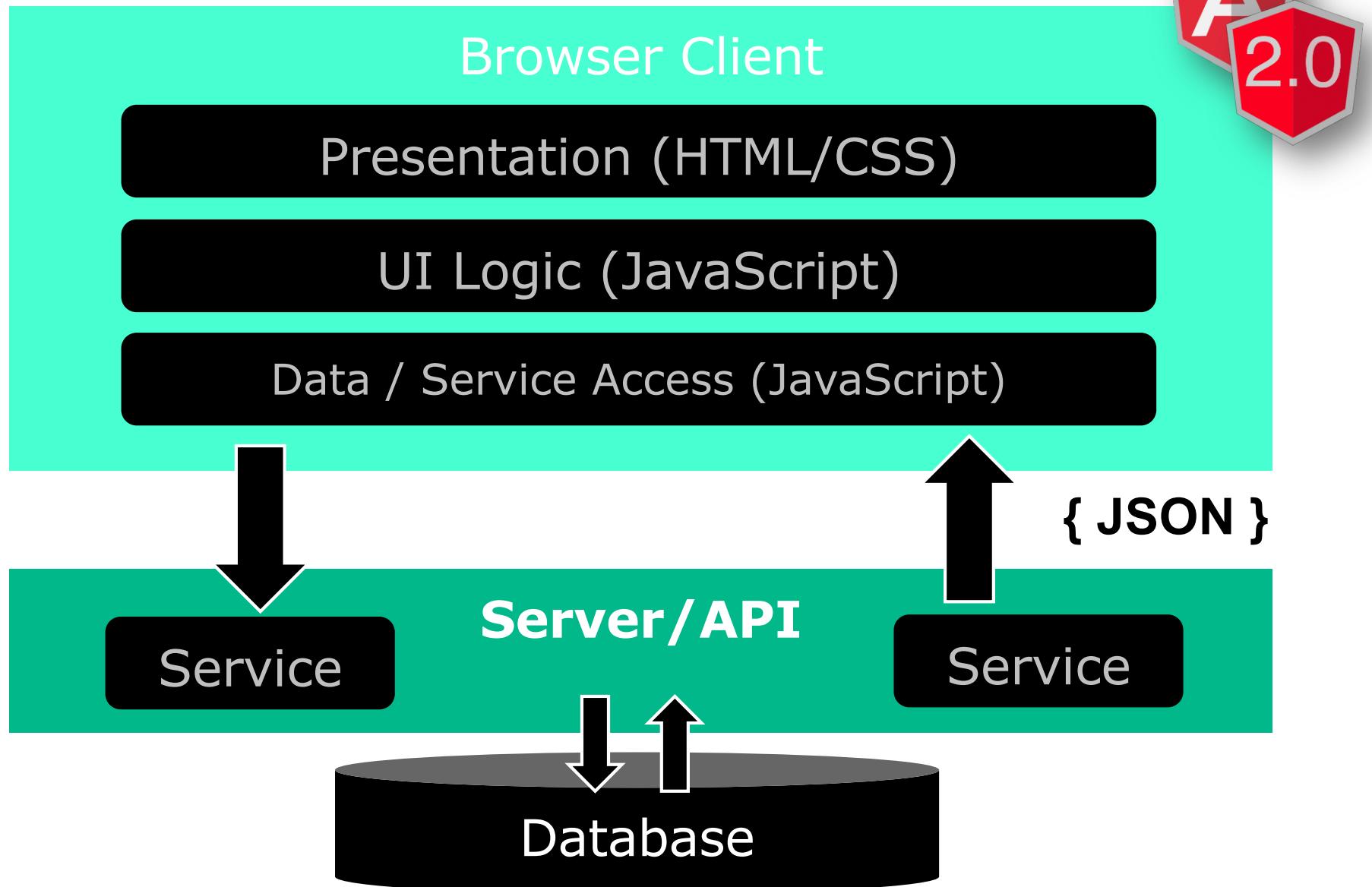
# Framework to Platform





# Single Page Application

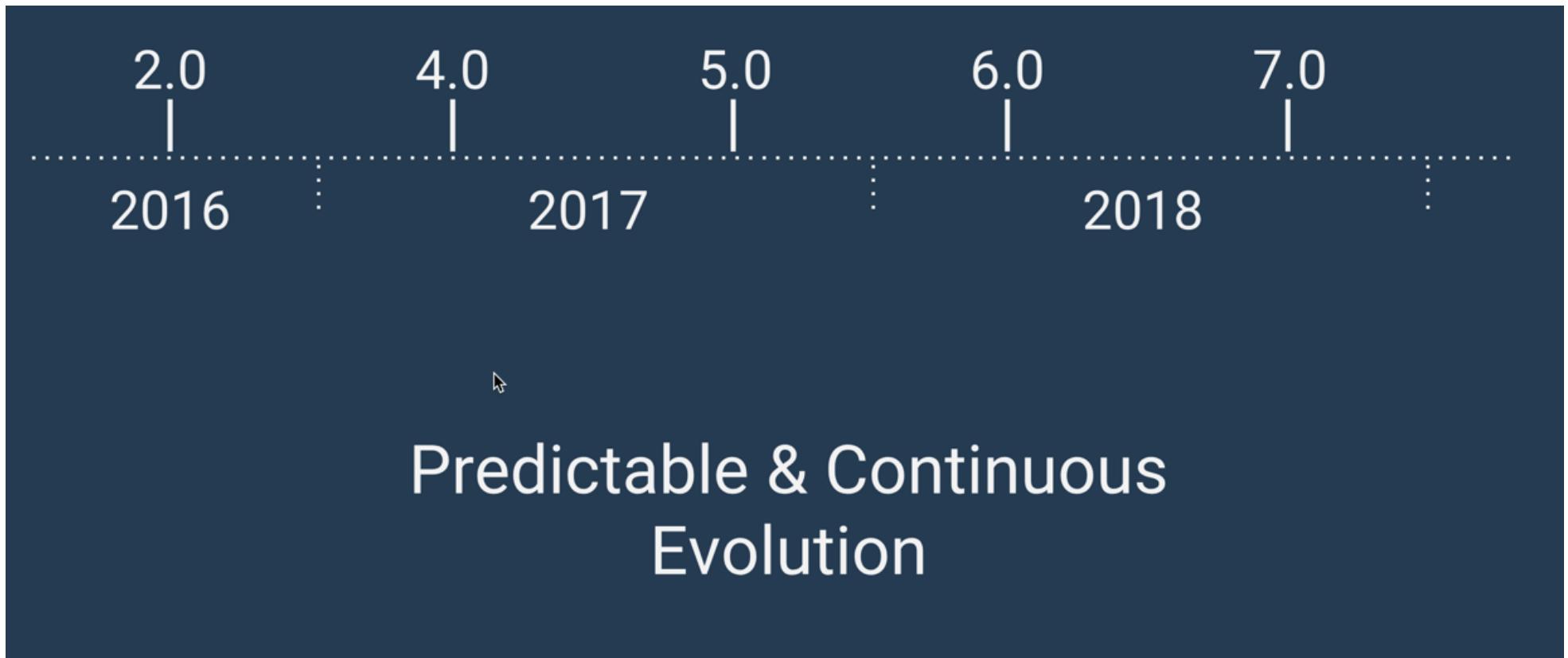
2010 – 20??



# On Versioning Numbering



## Planned – six months major release cycle





- Predictability
- Painless Updates
- Long Term Support

# Angular Versions and -Long Time Support

→ <https://angular.io/guide/releases>

The screenshot shows the Angular documentation page for releases. The left sidebar has sections like Getting Started, Tutorial, Fundamentals, Techniques (expanded to show Internationalization (i18n), Language Service, Security, Setup & Deployment, Service Workers, Keeping Up-to-Date (expanded to show Updating Your Projects, Angular Releases, and Upgrading from AngularJS), Server-side Rendering, Visual Studio 2015 QuickStart, Style Guide, Glossary, API (stable (v6.0.4)), and a footer note about v6.0.4. The main content area starts with a bulleted list about support frequency and LTS. Below that is a table showing support status and key dates for Angular 4.0.0, 5.0.0, and 6.0.0. To the right is a sidebar with links to Angular versioning, release frequency, release schedule, support policy, deprecation practices, public API surface, and Angular Labs.

Version	Status	Release Date	LTS Start Date	LTS End Date
^4.0.0	LTS	March 23, 2017	September 23, 2017	September 23, 2018
^5.0.0	LTS	November 1, 2017	May 1, 2018	May 1, 2019
^6.0.0	Active	May 3, 2018	November 3, 2018	November 3, 2019

## Deprecation practices

Sometimes "breaking changes", such as the removal of support for select APIs and features, are necessary to innovate and stay current with new best practices, changing dependencies, or changes in the (web) platform itself.

To make these transitions as easy as possible, we make two commitments to you:

- We work hard to minimize the number of breaking changes and to provide migration tools when possible.
- We follow the deprecation policy described here, so you have time to update your apps to the latest APIs and best practices.

<https://update.angular.io/>

Angular Update Guide

Select the options matching your project:

Angular Version

4.0 6.0

App Complexity

Basic Medium Advanced

ngUpgrade

I use ngUpgrade

Package Manager

npm yarn

Show me how to update!

Warning: We do not recommend moving across multiple major versions.

"It's just

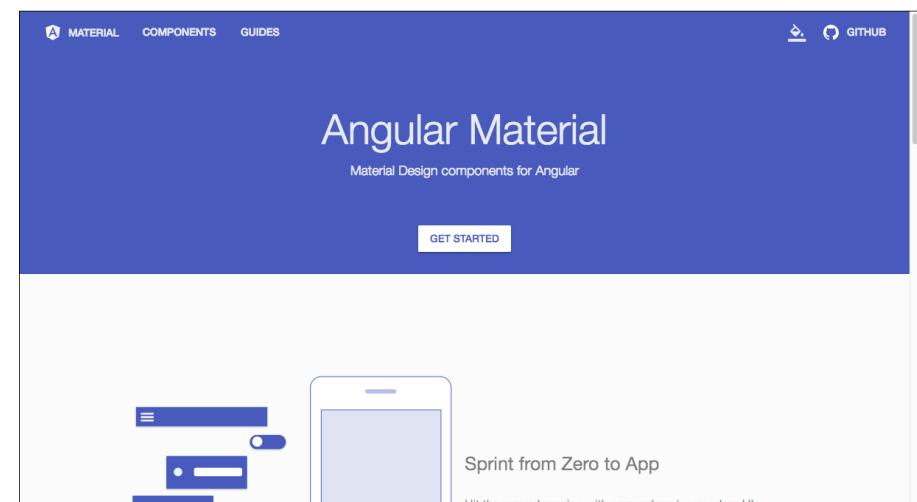
*Angular*

# Angular as a Platform



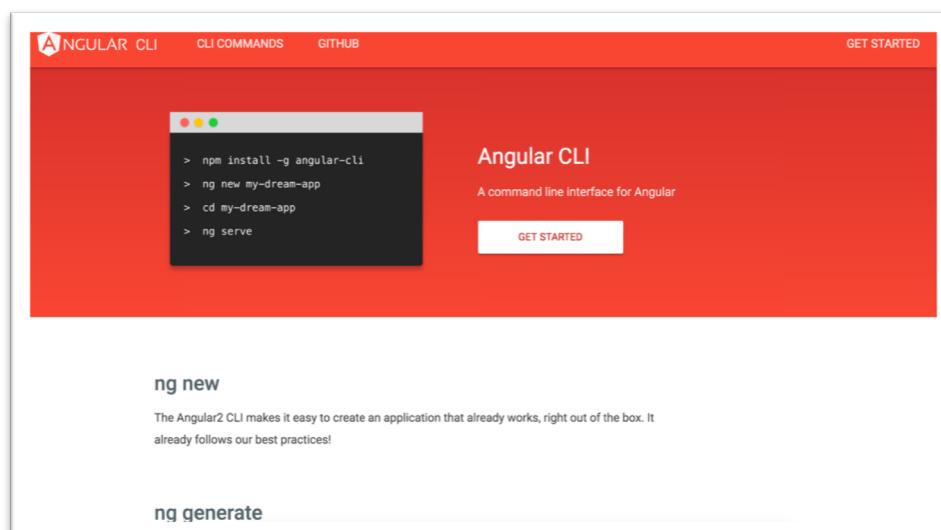
The Angular homepage features a large red hexagonal logo with a white 'A'. Below it, the text 'One framework. Mobile & desktop.' is displayed. A 'GET STARTED' button is located at the bottom right of the main section. At the bottom left, there's a callout for 'ANGULAR MIX' with a 'LEARN MORE' button.

<https://angular.io/>



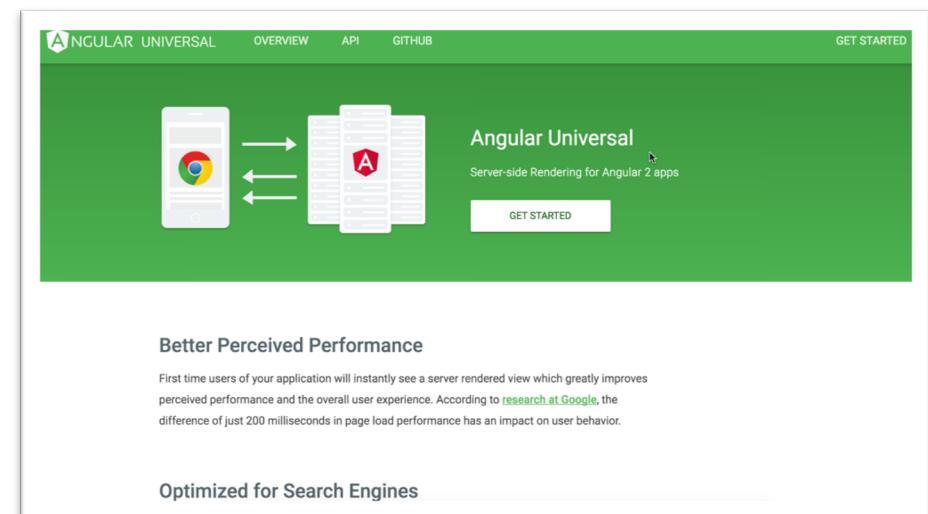
The Angular Material homepage has a blue header with navigation links for 'MATERIAL', 'COMPONENTS', and 'GUIDES'. It features a large 'Angular Material' title and a subtext 'Material Design components for Angular'. A 'GET STARTED' button is in the center. Below the header, there's a graphic of a smartphone and some UI components. A 'Sprint from Zero to App' section with a 'Hit the ground running with comprehensive, modern UI' subtext is also present.

<https://material.angular.io/>



The Angular CLI homepage has a red header with 'ANGULAR CLI', 'CLI COMMANDS', and 'GITHUB' links. It features a 'GET STARTED' button and a terminal window showing command examples like 'npm install -g angular-cli', 'ng new my-dream-app', 'cd my-dream-app', and 'ng serve'. Below the header, there's a 'ng new' section with a description of the CLI's purpose and a 'GET STARTED' button. The bottom section contains 'ng generate' information.

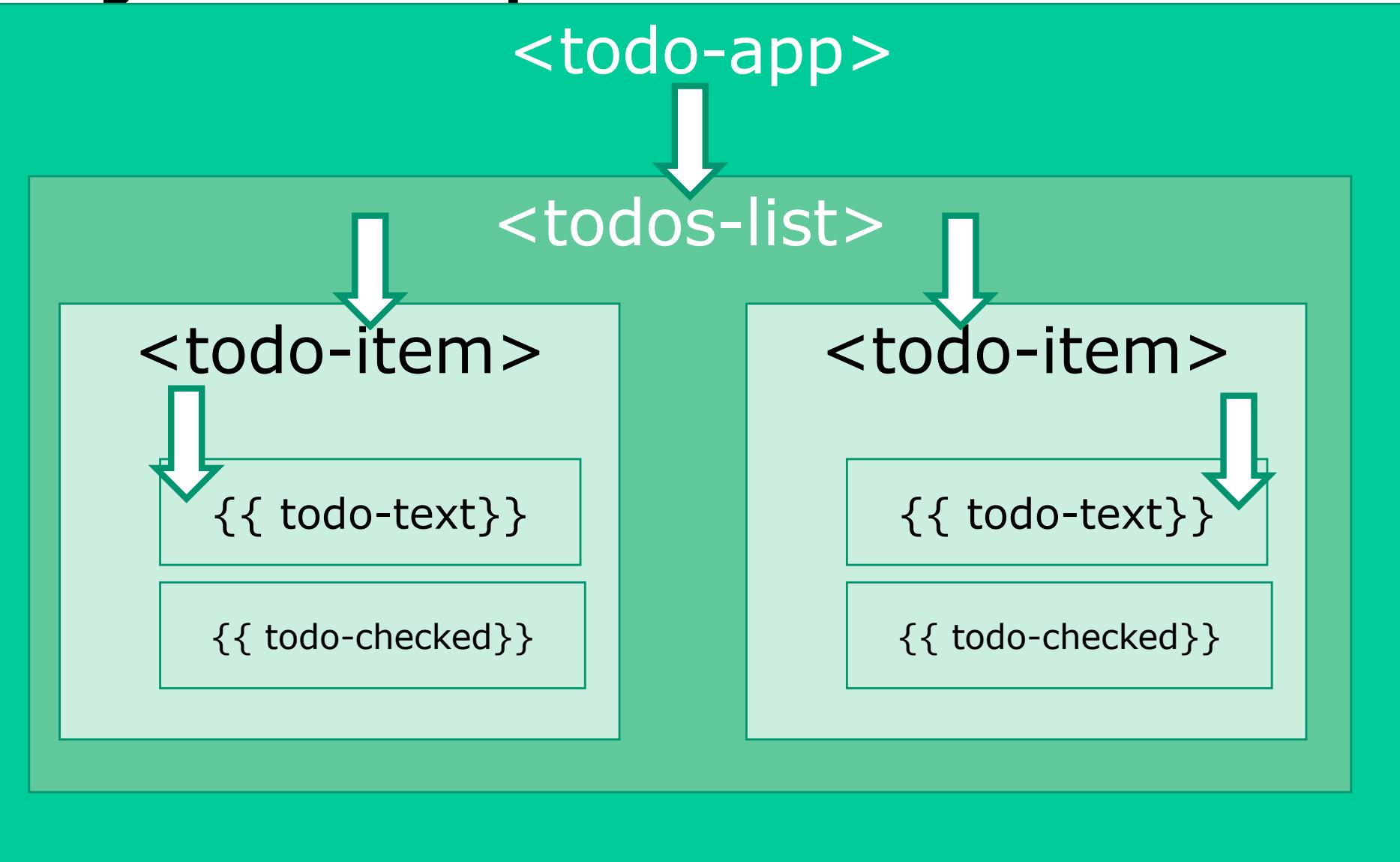
<https://cli.angular.io/>



The Angular Universal homepage has a green header with 'ANGULAR UNIVERSAL', 'OVERVIEW', 'API', and 'GITHUB' links. It features a 'GET STARTED' button and a diagram showing a browser connected to a server. Below the header, there's a 'Better Perceived Performance' section with a description and a 'GET STARTED' button. The bottom section contains 'Optimized for Search Engines' information.

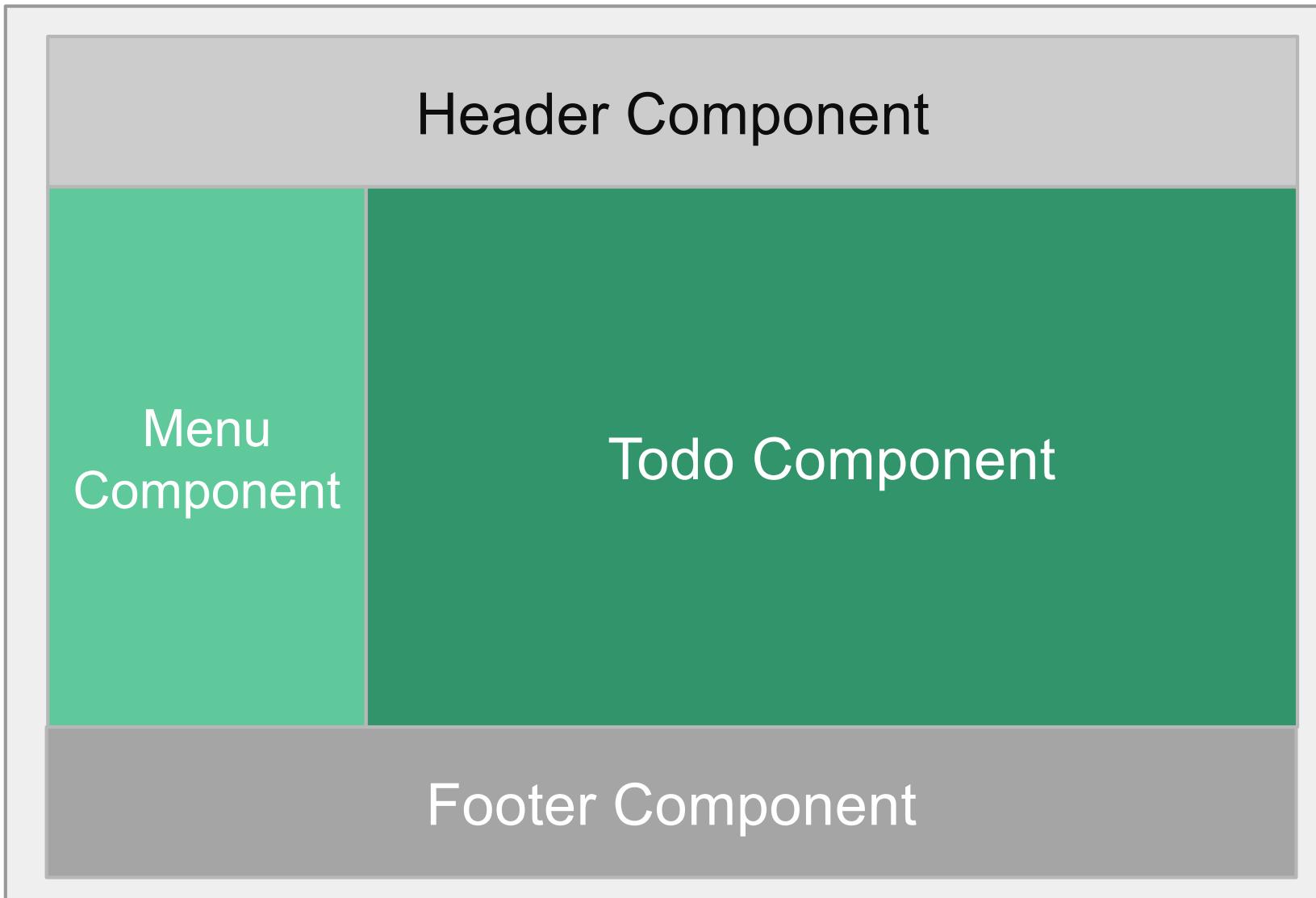
<https://universal.angular.io/>

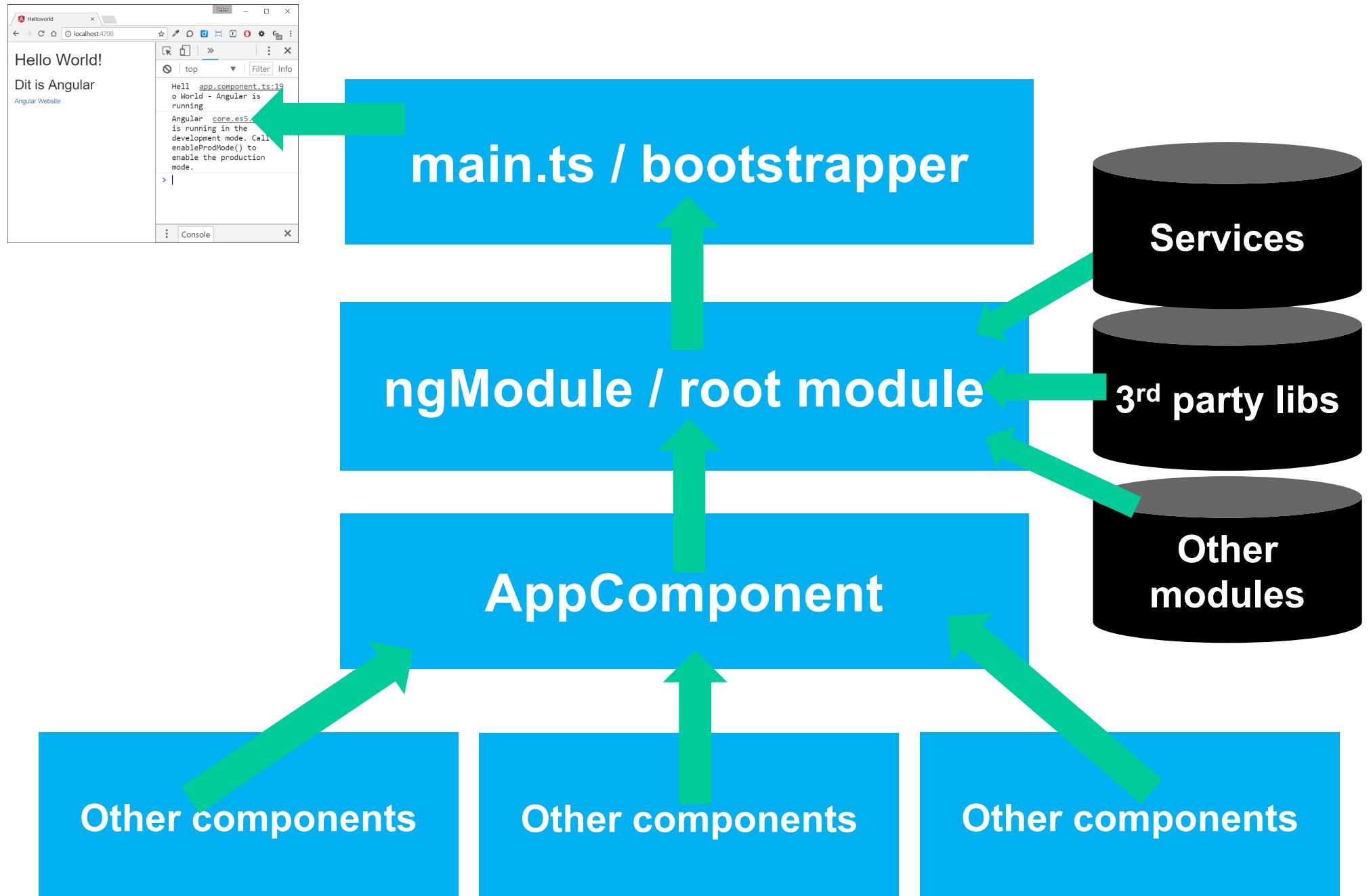
# Angular 2 - components



*"An Angular-app is a  
tree  
of components"*

# Components – visually







# Let's look at some code

## 'Hello World' in Angular

## **Angular 1:**

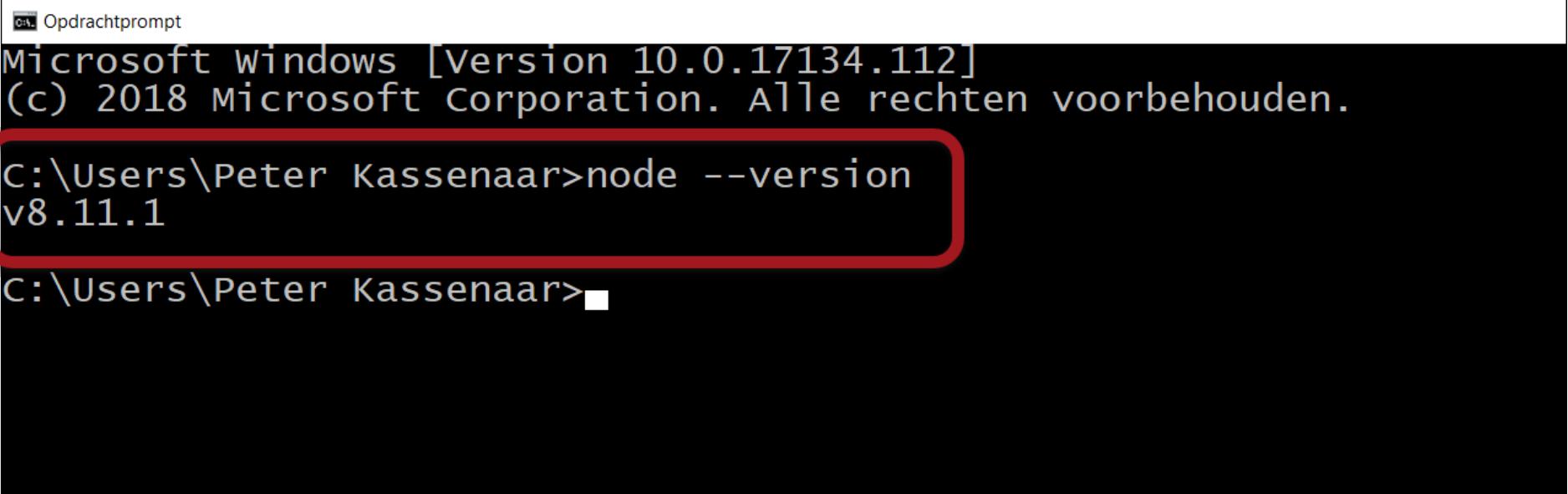
```
<script src="angular.min.js"></script>
```

# Angular development dependency: NodeJS 8.0+



The screenshot shows the official Node.js website. At the top, there's a dark navigation bar with the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted with a green background. Below the navigation, a large text area states: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." A green callout box says "Important June 2018 security upgrades now available". The main content area features a "Download for Windows (x64)" heading. It offers two options: "8.11.3 LTS" (Recommended For Most Users) and "10.6.0 Current" (Latest Features). Below these are links for "Other Downloads | Changelog | API Docs" for each version. Further down, it says "Or have a look at the [Long Term Support \(LTS\) schedule](#)." A call to action encourages users to "Sign up for [Node.js Everywhere](#), the official Node.js Weekly Newsletter." At the bottom, there's a footer with the Linux Foundation logo, a link to "COLLABORATIVE PROJECTS", and links for reporting issues or getting help. It also includes a copyright notice: "© Node.js Foundation. All Rights Reserved. Portions of this site originally © Joyent."

# Node – check your version



A screenshot of a Windows Command Prompt window titled "Opdrachtprompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Peter Kassenaar>node --version
v8.11.1

C:\Users\Peter Kassenaar>
```

The command "node --version" and its output "v8.11.1" are highlighted with a red rounded rectangle.

# Exercise

- Download or clone  
<https://github.com/PeterKassenaar/voorbeeldenAngular2>
- Unpack the repository and cd into /voorbeeldenAngular-master

```
cd examples
```

```
cd 100-helloworld
```

```
npm install
```

```
npm start
```

- Go to browser: <http://localhost:4200>

The screenshot illustrates the development environment for an Angular application. On the left, a browser window shows the application running at [localhost:4200](http://localhost:4200), displaying the text "Hello World!" and "This is Angular". Below the browser is a blue header bar with the text "Angular Website". On the right, an IDE interface displays the project structure and the source code for the `app.component.ts` file.

**Project Structure:**

- voorbeeldenAngular2** (Project root)
  - Examples
    - 100-helloworld
      - node\_modules (library root)
      - src
        - app
          - app.component.ts
          - app.module.ts
        - assets
        - environments
        - favicon.ico
        - index.html
        - main.ts
        - polyfills.ts
        - styles.css
        - tsconfig.app.json
      - .angulardoc.json
      - .gitignore
      - angular.json
      - package.json
      - package-lock.json
      - tsconfig.json
      - yarn.lock

```
Project voorbeeldenAngular2 C:\Users\Peter Kassenaar\Desktop\voorbeeldenAngular2\Examples\100-helloworld\node_modules library root
src\app\app.component.ts
import {Component, OnInit} from '@angular/core';
@Component({
  selector: 'hello-world',
  template: `
    

# Hello World!



## This is Angular

Angular
  `
})
export class AppComponent implements OnInit {
  constructor() {}
}
```

# Boilerplate code for Hello World

## Steps

1. Set up environment, boilerplate & libraries
2. Write Angular Root Component
3. Write @NgModule Component
4. Bootstrap module
5. Write HTML-pagina (`index.html`)



# Boilerplate files #1 - package.json

```
{  
  "name": "hello-angular",  
  "description": "Voorbeeldproject bij de training Angular (C) - info@kassenaar.com",  
  "version": "0.0.1",  
  "license": "MIT",  
  "scripts": {  
    "ng": "ng",  
    "start": "ng serve",  
    "build": "ng build",  
  },  
  "private": true,  
  "dependencies": {  
    "@angular/animations": "7.0.0",  
    "@angular/common": "7.0.0",  
    "@angular/compiler": "7.0.0",  
    "@angular/core": "7.0.0",  
    "@angular/forms": "7.0.0",  
    "rxjs": "^6.1.0",  
    "zone.js": "^0.8.26"  
  },  
  "devDependencies": {  
    "@angular-devkit/build-angular": "~0.6.0",  
    "@angular/cli": "6.0.0",  
    "typescript": "3.2.1"  
  },  
  "author": "Peter Kassenaar <info@kassenaar.com>"  
}
```

## Boilerplate files #2 - tsconfig.json

```
{  
  "compileOnSave": false,  
  "compilerOptions": {  
    "outDir": "./dist/out-tsc",  
    "baseUrl": "src",  
    "sourceMap": true,  
    "declaration": false,  
    "moduleResolution": "node",  
    "emitDecoratorMetadata": true,  
    "experimentalDecorators": true,  
    "target": "es5",  
    "typeRoots": [  
      "node_modules/@types"  
    ],  
    "lib": [  
      "es2016",  
      "dom"  
    ]  
  }  
}
```

## Boilerplate files #3 - angular.json

```
{  
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",  
  "version": 1,  
  "newProjectRoot": "projects",  
  "projects": {  
    "helloworld": {  
      "root": "",  
      "sourceRoot": "src",  
      "projectType": "application",  
      "architect": {  
        "build": {  
          "builder": "@angular-devkit/build-angular:browser",  
          "options": {  
            "outputPath": "dist",  
            "index": "src/index.html",  
            "main": "src/main.ts",  
            "tsConfig": "src/tsconfig.app.json",  
            ...  
          }  
        }  
      }  
    }  
  }  
}
```

# "Nice to have" - non-essential files

The screenshot shows a GitHub repository page for 'angular / quickstart'. The repository has 250 forks and 2,567 stars. The 'Code' tab is selected. A commit by 'wardbell' is shown, adding 'bs-config.e2e.json' to the file. The file has 16 lines (15 sloc) and 226 Bytes. The contents of the file are listed below:

```
1 .git
2 .gitignore
3 .travis.yml
4 *.spec*.ts
5 bs-config.e2e.json
6 CHANGELOG.md
7 e2e
8 favicon.ico
9 karma.conf.js
10 karma-test-shim.js
11 LICENSE
12 non-essential-files.txt
13 non-essential-files.osx.txt
14 protractor.config.js
15 README.md
```

<https://github.com/angular/quickstart/blob/master/non-essential-files.txt>

# Step 2 – Component

Convention - components in directory /src/app

Or: edit in angular.json

Filename: src/app/app.component.ts

```
import {Component} from '@angular/core';
@Component({
  selector: 'hello-world',
  template: '<h1>Hello Angular</h1>'
})
export class AppComponent { }
```

# Step 3 – @NgModule

Convention - filename: /src/app.module.ts

```
// Angular Modules
import {NgModule}      from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';

// Custom Components
import {AppComponent} from './app.component';

// Module declaration
@NgModule({
  imports      : [BrowserModule],
  declarations: [AppComponent],
  bootstrap    : [AppComponent]
})
export class AppModule {
```

Root Module of the application

# Some background info on Root Module

The screenshot shows a blog post titled "Introducing Angular Modules - Root Module" by John Papa. The post is dated September 5, 2016. The content discusses the `@NgModule` decorator and its purpose in organizing Angular applications. The sidebar features a bio of John Papa and links to his most recent posts.

**Introducing Angular Modules - Root Module**

05 SEPTEMBER 2016

The `@NgModule` is a new decorator that has recently been added in Angular 2. `NgModule` defines an Angular Module, which (from the official docs) are defined as "Angular Modules help organize an application into cohesive blocks of functionality."

**John**

Hi, I'm John Papa. I author this blog, create courses for Pluralsight and am a Google Developer Expert and Microsoft Regional Director. I speak at events and I train technology thought leaders →

**Most Recent**

- [Introducing Angular Modules - Root Module](#)
- [Learning Angular 2 this Fall](#)
- [The Ultimate Angular 2 Workshop in Ft](#)

<https://johnpapa.net/introducing-angular-modules-root-module/>

# Step 4 - bootstrap component

Best practice: bootstrap app in separate component

Convention: main.ts, **of** app.main.ts.

```
import {enableProdMode} from '@angular/core';
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

import {AppModule} from './app/app.module';
import {environment} from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

# Step 5 – index.html

index.html - simple HTML file - expanded at runtime by WebPack

```
<html>

<head>
  <meta charset="utf-8">
  <title>Helloworld</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
```

# Body of index.html

Element reference (selector) of root-component:

```
<body>  
  <hello-world>  
    loading...  
  </hello-world>  
</body>
```

# Run the app

npm start - run start script from package.json.

ng serve - start global angular-cli instance

The terminal window shows the following output:

```
Terminal
+ chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4} [initial] [rendered]
x
chunk {1} main.bundle.js, main.bundle.js.map (main) 3.73 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 9.78 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.1 MB [initial] [rendered]
chunk {4} main.bundle.js, main.bundle.js.map (main) 3.73 kB {1} [initial] [rendered]
webpack: C:\Users\...{[redacted]}
```

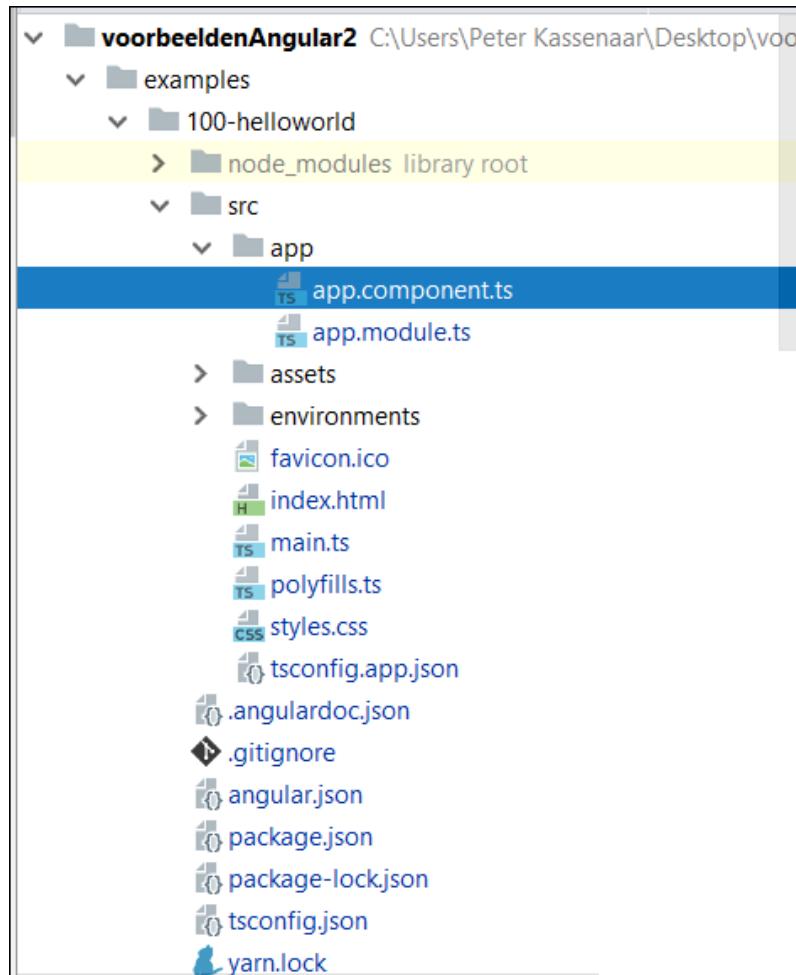
The browser screenshot shows the Angular application running at `localhost:4200`. The page displays "Hello World!" and "Dit is Angular". The developer tools console shows the following logs:

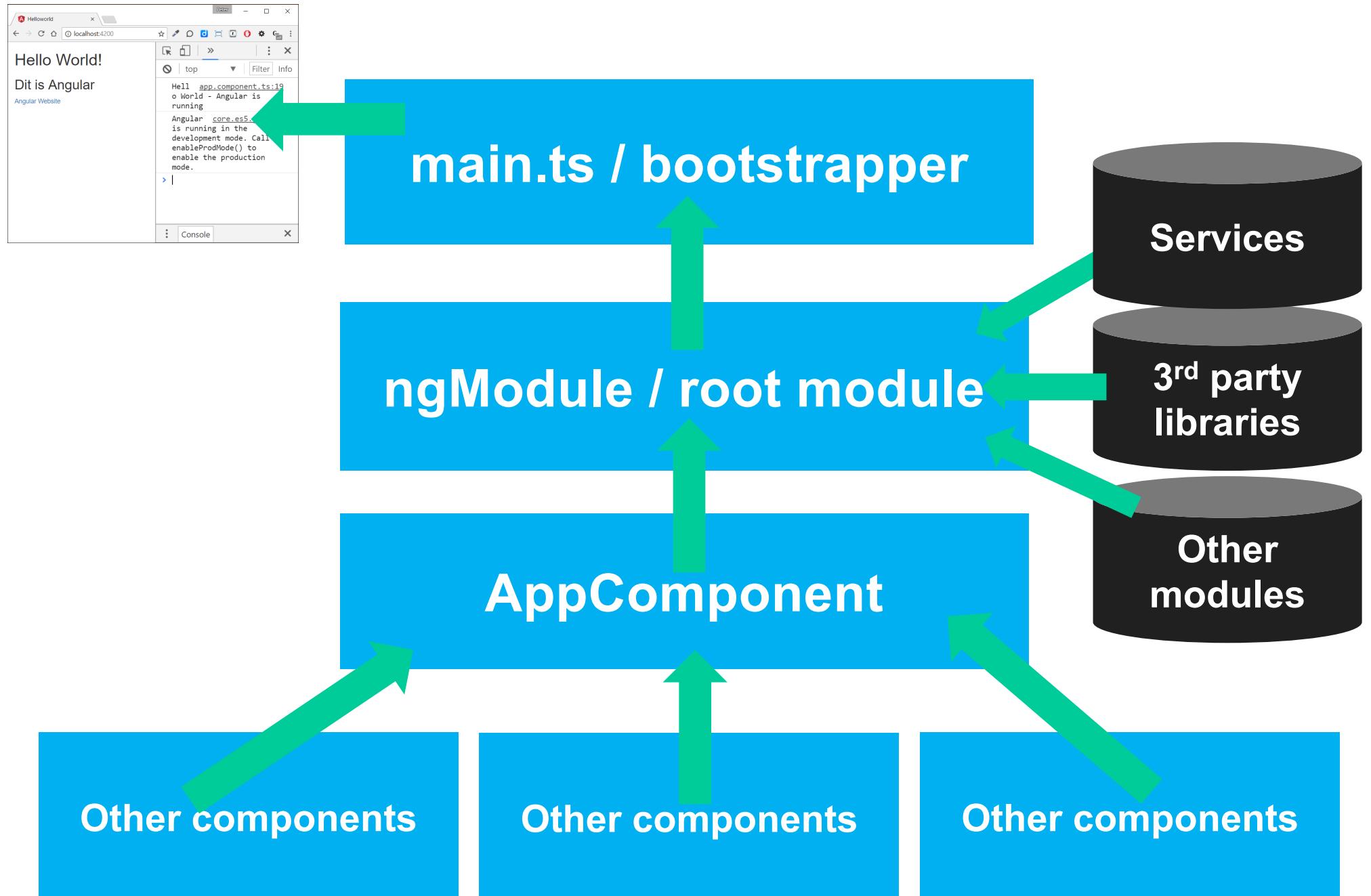
```
localhost:4200
Elements Console Sources Network Performance Memory > | : X
Filter Info
Hello World - Angular is running
Angular is running in the development mode. Call enableProdMode() to enable the production mode.
app.component.ts:19
core.es5.js:3025
```

After that: edit `app.component.ts`

– Automagically refreshed through Live Reload

# So, a Basic Project Structure and Architecture





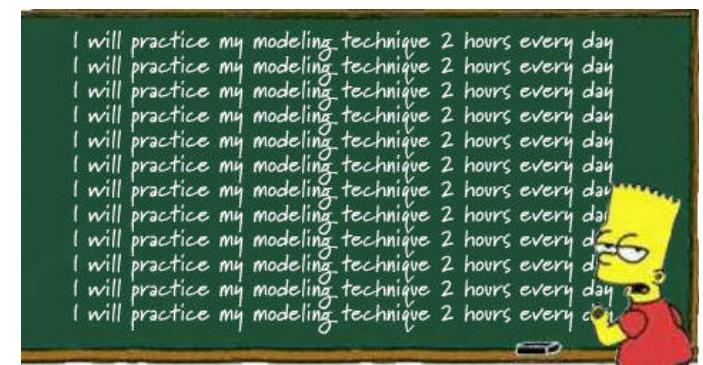
# Checkpoint

You need a lot of boilerplate code to start an Angular project.

(At least) Five steps:

1. Set up environment, boilerplate & libraries
2. Write Angular Root Component for app
3. Bootstrap component (`main.ts`)
4. write HTML-pagina (`index.html`)
5. Run the app : `npm start`

Then: work on your components, services, etc.

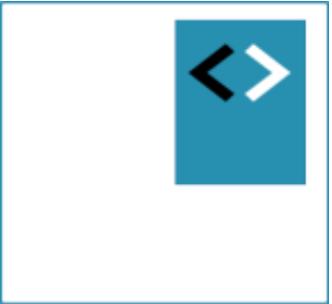


# Exercise....

# **Assets**

<https://github.com/PeterKassenaar/oce>

Exercises and example code



# Tooling - Angular CLI

Quickly set up new projects  
via command line interface

# Angular-CLI to the rescue

- It *is* possible to start new Angular projects from scratch
- But by using the CLI it is *much* simpler
- CLI-options:
  - Scaffolding
  - Generating
  - Testing
  - Building
  - AOT-Compiling
  - ...

# Scaffolding - Angular CLI

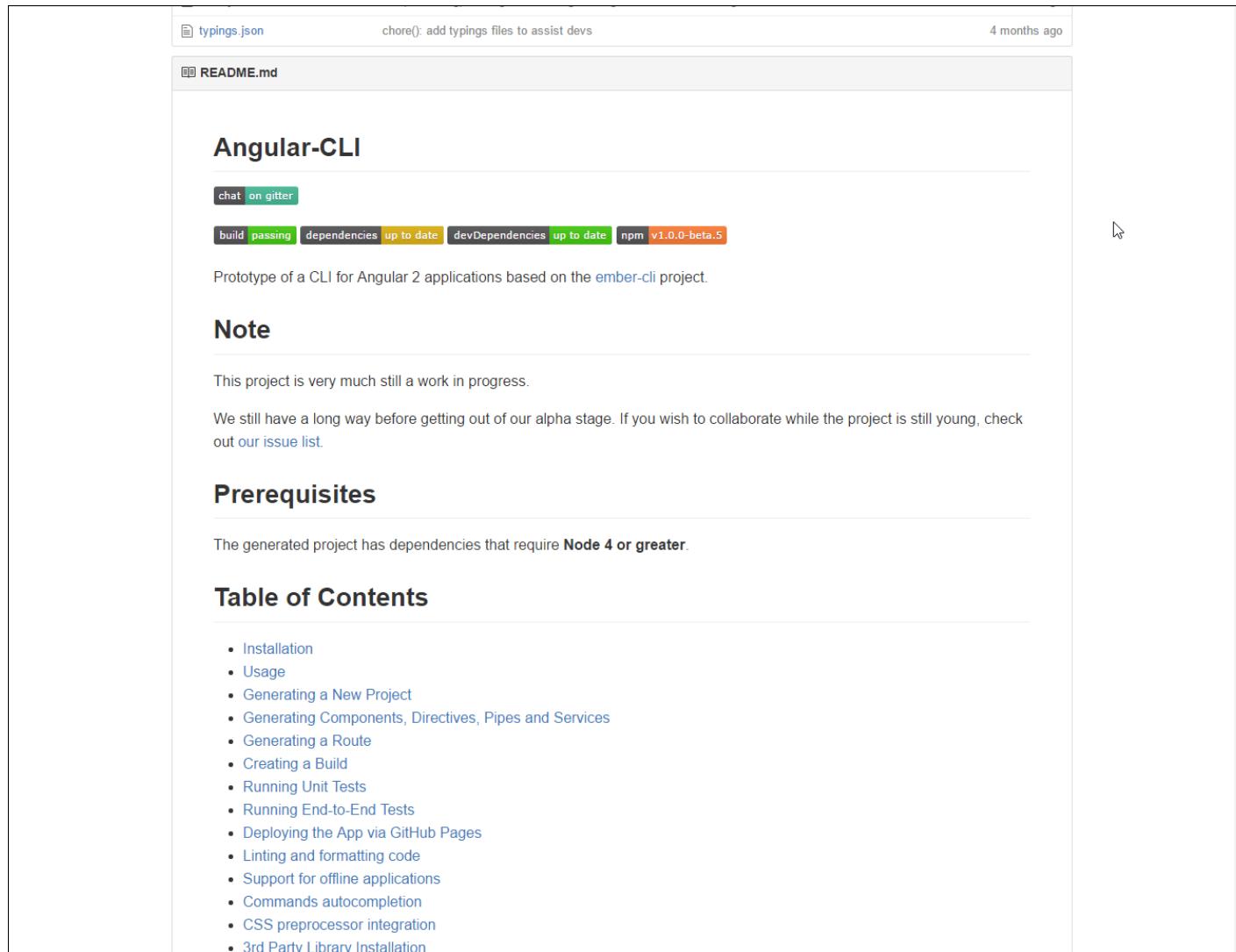
First : install CLI globally

<https://github.com/angular/angular-cli>

en

<https://cli.angular.io/>

```
npm install -g @angular/cli
```



A screenshot of the Angular-CLI GitHub repository's README page. The page includes a file tree header with 'typings.json' and a commit message 'chore(): add typings files to assist devs' from 4 months ago. Below the header is the 'README.md' content area.

## Angular-CLI

chat on gitter

build passing | dependencies up to date | devDependencies up to date | npm v1.0.0-beta.5

Prototype of a CLI for Angular 2 applications based on the [ember-cli](#) project.

### Note

This project is very much still a work in progress.

We still have a long way before getting out of our alpha stage. If you wish to collaborate while the project is still young, check out [our issue list](#).

### Prerequisites

The generated project has dependencies that require **Node 4 or greater**.

### Table of Contents

- Installation
- Usage
- Generating a New Project
- Generating Components, Directives, Pipes and Services
- Generating a Route
- Creating a Build
- Running Unit Tests
- Running End-to-End Tests
- Deploying the App via GitHub Pages
- Linting and formatting code
- Support for offline applications
- Commands autocomplete
- CSS preprocessor integration
- 3rd Party Library Installation

```
npm install -g @angular/cli
```

[DOCUMENTATION](#)[GITHUB](#)[GET STARTED](#)

```
> npm install -g @angular/cli
> ng new my-dream-app
> cd my-dream-app
> ng serve
```

## Angular CLI

A command line interface for Angular

[GET STARTED](#)

### ng new

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

### ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create

# ng --version

# Background info



<https://www.youtube.com/watch?v=wHZe6gGI5RY>

# Main commands

```
ng new PROJECT_NAME
```

```
cd PROJECT_NAME
```

```
ng serve
```

Project is served on <http://localhost:4200>

# More info

The screenshot shows the Scotch.io website. The left sidebar has a dark background with the Scotch logo (a glass with orange and yellow stones) and the text "scotch.io code on the rocks". It includes sections for "LEARN" (Tutorials, Bar Talk, Community), "Scotch School", and "OUR STUFF" (About, Shop, Join Us On Slack, Hire Us). The main content area has a red header with the Angular logo and the title "Use the Angular CLI for Faster Projects". Below this, there's a larger section with the same title and a subtitle: "Creating Angular 2 projects has never been faster than with the Angular CLI". The right sidebar features a "Popular Scotchers" list with profiles for Chris Sevilleja, Nicholas Cerminala, Holly Lloyd, Ado Kukic, Ryan Chenkie, Ken Wheeler, Alex Sears, and Chris Nwamba. At the bottom right of the sidebar is a button labeled "Become a Scotcher!". The top navigation bar includes a search bar, "Login / Sign Up", "Write a Post", and a "More" menu.

<https://scotch.io/tutorials/use-the-angular-cli-for-faster-angular-2-projects>

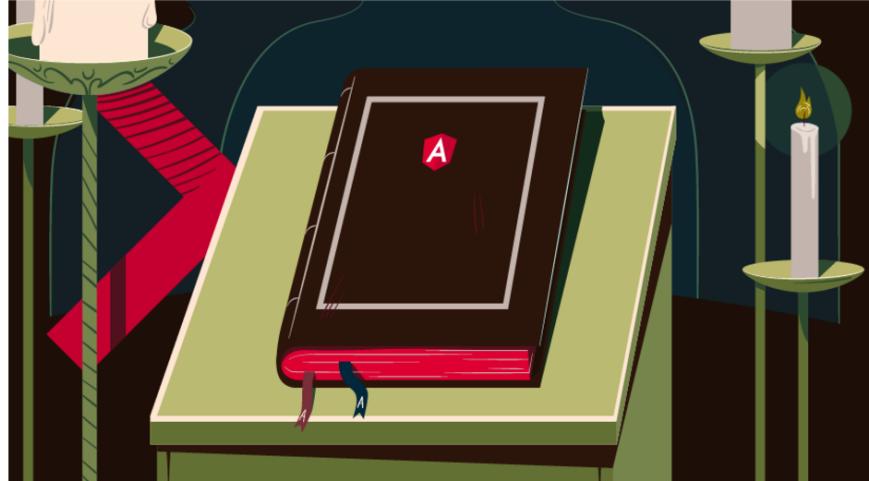
[≡](#)  sitepoint

Books & Courses   Community   [Login](#)   [Create Free Account](#)

HTML & CSS   JAVASCRIPT   PHP   DESIGN & UX   WEB   WORDPRESS   SEO BY WOORANK   MICROSOFT TECH

JavaScript - April 25, 2017 - By Jurgen Van de Moere

# The Ultimate Angular CLI Reference Guide



[f](#) [in](#) [t](#)

**2017.04.25:** As of March 24, Angular CLI v1.0 was released. This article has been updated to reflect the latest changes. If you want to add the latest features of Angular CLI v1.0 to your existing Angular project that was generated with an earlier version of Angular CLI, check out the [Angular CLI v1.0 migration guide](#).

**2017.02.17:** As of February 9, 2017, the `ng deploy` command has been removed

**Related Topics:**

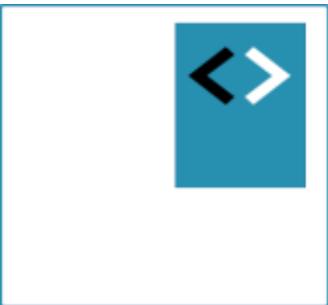
[React](#)   [Tools & Libraries](#)  
[Node.js](#)   [Raw JavaScript](#)  
[npm](#)   [More...](#)

**SPONSORS**

  
● JAVASCRIPT  
**An Introduction to Component Routing with Angular Router**  
Jurgen Van de Moere, 12 hours ago



<https://www.sitepoint.com/ultimate-angular-cli-reference/>

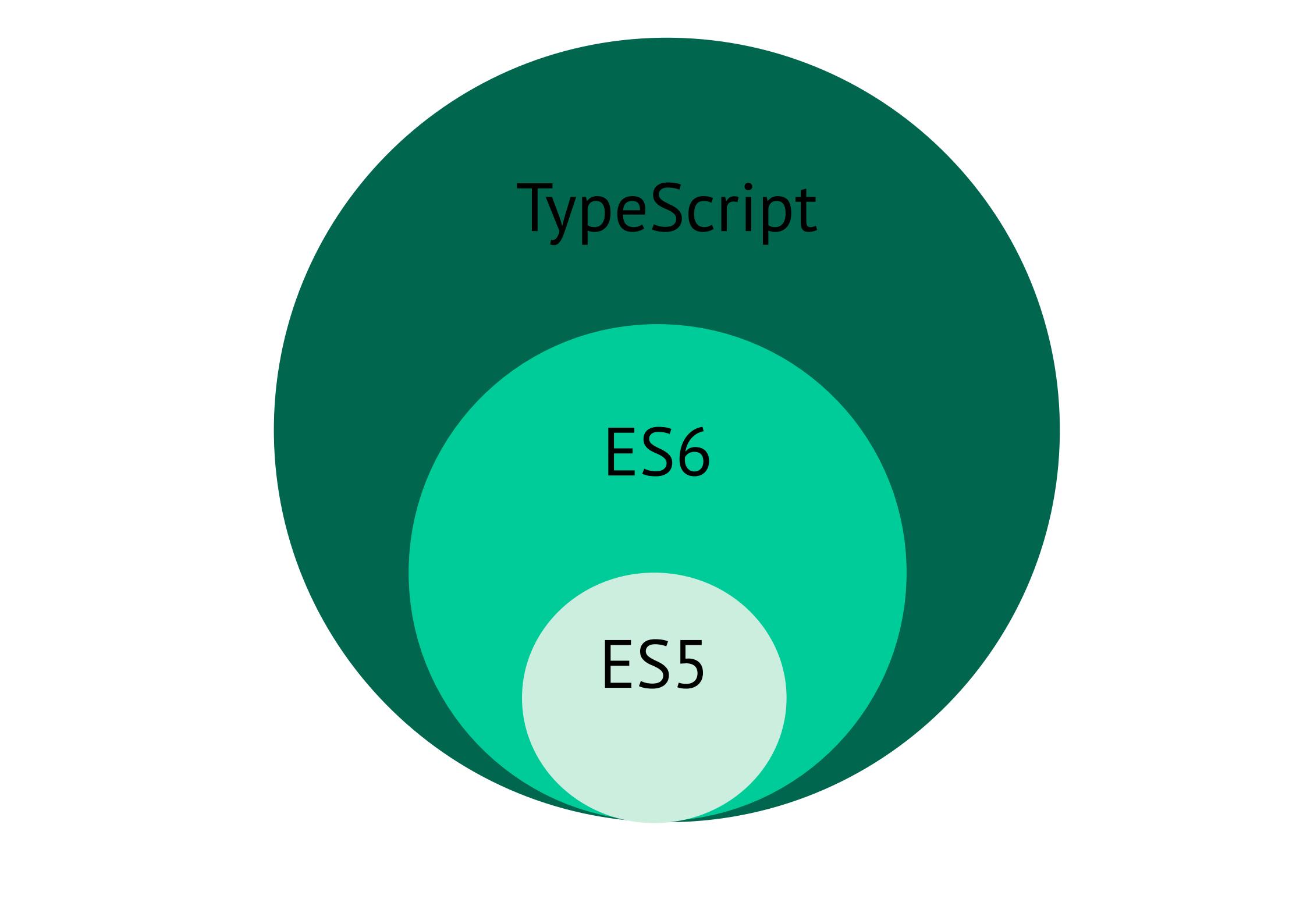


# Angular Code - “Backend”

On TypeScript en ES6

# Programming languages





TypeScript

ES6

ES5

# ES6 and TypeScript

The future of JavaScript is ES6/ES2015

Major update from JavaScript as a programming language

Modules, classes and more

Helps in developing Angular apps

TypeScript is a typed superset of ES6:

Annotations & types

Interfaces

Compiler

# TypeScript – tooling support

Types, Autocompletion, color coding.

Compile-time checking in editors.

Everything in TypeScript is  
*optional.*

You can always use just JavaScript



# Checkpoint

- Angular is a totally different beast than AngularJS
- Component-based vs. Page-based
- New Syntax
- New programming languages and design patterns
- Concepts are – mostly – the same.
- But: *you do need* a lot of boilerplate code to get started
- After that: never look around. Concentrate on components and other content