Angular Fundamentals
Module 1 - Introduction

Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

On Peter Kassenaar:

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

www.kassenaar.com/blog

info@kassenaar.com

Twitter: @PeterKassenaar

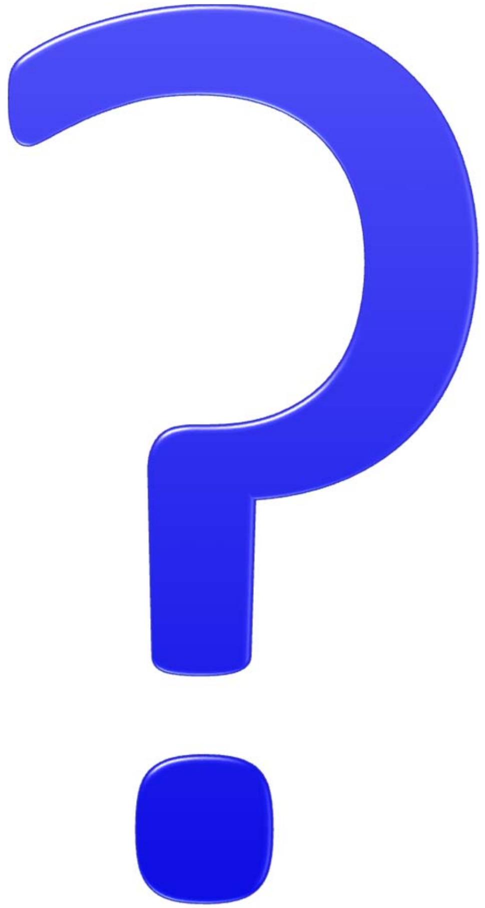
World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

www.angulartraining.nl

github.com/PeterKassenaar/cognizant

About you…



Introduce yourself shortly

Current knowledge, mobile apps, Angular apps?

Previous AngularJS 1.x- knowledge?

Other (web) languages?

Expectations of the training?

Specific or current projects?

Agenda - 3 days

- Introduction & short history – Why Angular?
- Angular 2/4/5/6 vs. Angular 1
- Hello World in Angular – Looking at the boilerplate-code - CLI
- Angular 2 in depth (modules):
  - Components
  - ECMAScript 2015 + TypeScript
  - Data binding
  - Dependency Injection (DI) – more components
  - Services and Http, Observables (RxJS)
  - Routing, Forms, Post training assessment/'exam'
- BEST PRACTICES / STYLE GUIDE

Date	Time	Session Title	Duration	Trainer
16-Jul-18	*9.00 - 9.30*	**Introduction by CDB leadership team**	0.5 hrs	CDB Leadership
	9.30 - 12.30	REACT : Module 1 : ReactJS Intro ; Module 2 : Components ; Lab: Case Study 1	3 hours	Henry Spiekerman
	12:30 - 13:15	*Lunch Break*		
	13:15 - 16:30	REACT : Lab: Case Study 1 (cont) ; Module 3 : Composition and LifeCycle ; Lab: Case Study 2	3 hours	Henry Spiekerman
	16:30 - 17:00	*Feedback Time - What went well, any changes/ improvements required ?*	0.5 hours	
17-Jul-18	9.00 - 12:30	REACT : Lab: Case Study 2 (cont) ; Module 4 : Forms and Validation ; Module 5 : ReactJS Routing ;Lab: Case Study 3	3 hours	Henry Spiekerman
	12.30 - 1.15	*Lunch Break*		
	13:15 - 16:30	REACT : Lab: Case Study 3 (cont) ; Module 6 : Redux Framework ; Lab: Case Study 4 (only, if time permits)	3 hours	Henry Spiekerman
	16:30 - 17:00	[illegible]	0.5 hours	
18-Jul-18	9.00 - 12:30	ANGULAR : Module 1: Introduction, context and concepts	3 hours	Peter Kassenaar
	12:30 - 13:15	*Lunch Break*		
	13:15 - 16:30	ANGULAR : Module 2: data binding and using client-sided Models	3 hours	Peter Kassenaar
	16:30 - 17:00	*Feedback Time - What went well, any changes/ improvements required ?*	0.5 hours	
19-Jul-18	9.00 - 12.30	ANGULAR : Module 3: Services and RxJS	3 hours	Peter Kassenaar
	12:30 - 13:15	*Lunch Break*		
	13:15 - 16:30	ANGULAR :Module 4: Building apps with multiple components	3 hours	Peter Kassenaar
	16:30 - 17:00	*Feedback Time - What went well, any changes/ improvements required ?*		
20-Jul-18	9.00 - 12.30	ANGULAR - Module 5 : Angular Routing	3 hours	Peter Kassenaar
	12:30 - 13:15	*Lunch Break*		
	13:15 - 16:30	ANGULAR - Module 6 : Forms, and a look at Angular Next Steps	3 hours	Peter Kassenaar
	16:30 - 17:00	**Post Training Assessment / Quiz**	0.5 hrs	Peter Kassenaar

So…keywords

Angular Structure and Architecture

Tooling

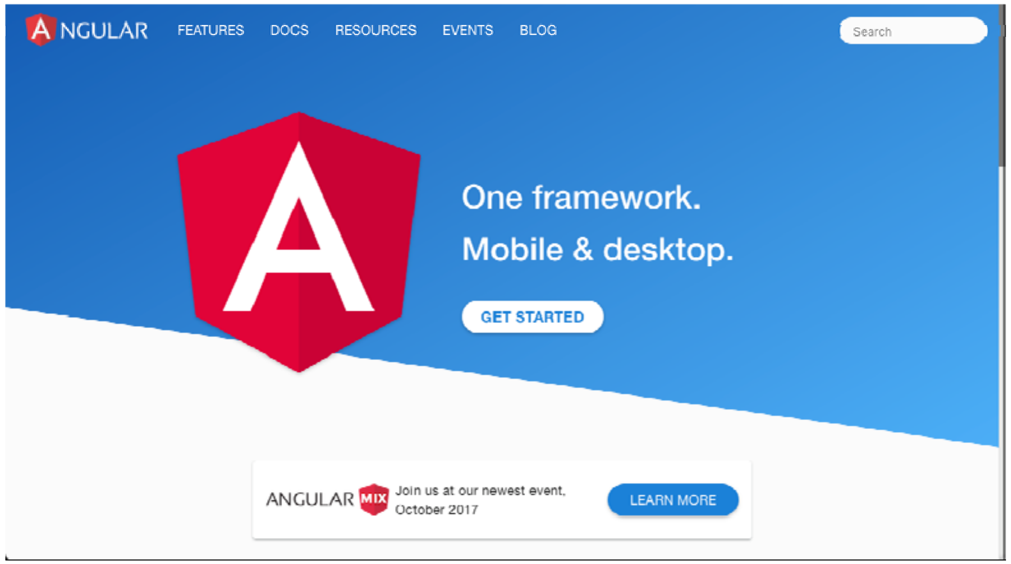
Best Practices

Components and Decorators

Communication in your app

Materials

Software	(downloads)
Handouts	(Github, Cognizant Repo)
Exercises	(Github)
Websites	(online)



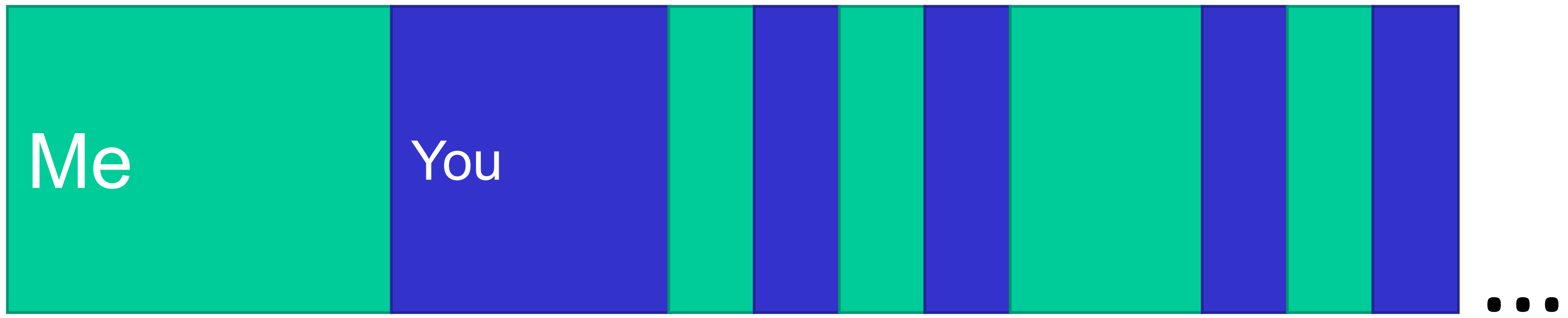
angular.io/

2 Guidelines

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

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

How I work...



Questions?

Angular**JS** vs. Angular

Features, differences

And similarities

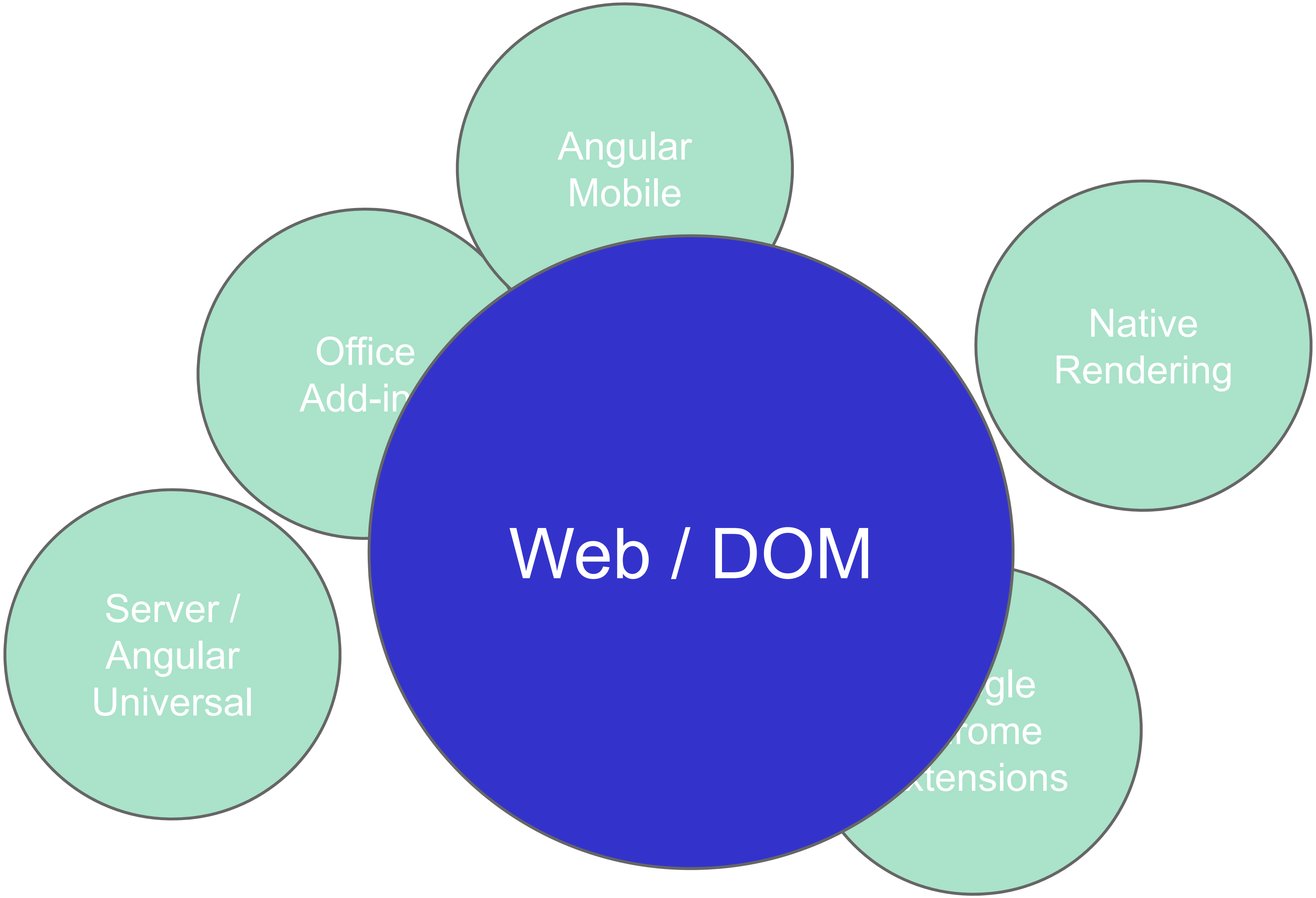
MV* Framework

2.0

Platform

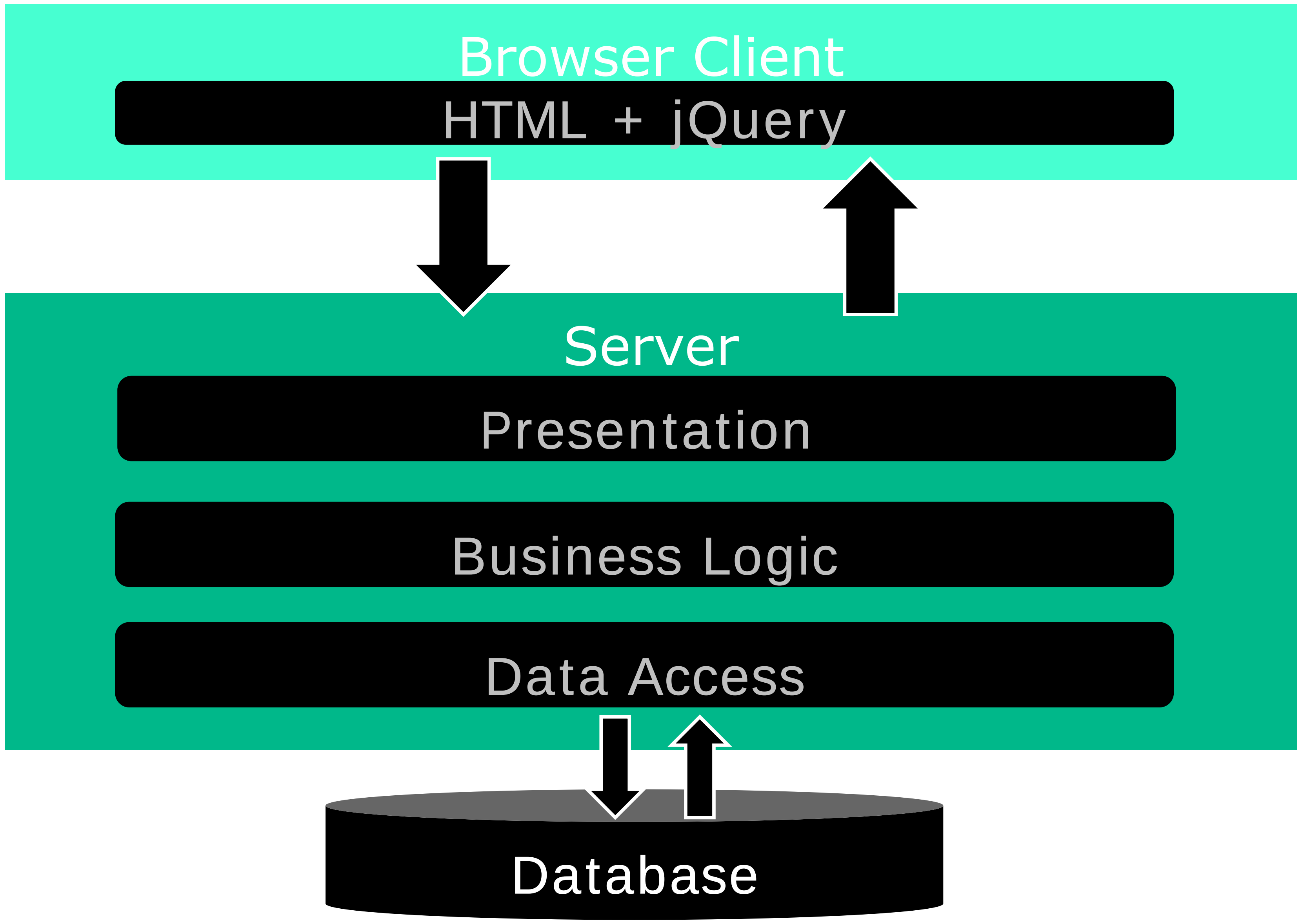
Framework to Platform

	Scaffolding	Code completion & Refactoring	Debugging
Tooling	Angular CLI	Language Services	Augury
Libraries	Material 2	Mobile	Universal
	Compile	Change Detection	Renderer
Core	Components & Dependency Injection	Decorators	Zones


Angular Mobile
Native Rendering
Office Add-in
Web / DOM
Server / Angular Universal
gle rome xtensions

Conventional Web App

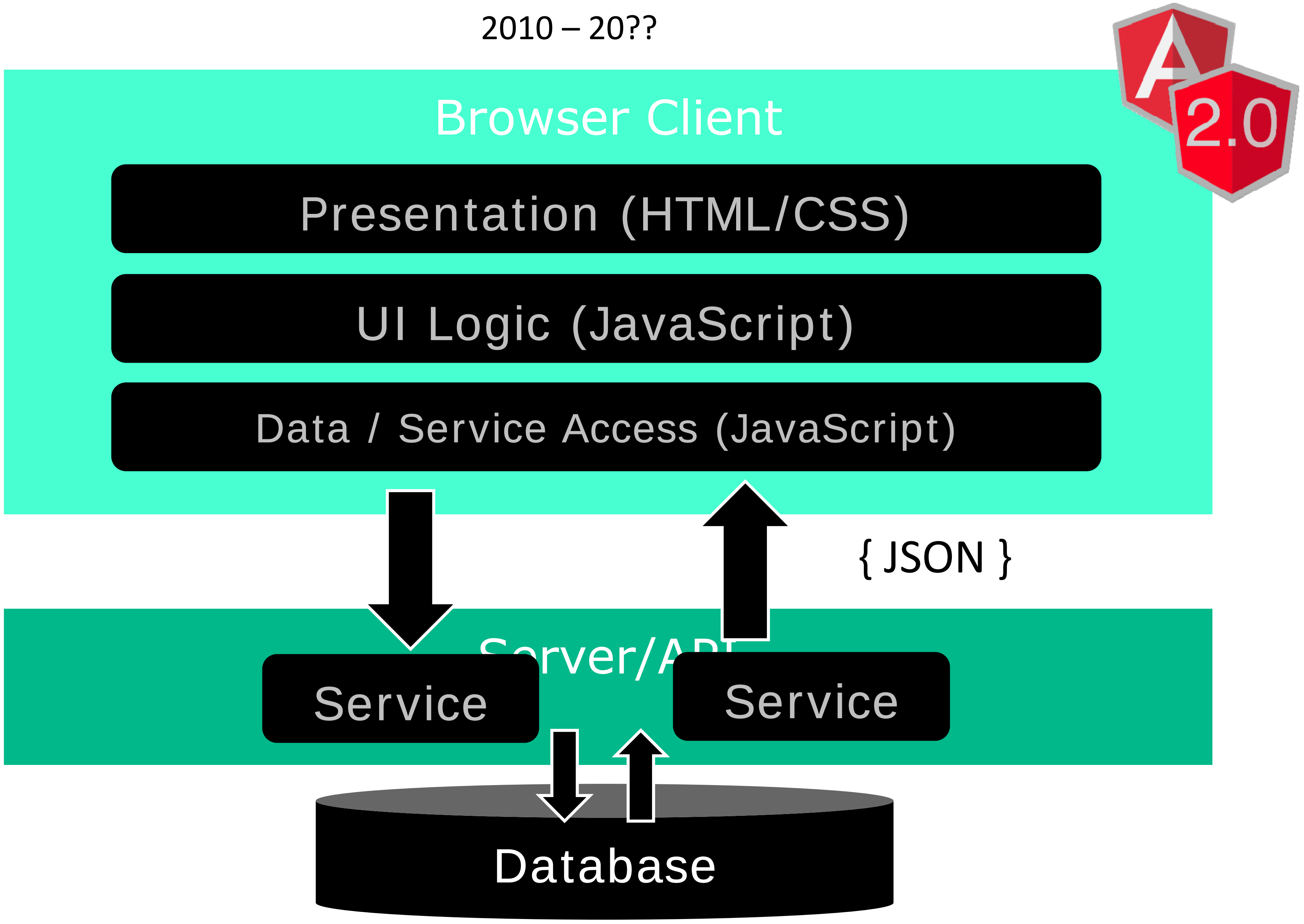
2000 - 2013

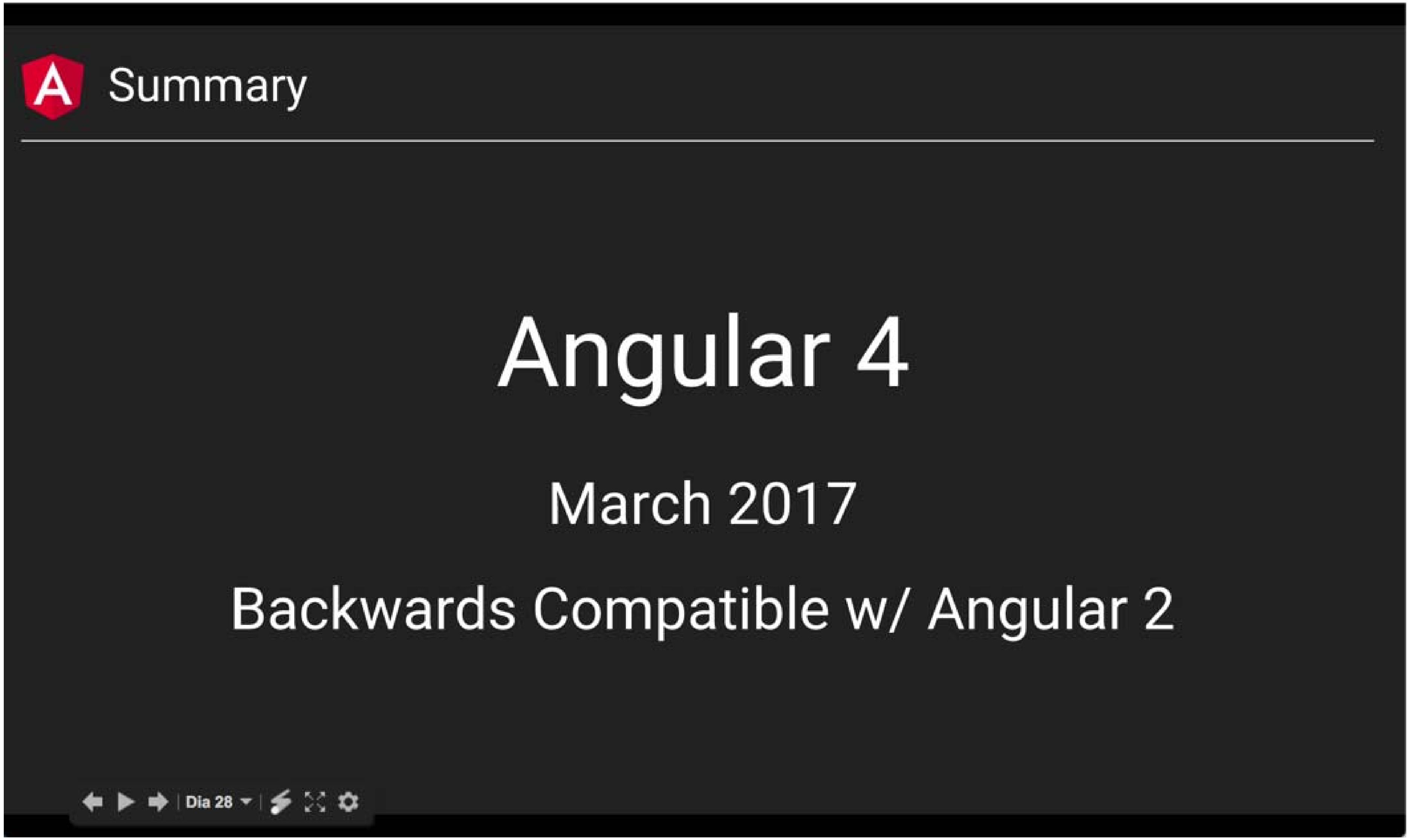


However, ca. 2010:

Single Page Application

2010 – 20??





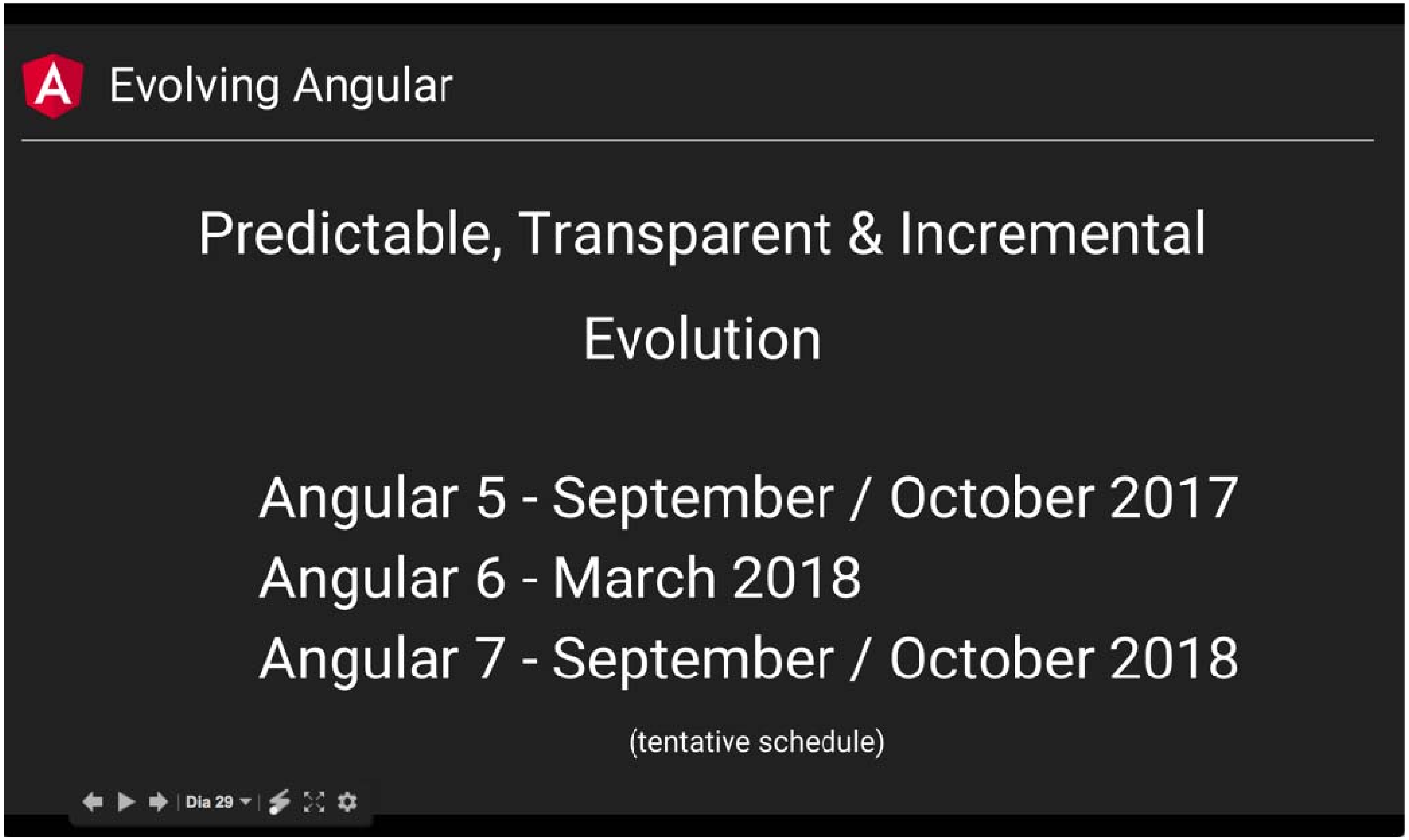
Summary

Angular 4

March 2017

Backwards Compatible w/ Angular 2

@IgorMinar, https://www.youtube.com/watch?v=aJIMoLgqU_o
Dec. 9, 2016



Evolving Angular

Predictable, Transparent & Incremental Evolution

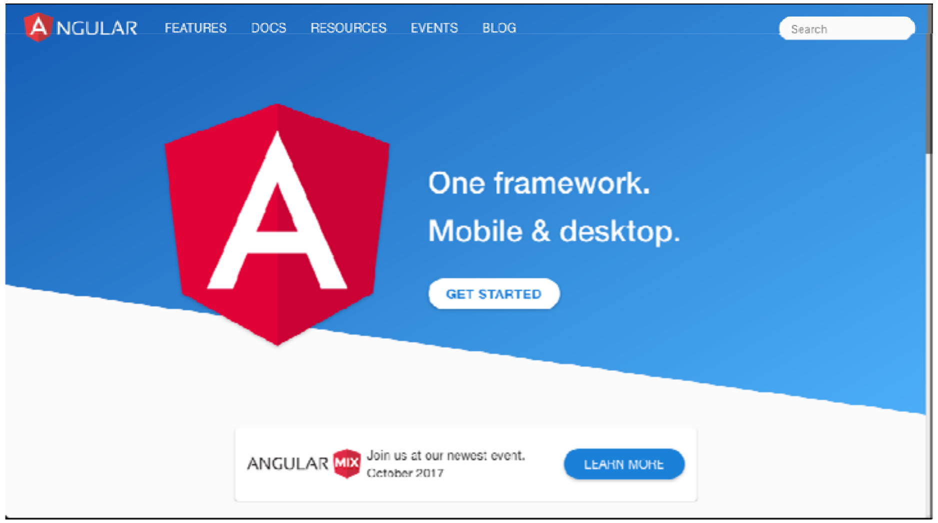
Angular 5 - September / October 2017
Angular 6 - March 2018
Angular 7 - September / October 2018

(tentative schedule)

"It's just

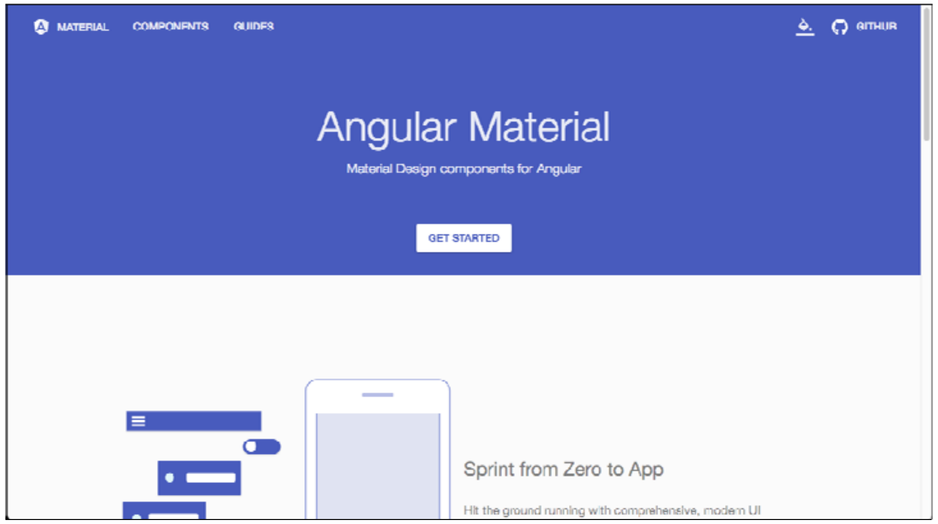
Angular

Angular as a Platform



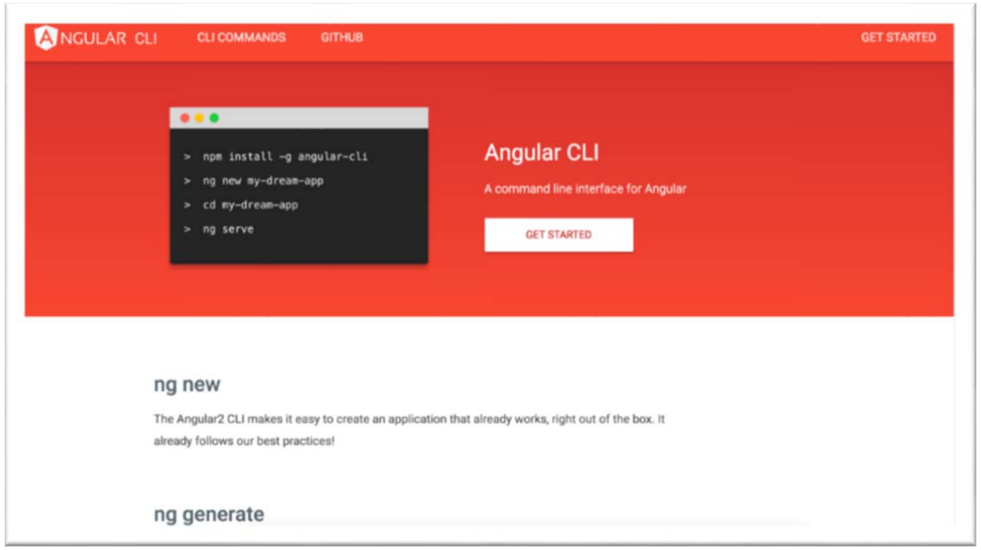
Join us at our newest event.
October 2017

https://angular.io/



Hit the ground running with comprehensive, modern UI

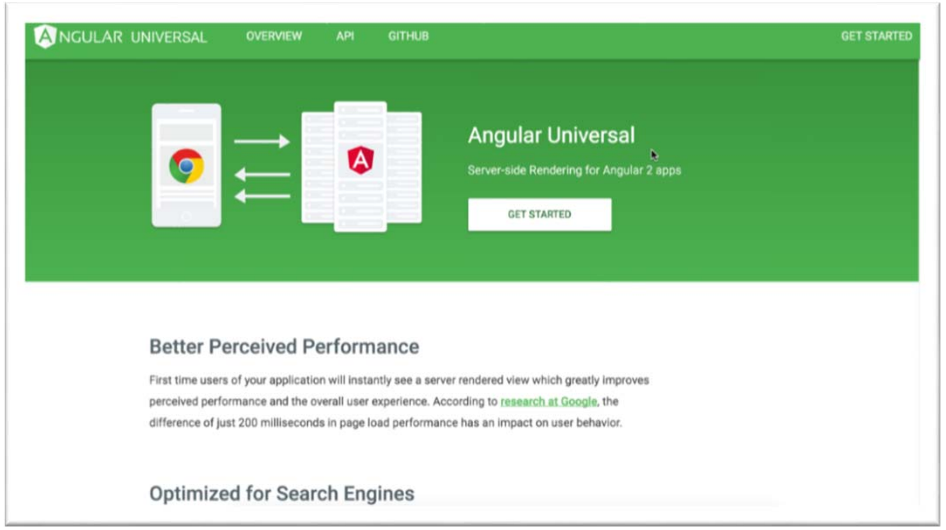
https://material.angular.io/



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

The Angular2 CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

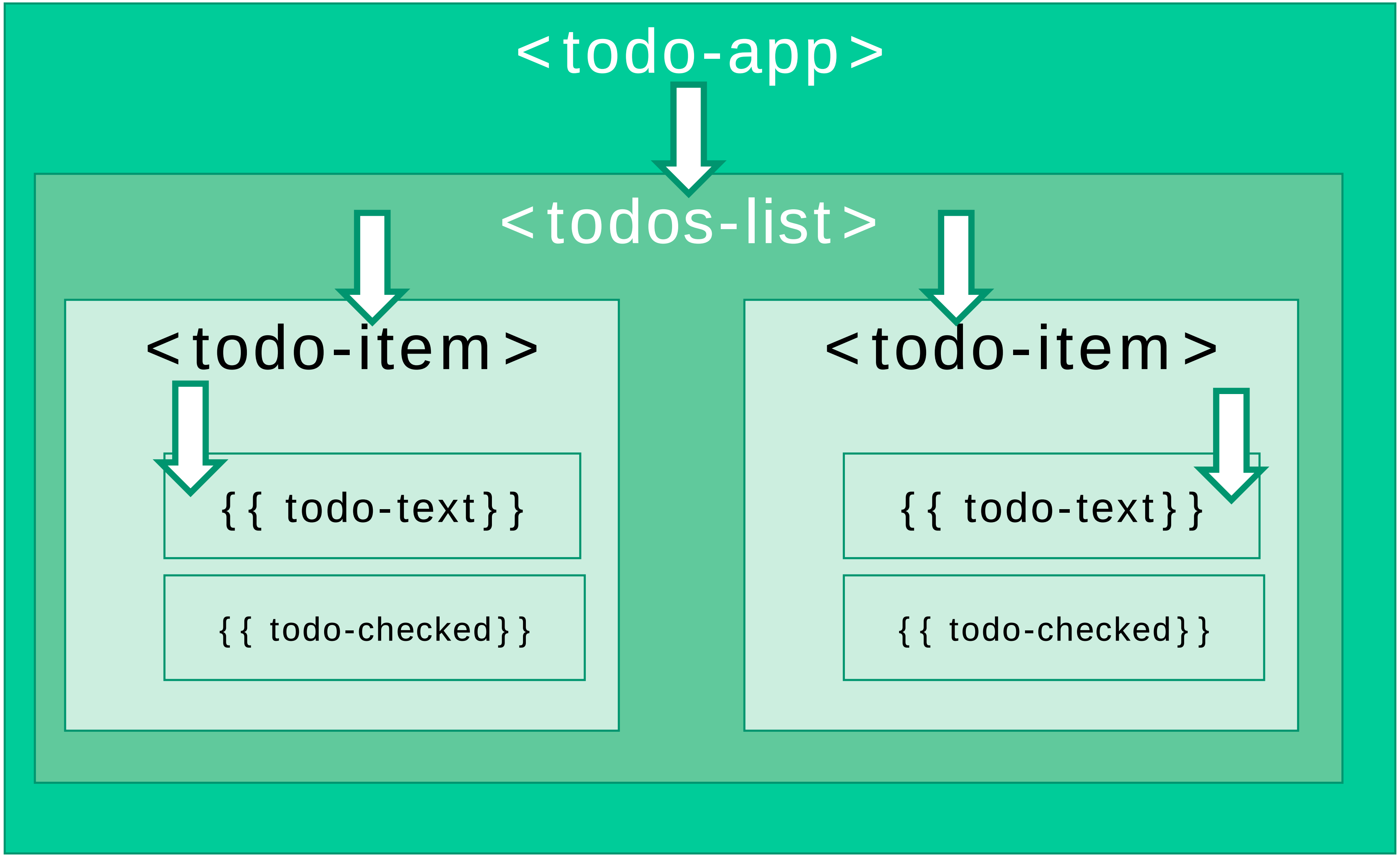
https://cli.angular.io/


ANGULAR UNIVERSAL
OVERVIEW
API
GITHUB
GET STARTED
Angular Universal
Server-side Rendering for Angular 2 apps
GET STARTED
Better Perceived Performance
Optimized for Search Engines

First time users of your application will instantly see a server rendered view which greatly improves perceived performance and the overall user experience. According to research at Google, the difference of just 200 milliseconds in page load performance has an impact on user behavior.

https://universal.angular.io/

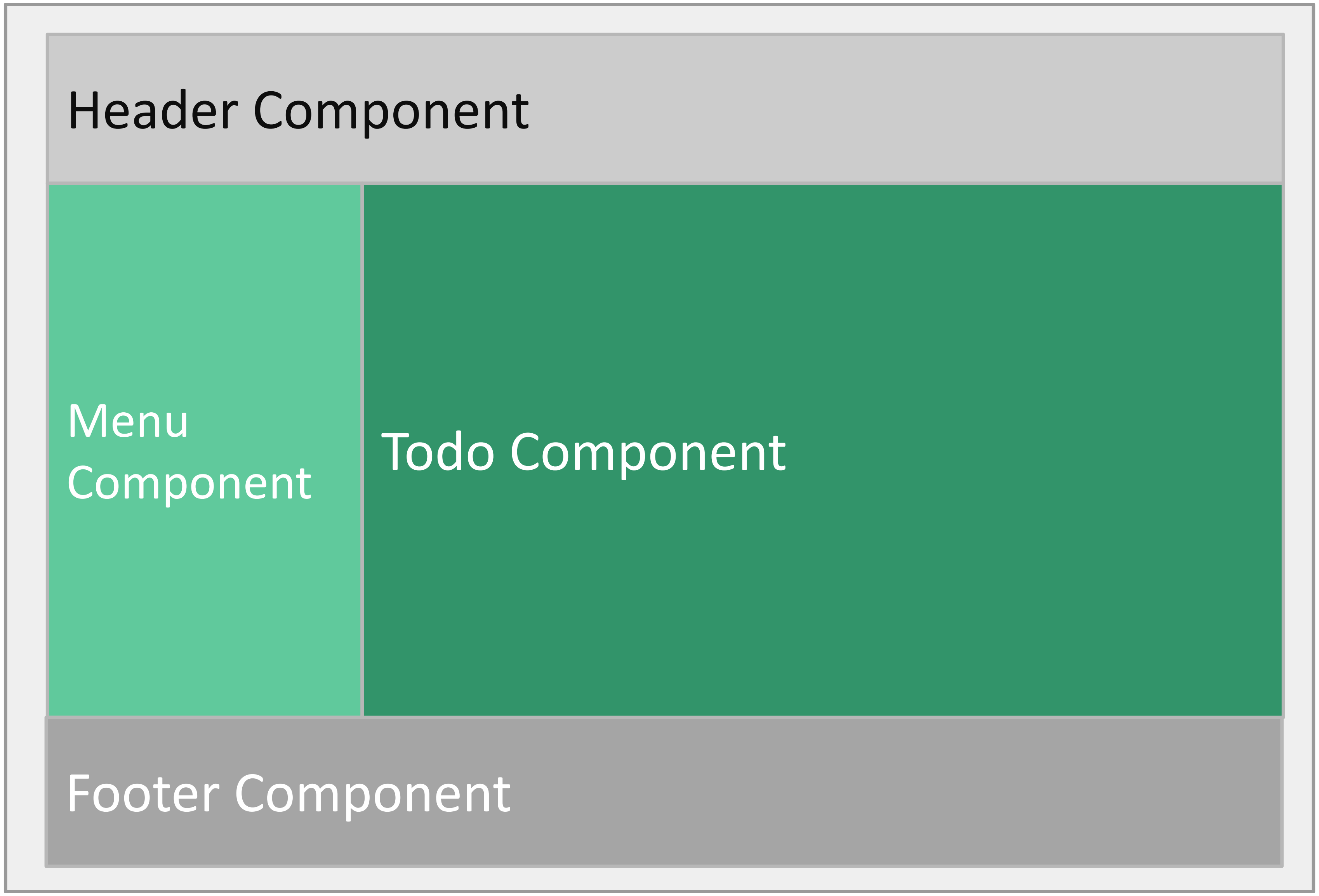
Angular 2 - components

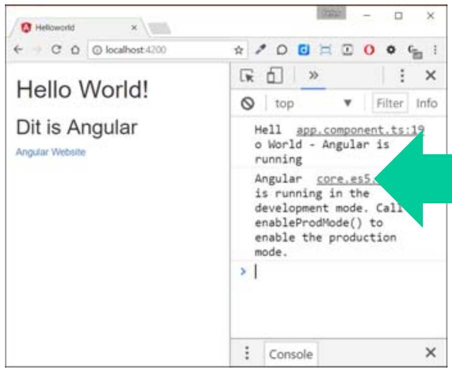


“An Angular-app is a tree

of components

Components – visually



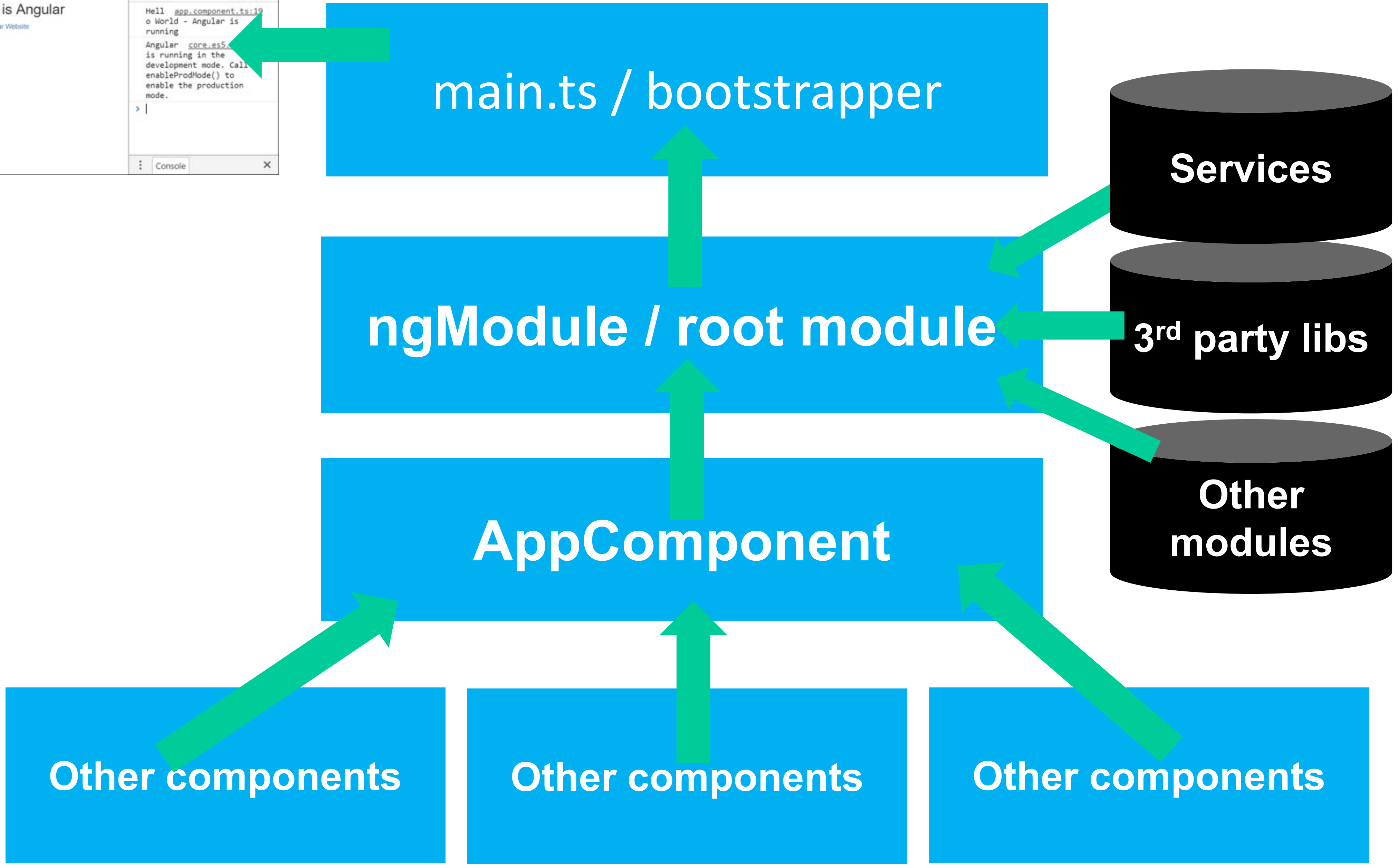


Hello World!

Dit is Angular

Angular Website

```
Hell  app.component.ts:19
o World - Angular is
running
Angular  core.es5.
is running in the
development mode. Call
enableProdMode() to
enable the production
mode.
> |
```



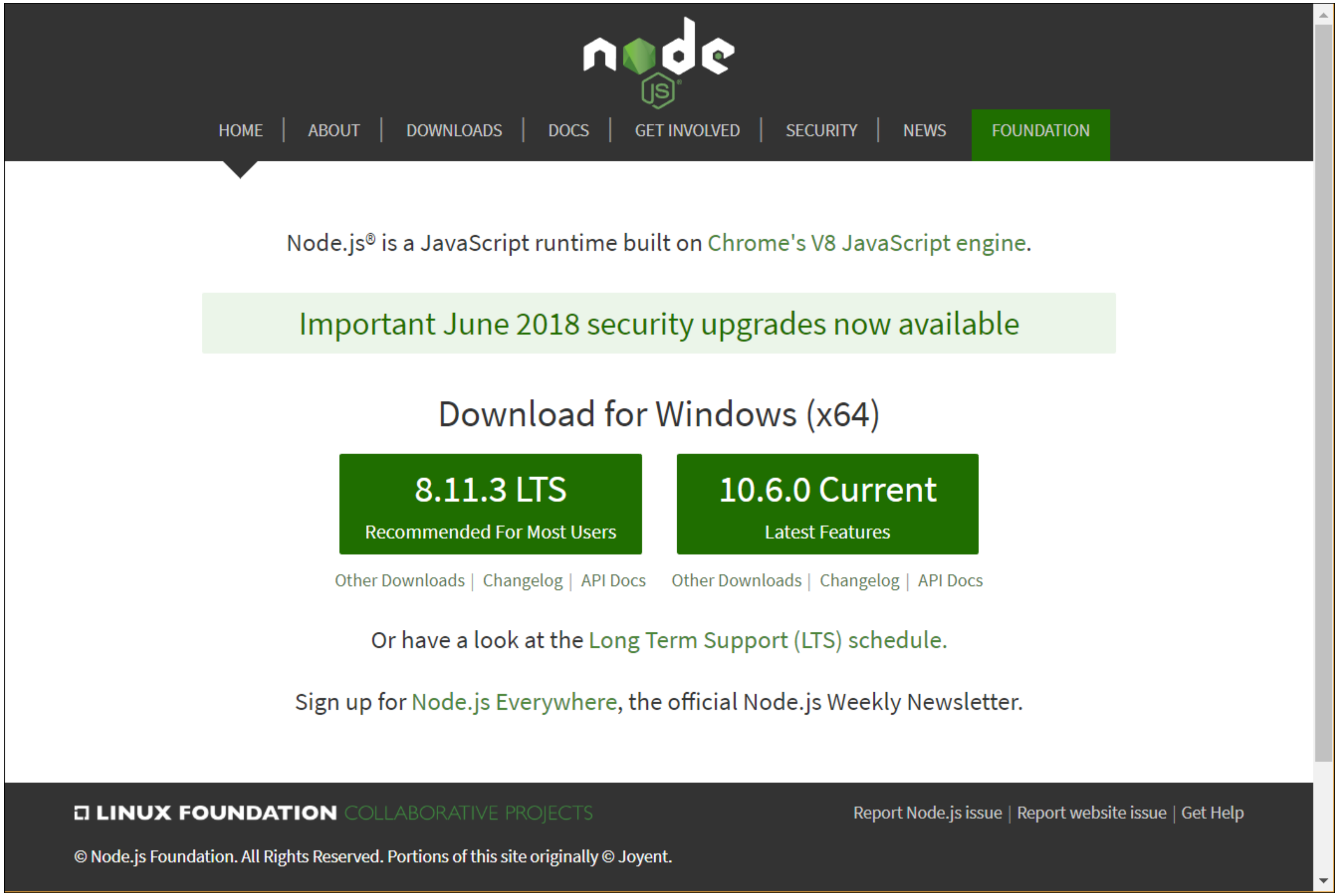
Let’s write some code

Hello World in Angular

Angular 1:

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

Angular development dependency: NodeJS 8.0+


HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Important June 2018 security upgrades now available

Download for Windows (x64)

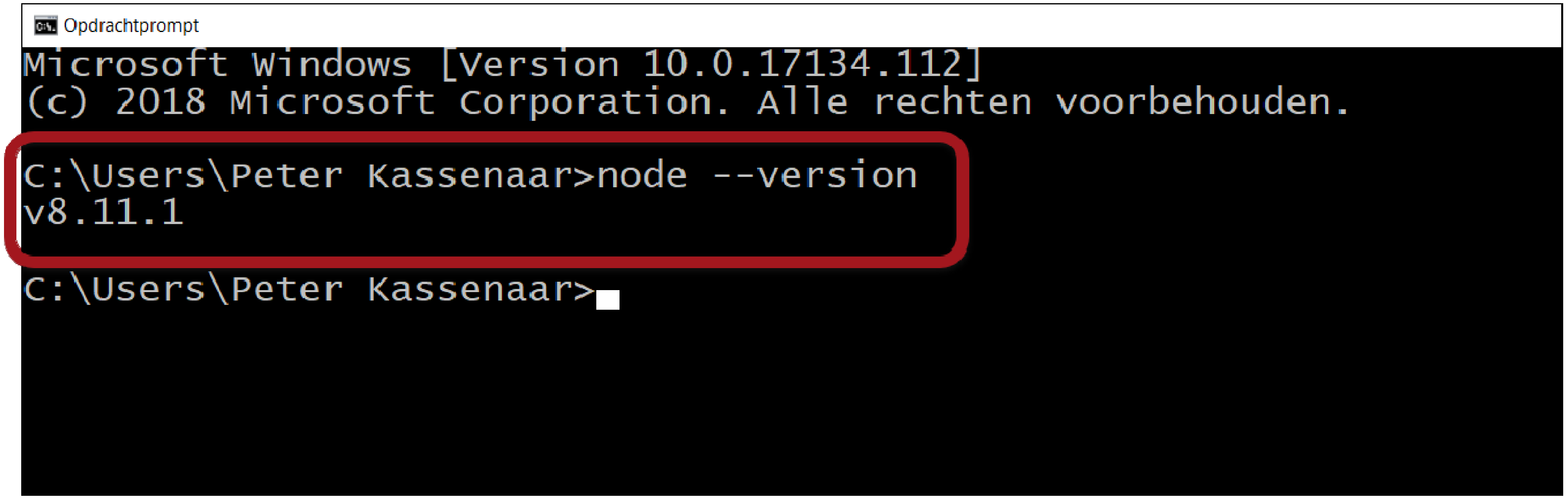
8.11.3 LTS
Recommended For Most Users
10.6.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

Sign up for Node.js Everywhere, the official Node.js Weekly Newsletter.

LINUX FOUNDATION COLLABORATIVE PROJECTS
Report Node.js issue | Report website issue | Get Help

Node – check your version



```
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>
```

Exercise

- Download or clone https://github.com/PeterKassenaar/voorbeeldenAngular2

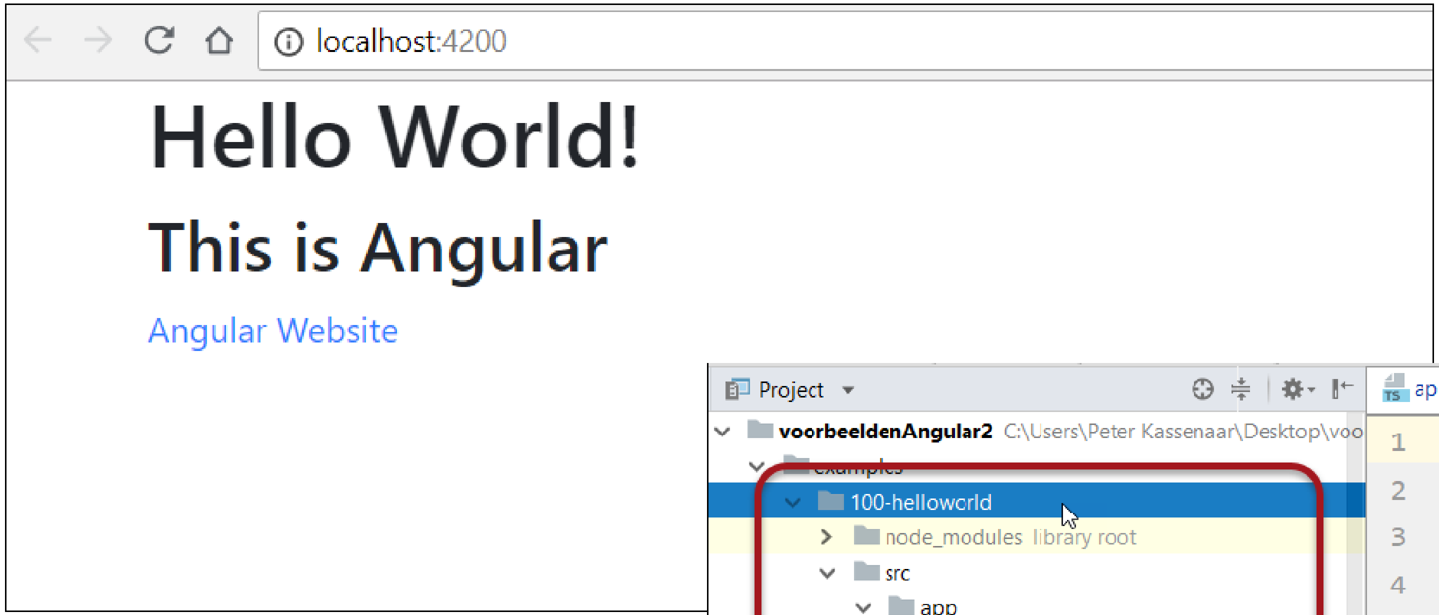
```
cd examples

cd 100-helloworld

npm install

npm start
```

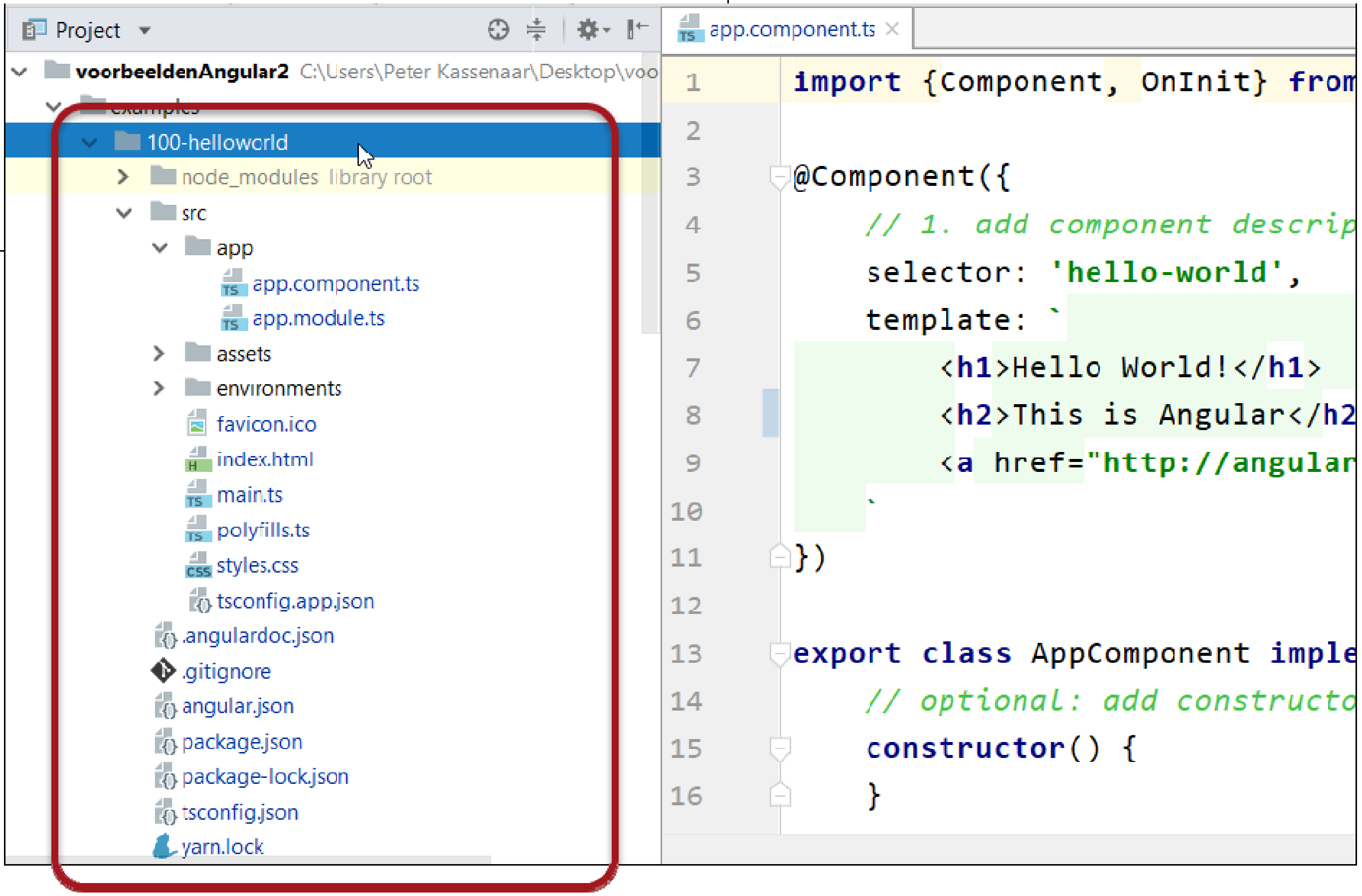
- Go to browser: http://localhost:4200


localhost:4200

Hello World!

This is Angular

Angular Website


Project

voorbeeldenAngular2 C:\Users\Peter Kassenaar\Desktop\voo
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

app.component.ts

```
import {Component, OnInit} from

@Component({
    // 1. add component descrip
    selector: 'hello-world',
    template: `
        <h1>Hello World!</h1>
        <h2>This is Angular</h2
        <a href="http://angular
    `
})

export class AppComponent imple
    // optional: add constructo
    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

```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": "6.0.0",
    "@angular/common": "6.0.0",
    "@angular/compiler": "6.0.0",
    "@angular/core": "6.0.0",
    "@angular/forms": "6.0.0",
    "rxjs": "^6.1.0",
    "zone.js": "^0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.6.0",
    "@angular/cli": "6.0.0",
    "typescript": "2.7.2"
  },
  "author": "Peter Kassenaar <info@kassenaar.com>"
}
```

Boilerplate files #2 - **`tsconfig.json`**

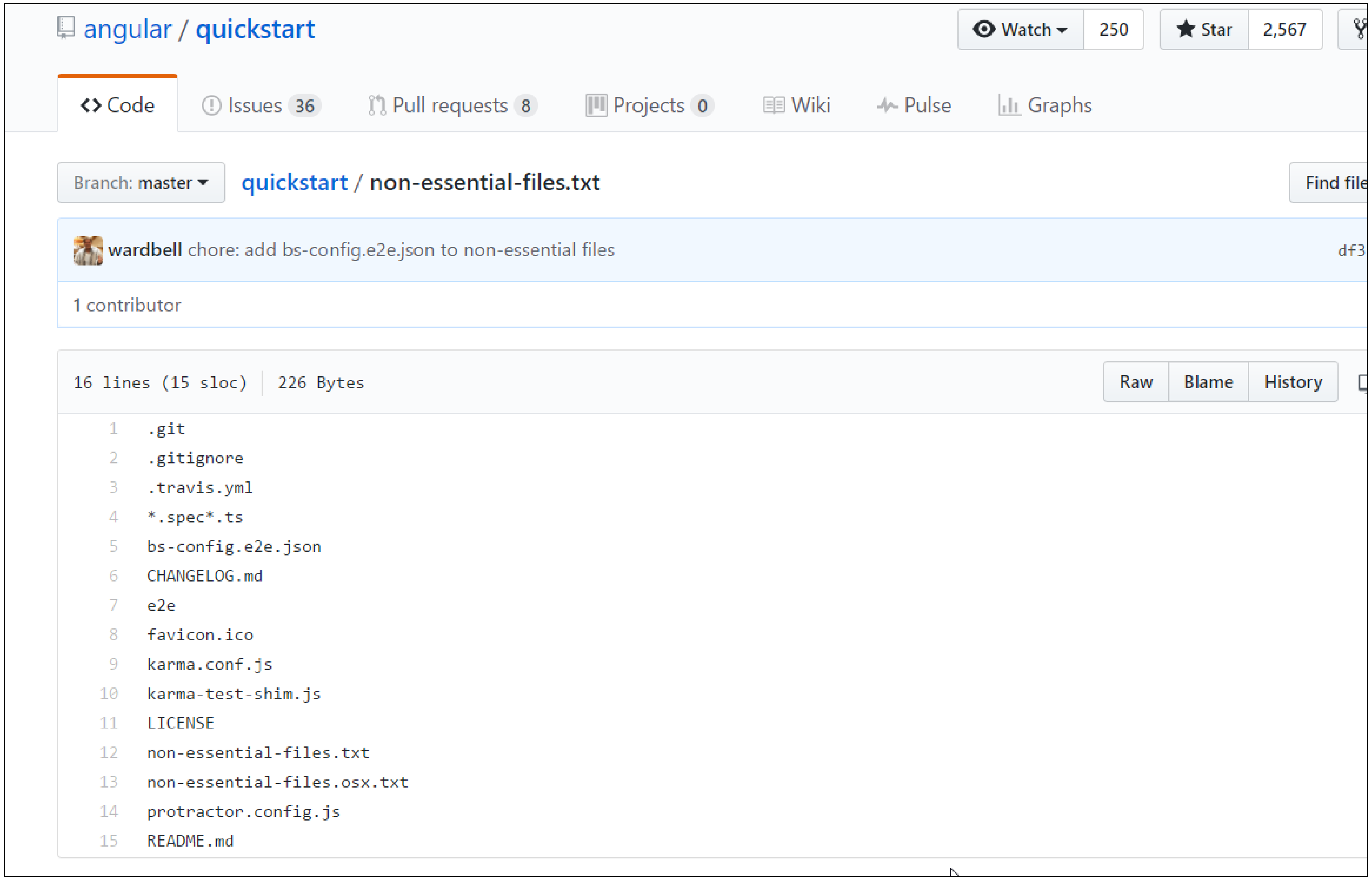
```
{
  "compileOnSave"  : false,
  "compilerOptions": {
   "outDir"                   : "./dist/out-tsc",
   "baseUrl"                  : "src",
   "sourceMap"                : true,
   "declaration"              : false,
   "moduleResolution"         : "node",
   "emitDecoratorMetadata"    : true,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

Boilerplate files #3 - **`angular.json`**

```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



```
.git
.gitignore
.travis.yml
*.spec*.ts
bs-config.e2e.json
CHANGELOG.md
e2e
favicon.ico
karma.conf.js
karma-test-shim.js
LICENSE
non-essential-files.txt
non-essential-files.osx.txt
protractor.config.js
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`

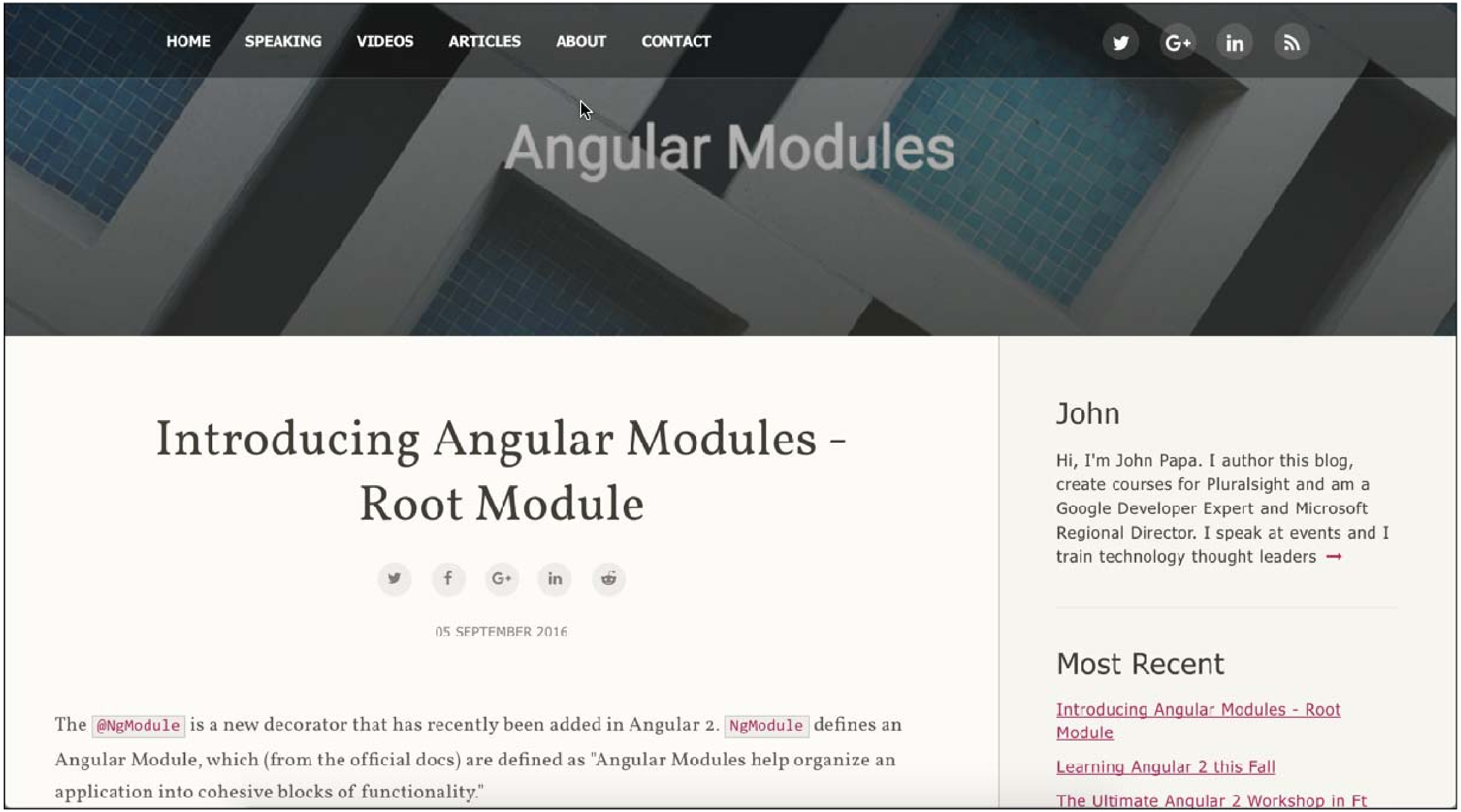
```
// Angular Modules
import {NgModule}       from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';

// 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



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
<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

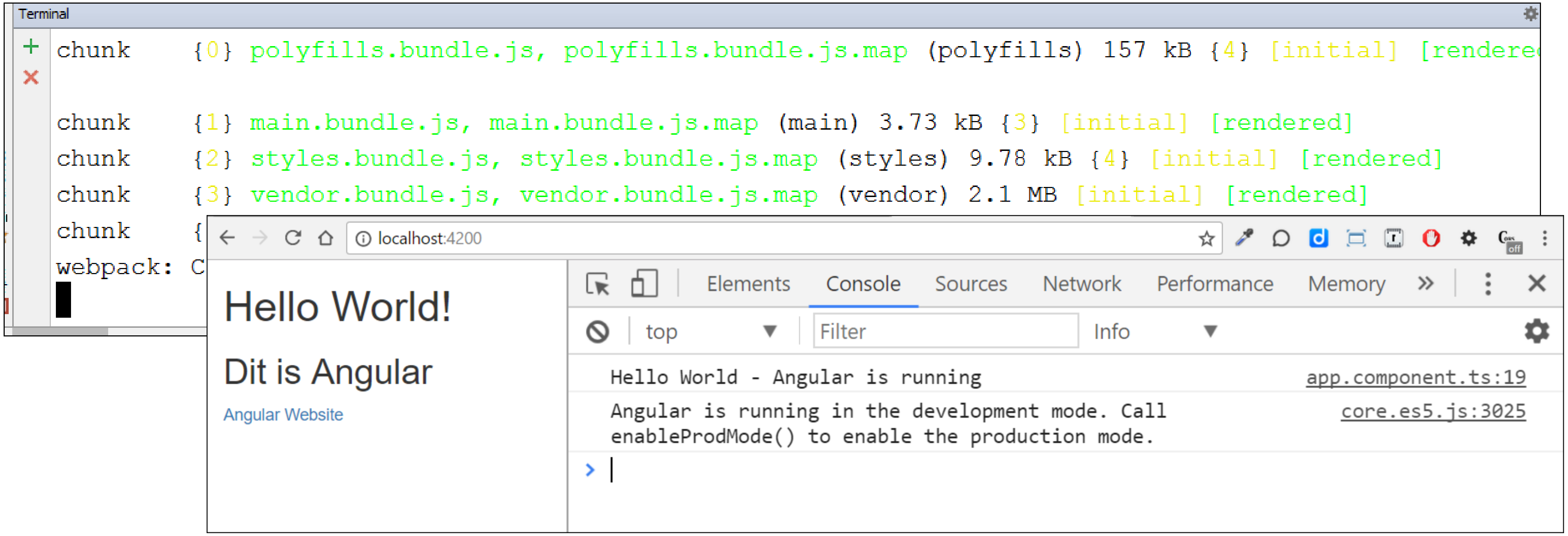
Element reference (`selector`) of root-component:

```
<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



```
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4} [initial] [rendere

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    {
webpack: C
```

localhost:4200

Hello World!

Dit is Angular

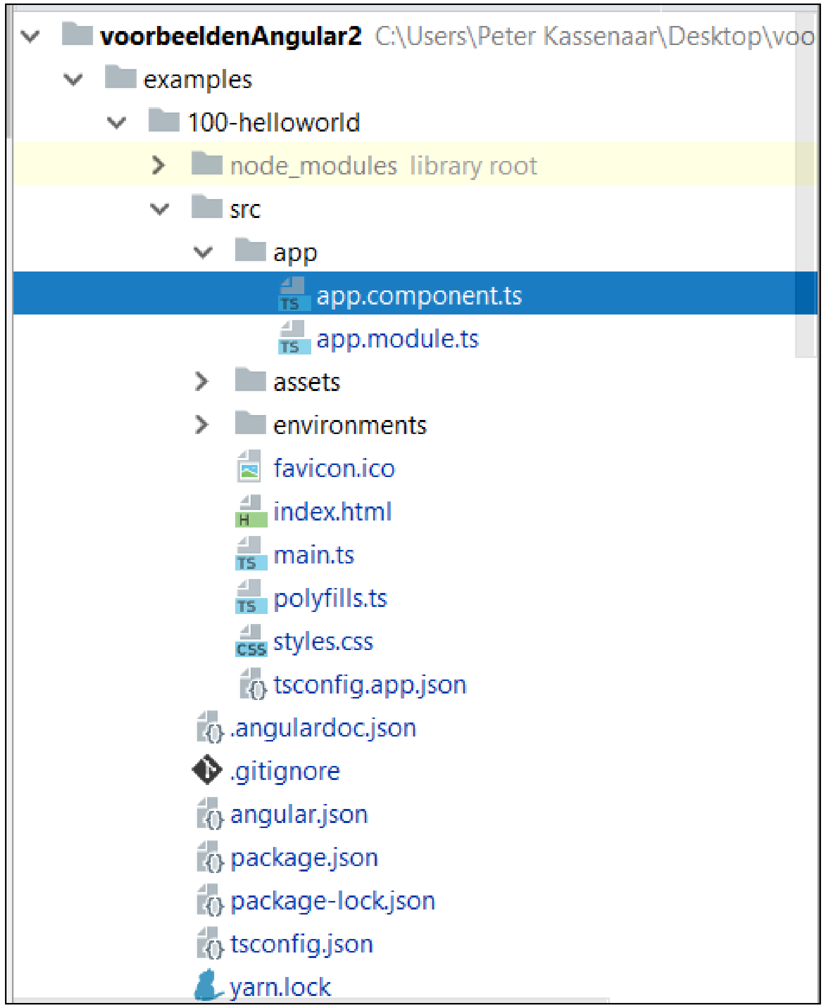
Angular Website

```
Hello World - Angular is running                               app.component.ts:19
Angular is running in the development mode. Call               core.es5.js:3025
enableProdMode() to enable the production mode.
```

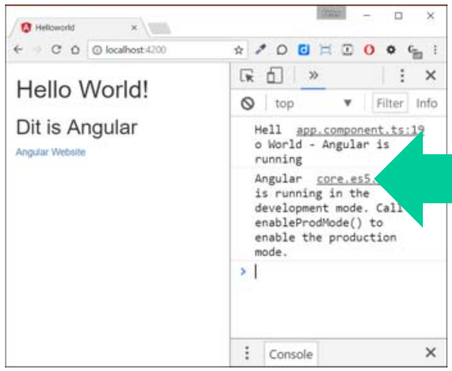
After that: edit `app.component.ts`

– Automagically refreshed through Live Reload

Basic Project Structure



```
voorbeeldenAngular2  C:\Users\Peter Kassenaar\Desktop\voo
  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
```

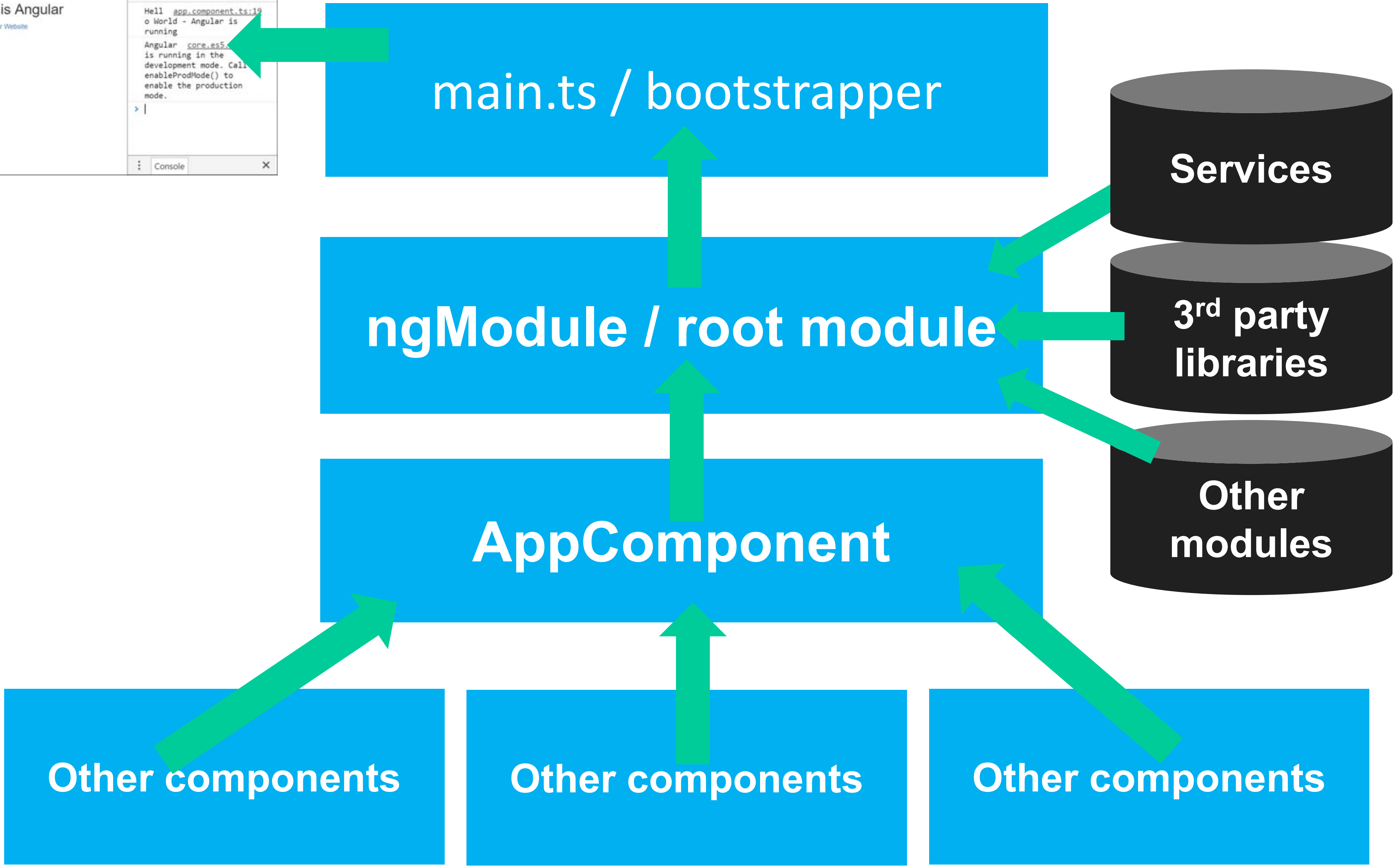


Hello World!

Dit is Angular

Angular Website

```
Hell  app.component.ts:19
o World - Angular is
running
Angular  core.es5.
is running in the
development mode. Call
enableProdMode() to
enable the production
mode.
```

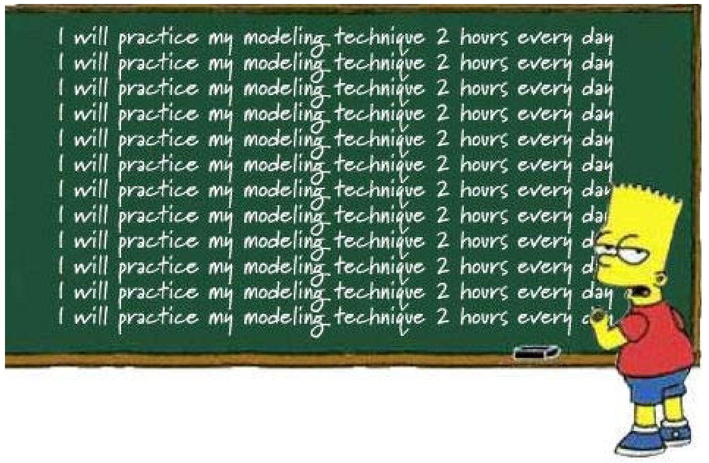


You need a lot of boilerplate code to start an Angular project.

(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

github.com/PeterKassenaar/voorbeeldenAngular2

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

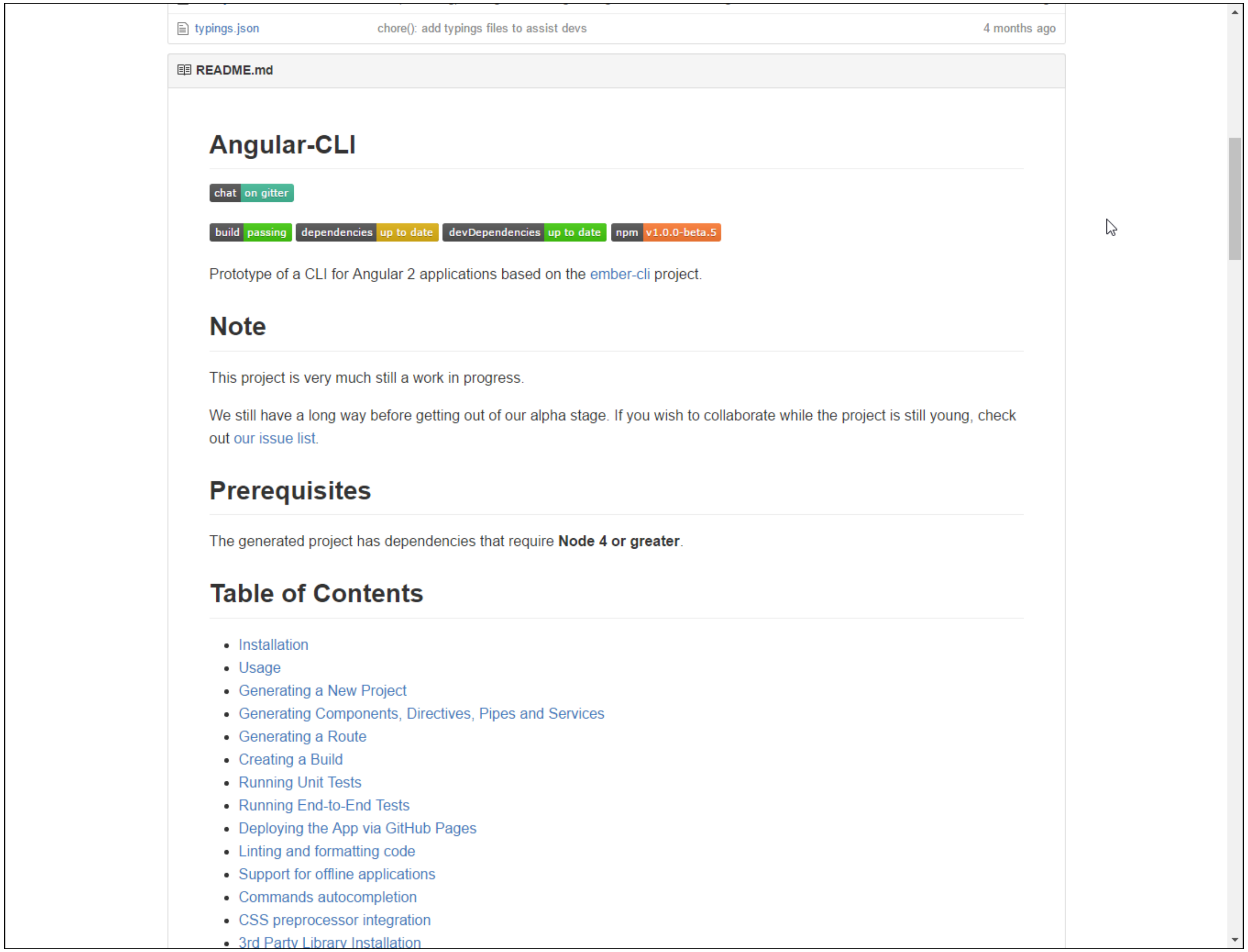
First : install CLI globally

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

en

https://cli.angular.io/

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



typings.json chore(): add typings files to assist devs 4 months ago

README.md

Angular-CLI

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.

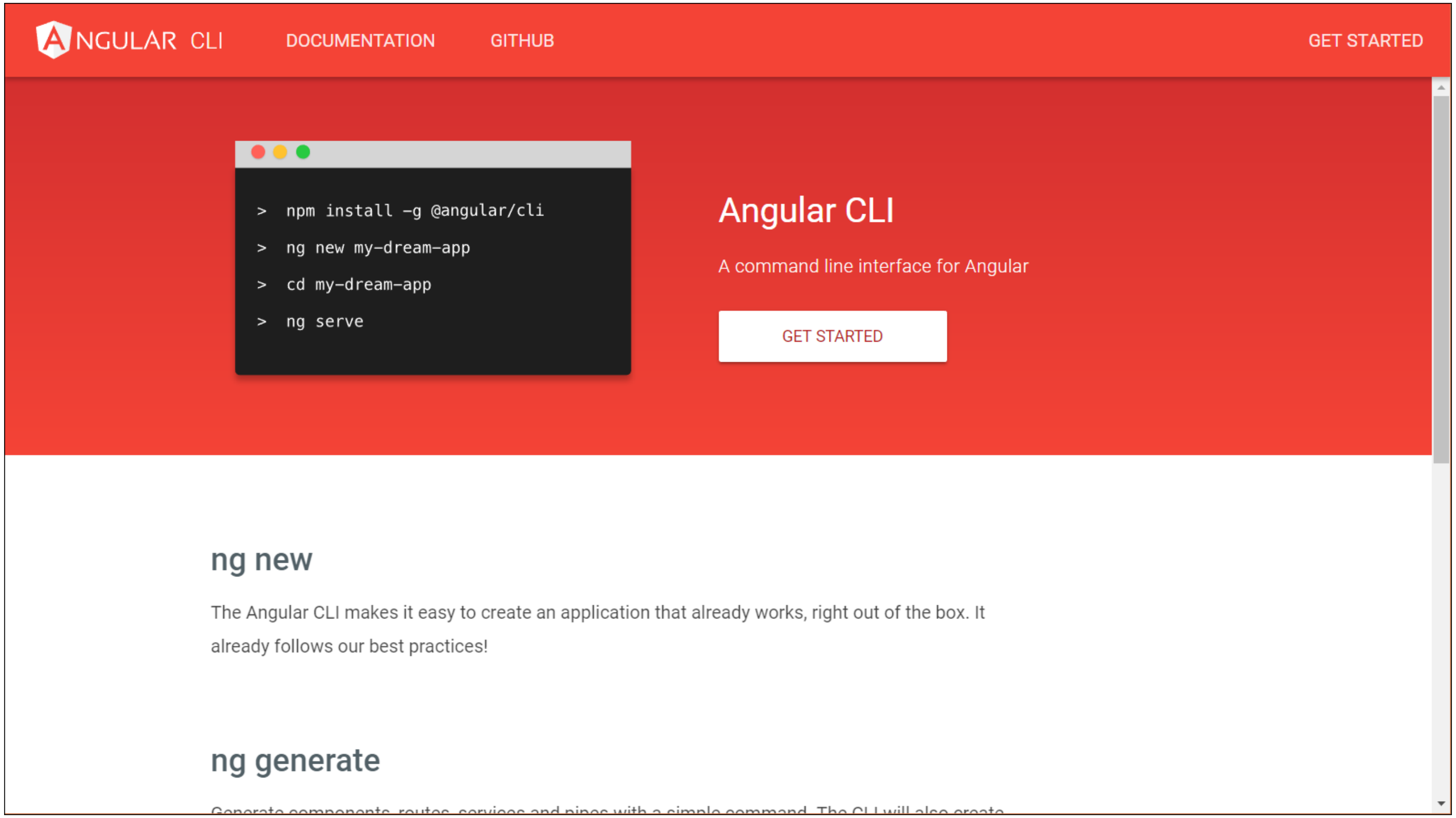
We still have a long way before getting out of our alpha stage. If you wish to collaborate while the project is still young, check out our issue list.

Prerequisites

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

Table of Contents

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



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

Angular CLI

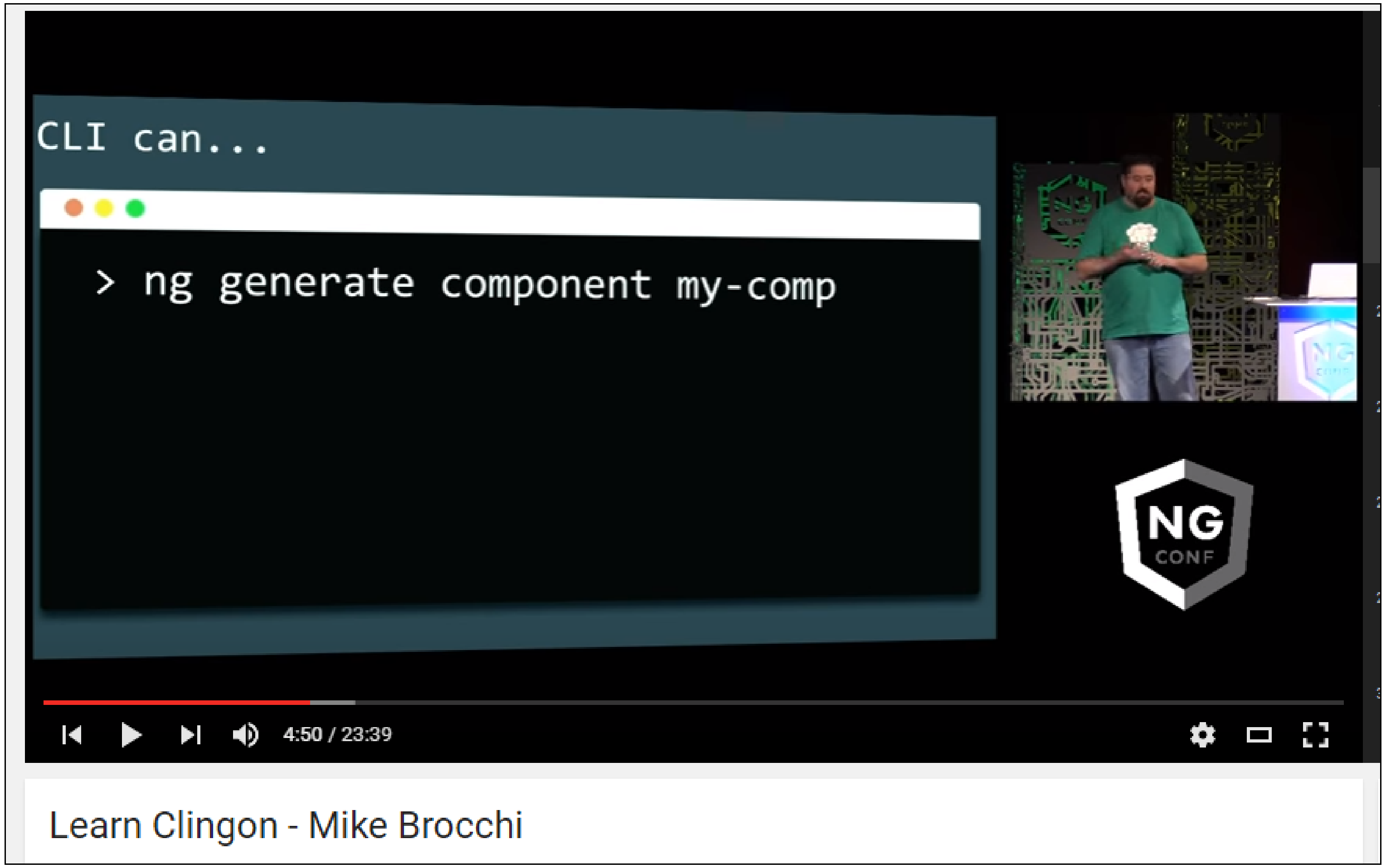
A command line interface for Angular

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



CLI can...

```
> ng generate component my-comp
```

Learn Clingon - Mike Brocchi

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

Main commands

```
ng new PROJECT_NAME

cd PROJECT_NAME

ng serve
```

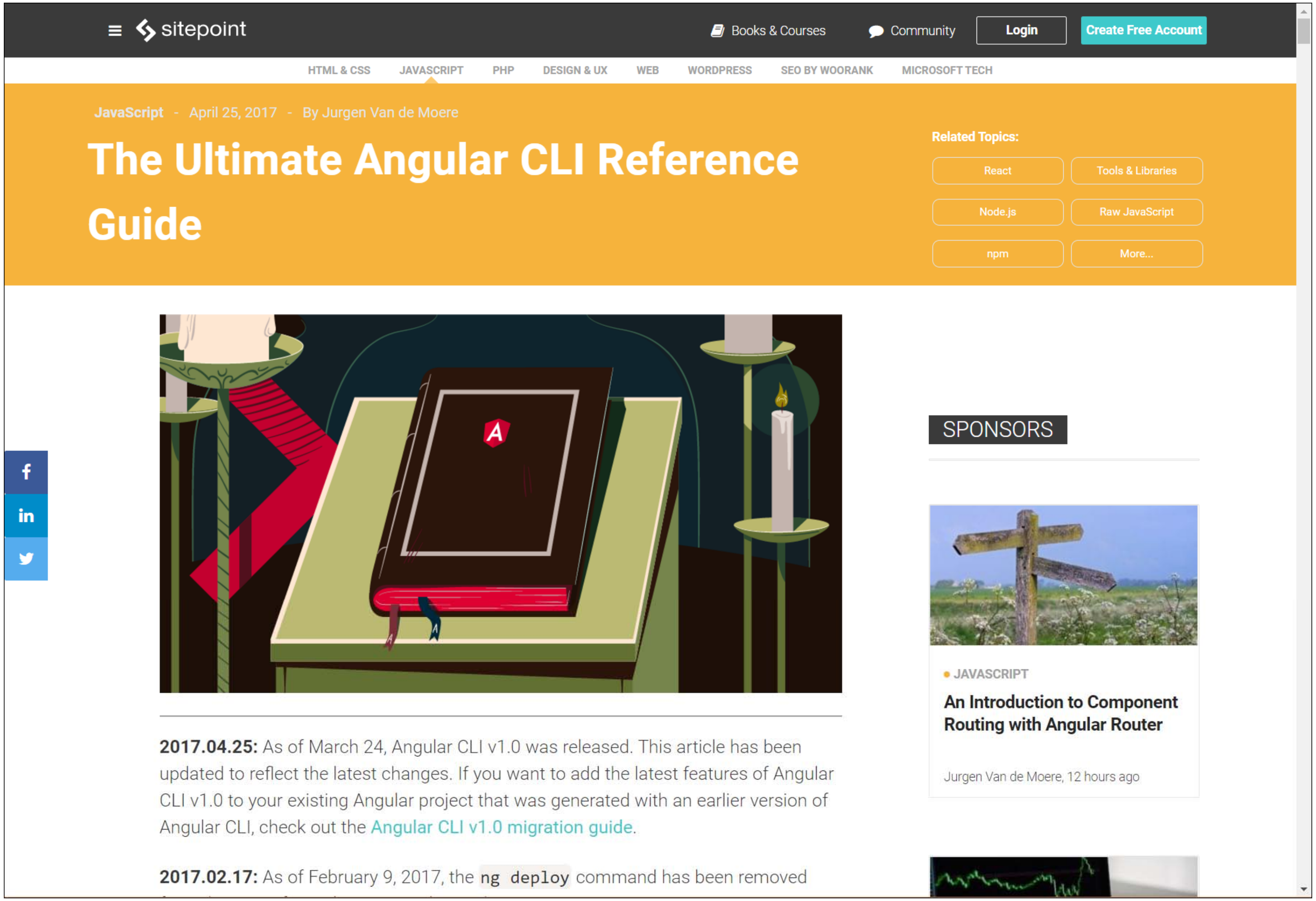
Project is served on `http://localhost:4200`

More info

Use the Angular CLI For Faster Angular 2 Projects

Creating Angular 2 projects has never been faster than with the Angular CLI

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



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

The Ultimate Angular CLI Reference Guide

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

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

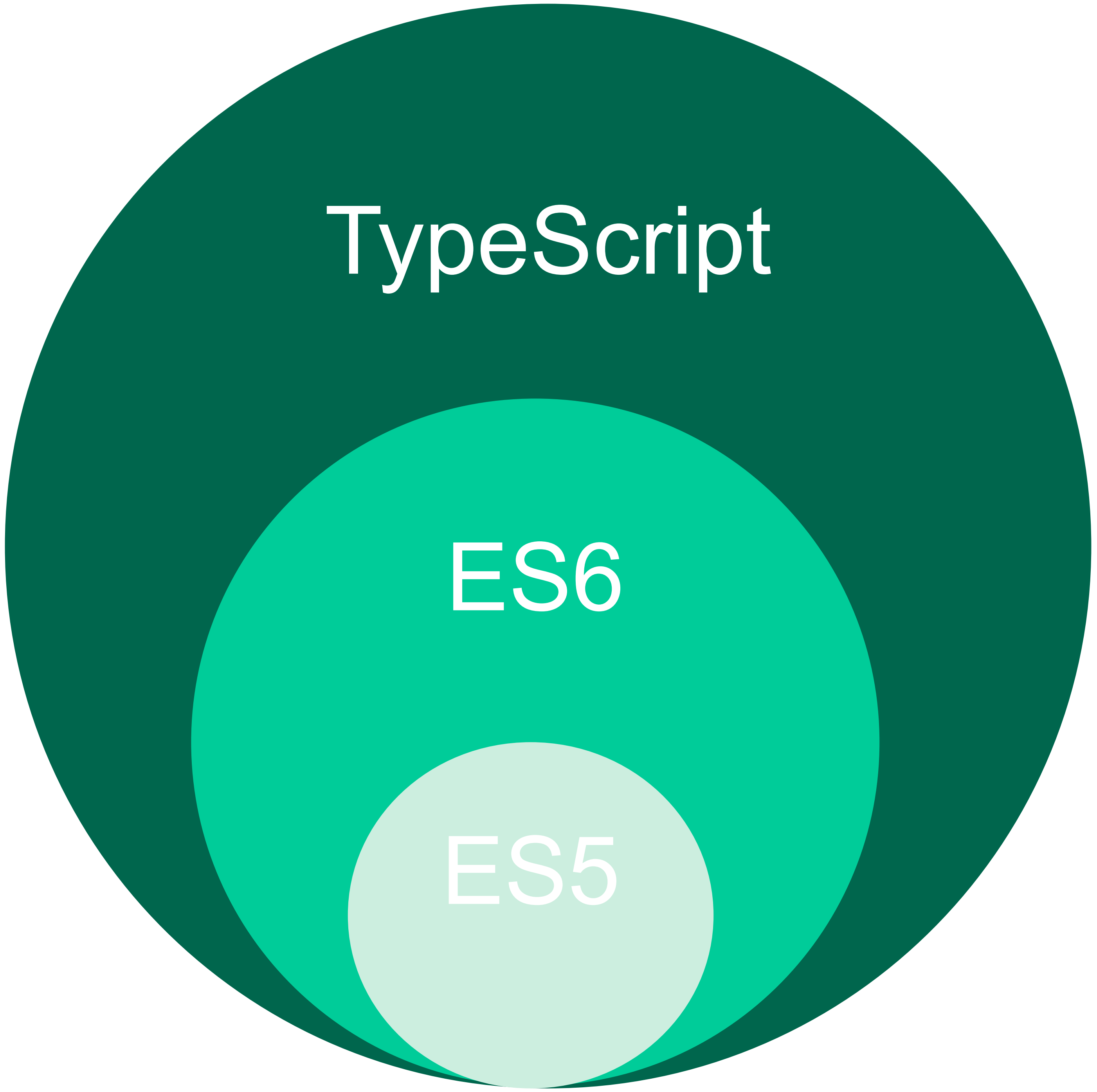
Angular Code - “Backend”

On TypeScript en ES6

Programming languages

ES6

TS


TypeScript
ES6
ES5

ES6 en 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.

Everyting 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 need a lot of boilerplate code to get started
- After that: never look around. Concentrate on components and other content