Angular Fundamentals
Module 1 – Core

Rabobank

Peter Kassenaar
info@kassenaar.com

Peter Kassenaar

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

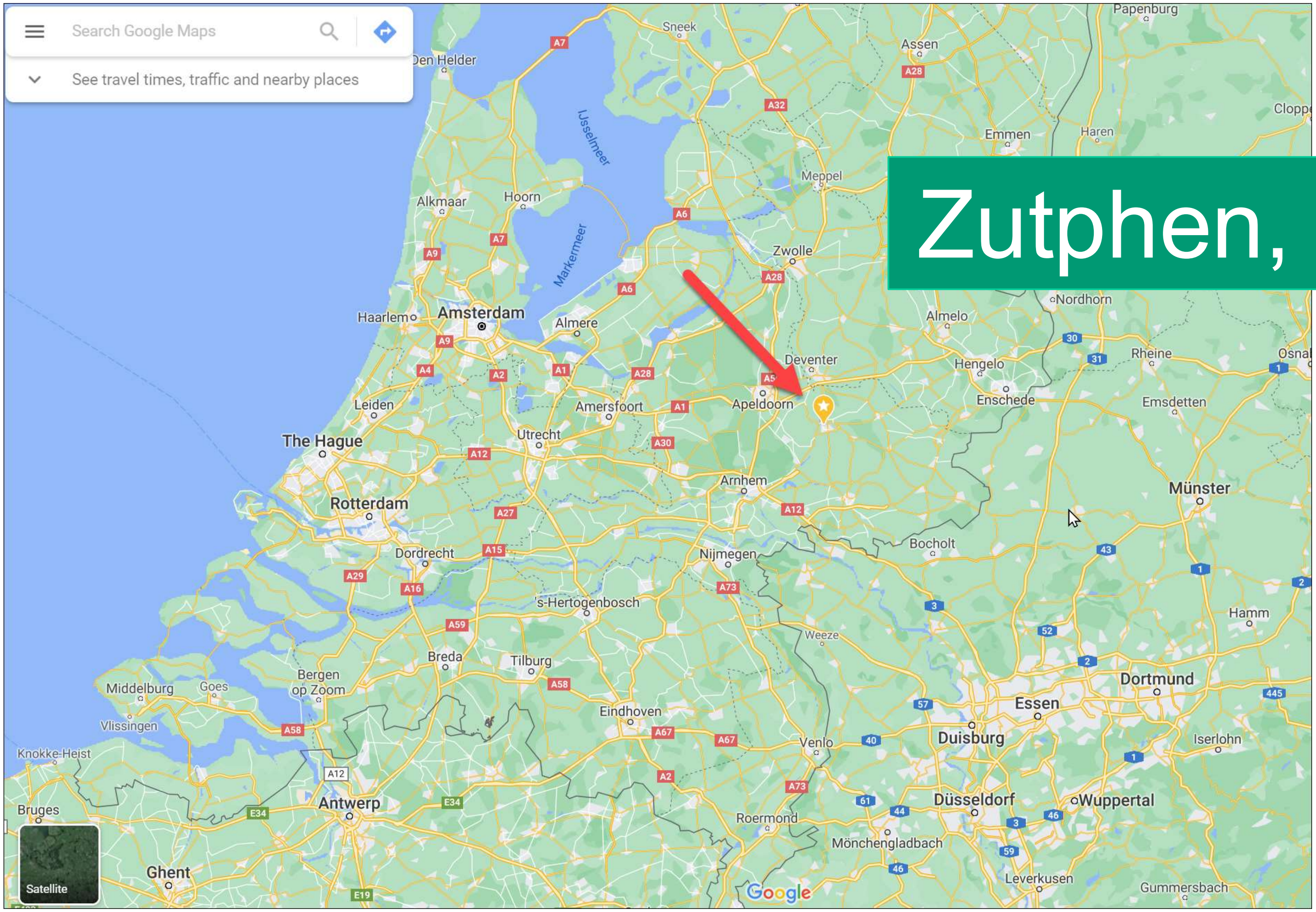
www.kassenaar.com/blog

info@kassenaar.com

Twitter: @PeterKassenaar

ING OHRA euricom A DIMENSION DATA COMPANY sanoma delta lloyd

zenito BETERE ZEKERHEID VOOR ONDERNEMERS AtoS woonbron OBERON INTERACTIVE ROC West-Brabant the eforum FACTORY


Search Google Maps
See travel times, traffic and nearby places
Satellite
Google
Den Helder
Sneek
Assen
Papenburg
Emmen
Haren
Cloppe
Meppel
IJsselmeer
Markermeer
Alkmaar
Hoorn
Zwolle
Nordhorn
Haarlem
Amsterdam
Almere
Almelo
Deventer
Hengelo
Rheine
Osnal
Enschede
Emsdetten
Leiden
Amersfoort
Apeldoorn
The Hague
Utrecht
Arnhem
Münster
Rotterdam
Bocholt
Dordrecht
Nijmegen
's-Hertogenbosch
Hamm
Weeze
Breda
Tilburg
Dortmund
Middelburg
Goes
Bergen op Zoom
Eindhoven
Essen
Vlissingen
Venlo
Duisburg
Iserlohn
Knokke-Heist
Antwerp
Düsseldorf
Wuppertal
Bruges
Roermond
Mönchengladbach
Ghent
Leverkusen
Gummersbach

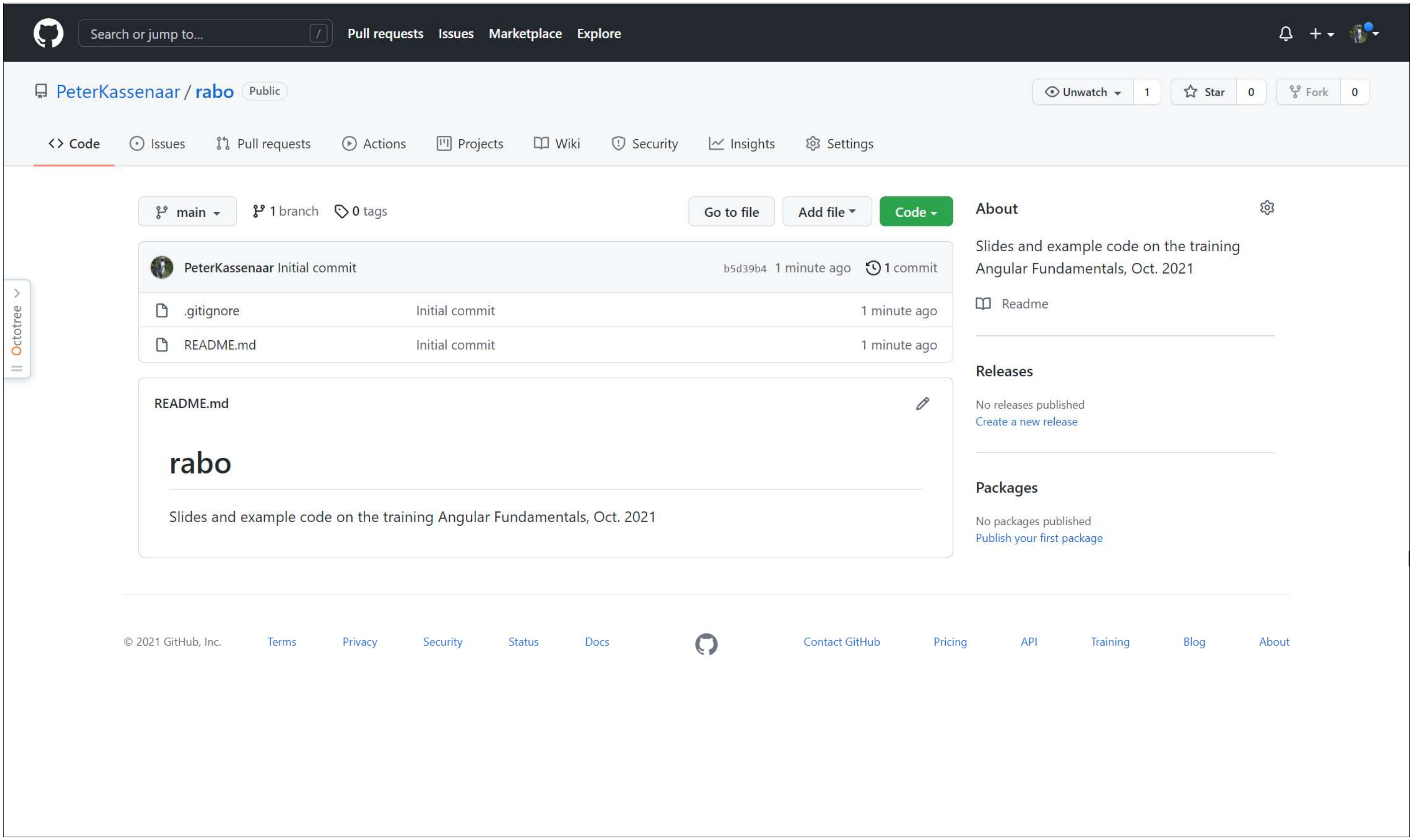
Zutphen, NL

World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

www.angulartraining.nl

github.com/PeterKassenaar/rabo



PeterKassenaar Initial commit b5d39b4 1 minute ago 1 commit

.gitignore	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

README.md

rabo

Slides and example code on the training Angular Fundamentals, Oct. 2021

About

Slides and example code on the training Angular Fundamentals, Oct. 2021

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

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?

Specific questions or techniques you want to learn about?

Goals of this training

*You're **not** going to be an Angular wizard in 3 days (sorry)*

but….

Goals

1. You *will* learn about the structure and architecture of Angular Apps. From a small `hello-world` app to the largest enterprise applications.

2. You are familiar with the main Angular concepts of the framework. You can always google the code details yourself.

3. You will have some hands-on experience regarding creating apps and components, services, API's/backends, security concepts, routing and forms.

4. You will have a general understanding of the way modern web apps are created using Angular, TypeScript and build tools.

Schedule - globally

27, 28, 29 October 2021 – Wed.-Fri

9:00 ~ 12:00 Morning session

Coffee/tea break

12:00 – 12:45 Lunch

12:45 ~ 16:00/16:15 Afternoon session

Coffee/tea break

Friday: probably wrap up a little bit early

Angular Advanced - Next year

24, 25, 26 January 2022 – Wed.-Fri

Try to get some Angular-experience before starting the *Advanced* course

(work- or hobby projects)

Agenda - Fundamentals, 3 days

- Introduction & short history – Why Angular?
- Key features of Angular 2 to 12
- 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), working with backend/API's
  - Routing, [Reactive] Forms
- BEST PRACTICES / STYLE GUIDE

Materials

Software	(Angular, NodeJS & NPM, Editor, browser)
Handouts	(Github)
Workshops	(Github)
Websites	(online)

angular.io/

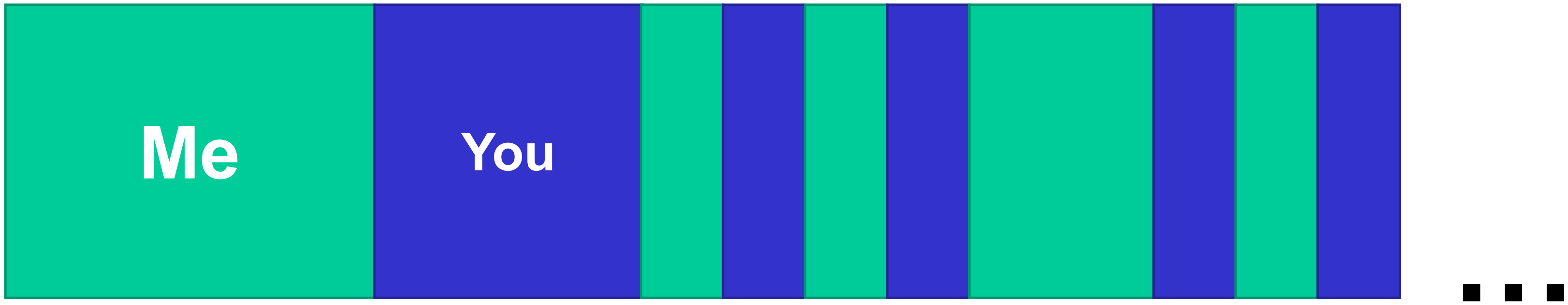
2 Guidelines

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

2. Example code – Updated to Angular V12.x.
   - To support the exercises – ready made examples
   - Work in progress – check Angular-site!
   - https://github.com/PeterKassenaar/voorbeeldenAngular2

     (Mainly Dutch)

How I work...



Advanced warning - First morning

Not so much code…

Concepts, architecture, structure

Questions?

Angular**JS** vs. Angular 2-12

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:

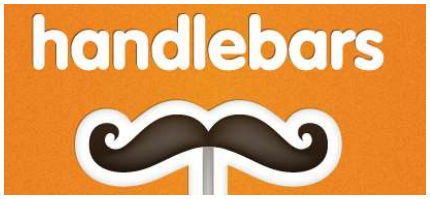
“Do we want to go back

to the jQuery days?”

speed, consistency, not re-inventing the wheel, community, performance, testing….

Old school web apps

HTML + templates



Data Binding

Routing

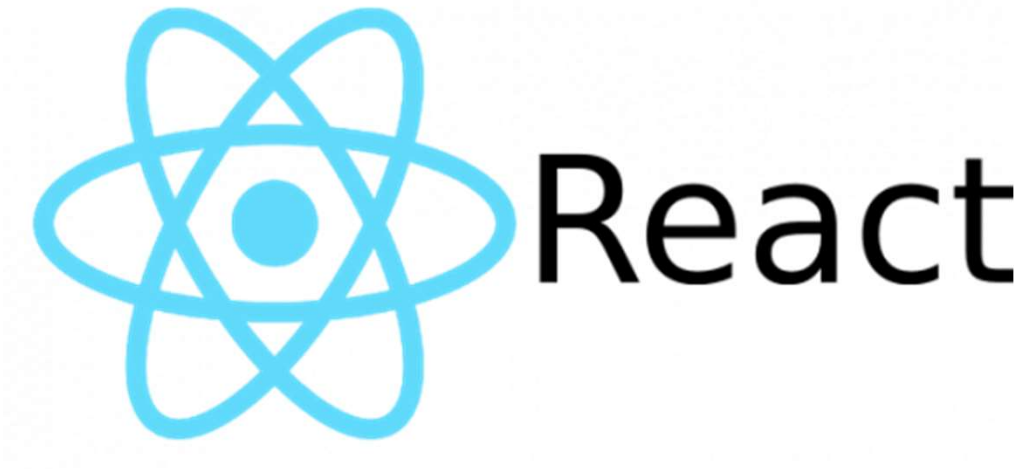
DOM-manipulation

Mobile development

...

“The Frankenstein Framework”

Front-end Frameworks – the big four



Similarities

- Creating Single Page Applications
- Based on components
- Data binding, props, events, routing, state management, …
- Huge ecosystem
- Huge community
- High adaptation rate

Differences (apart from syntax)

- Point of departure: HTML template, enhanced with framework specific tags and attributes
- One-stop-shop / solution

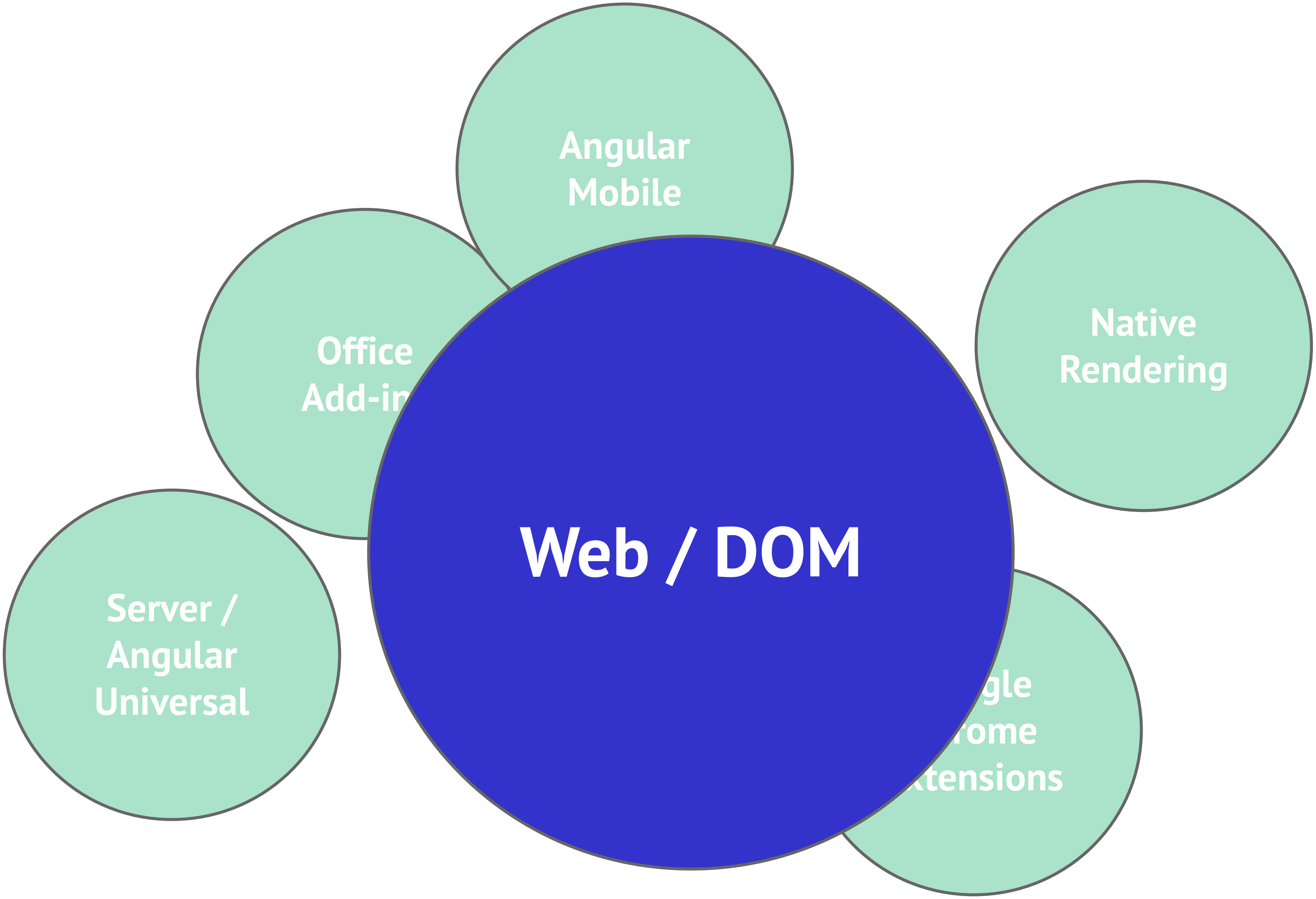
React

- Point of departure: JavaScript, JavaScript, JavaScript (JSX)
- Build-all-yourself / choice anxiety

Platform

Framework to Platform

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



Angular Mobile

Office Add-in

Native Rendering

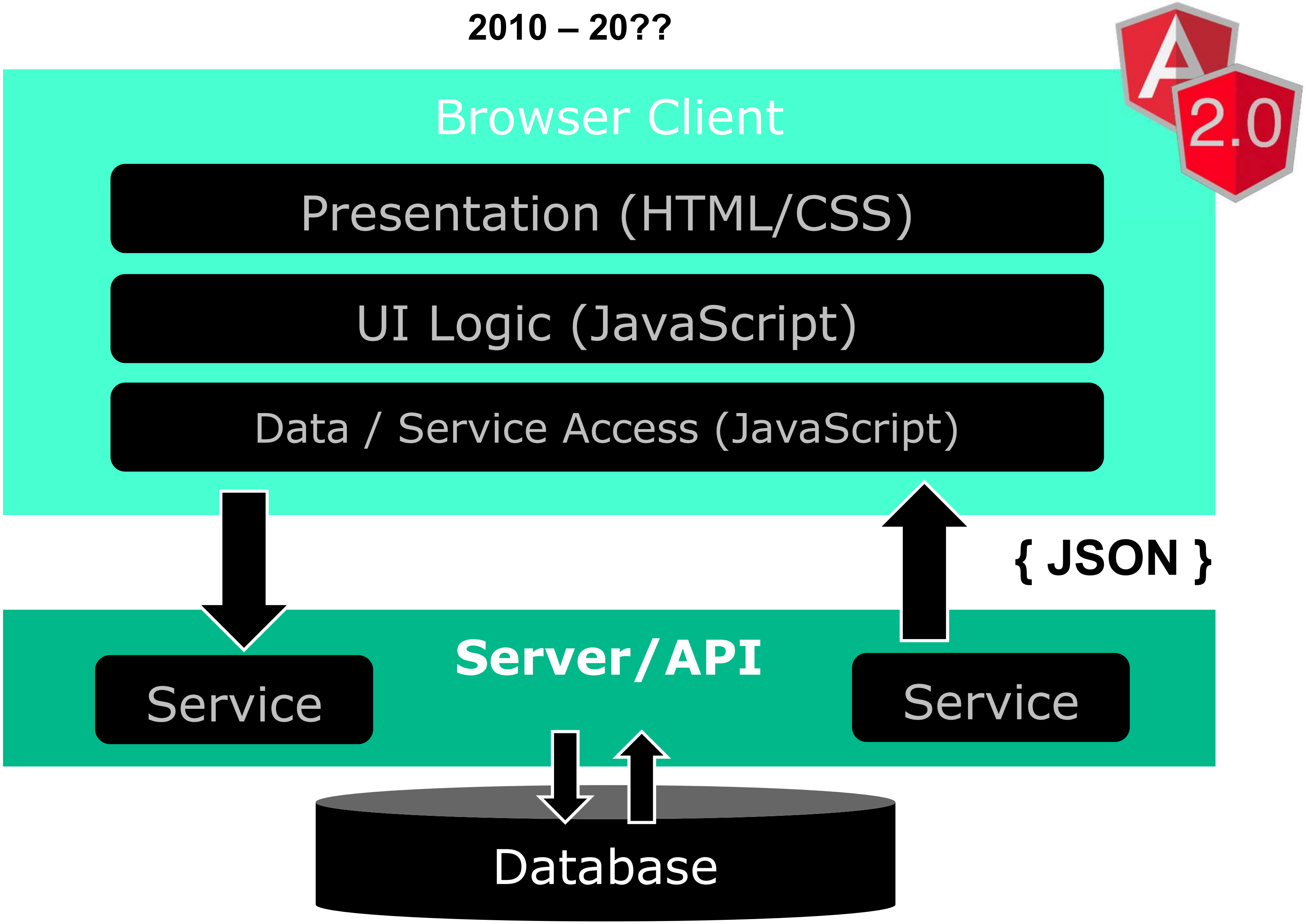
Web / DOM

Server / Angular Universal

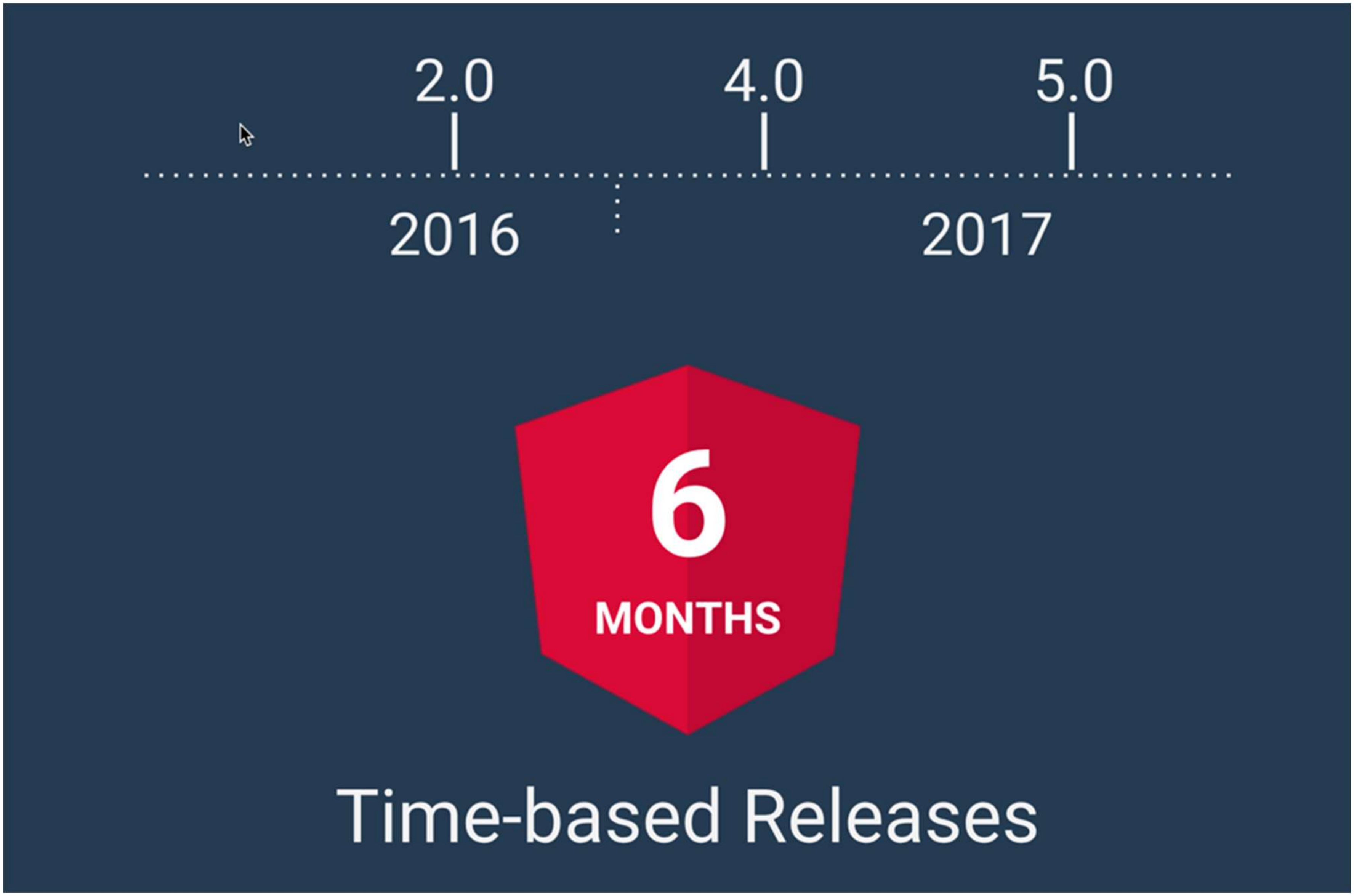
gle rome xtensions

Single Page Application

2010 – 20??



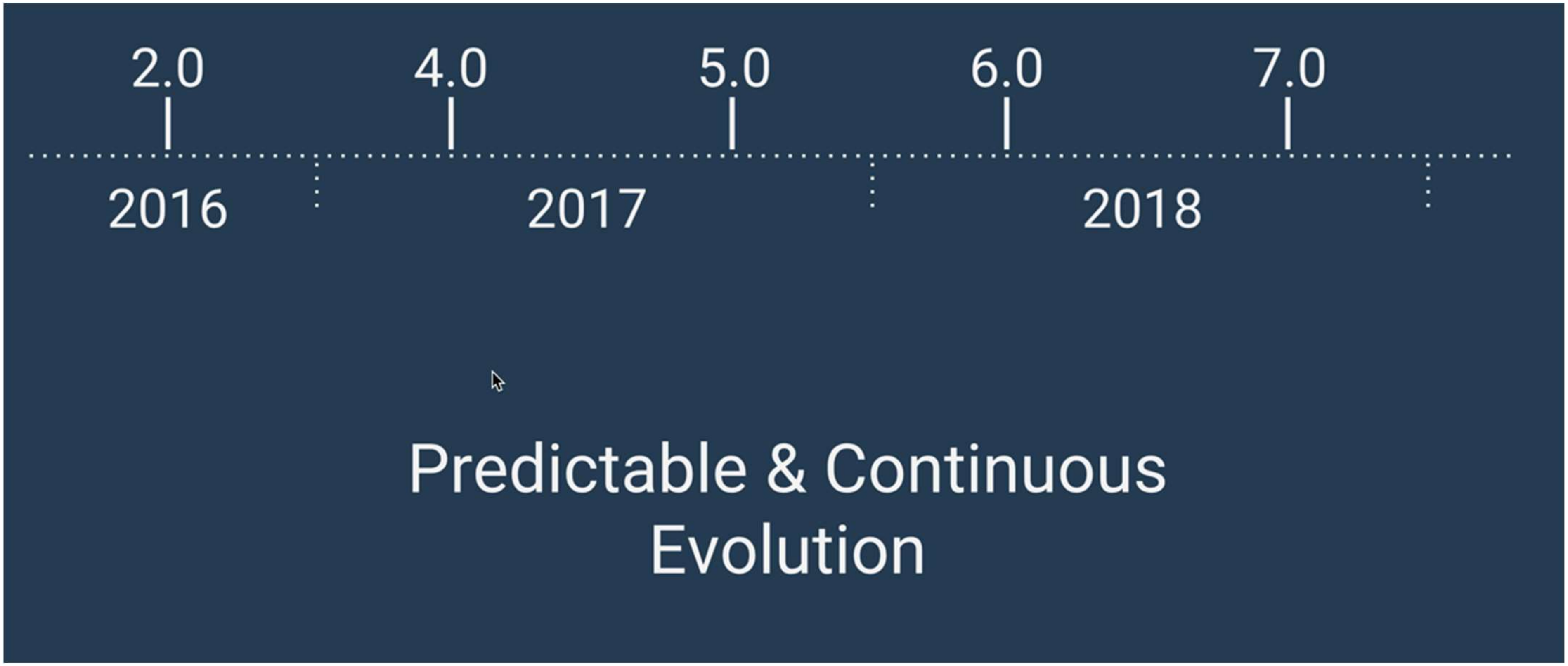
On Versioning Numbering



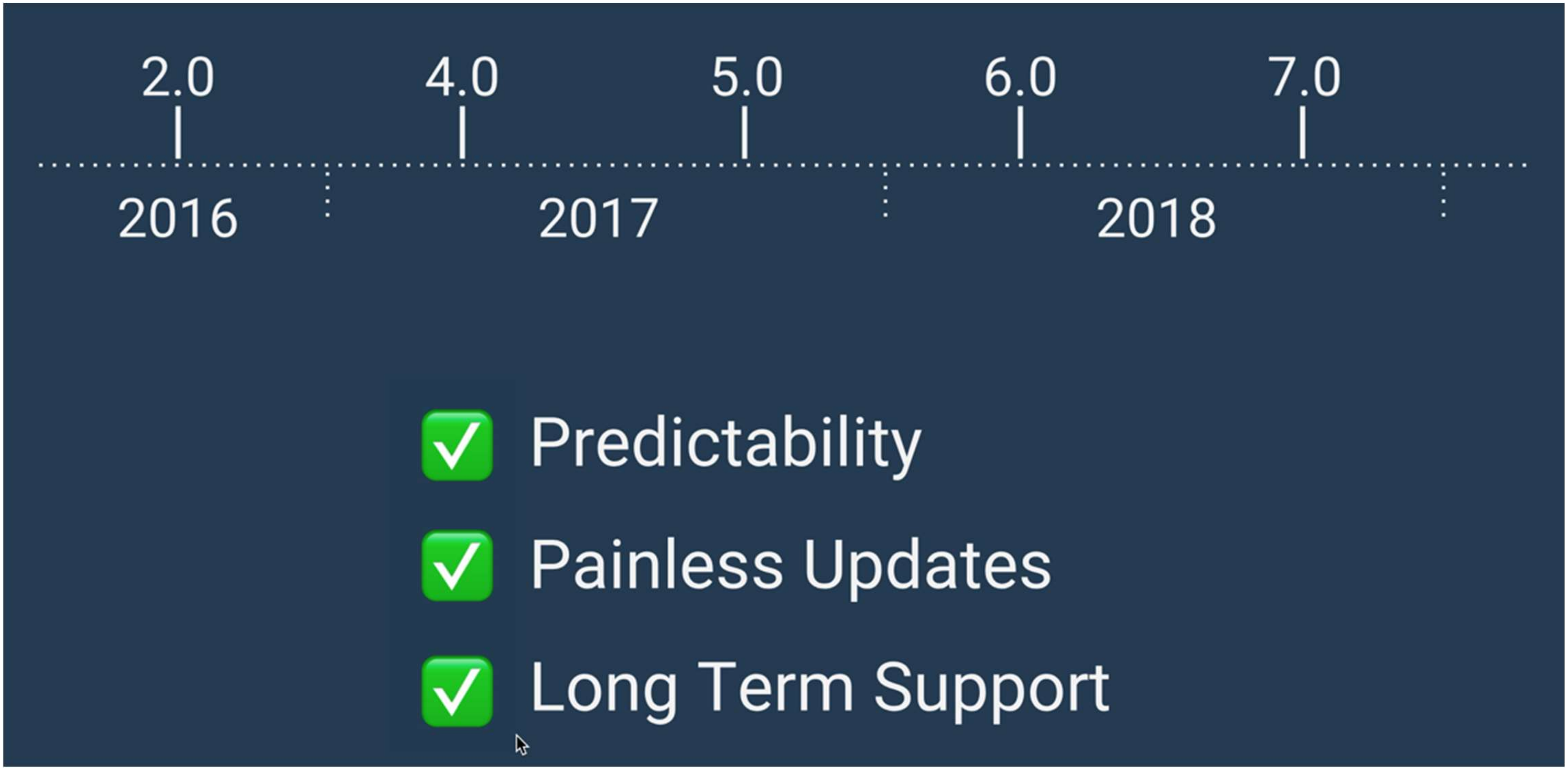
Time-based Releases

@IgorMinar, Nov. 9, 2018

Planned – six months major release cycle



Predictable & Continuous Evolution



2.0

4.0

5.0

6.0

7.0

2016

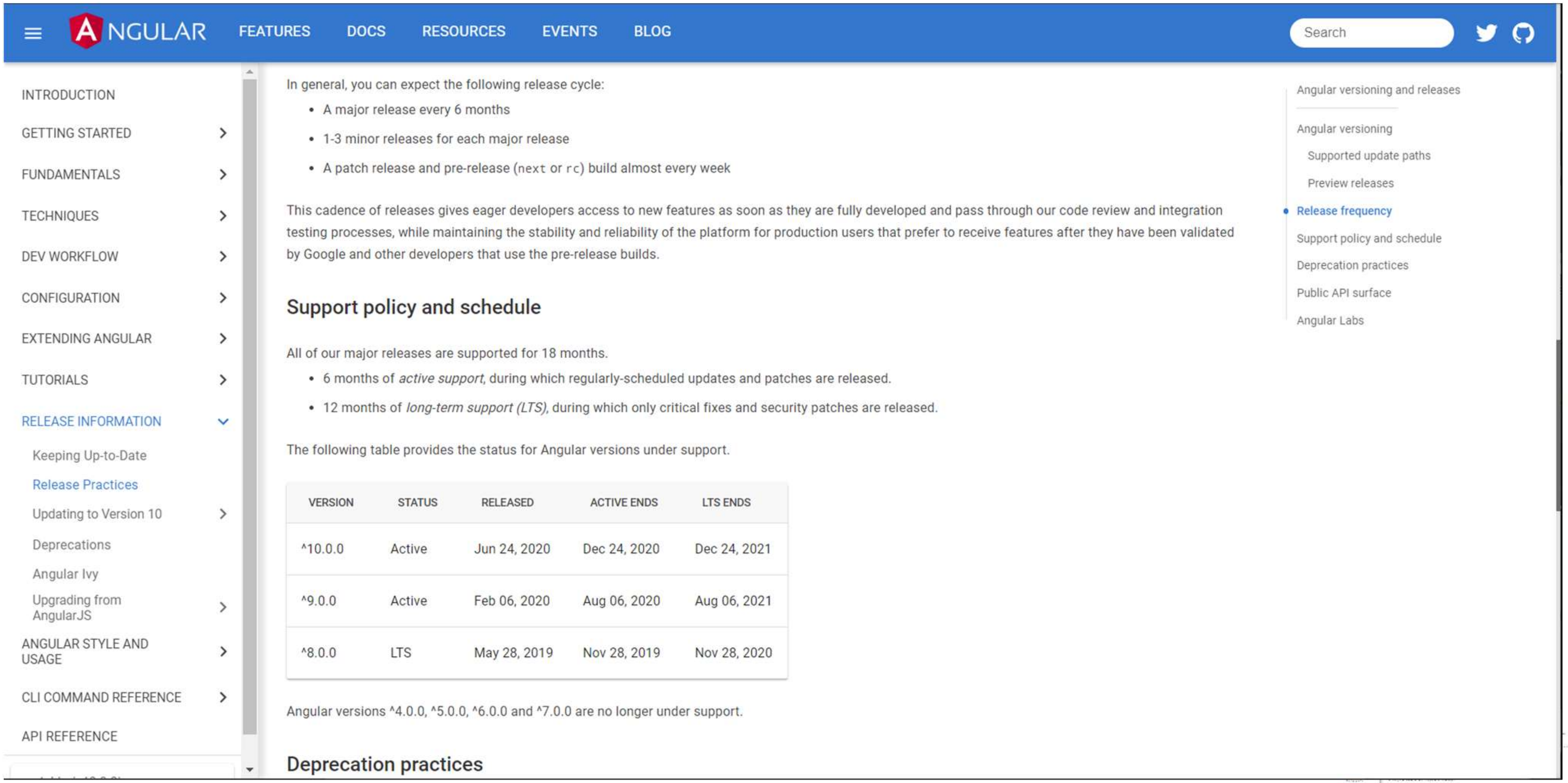
2017

2018

- ✅ Predictability
- ✅ Painless Updates
- ✅ Long Term Support

Angular Versions and -Long Time Support

→ https://angular.io/guide/releases



In general, you can expect the following release cycle:

- A major release every 6 months
- 1-3 minor releases for each major release
- A patch release and pre-release (`next` or `rc`) build almost every week

This cadence of releases gives eager developers access to new features as soon as they are fully developed and pass through our code review and integration testing processes, while maintaining the stability and reliability of the platform for production users that prefer to receive features after they have been validated by Google and other developers that use the pre-release builds.

Support policy and schedule

All of our major releases are supported for 18 months.

- 6 months of *active support*, during which regularly-scheduled updates and patches are released.
- 12 months of *long-term support (LTS)*, during which only critical fixes and security patches are released.

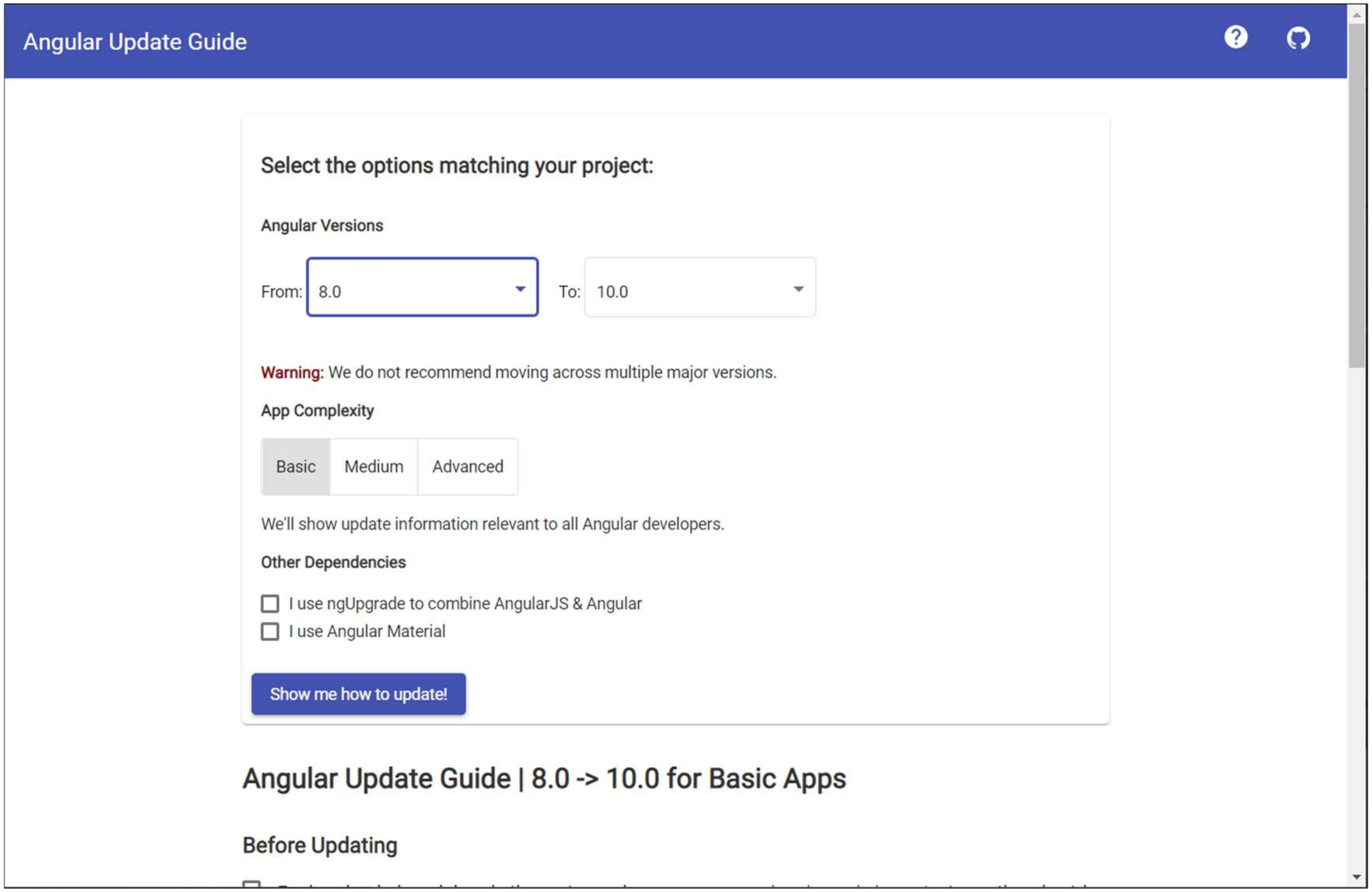
The following table provides the status for Angular versions under support.

VERSION	STATUS	RELEASED	ACTIVE ENDS	LTS ENDS
^10.0.0	Active	Jun 24, 2020	Dec 24, 2020	Dec 24, 2021
^9.0.0	Active	Feb 06, 2020	Aug 06, 2020	Aug 06, 2021
^8.0.0	LTS	May 28, 2019	Nov 28, 2019	Nov 28, 2020

Angular versions ^4.0.0, ^5.0.0, ^6.0.0 and ^7.0.0 are no longer under support.

Deprecation practices

https://update.angular.io/



Select the options matching your project:

Angular Versions

From: 8.0 To: 10.0

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

App Complexity

[✓ Basic] Medium Advanced

We'll show update information relevant to all Angular developers.

Other Dependencies

□ I use ngUpgrade to combine AngularJS & Angular
□ I use Angular Material

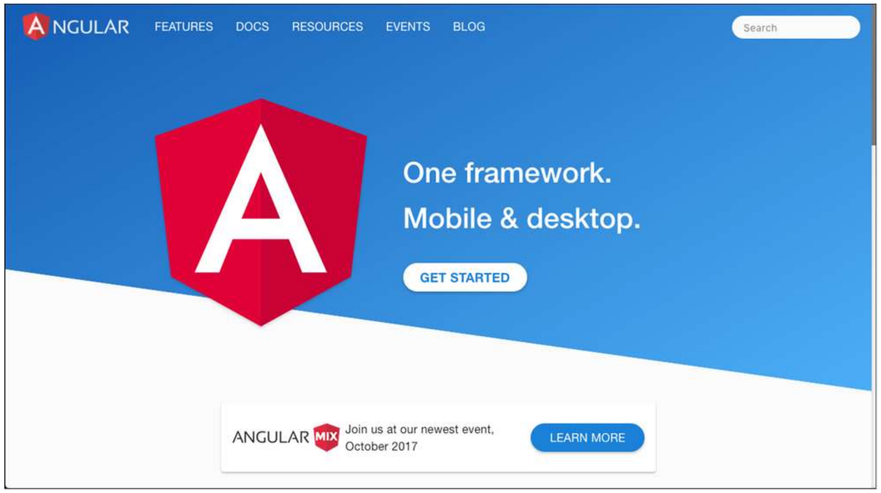
Angular Update Guide | 8.0 -> 10.0 for Basic Apps

Before Updating

"It's just

Angular

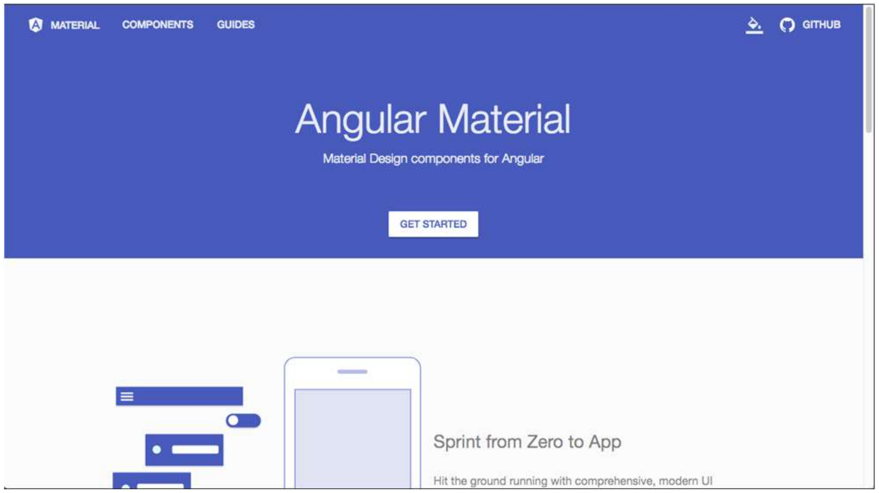
Angular as a Platform



One framework.
Mobile & desktop.

Join us at our newest event, October 2017

https://angular.io/



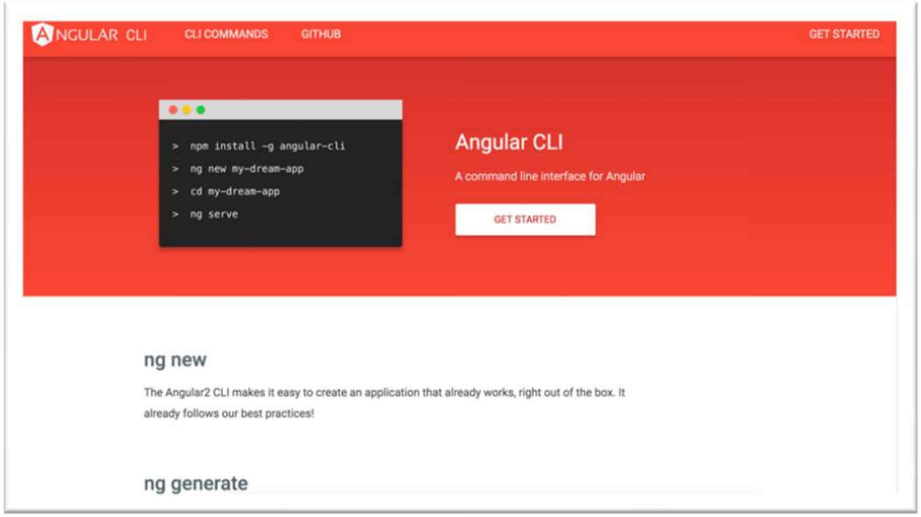
Angular Material

Material Design components for Angular

Sprint from Zero to App

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
```

Angular CLI

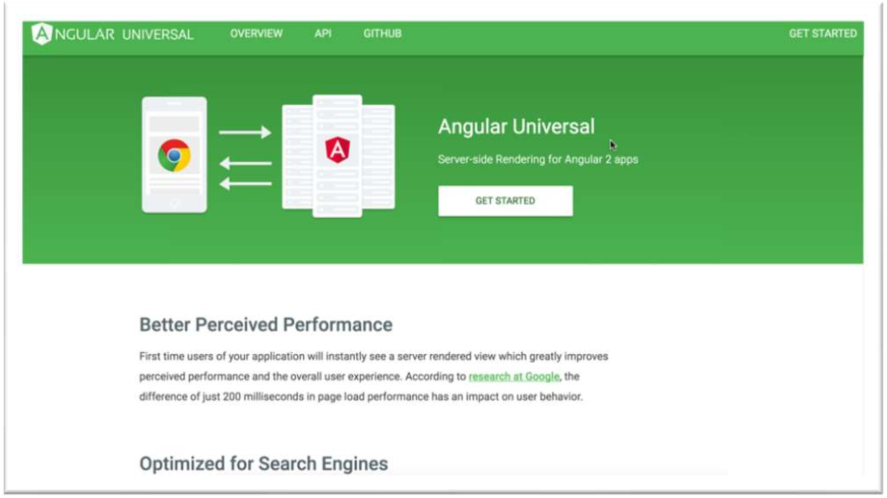
A command line interface for Angular

ng new

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

ng generate

https://cli.angular.io/



Angular Universal

Server-side Rendering for Angular 2 apps

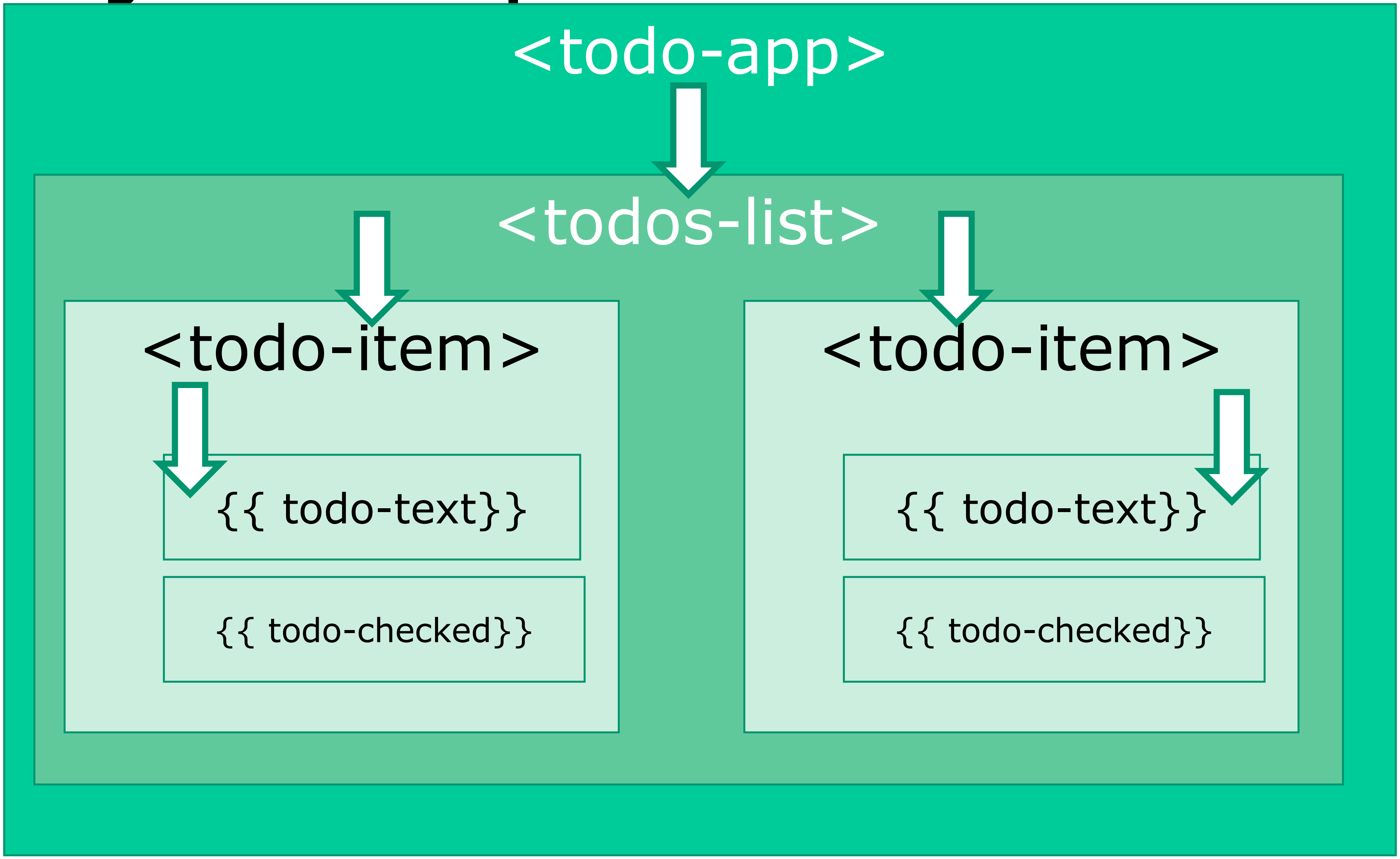
Better Perceived Performance

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.

Optimized for Search Engines

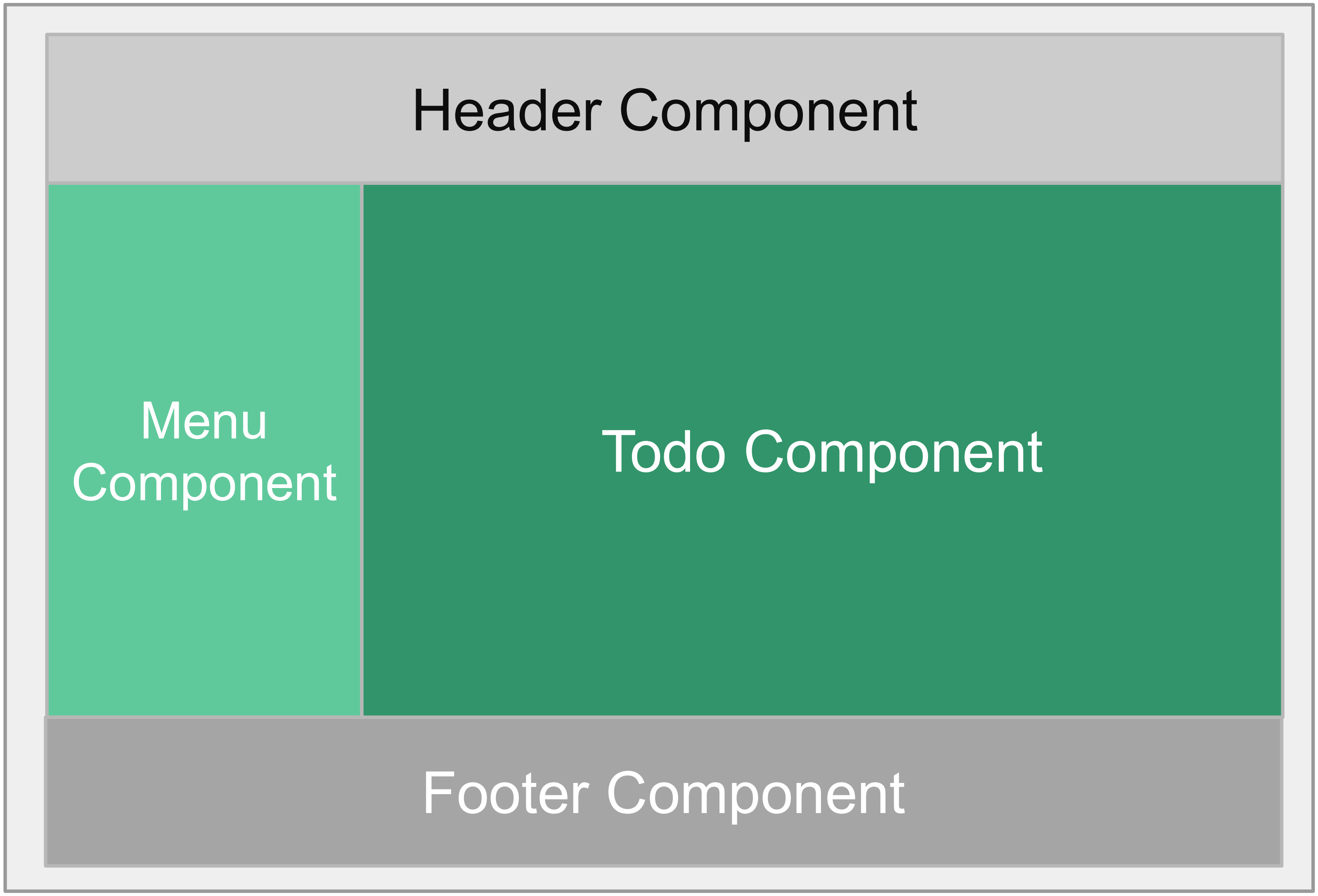
https://universal.angular.io/

Angular 2 - components



"An Angular-app is a tree of components”

Components – visually

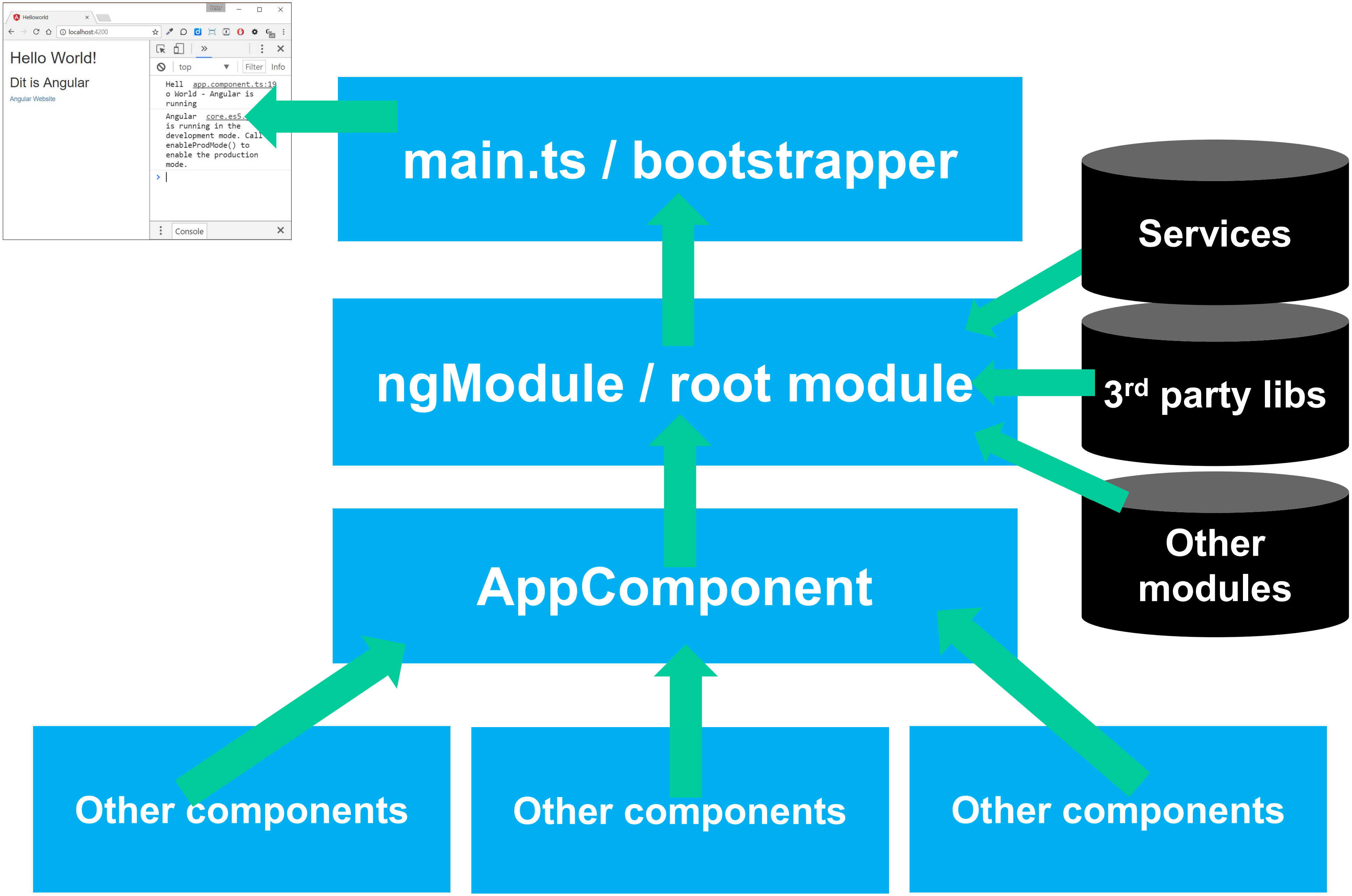


Header Component

Menu Component

Todo Component

Footer Component



Helloworld

localhost:4200

Hello World!

Dit is Angular

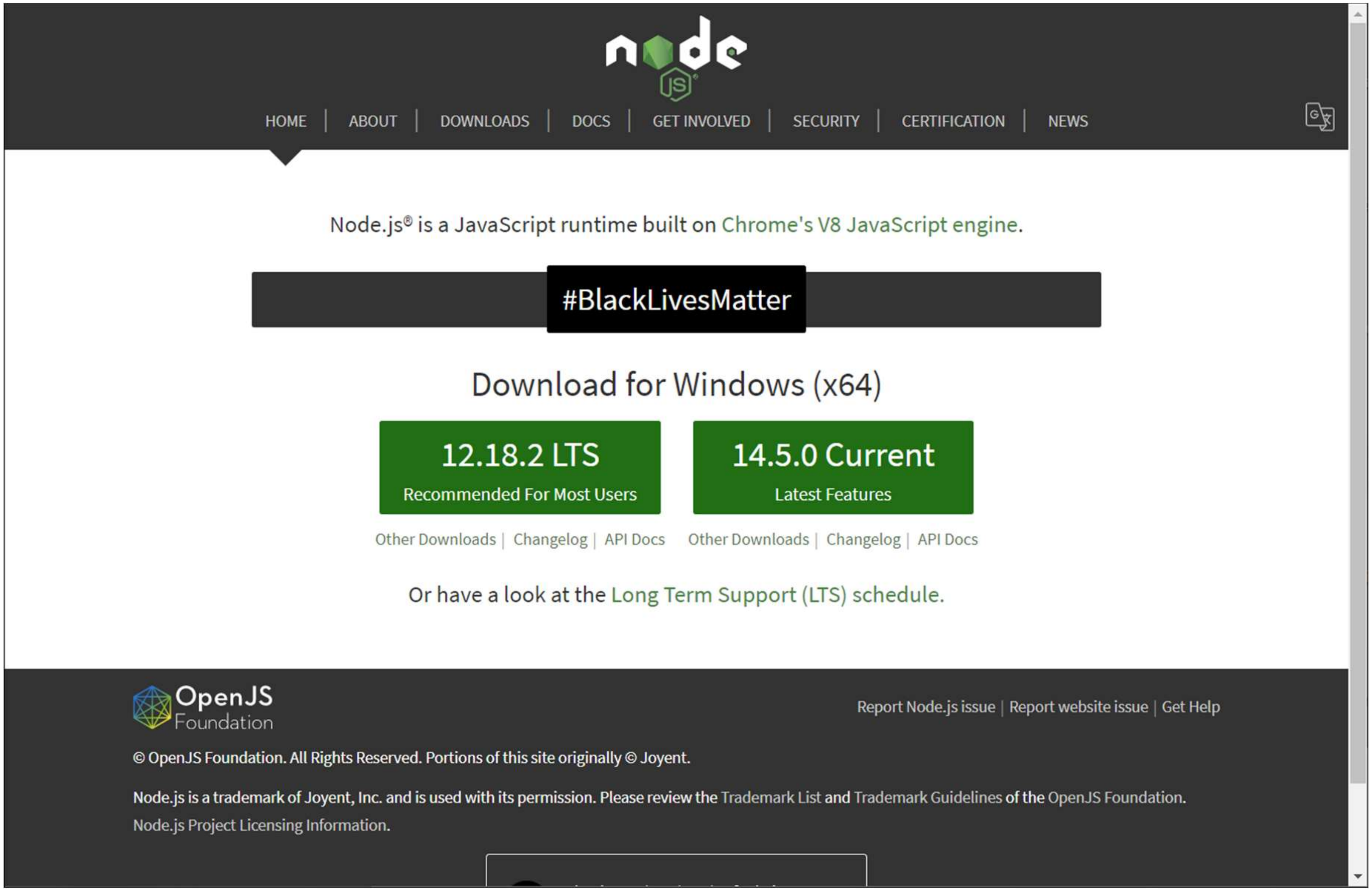
Angular Website

```
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 look at some code

‘Hello World’ in Angular

Angular development dependency: NodeJS 12+


HOME
ABOUT
DOWNLOADS
DOCS
GET INVOLVED
SECURITY
CERTIFICATION
NEWS
#BlackLivesMatter
12.18.2 LTS
Recommended For Most Users
14.5.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs
OpenJS Foundation
Report Node.js issue | Report website issue | Get Help

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

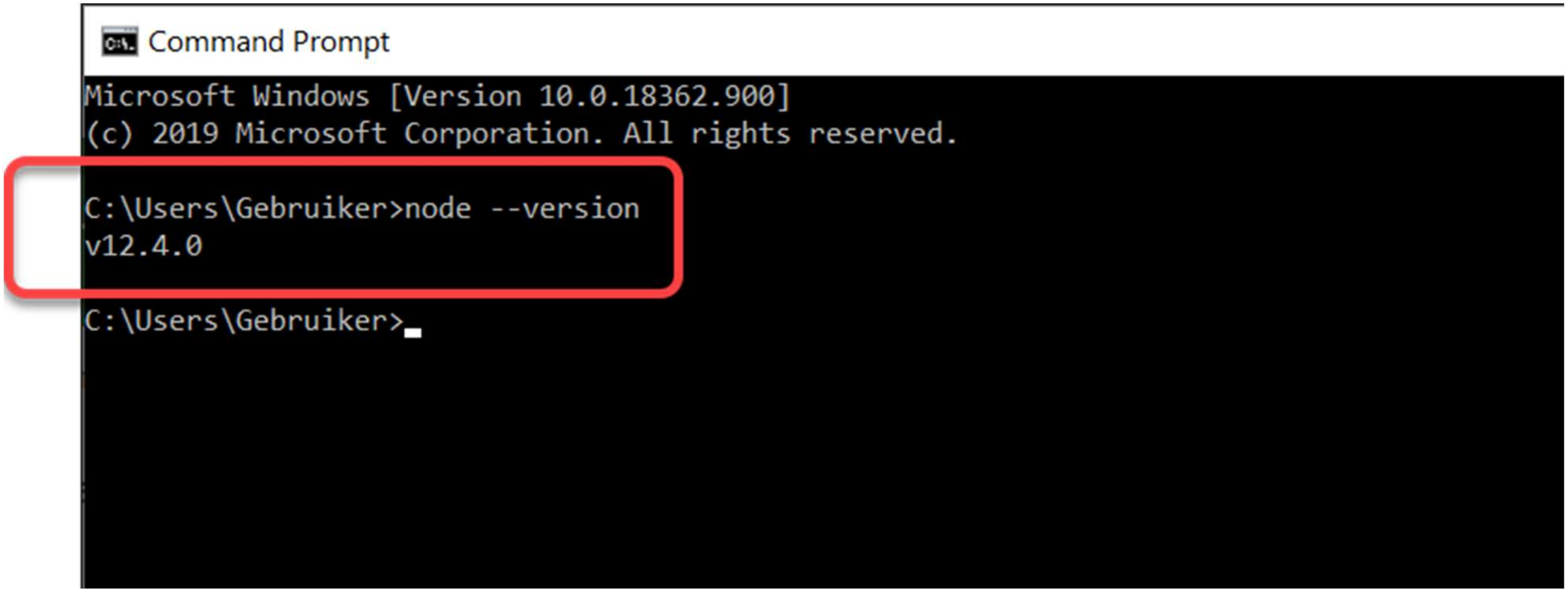
Download for Windows (x64)

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

© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark List and Trademark Guidelines of the OpenJS Foundation.
Node.js Project Licensing Information.

Node – check your version



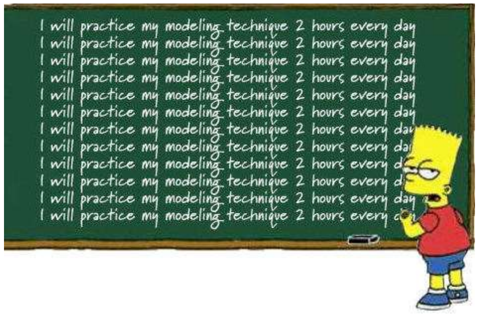
```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Gebruiker>node --version
v12.4.0

C:\Users\Gebruiker>
```

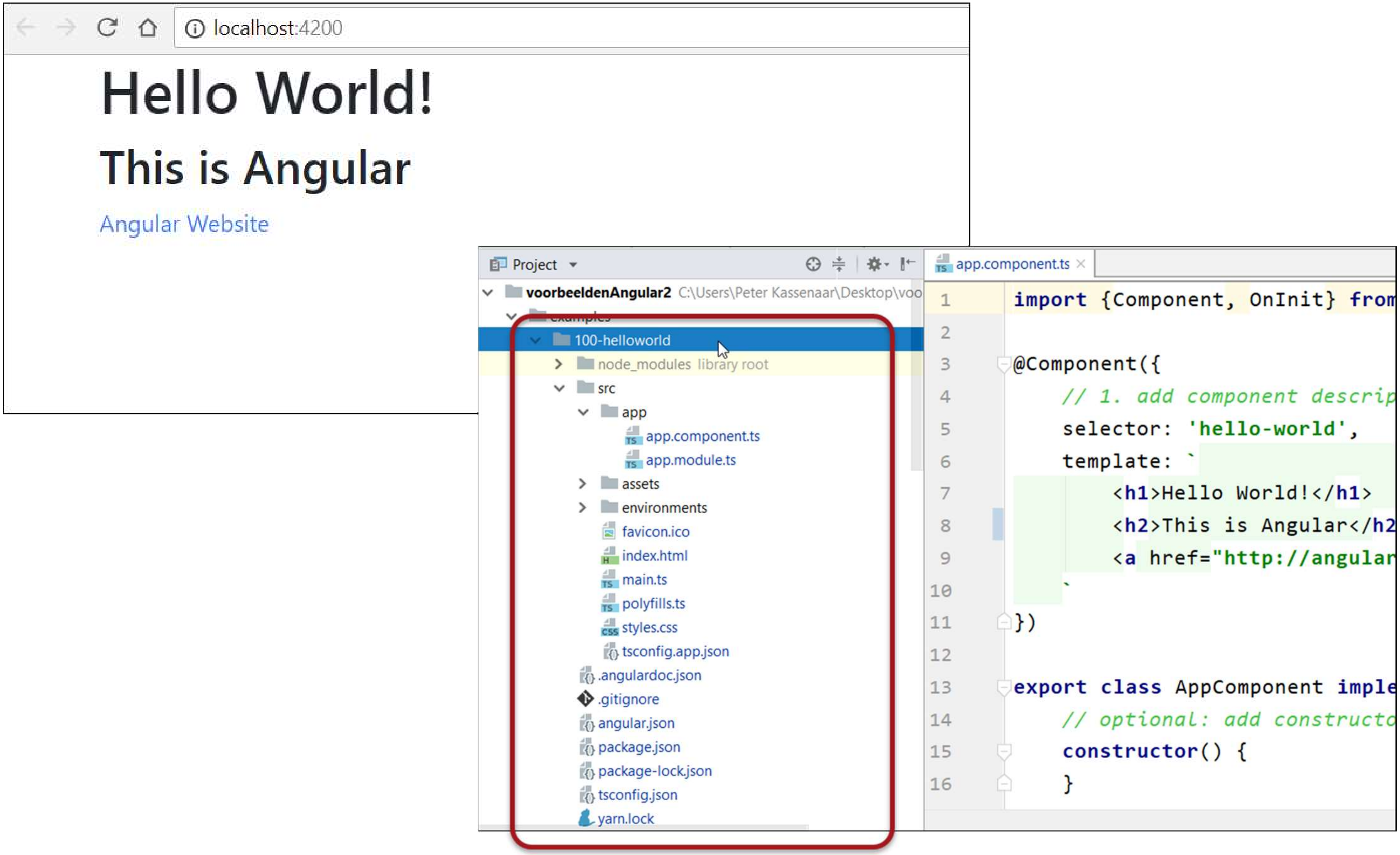
Mini-workshop

- 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


I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day



localhost:4200

Hello World!

This is Angular

Angular Website

```
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
```

```
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
   - Important configuration files
2. Angular Component(s) - `@Component`
3. Angular Module(s): `@ngModule`
4. Bootstrap our 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",
       …
}
```

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 {


}
```

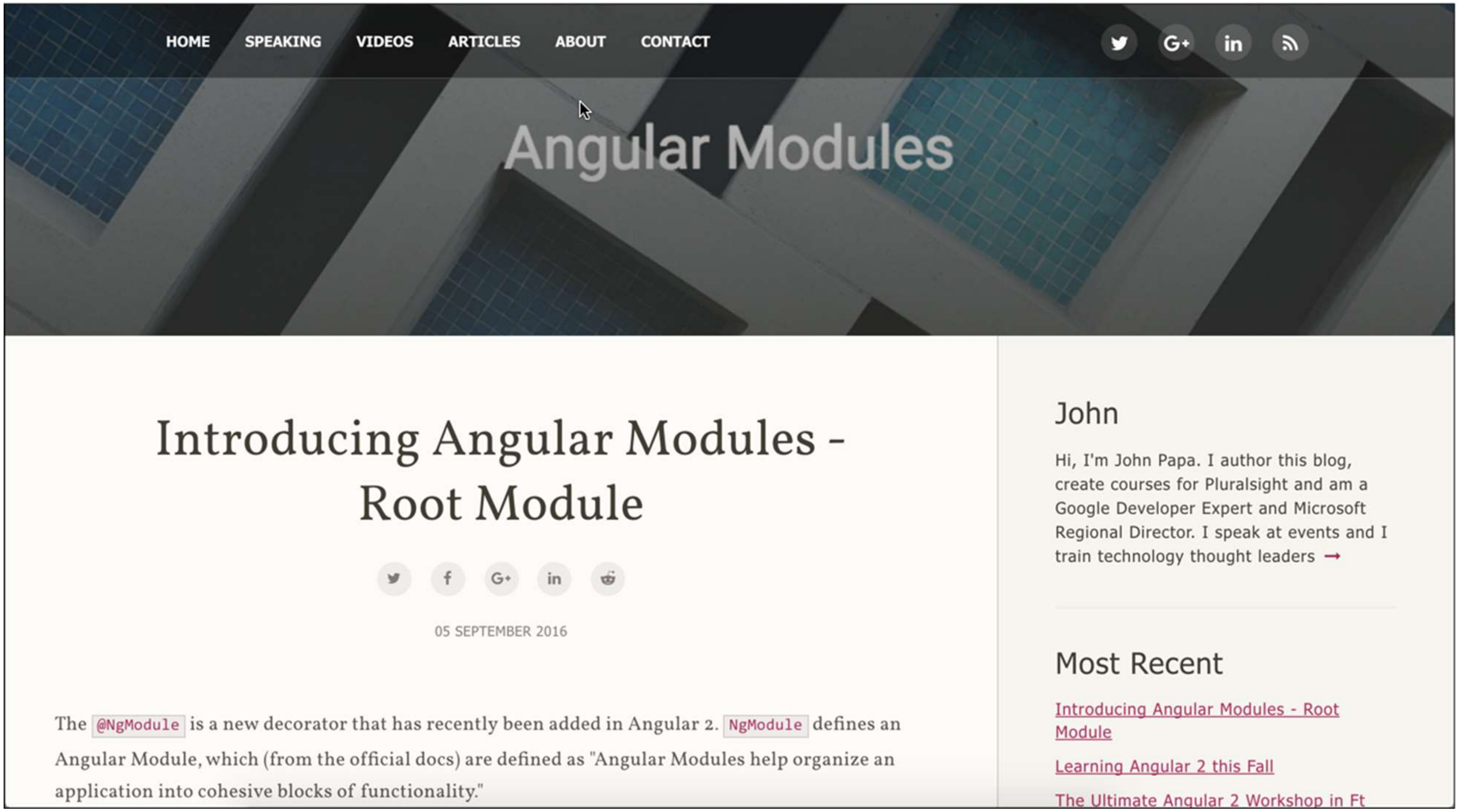
Step 3 – @ngModule

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>


<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

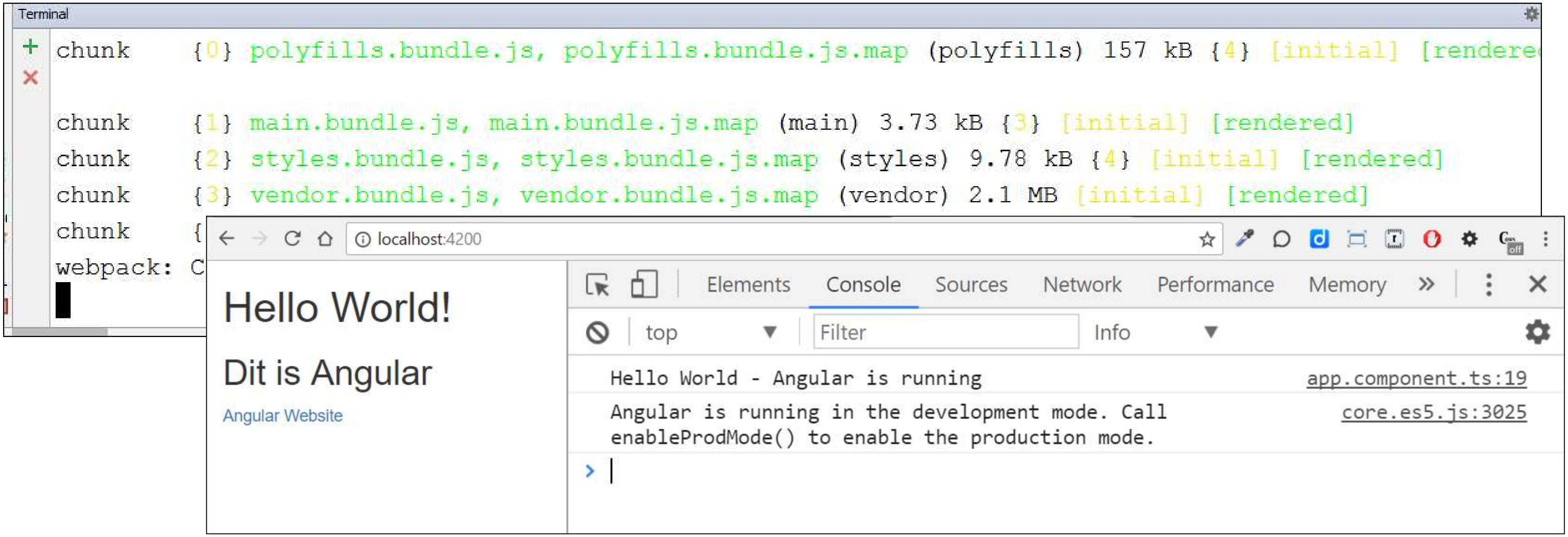
Element reference (`selector`) of root-component:

```
<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
```

Hello World!

Dit is Angular

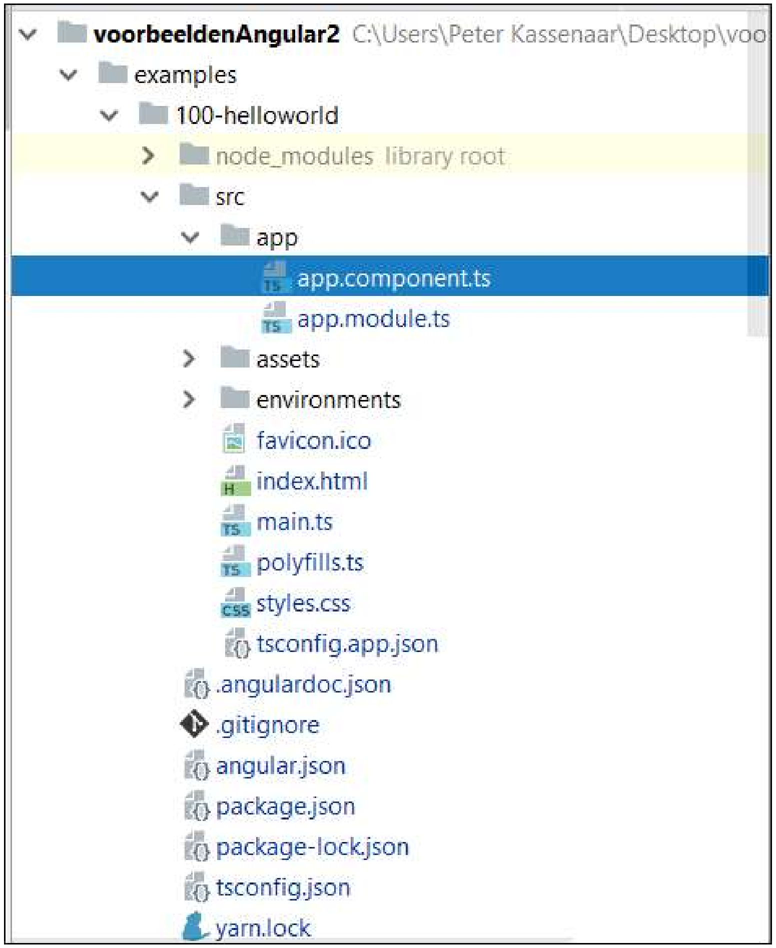
Angular Website

```
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.
>
```

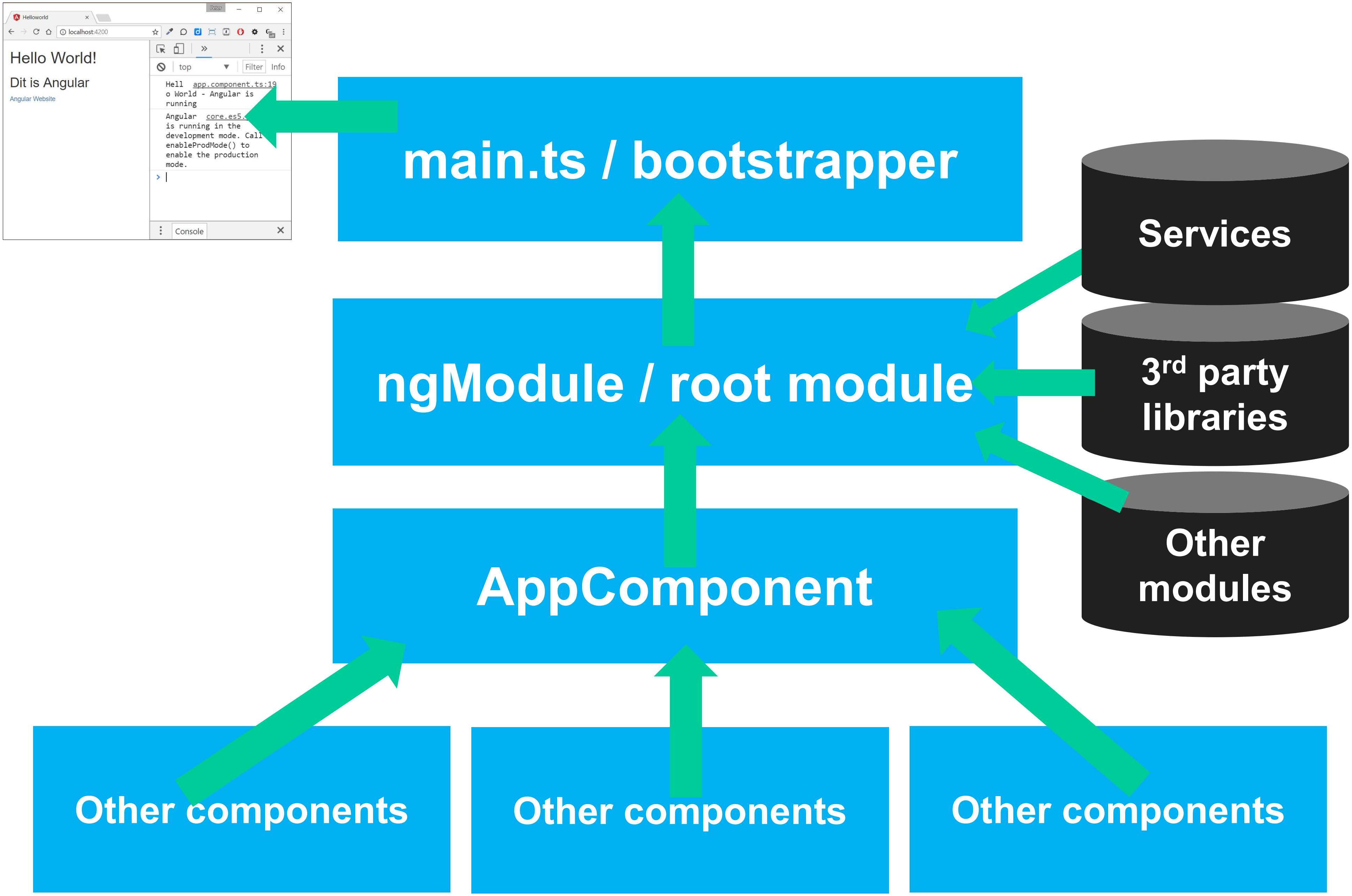
After that: edit `app.component.ts`

– Automagically refreshed through Live Reload

So, a Basic Project Structure and Architecture



```
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

```
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.
```

Checkpoint

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

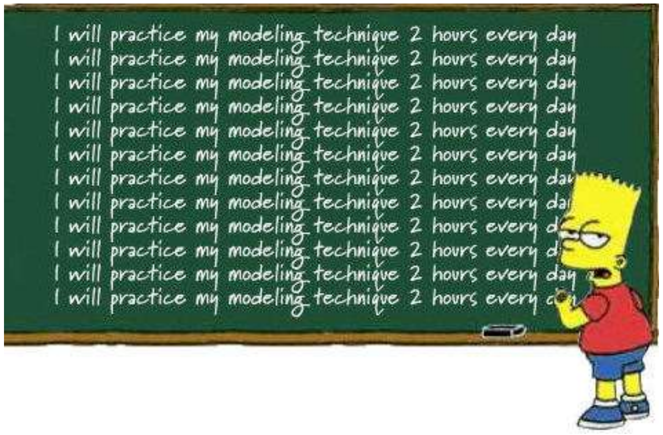
(At least) Five steps:

1. Set up environment, boilerplate & libraries
2. Write/edit 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.

Workshop `1a), 1b).` Optional: `1c), 1d).`

Workshop….


I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every da
I will practice my modeling technique 2 hours every da
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every d

Assets

github.com/PeterKassenaar/lendex

Workshops and links to example code

Tooling - Angular CLI & TypeScript

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

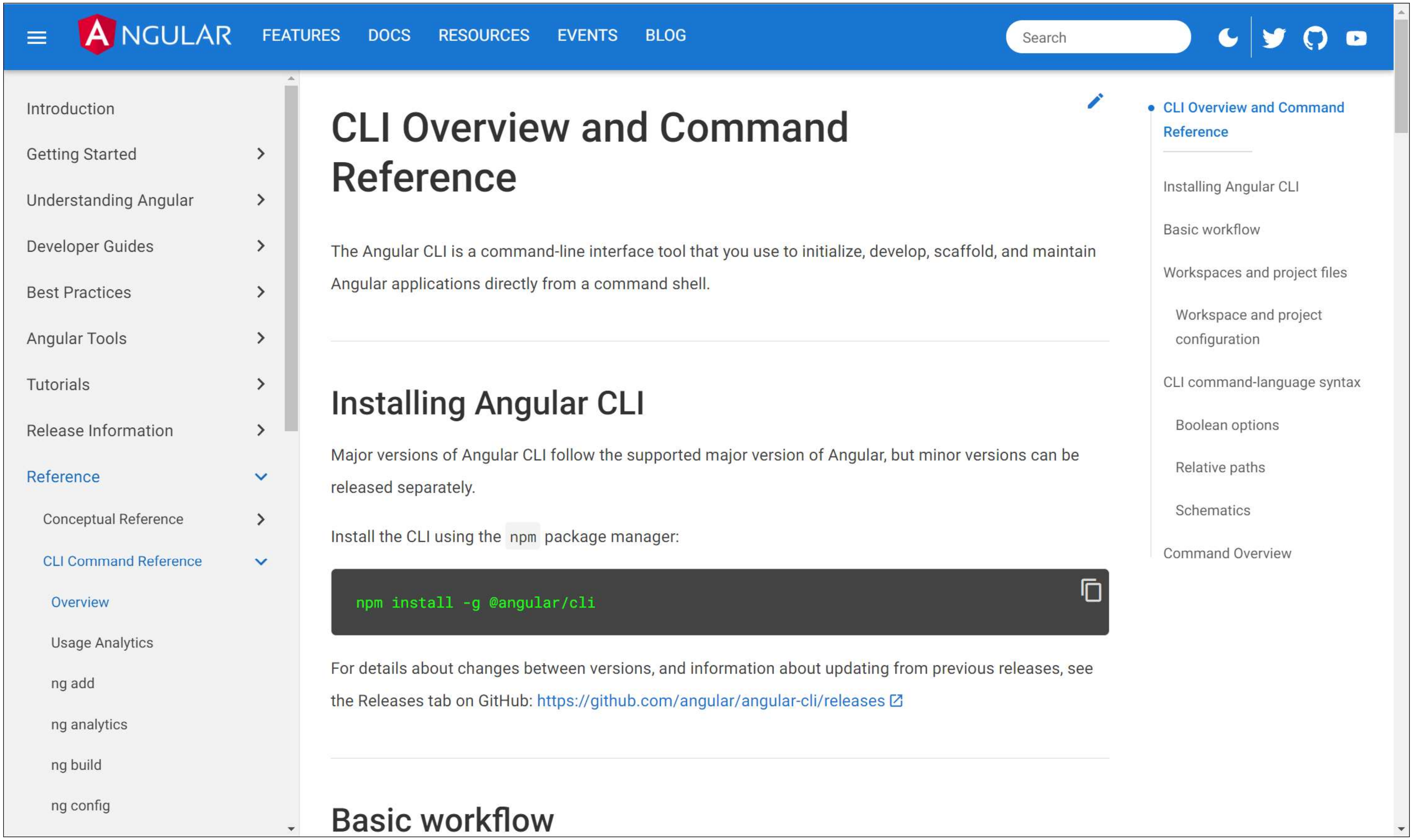
https://github.com/angular/angular-cli

or

https://cli.angular.io/

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

cli.angular.io



CLI Overview and Command Reference

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

Installing Angular CLI

Major versions of Angular CLI follow the supported major version of Angular, but minor versions can be released separately.

Install the CLI using the `npm` package manager:

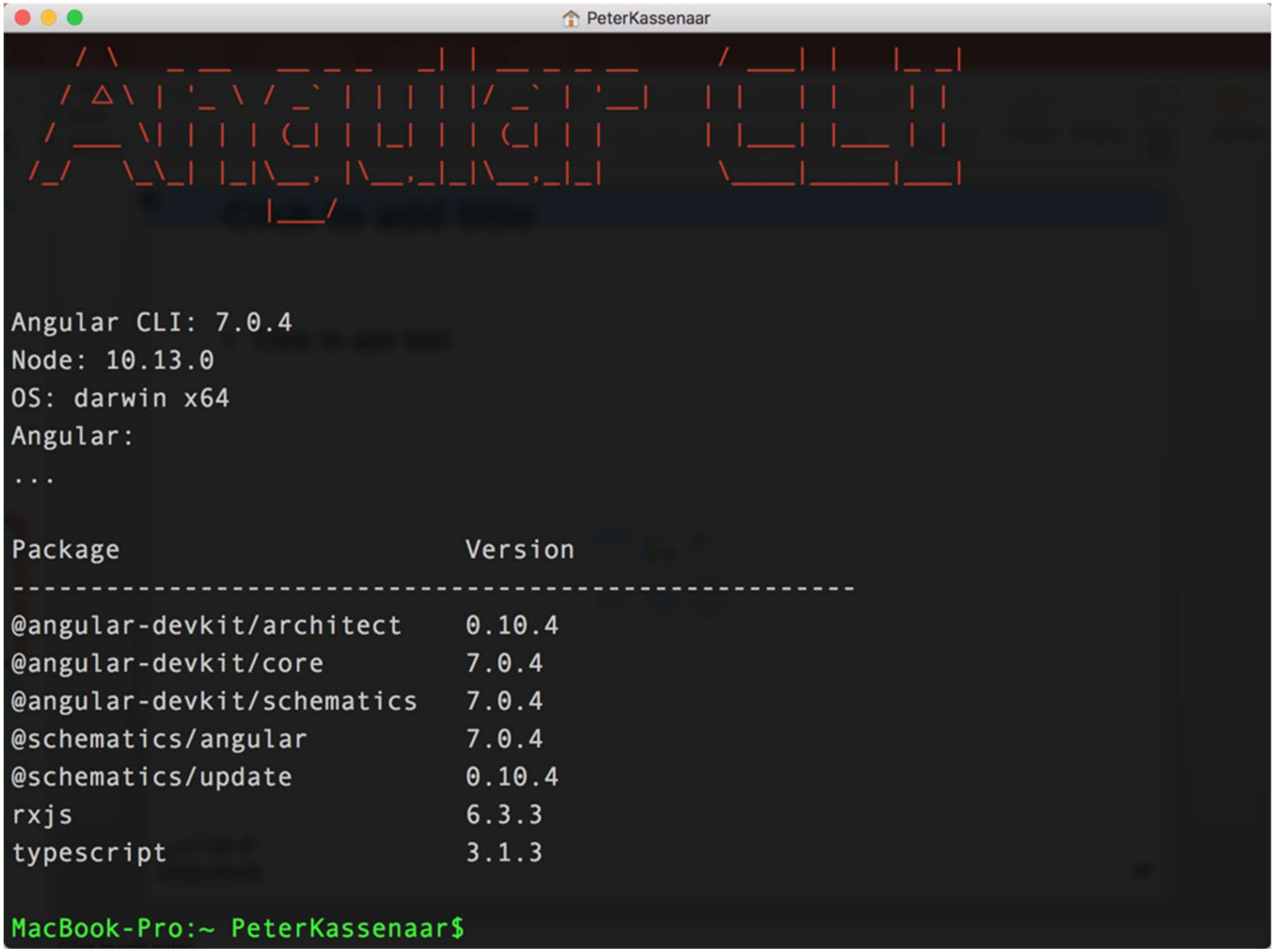
```
npm install -g @angular/cli
```

For details about changes between versions, and information about updating from previous releases, see the Releases tab on GitHub: https://github.com/angular/angular-cli/releases

Basic workflow

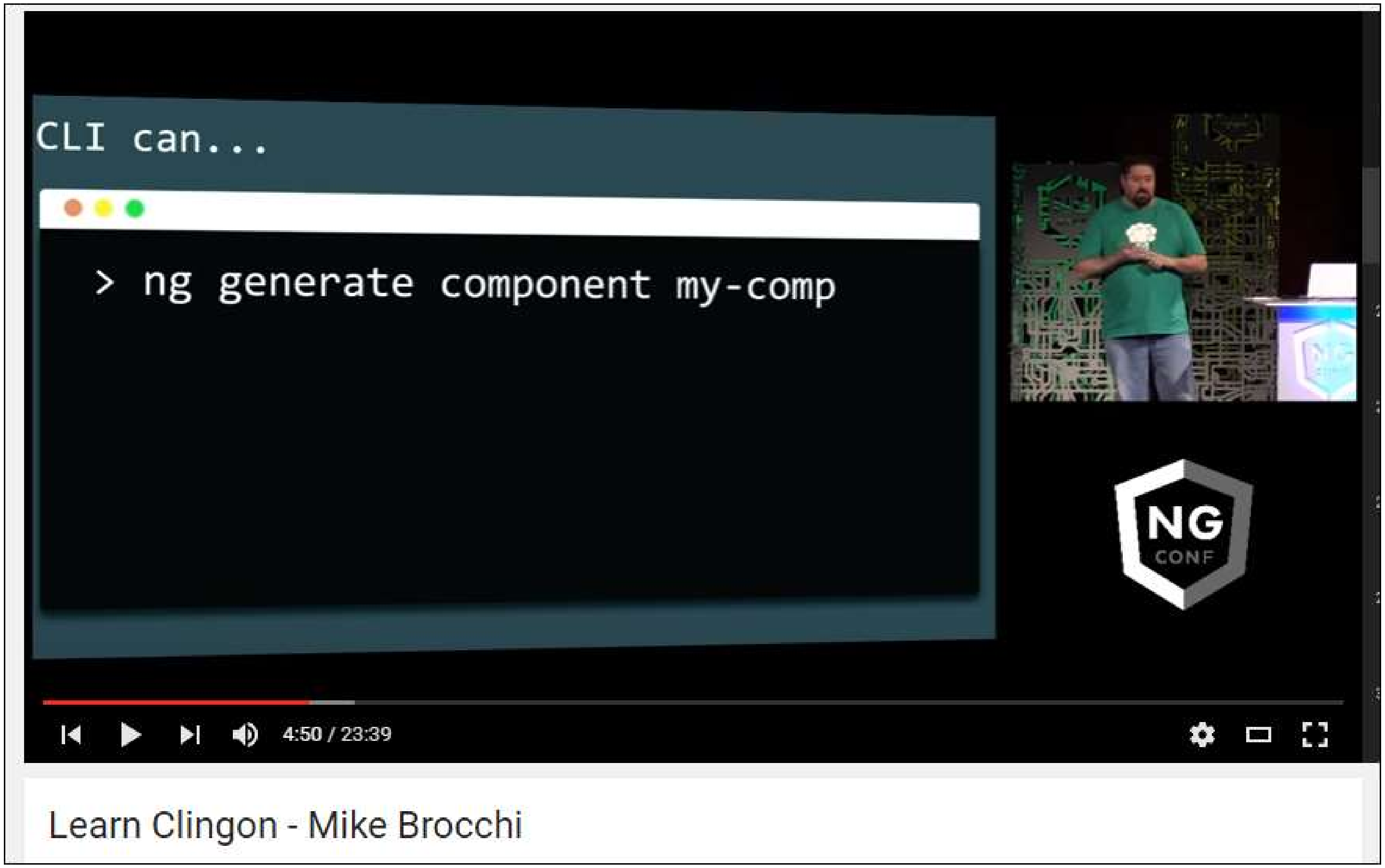
```
npm install -g @angular/cli
```

ng --version



```
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / △ \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI: 7.0.4
Node: 10.13.0
OS: darwin x64
Angular:
...

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.10.4
@angular-devkit/core         7.0.4
@angular-devkit/schematics   7.0.4
@schematics/angular          7.0.4
@schematics/update           0.10.4
rxjs                         6.3.3
typescript                   3.1.3

MacBook-Pro:~ PeterKassenaar$
```

Background info



CLI can...

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

Learn Clingon - Mike Brocchi

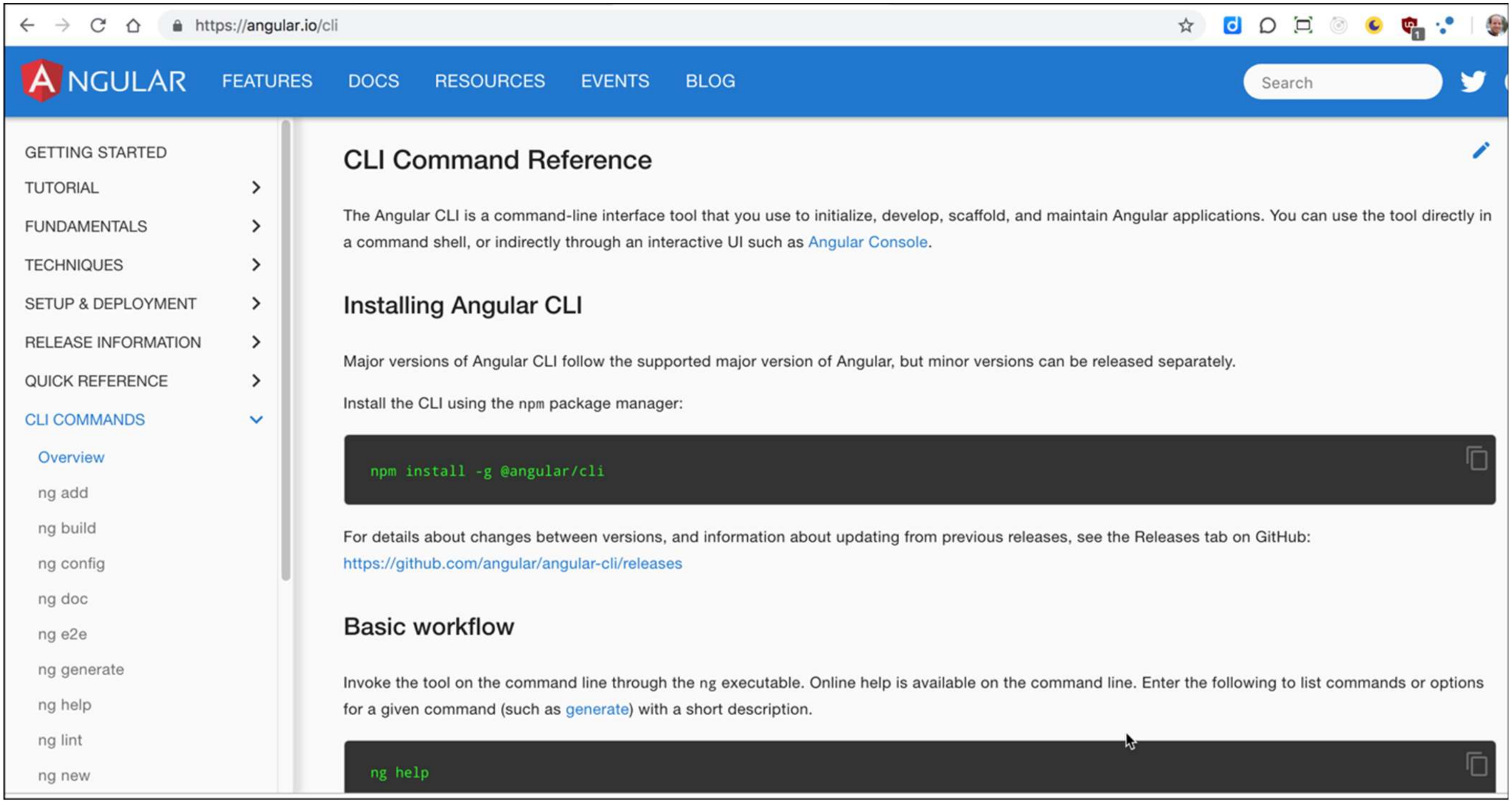
https://www.youtube.com/watch?v=wHZe6gGI5RY

Main commands

```
ng new PROJECT_NAME
cd PROJECT_NAME
ng serve
```

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

Documentation - in the Angular Docs



CLI Command Reference

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications. You can use the tool directly in a command shell, or indirectly through an interactive UI such as Angular Console.

Installing Angular CLI

Major versions of Angular CLI follow the supported major version of Angular, but minor versions can be released separately.

Install the CLI using the `npm` package manager:

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

For details about changes between versions, and information about updating from previous releases, see the Releases tab on GitHub: https://github.com/angular/angular-cli/releases

Basic workflow

Invoke the tool on the command line through the `ng` executable. Online help is available on the command line. Enter the following to list commands or options for a given command (such as generate) with a short description.

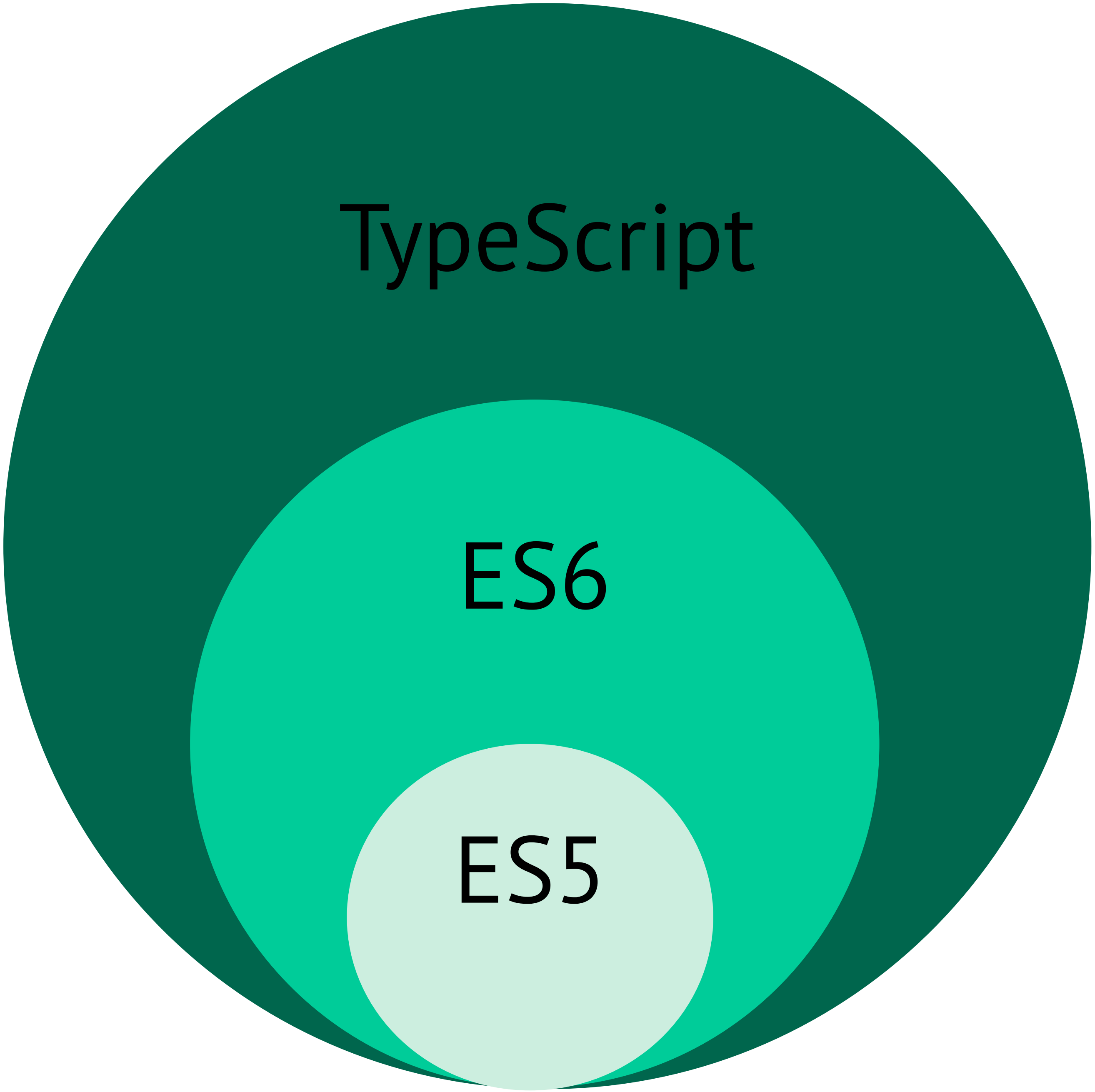
```
ng help
```

https://angular.io/cli

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

.

Architecture of your Component Class

imports

```
import { Component } from ’@angular/core';
import { DataService } from './services/data-service';
```

annotations

```
@Component({
  selector: 'orders',
  directives: [DataService],
  templateUrl: 'orders-component.html',
})
```

class

```
export class OrdersComponent {
          …
}
```

Checkpoint

- Angular 2+ 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