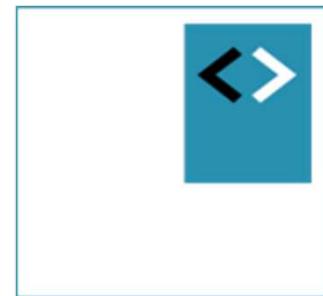




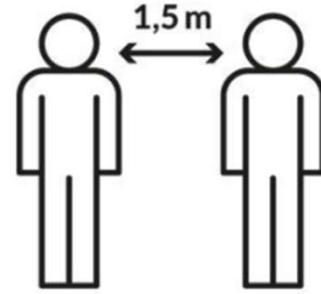
Angular Fundamentals

Module 1 - Inleiding



Peter Kassenaar
info@kassenaar.com

Hoe ziet een opleiding er uit?



Symptomen? Blijf thuis!

Heb je symptomen die mogelijk duiden op CoVID-19, blijf zeker thuis. Indien je op de opleiding verschijnt met symptomen, word je naar huis gestuurd.

Mondmaskers verplicht

Het gebruik van mondmaskers is **verplicht**. Enkel waar voldoende afstand gegarandeerd kan worden, is het dragen van een mondmasker optioneel. Neem deze dus zeker mee naar de opleiding.

Handhygiëne

Voor de les begint en voor de lunchpauze ben je verplicht om je handen te wassen. Er is voldoende water en zeep en/of handgel beschikbaar.

Afstand

Elke opleiding vindt plaats in een locatie die ruim genoeg is om voldoende fysieke afstand te kunnen houden. Hou altijd minstens **1,5 meter afstand**.



Blijf van je gezicht

Raak je gezicht niet aan en vermijd elk fysiek contact.

Volg de instructies

De lesgever informeert je over de **te volgen richtlijnen**. Luister altijd naar zijn/haar instructies. Deze dienen om jouw veiligheid en die van de lesgever te garanderen.

Materiaal

Al het materiaal dat je gebruikt is op voorhand **gereinigd** en wordt ook na elke les opnieuw gereinigd. Tijdens praktijklessen moet je steeds met (eigen) handschoenen werken.

Reiniging lokalen

Na elke les worden lessafels, stoelen, klinken en eventueel lesmateriaal opnieuw gereinigd.

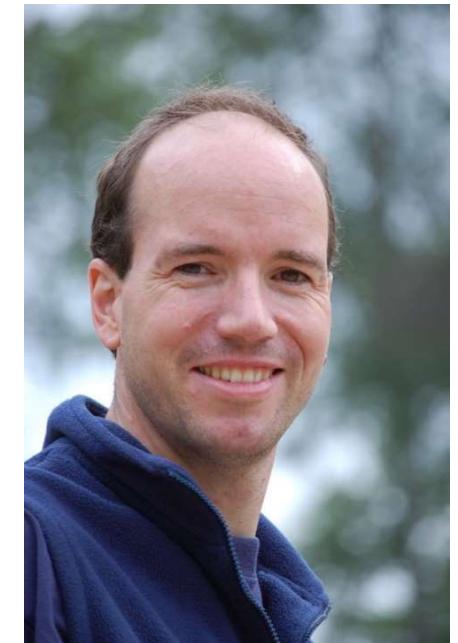
Peter Kassenaar

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

www.kassenaar.com

info@kassenaar.com

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



VANDUUREN
MEDIA

ING

OHRA

zenito
BETERE ZEKERHEID
VOOR ONDERNEEMERS

Atos

euricom
A DIMENSION DATA COMPANY

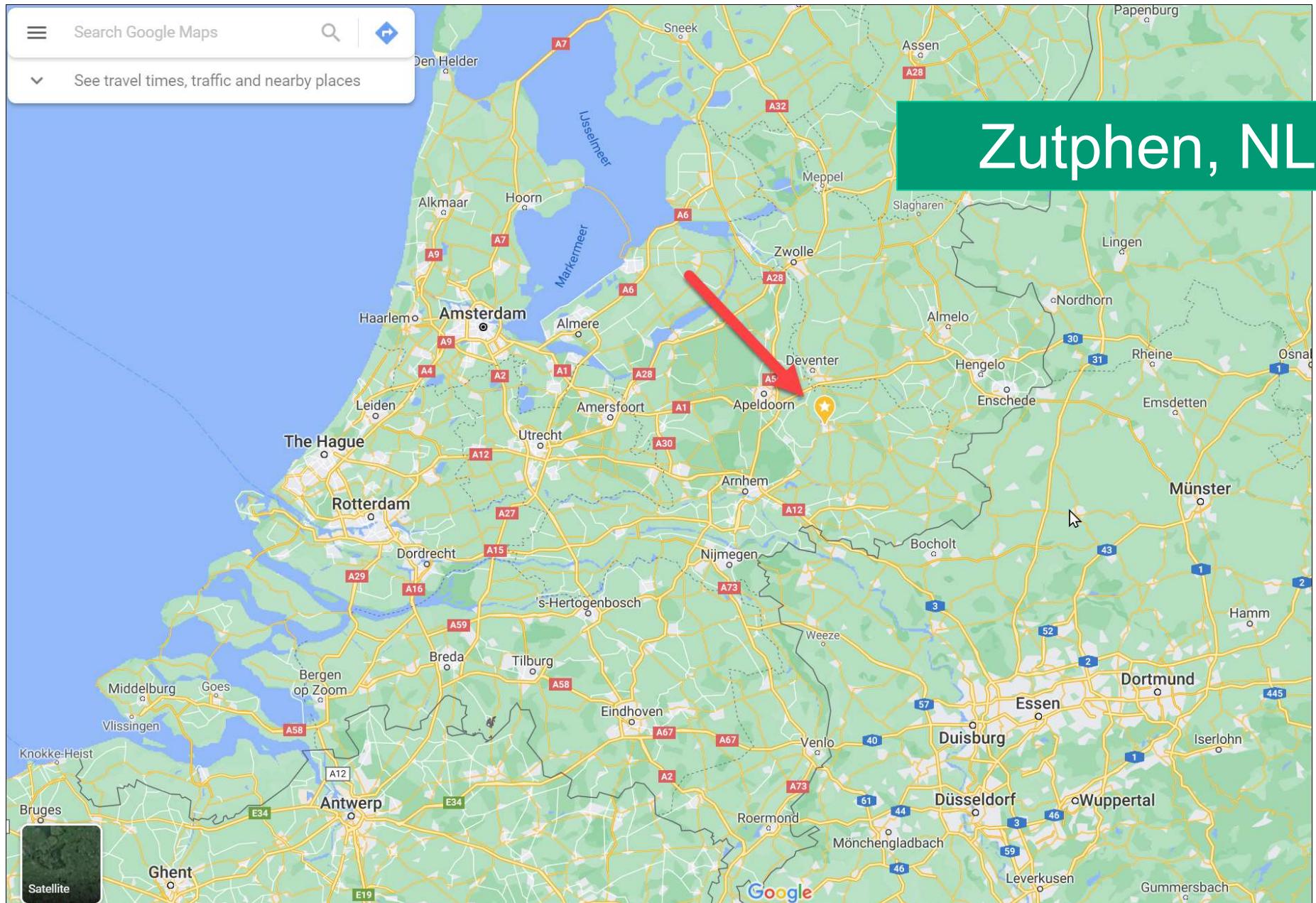
woonbron
OBERON INTERACTIVE

sanoma

ROC West-Brabant

delta lloyd

the eforum
FACTORY



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

github.com/PeterKassenaar/aarixa

The screenshot shows a GitHub repository page for the user PeterKassenaar with the repository name aarixa. The page has a dark theme.

Header: Search or jump to... / Pull requests Issues Marketplace Explore

Repository Information: PeterKassenaar / aarixa Public Unwatch 1 Star 0 Fork 0

Navigation: Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Code Overview: main 1 branch 0 tags Go to file Add file Code

Commits:

- PeterKassenaar Update README.md b2e6d5a 2 minutes ago 2 commits
- .gitignore Initial commit 3 minutes ago
- README.md Update README.md 2 minutes ago

File Content Preview: README.md

Description: Slides and example code on the training Angular - November 2021

Links:

- Generic repo with example code: <https://github.com/PeterKassenaar/voorbeeldenAngular2>
- ...

About: Slides and example code on the training Angular - November 2021

Readme:

Releases: No releases published Create a new release

Packages: No packages published Publish your first package

Over jullie



Stel jezelf kort voor

Voorkennis webdevelopment, (mobile/web-) apps?

(Kennis AngularJS 1.x?)

Voorkennis andere (web) talen?

Verwachtingen van de cursus?

Concrete projecten?

Specifieke vragen of technieken die je wilt behandelen?

Agenda - globaal

22 – 24 november 2021 – ma – wo

17 – 18 januari 2022 – *Advanced Topics*

~9:00 start

~ 10:30 Break

~12:00 lunch

~ 14:30 Break

~16:00-16:15 Einde

Woensdag – mogelijk iets eerder stoppen

Doel van de training

*Je wordt **geen** Angular wizard in 3
dagen (sorry)*

maar.....

Goals

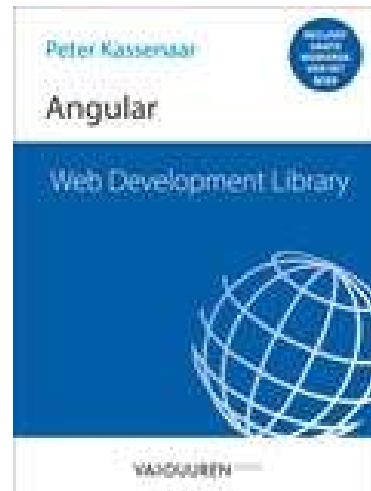
1. Je leert over de **structuur en architectuur** van Angular Apps. Van een **kleine** hello-world app tot een grote Enterprise applicatie.
2. Je bent bekend met de belangrijkste **Angular concepten** van het framework. Specifieke details kun je altijd Googelen.
3. Je hebt **enige hands-on ervaring** met het maken van apps en componenten, services, API's/backends, component communicatie.
4. Je hebt een **algemeen begrip** van de manier waarop moderne web apps worden gemaakt met Angular, TypeScript en build tools.

Agenda - 3 dagen

- Introductie & geschiedenis - waarom Angular?
- Kernbegrippen in Angular 2 – 10
- CLI, Hello World in Angular – inzicht in boilerplate-code
- Angular in depth (modules):
 - Components
 - ECMAScript 2015 + TypeScript
 - Data binding
 - Dependency Injection (DI) – more components
 - Services en http, Observables (RxJS), communicatie backend
 - Intercomponent communication
- BEST PRACTICES / STYLE GUIDE

Materialen

- Software (Angular, NodeJS + NPM, editor, browser)
- Handouts (Github - PDF)
- Oefeningen (Github)
- Websites (online)



The screenshot shows the official Angular website. At the top, there's a navigation bar with links for 'FEATURES', 'DOCS', 'ABOUT', and 'CONTRIBUTE'. Below the navigation is a large red 'A' logo with the text '2.0 NOW IN BETA!' underneath it. To the right of the logo, the text 'One framework. Mobile and desktop.' is displayed. A 'GET STARTED' button is located below this section. The bottom of the page is divided into three columns: 'Fast', 'Mobile', and 'Flexible'. Each column contains a brief description of the framework's capabilities.

Fast	Mobile	Flexible
Angular computes updates based on changes to data, not DOM, for fast updates that scale to the largest data sets with minimal memory overhead.	With Angular Universal for server-side rendering and Web Workers for smooth scrolling and transitions, Angular 2 solves the core challenges in mobile web performance.	Supports several languages including plain JavaScript, TypeScript, and Dart. Also supports both object-style data structure with POJO data-binding and functional reactive style with unidirectional data flow.

angular.io/

Vandaag

- Dag 1 – Intro & Data binding
 - Theory - Introductie & geschiedenis - why Angular
 - Hello World in Angular –boilerplate-code
 - Concepts, context & architecture
 - Angular CLI
 - Components
 - Data binding

2 Richtlijnen

1. Oefeningen

- Maar: neem ook vooral zijpaden, experimenteer, lees verder, maak een eigen project, app, website...

2. Voorbeeldcode

- Als ondersteuning bij de oefeningen, zie boven
- Work in progress – check de Angular-site!
- github.com/PeterKassenaar/voorbeeldenAngular2

Globale werkwijze



Vragen?



Angular vs. The Rest

Differences, similarities, new features

Addressing the “WHY” question!

WHY, would we want to use a frontend framework.

It is all **HTML, CSS** and **JavaScript** right?

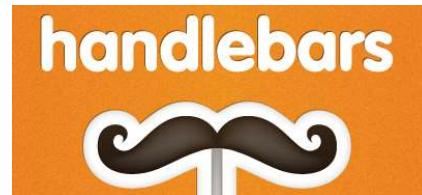
Rethorical question:

**“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

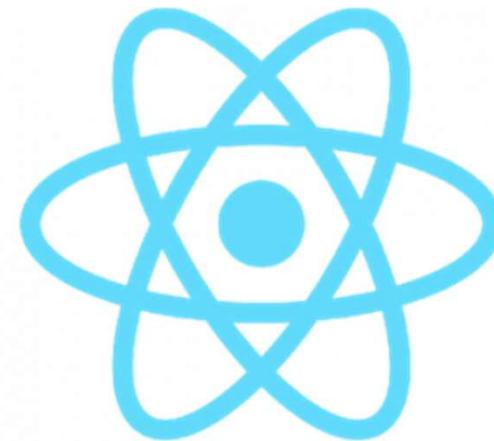


...

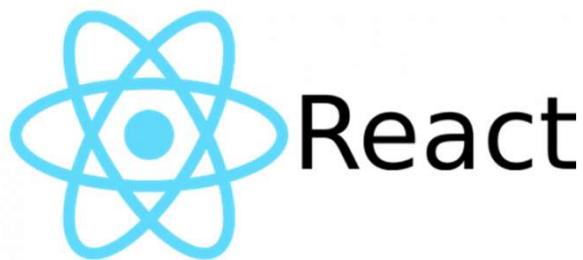
“The Frankenstein Framework”



Front-end Frameworks – the big four



Similarities

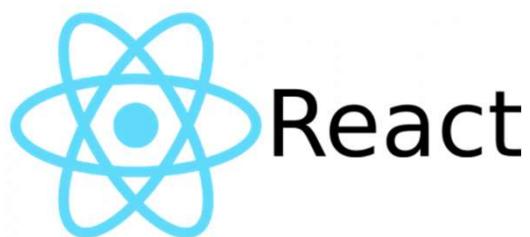


- Creating Single Page Applications
- Based on components
- Data binding, props, events, routing, state management, ...
- Huge ecosystem
- Huge community
- High adaptation rate

Differences (apart from syntax)



- Point of departure: **HTML template**, enhanced with framework specific tags and attributes
 - One-stop-shop / solution
-
- Point of departure: **JavaScript**, JavaScript, JavaScript (JSX)
 - Build-all-yourself / choice anxiety

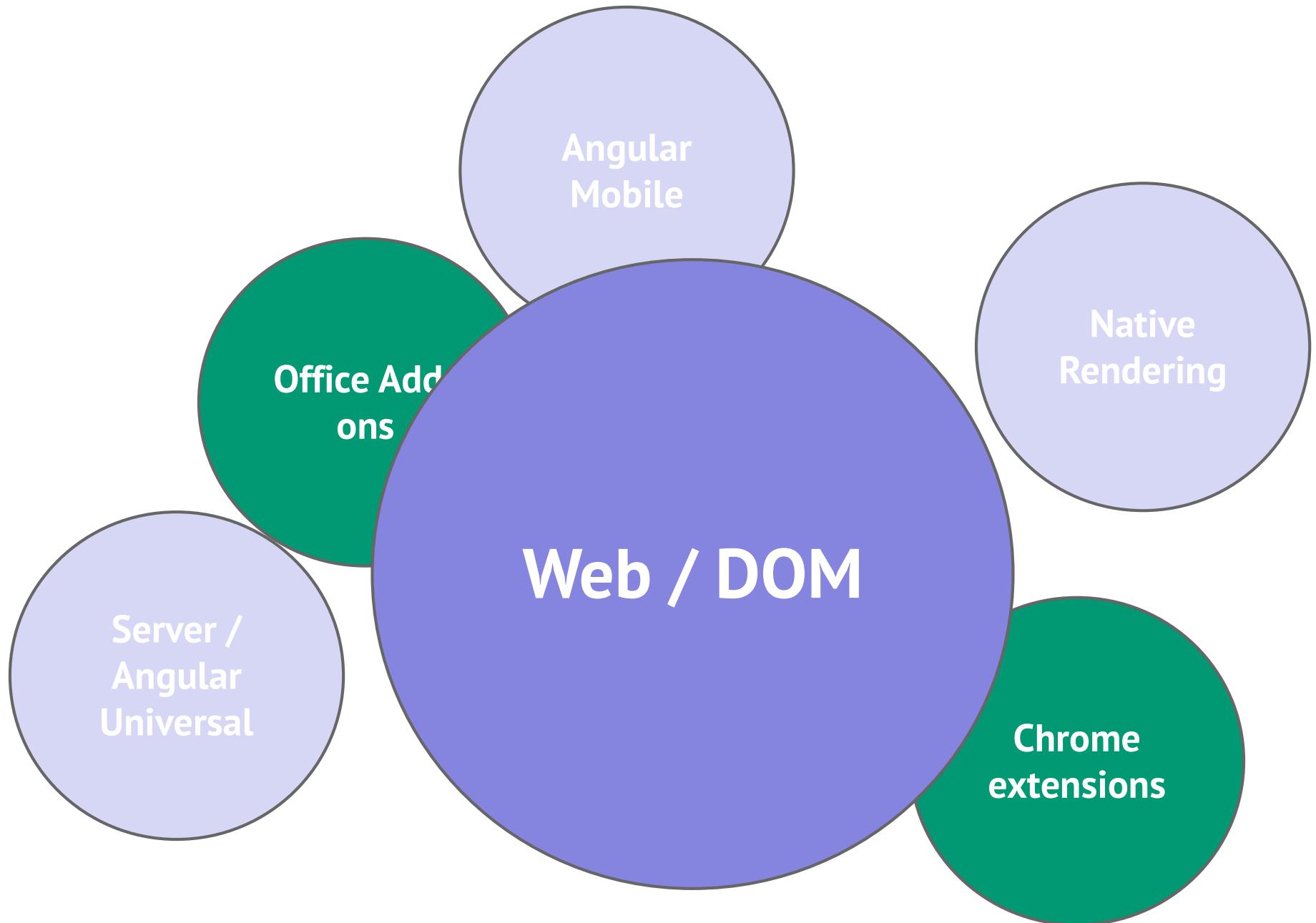




Platform

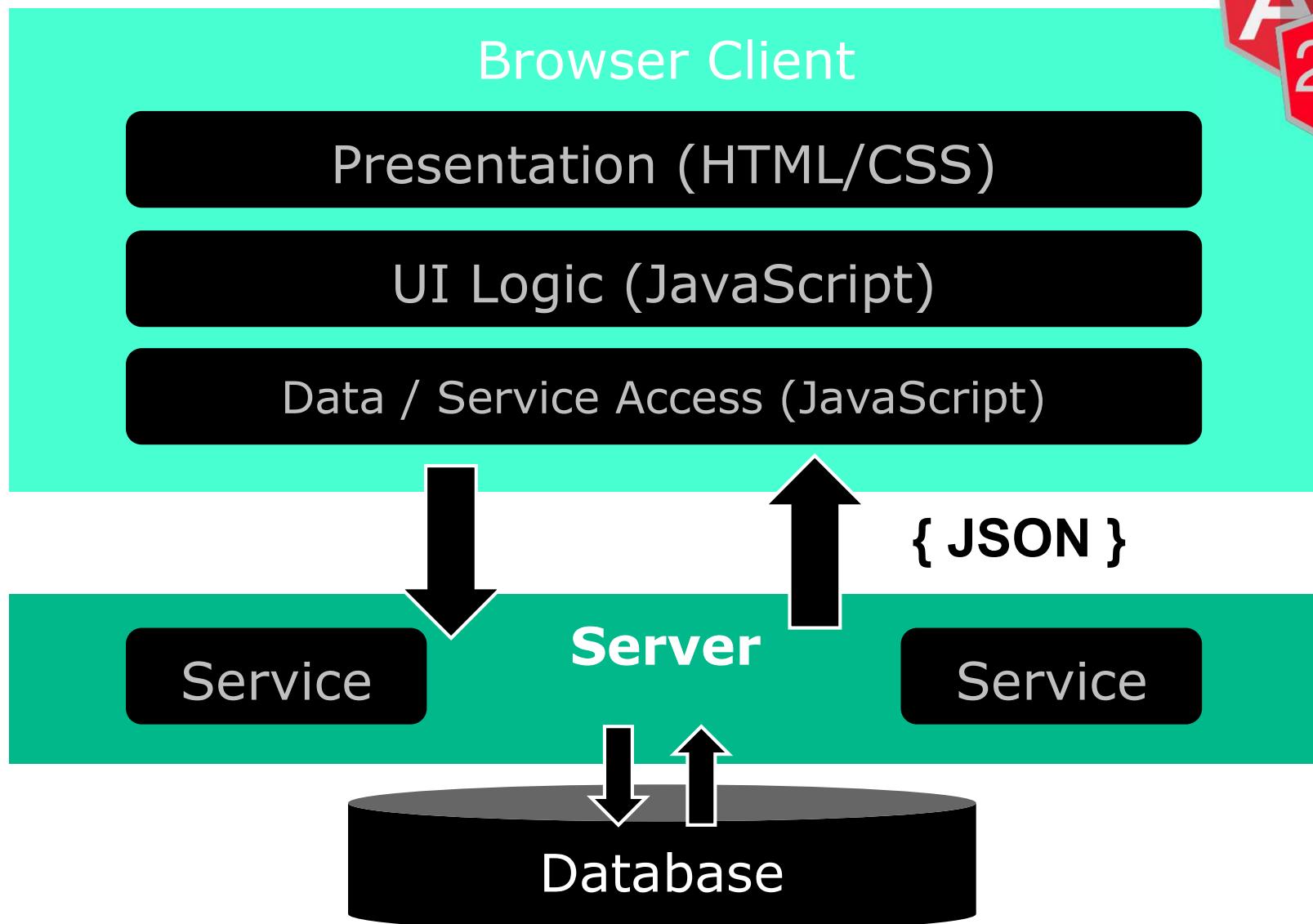
Platform Features

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



Single Page Application

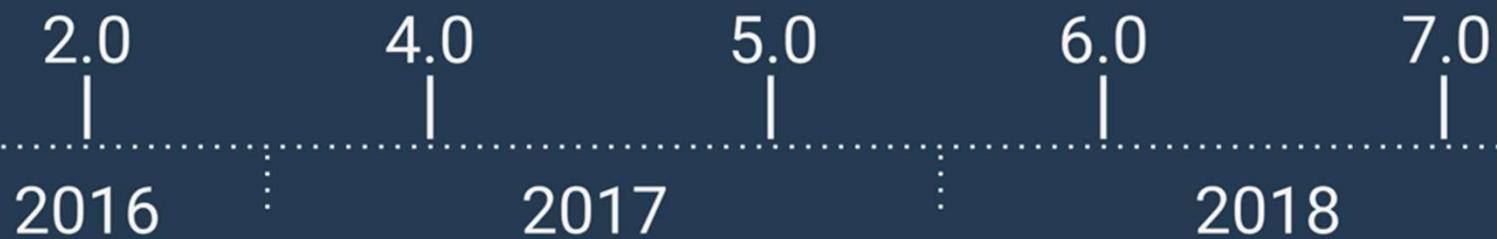
2010 – 20??



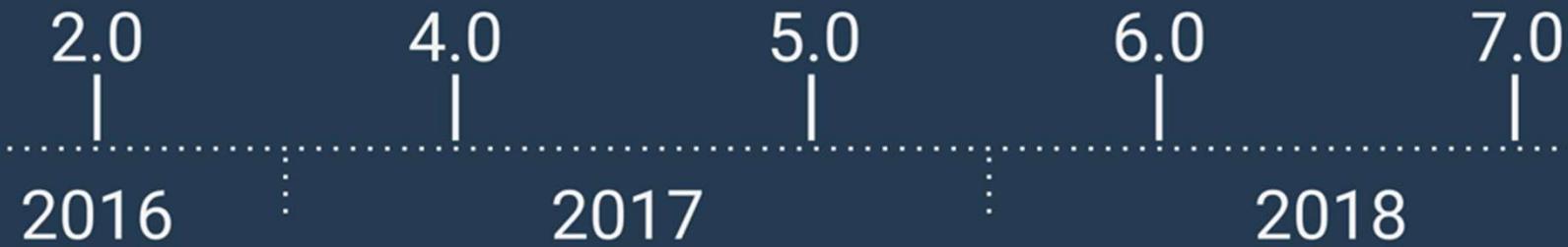


Time-based Releases

@IgorMinar, Nov. 9, 2018



Predictable & Continuous
Evolution



- Predictability
- Painless Updates
- Long Term Support

Angular Versies en -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, Language Service, Security, Setup & Deployment, Service Workers, Keeping Up-to-Date, Updating Your Projects, Angular Releases, Upgrading from AngularJS, Server-side Rendering, Visual Studio 2015 QuickStart, Style Guide, Glossary, and API. The main content area discusses support policy, noting 6 months of active support followed by 12 months of LTS. It includes a table for Angular 4.0.0, 5.0.0, and 6.0.0. A sidebar on the right lists Angular versioning, release frequency, schedule, support policy (selected), 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

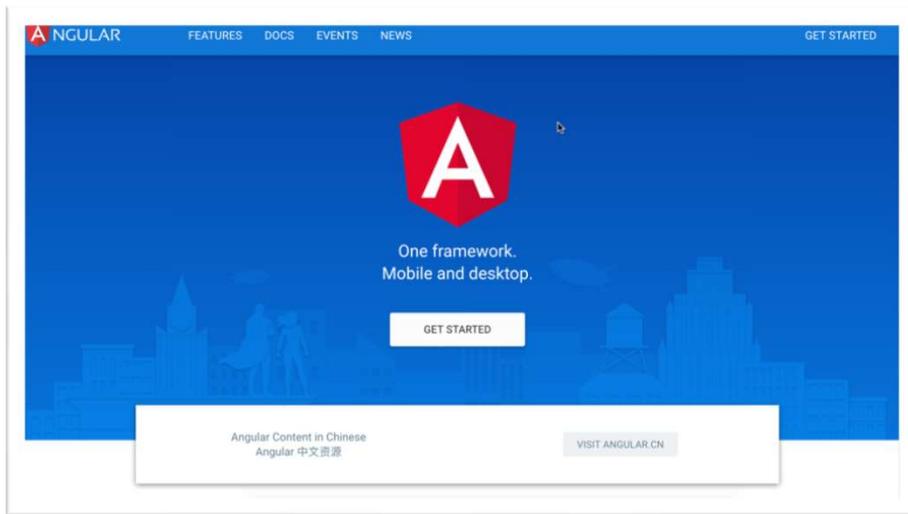
Show me how to update!

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

“It's just

Angular”

Angular as a Platform



<https://angular.io/>

The screenshot shows the Angular Material website. It has a teal header with the 'ANGULAR MATERIAL' logo and a 'PREVIEW ON GITHUB' link. Below the header, there's a preview of a mobile application interface showing various cards. The main text reads 'Angular Material' and 'Material Design components for Angular 2 apps', with a 'PREVIEW ON GITHUB' button. At the bottom, there's a section titled 'Sprint from Zero to App' with a sub-section 'Fast and Consistent'.

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

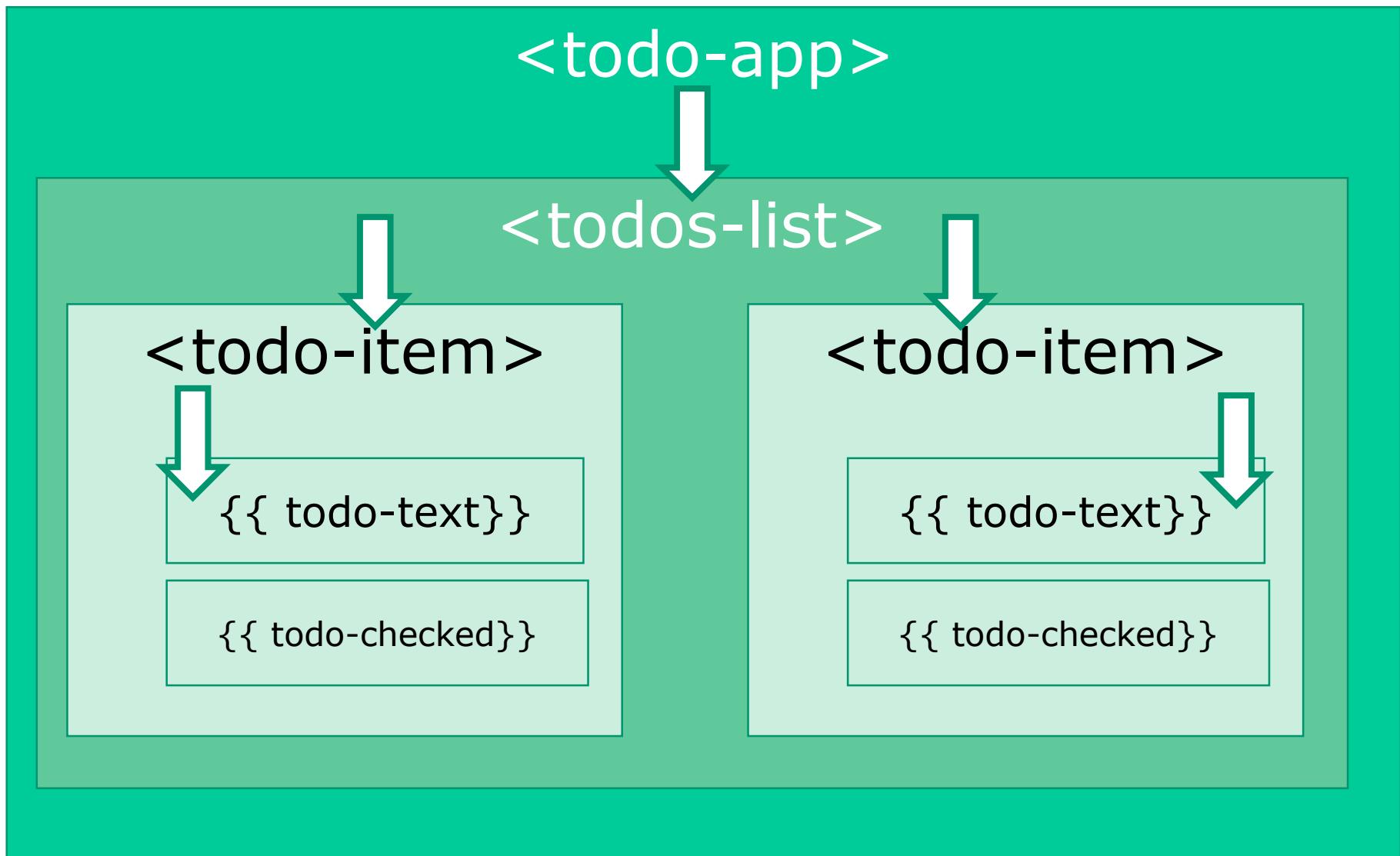
The screenshot shows the Angular CLI website. It has an orange header with 'ANGULAR CLI', 'CLI COMMANDS', 'GITHUB', and 'GET STARTED' links. Below the header, there's a terminal window showing command examples: 'npm install -g angular-cli', 'ng new my-dream-app', 'cd my-dream-app', and 'ng serve'. The main text reads 'Angular CLI' and 'A command line interface for Angular', with a 'GET STARTED' button. At the bottom, there are sections for 'ng new' (describing it as a command to create an application) and 'ng generate'.

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

The screenshot shows the Angular Universal website. It has a green header with 'ANGULAR UNIVERSAL', 'OVERVIEW', 'API', 'GITHUB', and 'GET STARTED' links. Below the header, there's a diagram showing a smartphone connected to a server icon with the Angular logo. The main text reads 'Angular Universal' and 'Server-side Rendering for Angular 2 apps', with a 'GET STARTED' button. At the bottom, there are sections for 'Better Perceived Performance' (describing server-side rendering) and 'Optimized for Search Engines'.

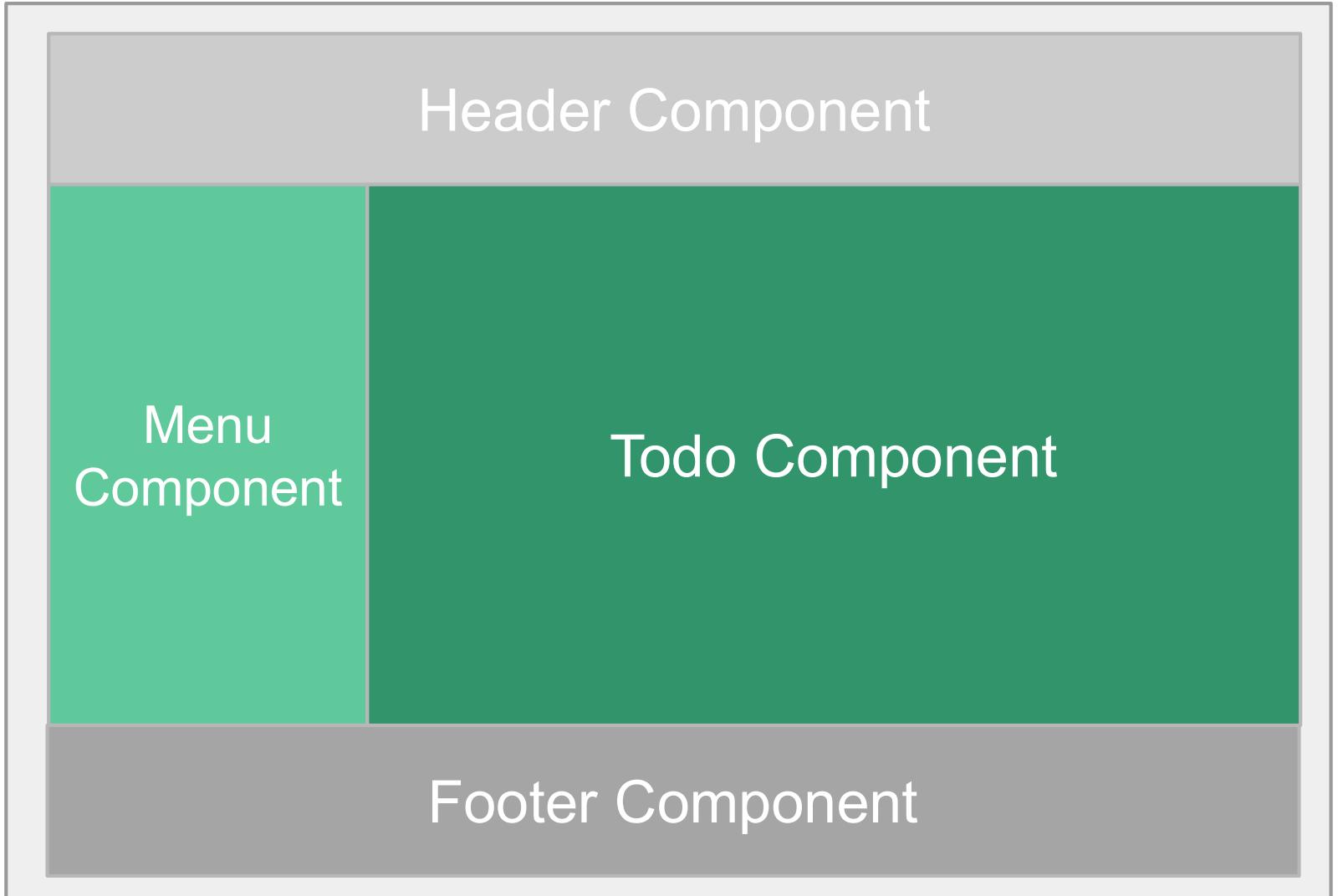
<https://universal.angular.io/>

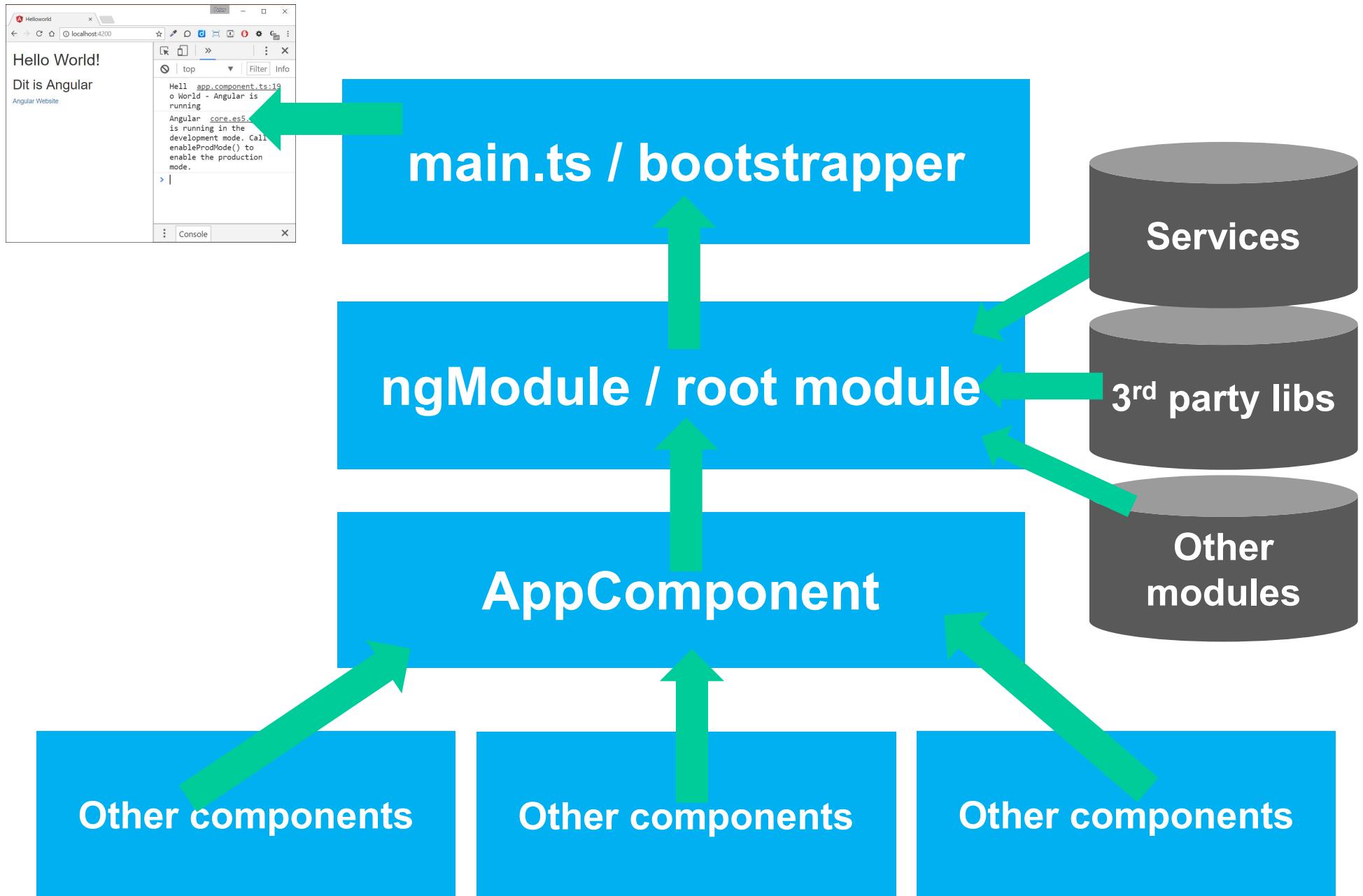
Angular 2 - components



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

Components – visual representation







Let's write some code

Hello World in Angular

Angular development dependency: NodeJS 12+

The screenshot shows the official Node.js website. At the top, there's a dark header with the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION (which is highlighted). Below the header, a main section features a green banner with the text "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Underneath, there are two prominent download buttons: one for "12.13.0 LTS" (labeled "Recommended For Most Users") and another for "13.0.1 Current" (labeled "Latest Features"). Below these buttons, there are links for "Other Downloads", "Changelog", and "API Docs" for both versions. Further down, there's a link to the "Long Term Support (LTS) schedule". A call-to-action encourages users to "Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter." At the bottom of the page, there's a footer with links to "Report Node.js issue", "Report website issue", and "Get Help". The footer also contains copyright information and trademarks.

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

[Download for macOS \(x64\)](#)

12.13.0 LTS
Recommended For Most Users

13.0.1 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 Monthly Newsletter.

LINUX FOUNDATION COLLABORATIVE PROJECTS

[Report Node.js issue](#) | [Report website issue](#) | [Get Help](#)

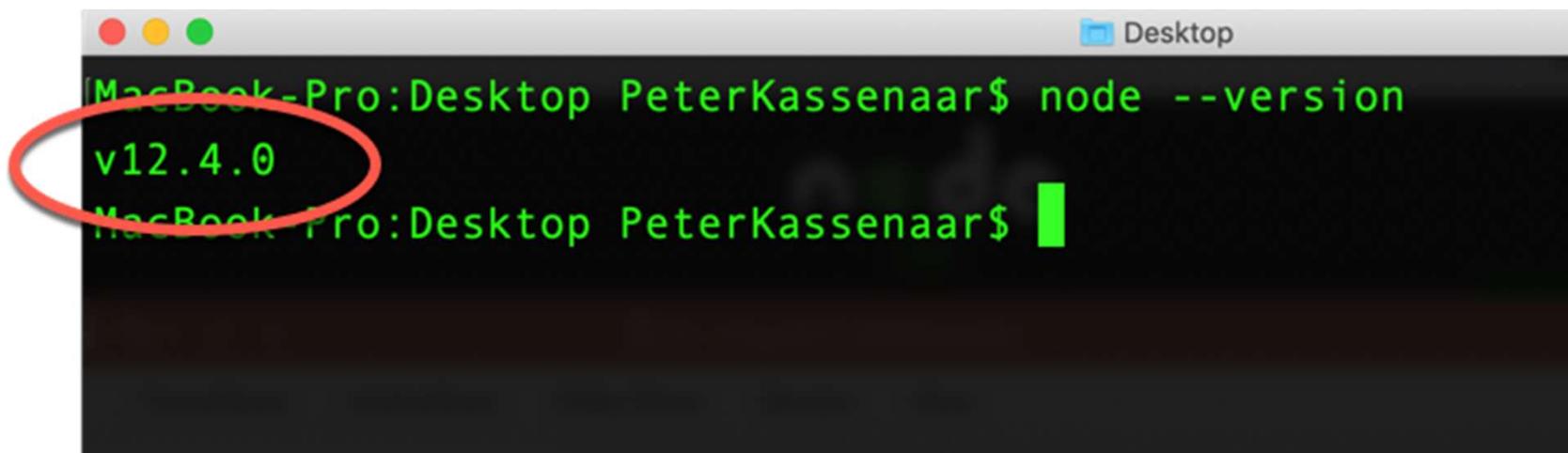
© Node.js Foundation. All Rights Reserved. Portions of this site originally © Joyent.

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

Linux Foundation is a registered trademark of The Linux Foundation.

Linux is a registered trademark of Linus Torvalds.

Node – check your version



A screenshot of a macOS terminal window titled "Desktop". The window contains the following text:

```
MacBook-Pro:Desktop PeterKassenaar$ node --version
v12.4.0
MacBook Pro:Desktop PeterKassenaar$
```

The output "v12.4.0" is highlighted with a red oval.

Mini workshop

- Download or clone

<https://github.com/PeterKassenaar/voorbeeldenAngular2>

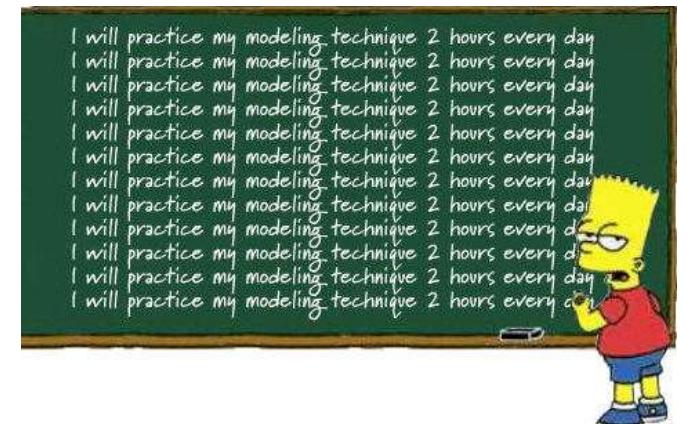
`cd examples`

`cd 100-helloworld`

`npm install`

`npm start`

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



← → C ⌂ ⓘ localhost:4200

Hello World!

This is Angular

Angular Website

The screenshot shows a browser window displaying the 'Hello World!' Angular application. Below the browser is an IDE interface. On the left is a 'Project' view showing a folder structure for 'voorbeeldenAngular2' containing 'examples' and '100-helloworld'. The '100-helloworld' folder is highlighted with a red rectangle. Inside it are 'node_modules' (library root), 'src' (which contains 'app' with 'app.component.ts' and 'app.module.ts', and other files like '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', and 'yarn.lock'), and 'angular.json', 'package.json', 'package-lock.json', 'tsconfig.json', and 'yarn.lock'. On the right is a code editor showing the 'app.component.ts' file. The code is as follows:

```
import {Component, OnInit} from '@angular/core';
@Component({
  selector: 'hello-world',
  template: `
    

# Hello World!</h1> <h2>This is Angular</h2> <a href="http://angular">Angular</a> ` }) export class AppComponent implements OnInit { constructor() {} }


```

```
import {Component, OnInit} from '@angular/core';
@Component({
  selector: 'hello-world',
  template: `
    

# Hello World!</h1> <h2>This is Angular</h2> <a href="http://angular">Angular</a> ` }) export class AppComponent implements OnInit { constructor() {} }


```

Boilerplate code for Hello World

Steps

1. Set up environment, boilerplate & libraries
 - Important configuration files
2. Angular Component(s)
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": "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

```
{  
  "$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 'Code' tab is selected. The file 'non-essential-files.txt' is displayed, listing various files and directories:

```
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 discusses the `@NgModule` decorator and its purpose in organizing applications. To the right, there is a sidebar with a bio for John Papa and a "Most Recent" section with links to his other posts.

Header: Angular Modules

Post Title: Introducing Angular Modules - Root Module

Author: John

bio: 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 →

Recent Posts:

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

Text in Post: 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."

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

Step 4 - bootstrap component

Best practice: bootstrap app in separate component

Convention: main.ts, or 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

Header:

```
<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 van index.html

Verwijzing naar de root-component:

```
<body>  
  <hello-world>  
    Bezig met laden...  
  </hello-world>  
</body>
```

App draaien

`npm start` – draait de scriptopdracht `start` uit `package.json`.

ng serve - start globale angular-cli instantie

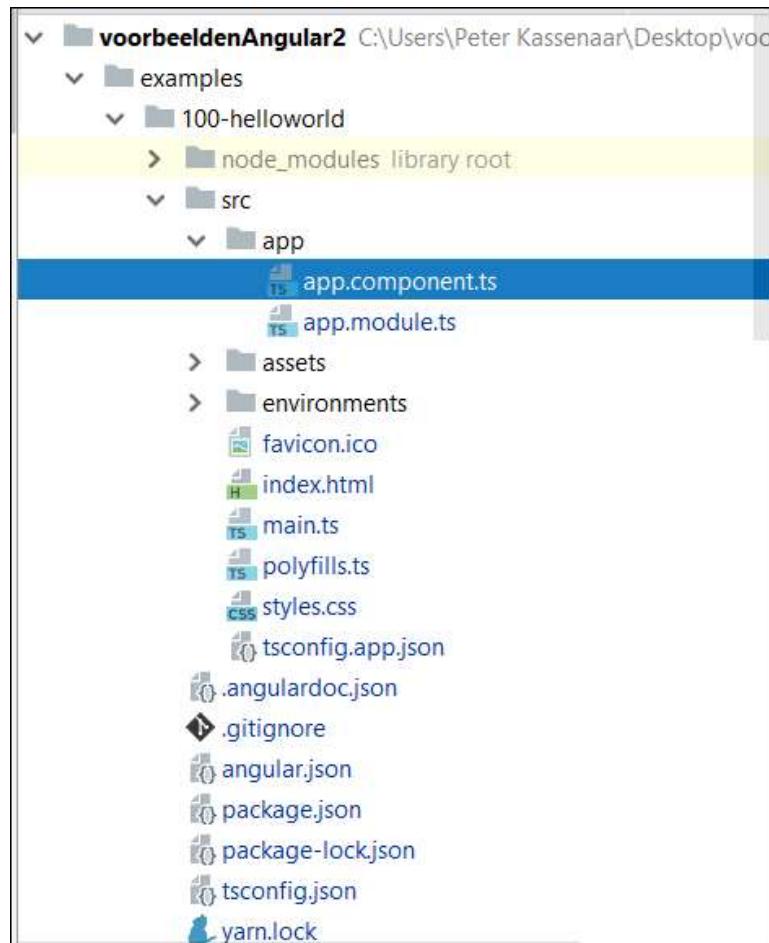
```
Terminal
+ chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4} [initial] [rendered]
x

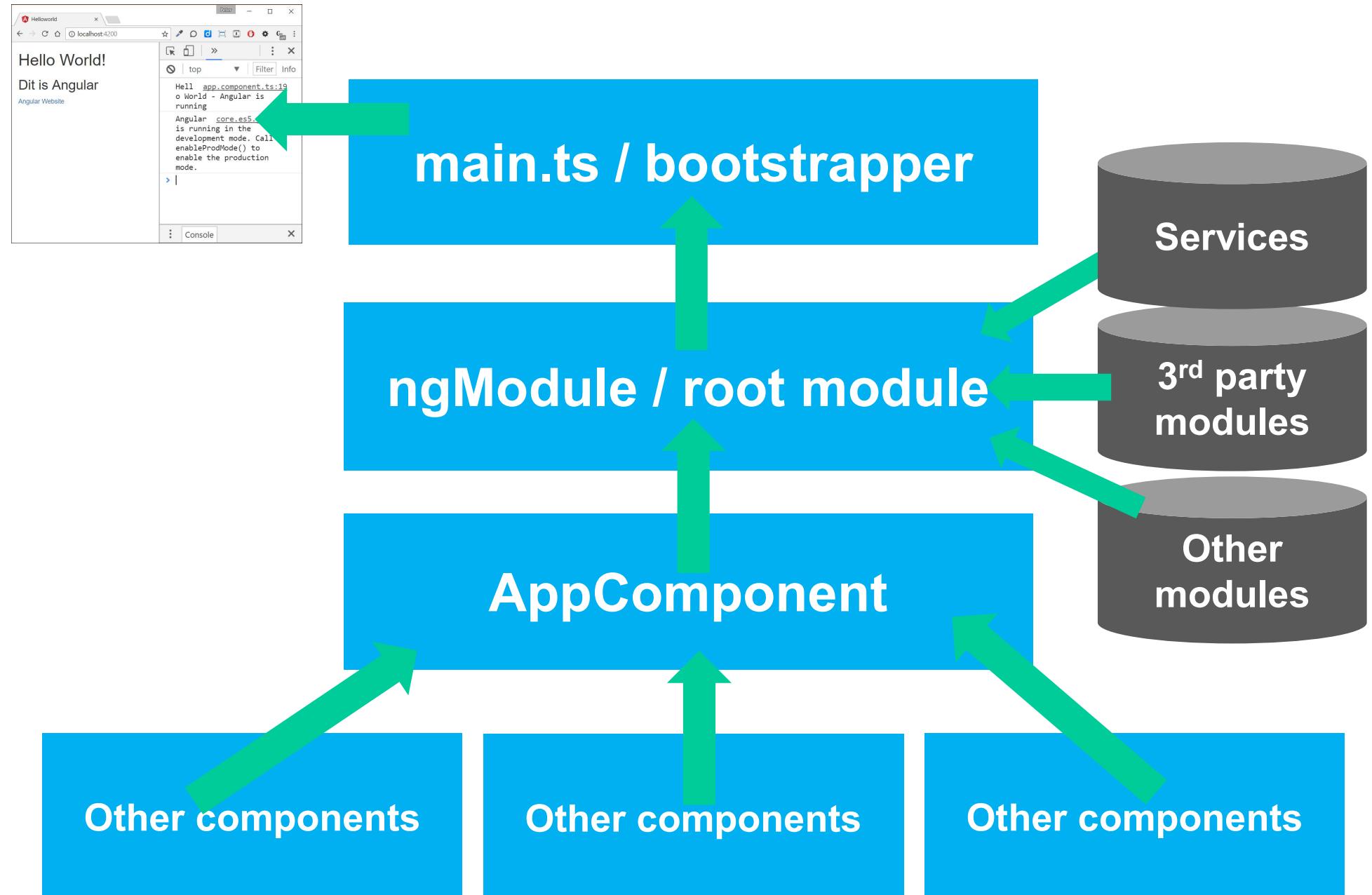
chunk {1} main.bundle.js, main.bundle.js.map (main) 3.73 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 9.78 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.1 MB [initial] [rendered]
chunk {4} runtime~main.bundle.js, runtime~main.bundle.js.map (runtime~main) 1.03 kB [initial] [rendered]
webpack: C
localhost:4200
Hello World!
Dit is Angular
Angular Website
Elements Console Sources Network Performance Memory > ...
top 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
> |
```

Daarna: wijzigingen aanbrengen in app.component.ts

– worden opgepikt door Live Reload

Basic Project Structure





Assets

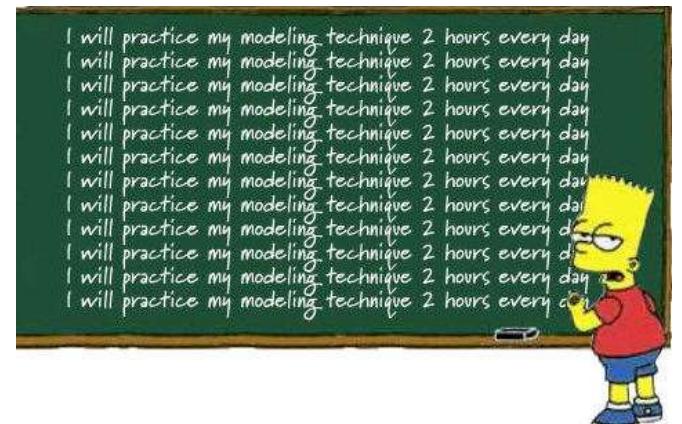
github.com/PeterKassenaar/voorbeeldenAngular2

Oefeningen en meer voorbeeldcode

Checkpoint

- Er is aardig wat boilerplate code nodig om een Angular-app te starten
- Vier stappen
 1. Set up environment, boilerplate & libraries
 2. Schrijf Angular Root Component voor de app
 3. Bootstrap de component
 4. Schrijf HTML-pagina (`index.html`)
- Daarna: app gaan uitbreiden
- Oefening 1a), 1b), 1c), 1d)

Oefening....





Angular CLI

Snel nieuwe projecten instellen via de command line

Angular-CLI to the rescue

- Het *is* mogelijk nieuwe Angular-projecten from scratch te starten.
- Met de CLI is eenvoudiger.
- CLI-options:
 - Scaffolding
 - Generating
 - Testing
 - Building
 - AOT-Compiling
 - ...

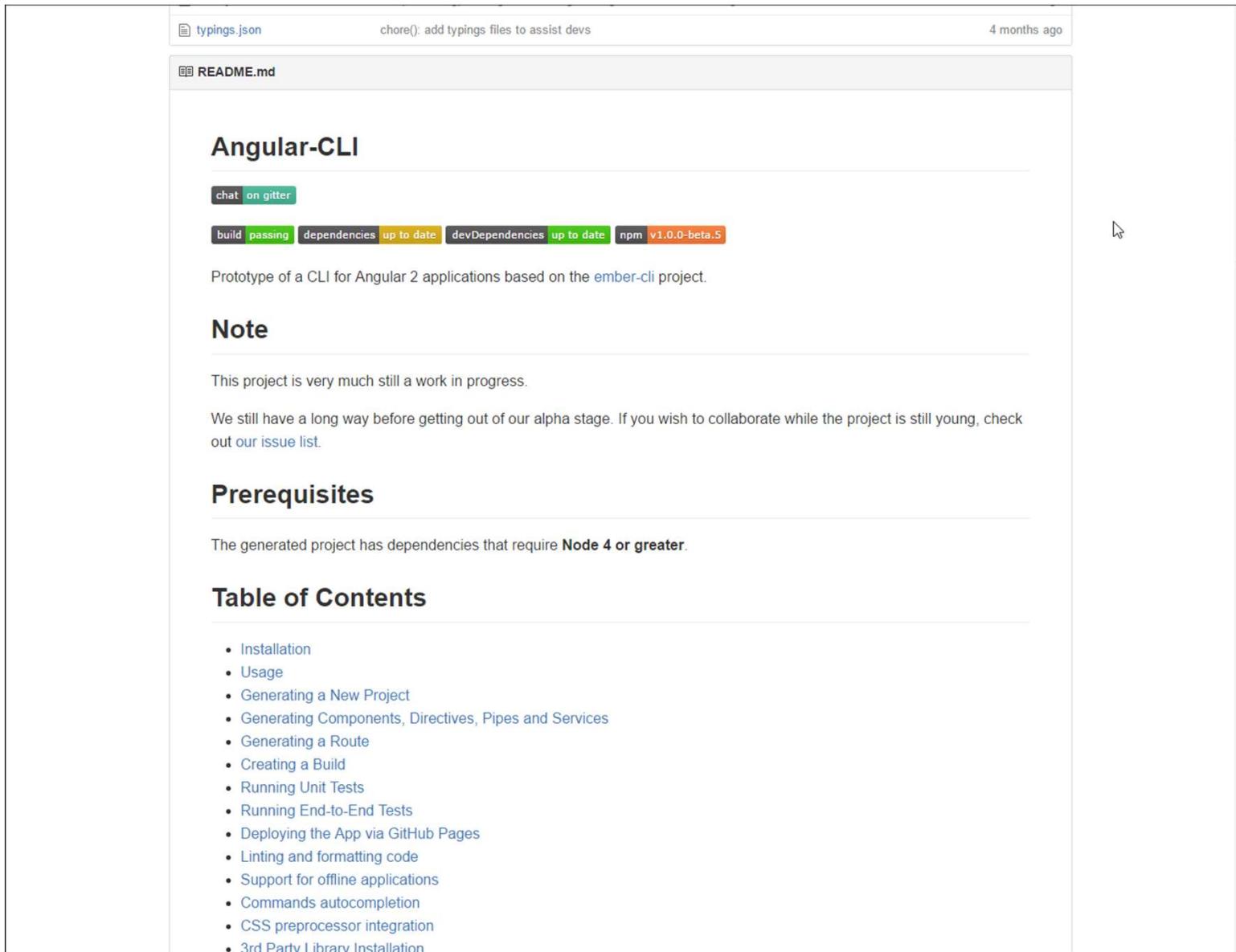
Scaffolding - Angular CLI

Projecten, componenten, routes en meer definiëren
vanaf de command line

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

en

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



A screenshot of the Angular-CLI project's GitHub README page. The page features a header with links to 'typings.json' (chore: add typings files to assist devs, 4 months ago), 'README.md', and a 'chat on gitter' button. Below the header, there are several badge indicators: 'build passing', 'dependencies up to date', 'devDependencies up to date', 'npm v1.0.0-beta.5'. A note below the badges states: 'Prototype of a CLI for Angular 2 applications based on the ember-cli project.' The main content area contains sections for 'Note' (warning: 'This project is very much still a work in progress.' and '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' (mentioning Node 4 or greater), and 'Table of Contents' (listing various command-line options like Installation, Usage, Generating a New Project, etc.).

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

Introduction
Getting Started >
Understanding Angular >
Developer Guides >
Best Practices >
Angular Tools >
Tutorials >
Release Information >
Reference >▼
Conceptual Reference >
CLI Command Reference >▼
Overview
Usage Analytics
ng add
ng analytics
ng build
ng config

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

• CLI Overview and Command Reference

Installing Angular CLI

Basic workflow

Workspaces and project files

Workspace and project configuration

CLI command-language syntax

Boolean options

Relative paths

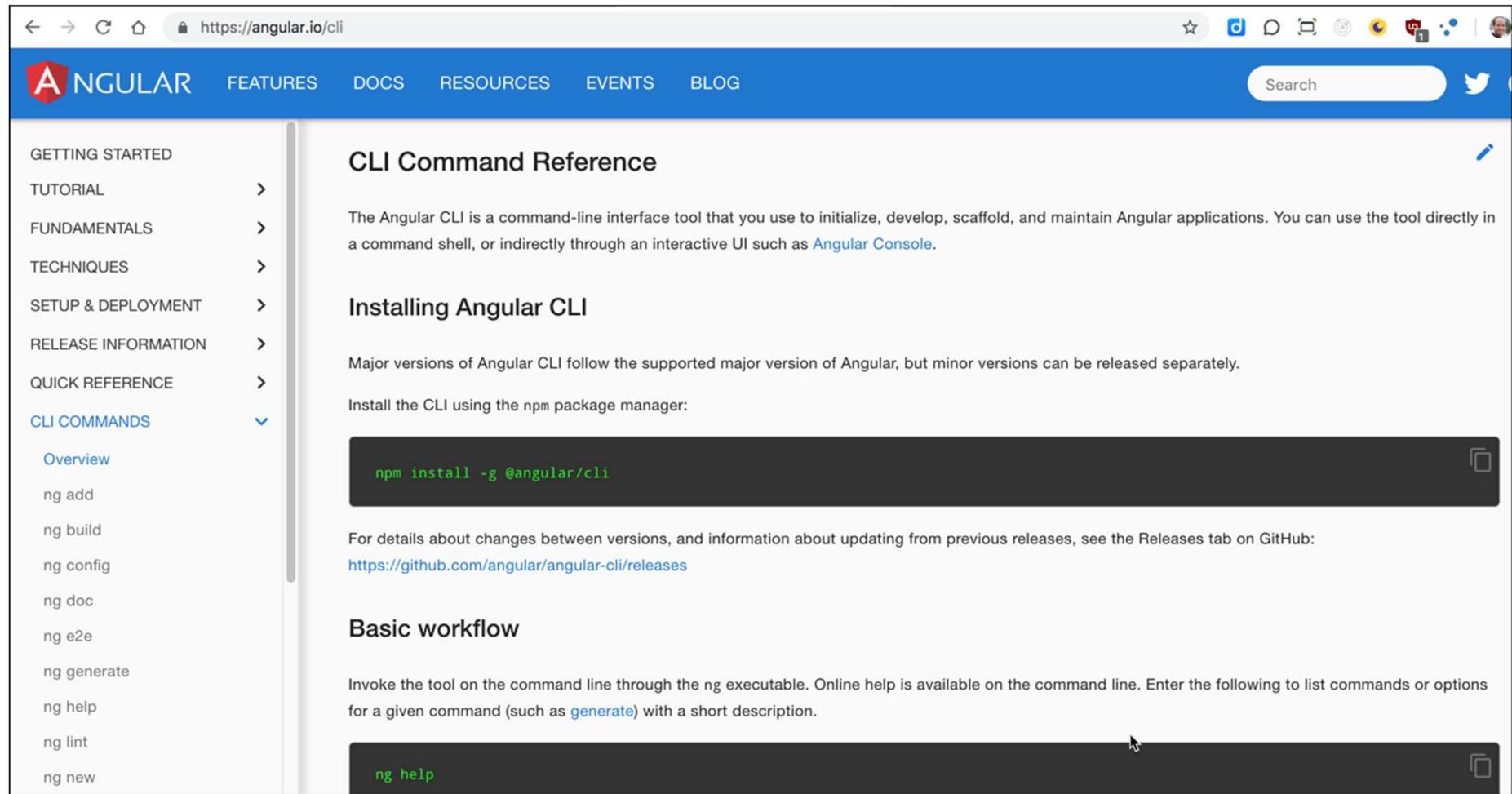
Schematics

Command Overview



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

Documentatie - in de Angular Docs



The screenshot shows a web browser displaying the Angular CLI documentation at <https://angular.io/cli>. The page has a blue header with the Angular logo and navigation links for FEATURES, DOCS, RESOURCES, EVENTS, and BLOG. A search bar and social sharing icons are also present. The main content area is titled "CLI Command Reference". It describes the Angular CLI as a command-line interface tool for initializing, developing, scaffolding, and maintaining Angular applications. It includes sections on "Installing Angular CLI", "Basic workflow", and command examples like "npm install -g @angular/cli" and "ng help". A sidebar on the left lists various Angular CLI commands: TUTORIAL, FUNDAMENTALS, TECHNIQUES, SETUP & DEPLOYMENT, RELEASE INFORMATION, QUICK REFERENCE, and a expanded section for CLI COMMANDS (Overview, ng add, ng build, ng config, ng doc, ng e2e, ng generate, ng help, ng lint, ng new).

<https://angular.io/cli>

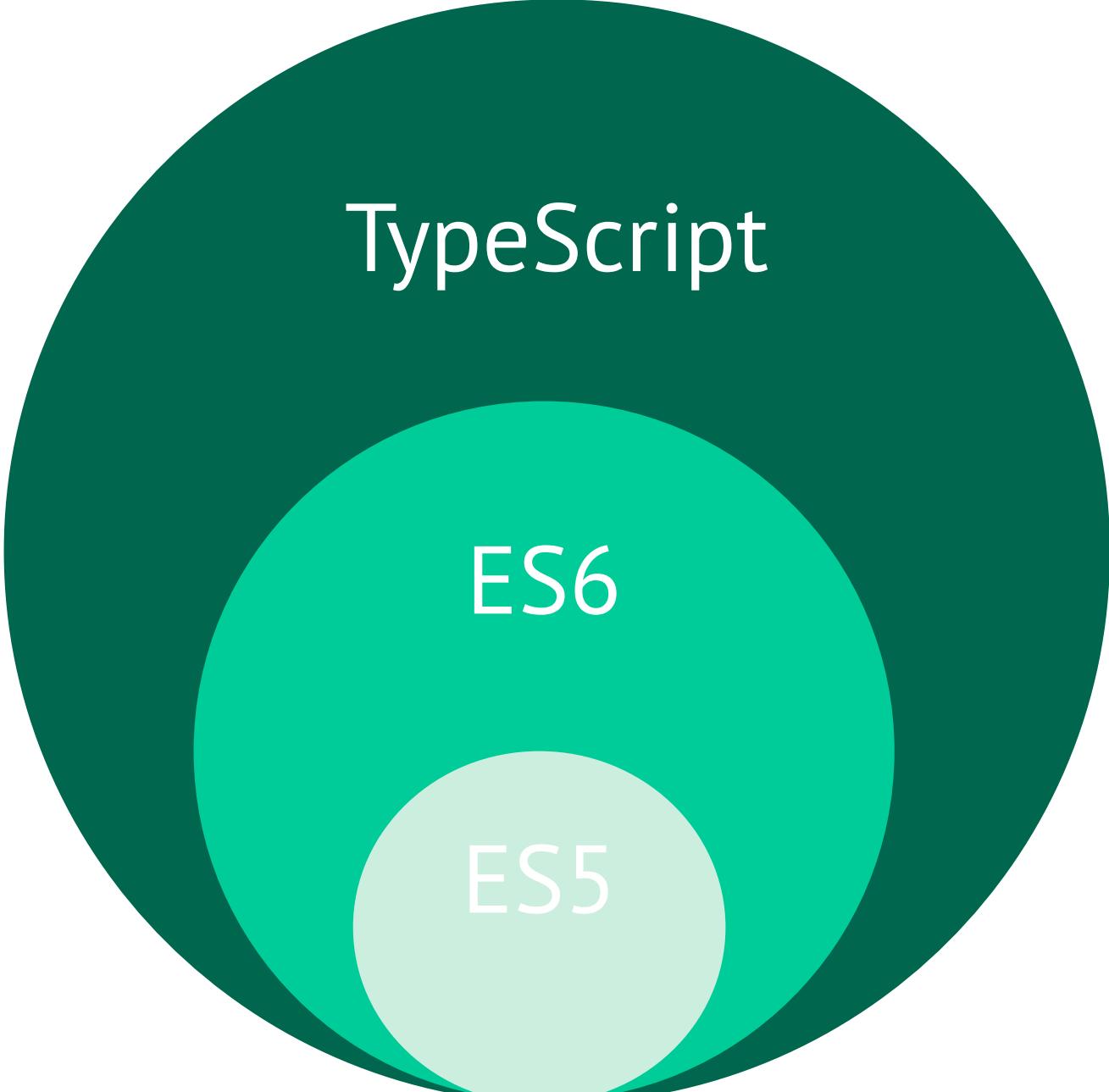


Angular 2 Code - Backend

Kort over TypeScript en ES6

Programmeertalen





TypeScript

ES6

ES5

ES6 en TypeScript

De toekomst van JavaScript is ES6/ES2015

Major update van JavaScript als programmeertaal

Modules, classes en meer

Helpt bij het ontwikkelen in Angular 2

TypeScript breidt ES6 verder uit

Annotaties & types

Interfaces

Compiler

TypeScript – tooling support

Types, Autocompletion.

Compile-time checking in editors.

Alles is *optioneel*. Je kunt altijd nog gewoon
JavaScript gebruiken.

Onderdelen van een 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 een totaal ander framework dan Angular 1
- Component-based vs. Page-based
- Nieuwe syntaxis
- Nieuwe programmeertalen en andere nieuwe kenmerken
- Concepten komen – grotendeels – overeen
- Veel boilerplate-code nodig voor een Quickstart
- Daarna: niet meer naar omkijken. Concentreren op de componenten