



Global Knowledge®

Angular + TypeScript

Module 1 - Inleiding

Peter Kassenaar –
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

- 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

euricom
A DIMENSION DATA COMPANY

s a n o m a

delta lloyd

zenito
BETERE ZEKERHEID
VOOR ONDERNEMERS

Atos

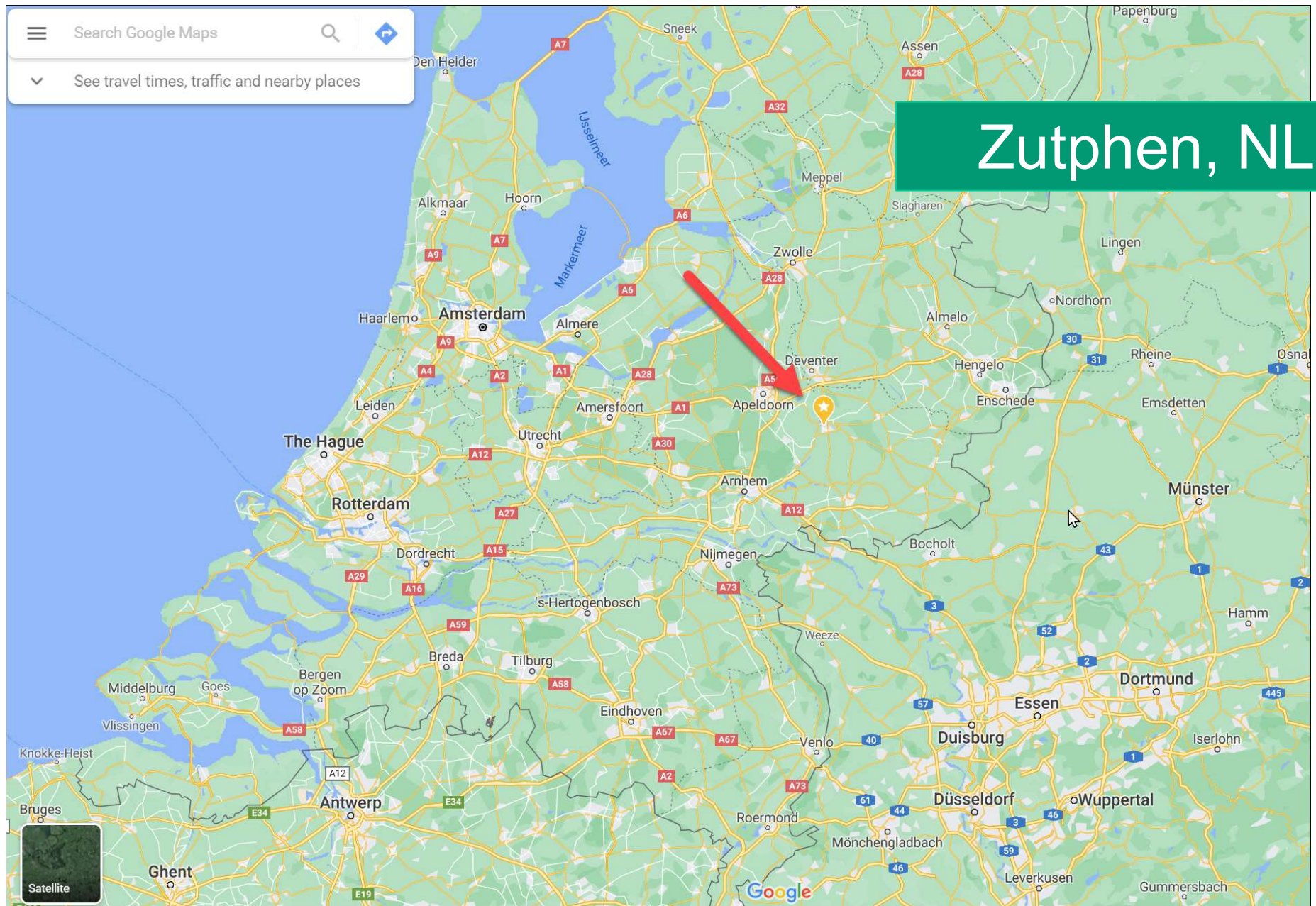


OBERON INTERACTIVE

woonbron

ROC West-Brabant

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, // this on to for console logging of route events
  preloadingStrategy: PreloadingStrategy.NoPreload,
};

@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/ng-global

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

PeterKassenaar / ng-global Public Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main ng-global / README.md Go to file

PeterKassenaar Update README.md Latest commit ceb37c1 now History

1 contributor

7 lines (6 sloc) 283 Bytes <> Raw Blame

ng-global

Slides and example code on the training Angular + TypeScript, Global Knowledge, November 2022

Links

- Repository with example code: <https://github.com/PeterKassenaar/voorbeeldenAngular2>
- Please study the blogs and sites mentioned in the various presentations!
- ...

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

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 – 3 dagen

21 - 23 November 2022 – ma. – wo.

- ~ 9:00 start
 - ~ 10:15 short break
- ~ 12:30 lunch
- ~ 13:15 middagsessie
 - ~ 14:15 short break
- ~ 16:15 Einde

Doel van de training

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

maar....

Doelen

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.

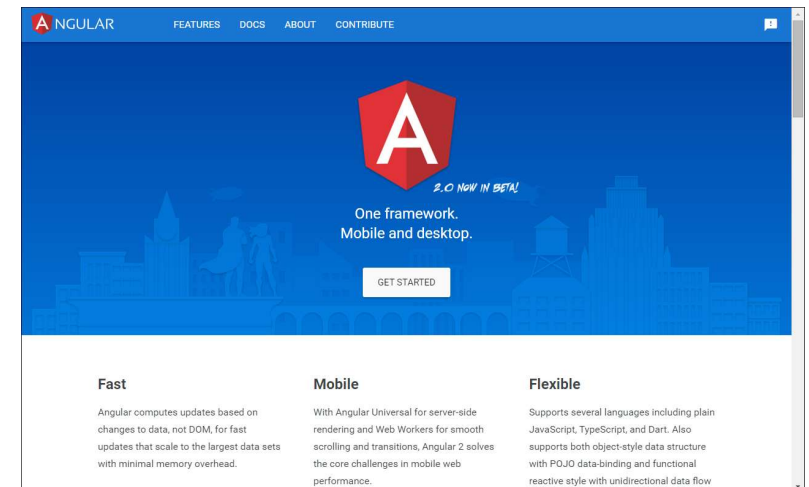
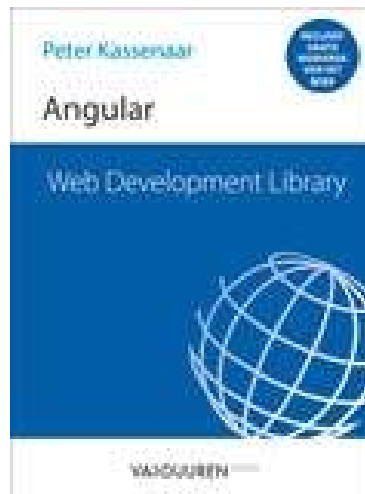
Agenda - 3 dagen

- **Introductie & geschiedenis** - waarom Angular, TypeScript?
- **Kernbegrippen** in Angular 2 – 14
- 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**



Materialen

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



angular.io/

Vandaag

- Dag 1 – Intro & Data binding
 - Theorie - Introductie & geschiedenis – waarom Angular
 - Hello World in Angular –boilerplate-code
 - Components + Modules
 - Concepts, context & architecture
 - Angular CLI, TypeScript
 - Data binding

2 Richtlijnen

1. Oefeningen

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

2. Voorbeeldcode

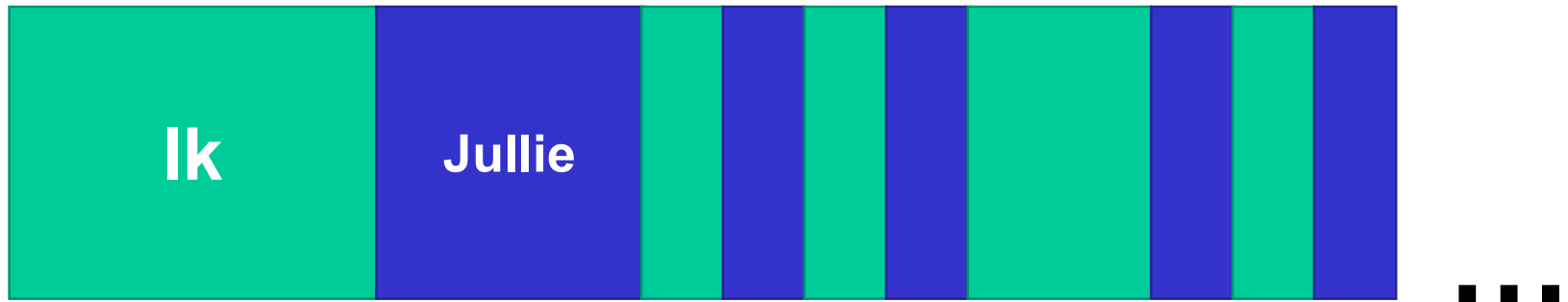
- Als ondersteuning bij de oefeningen, zie boven
- Work in progress – check de Angular-site!
- github.com/PeterKassenaar/voorbeeldenAngular2

Advanced warning - First morning

Not so much code...

*Concepts, architecture,
structure*

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?

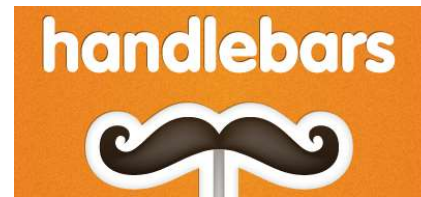
Rethorical question:

*“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

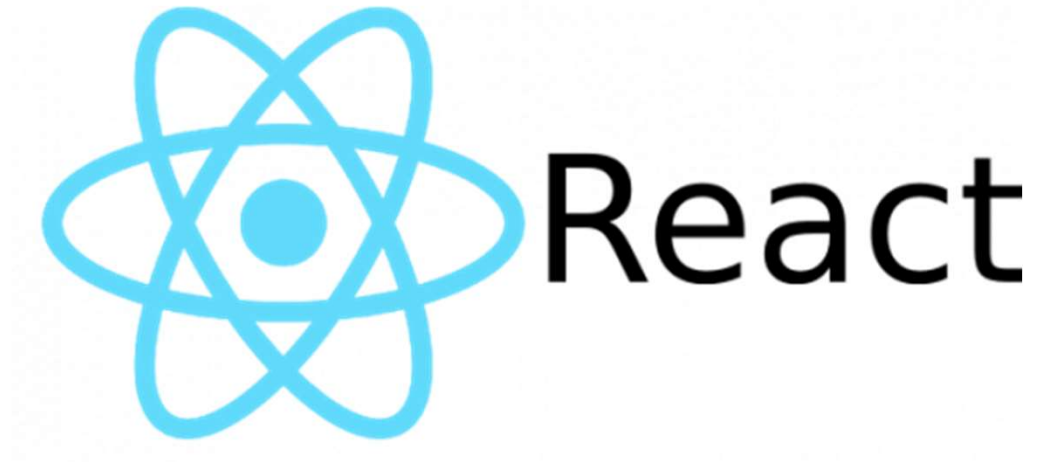


...

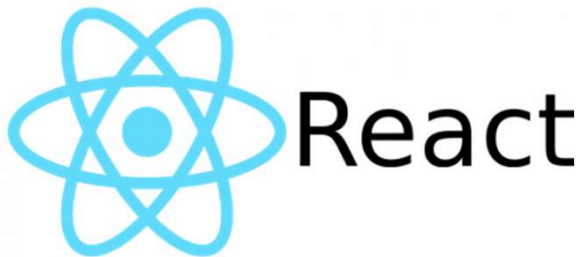
“The Frankenstein Framework”



Front-end Frameworks – the big four



Similarities

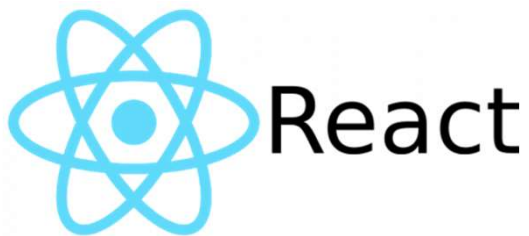


- Creating **Single Page Applications**
- Based on **components**
- Data **binding**, **props**, **events**, **routing**,
state management, ...
- Huge **ecosystem**
- Huge **community**
- High **adaptation** rate

Differences (apart from syntax)



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



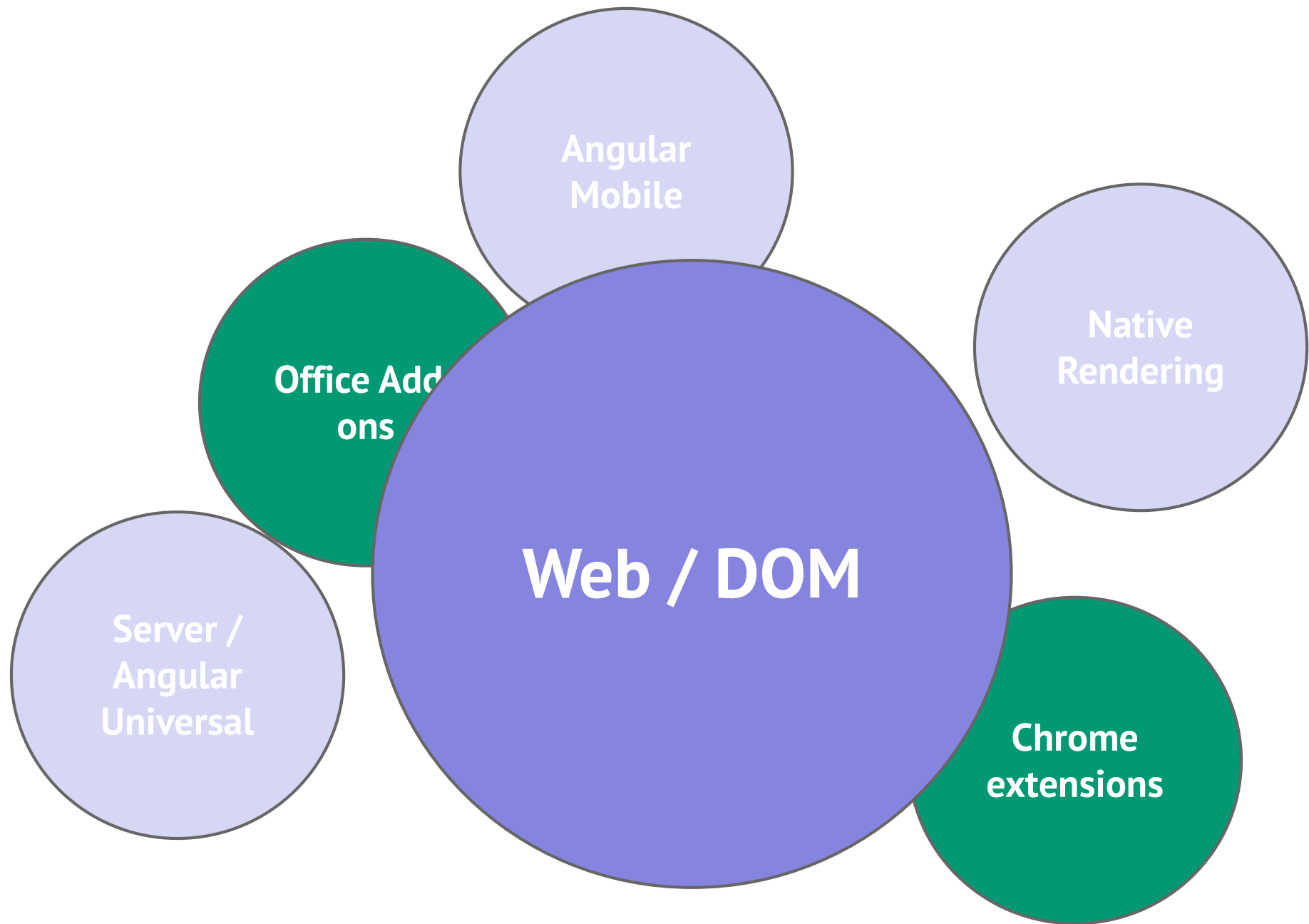
- Point of departure: **JavaScript**, JavaScript, JavaScript (JSX)
- Build-all-yourself / choice anxiety



Platform

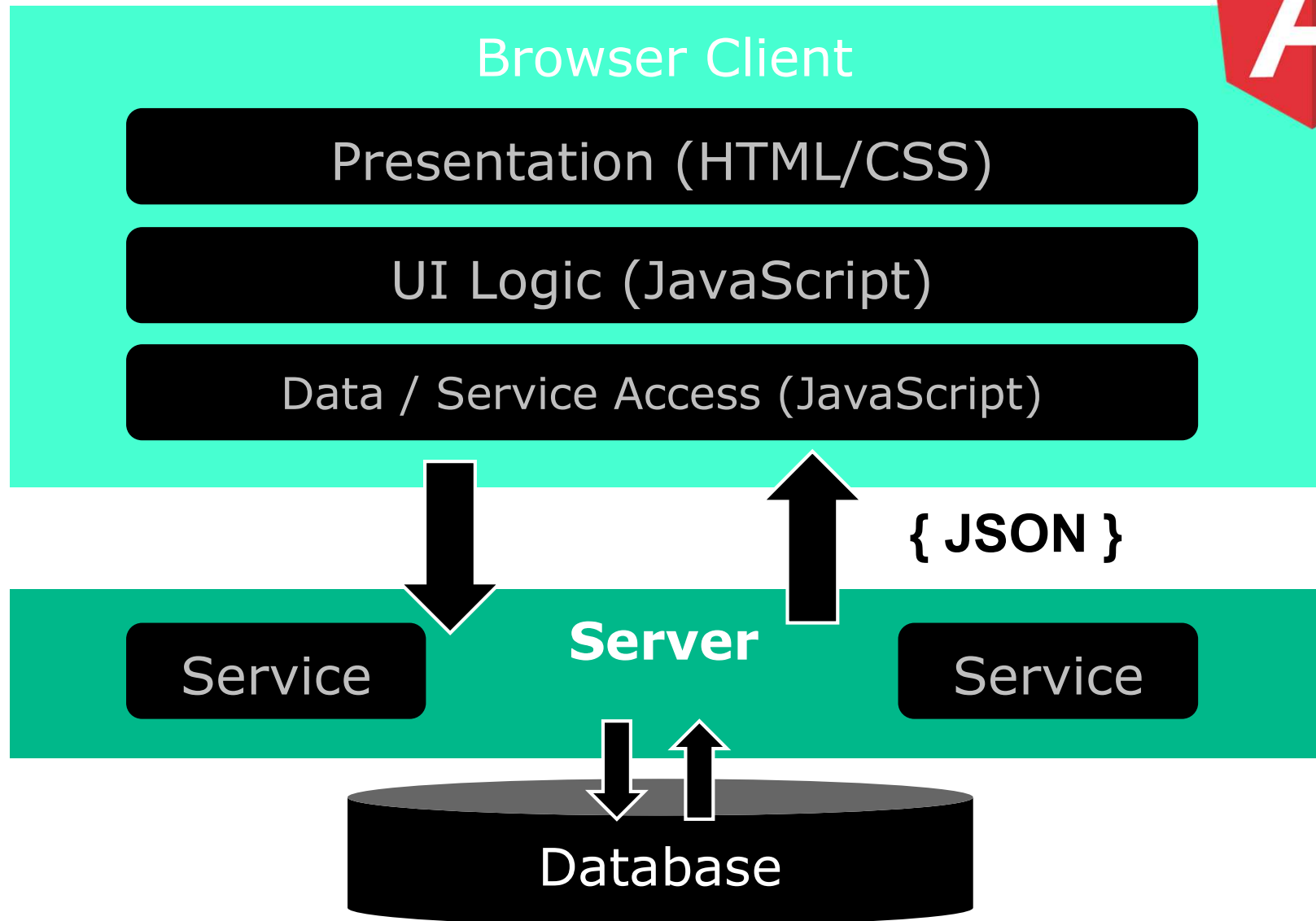
Platform Features

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



Single Page Application

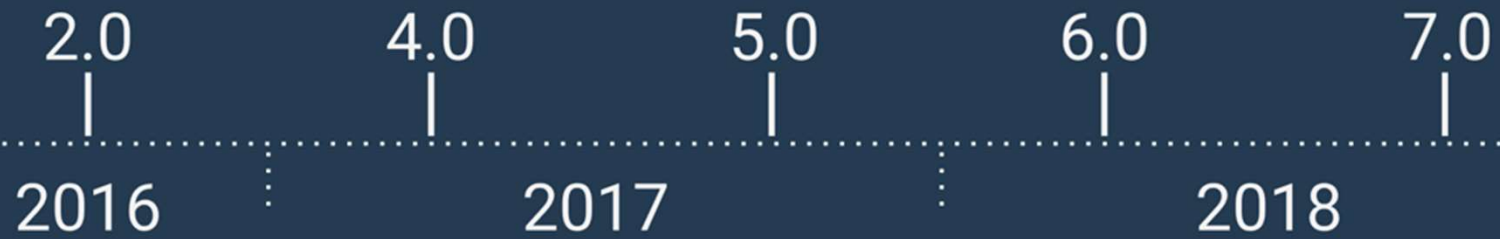
2010 – 20??





Time-based Releases

@IgorMinar, Nov. 9, 2018



Predictable & Continuous
Evolution

Angular updates

- Angular 2 – 14 **Grotendeels 'meer van hetzelfde'**
 - Soms zijn er breaking changes
 - Zelf checken: <https://update.angular.io/>
 - Updates van verwante libraries (RxJS, Angular Material)
- Belangrijke milestones
 - Angular 2 – breekt met structuur en architectuur van Angular JS (1.x)
 - Angular 6 – Introductie modern Angular CLI
 - Angular 9 – Ivy Compiler
 - Angular 14 – Standalone components



<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

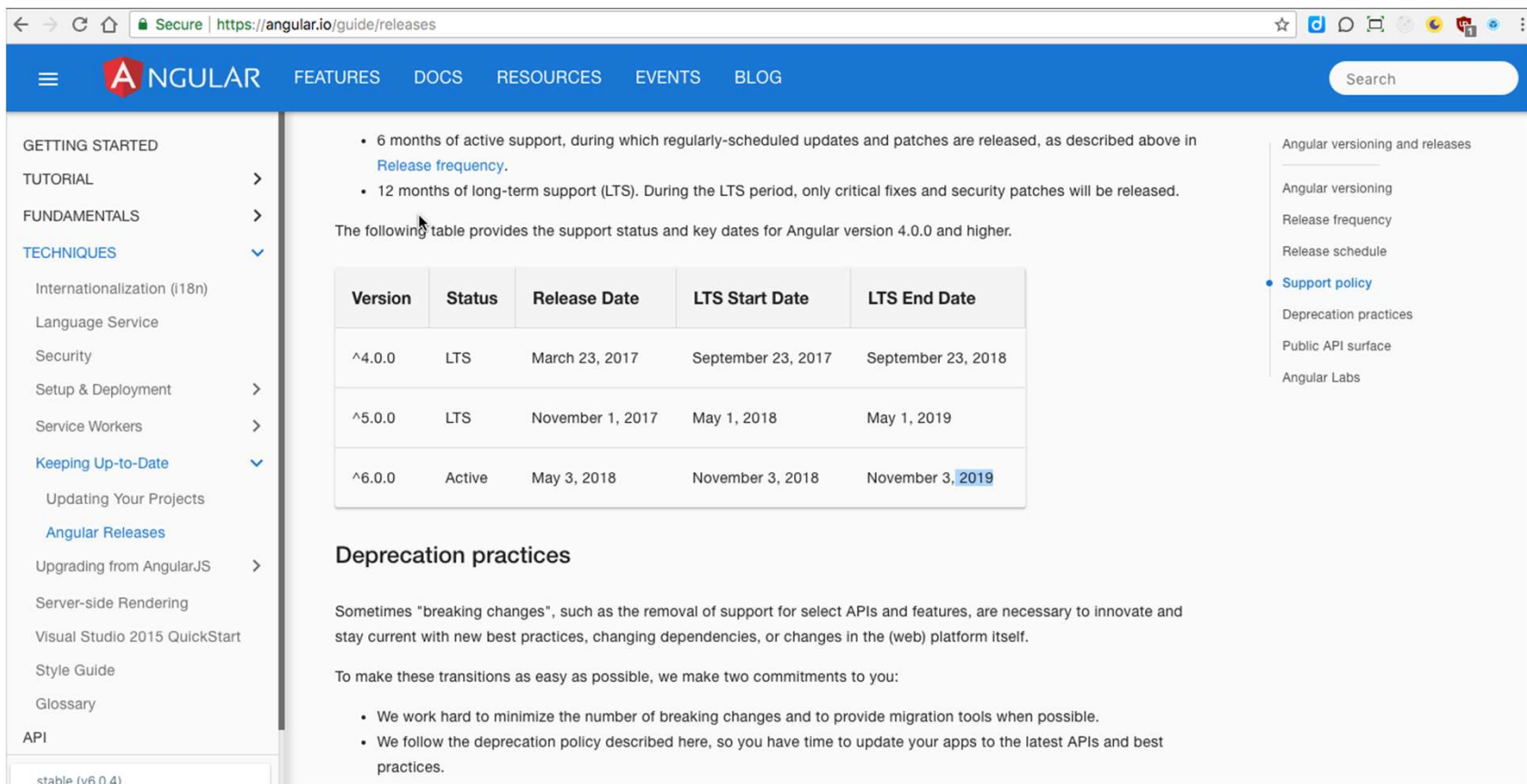
yarn

Show me how to update!

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

Angular Versions - Long Time Support

→ <https://angular.io/guide/releases>



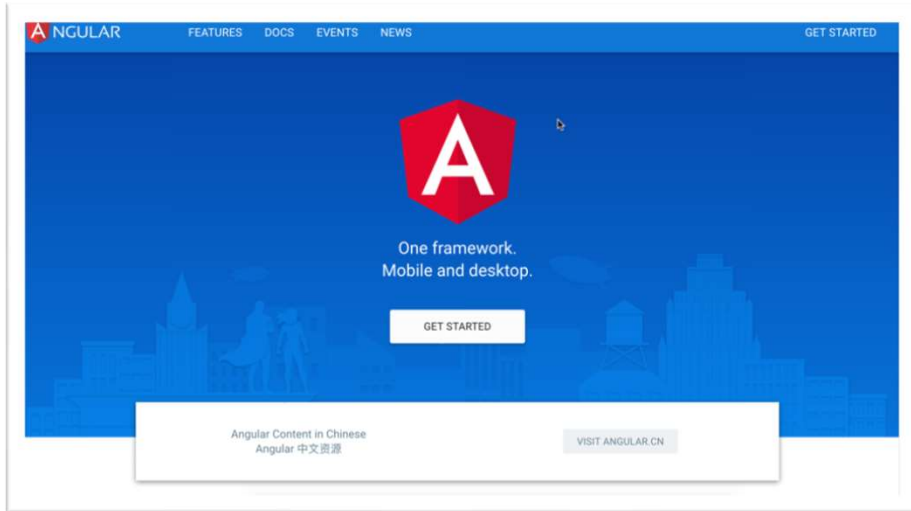
The screenshot shows the Angular.io website's 'Releases' page. The page is titled 'Angular Versions - Long Time Support'. It features a navigation menu on the left with categories like 'GETTING STARTED', 'TUTORIAL', 'FUNDAMENTALS', 'TECHNIQUES', and 'API'. The main content area is titled 'Angular versioning and releases' and contains a table of release dates. The table has columns for 'Version', 'Status', 'Release Date', 'LTS Start Date', and 'LTS End Date'. The table lists three versions: ^4.0.0 (LTS, March 23, 2017 to September 23, 2018), ^5.0.0 (LTS, November 1, 2017 to May 1, 2019), and ^6.0.0 (Active, May 3, 2018 to November 3, 2019). The page also includes a sidebar on the right with links to 'Angular versioning and releases', '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

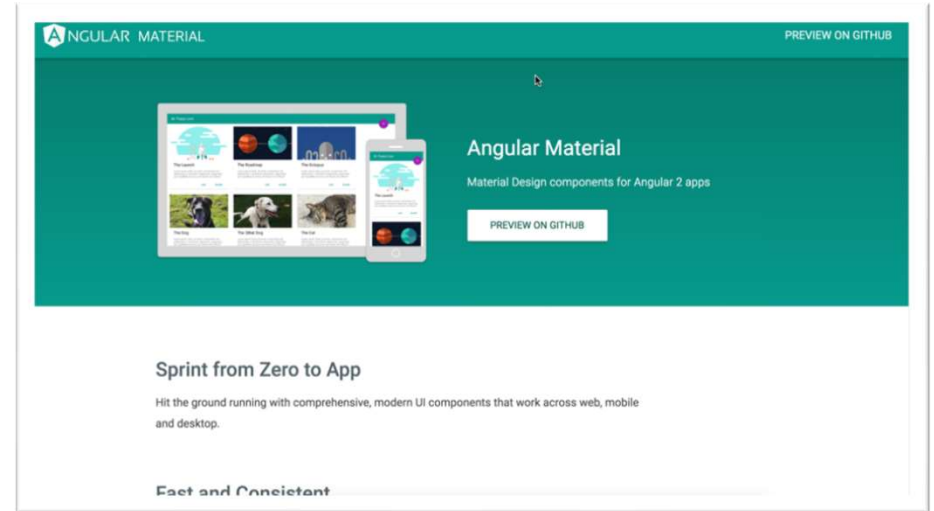
“It's just

Angular”

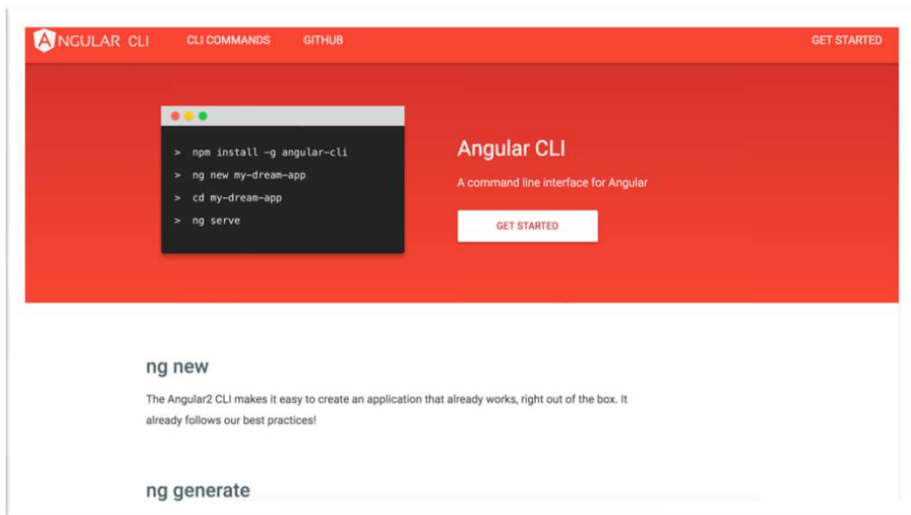
Angular as a Platform



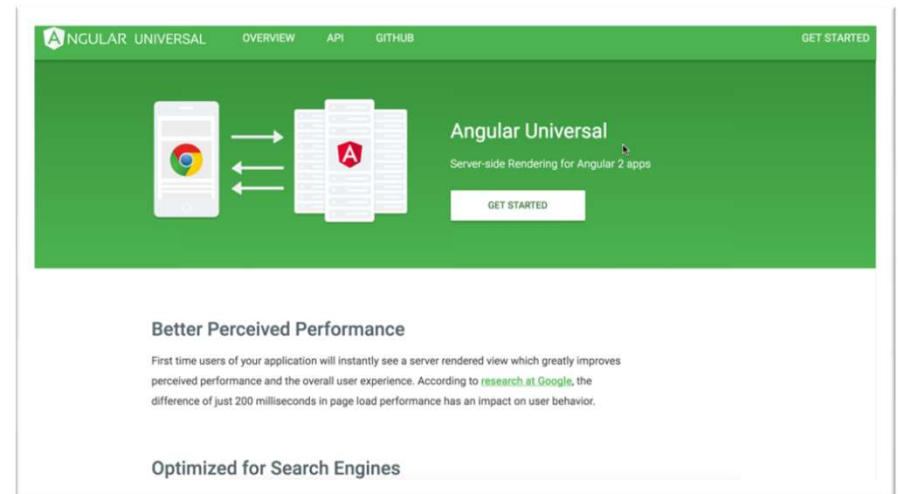
<https://angular.io/>



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

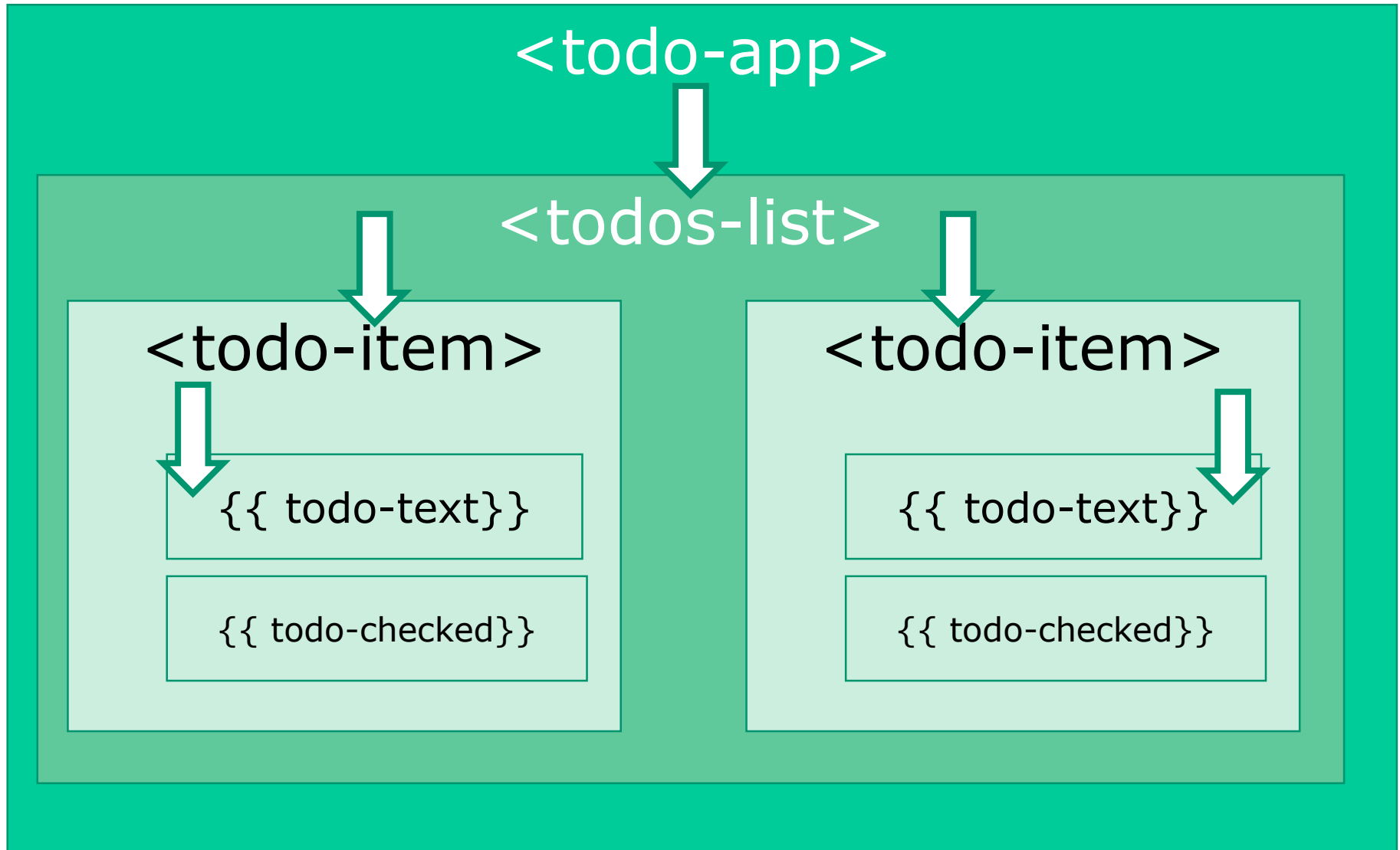


<https://cli.angular.io/>



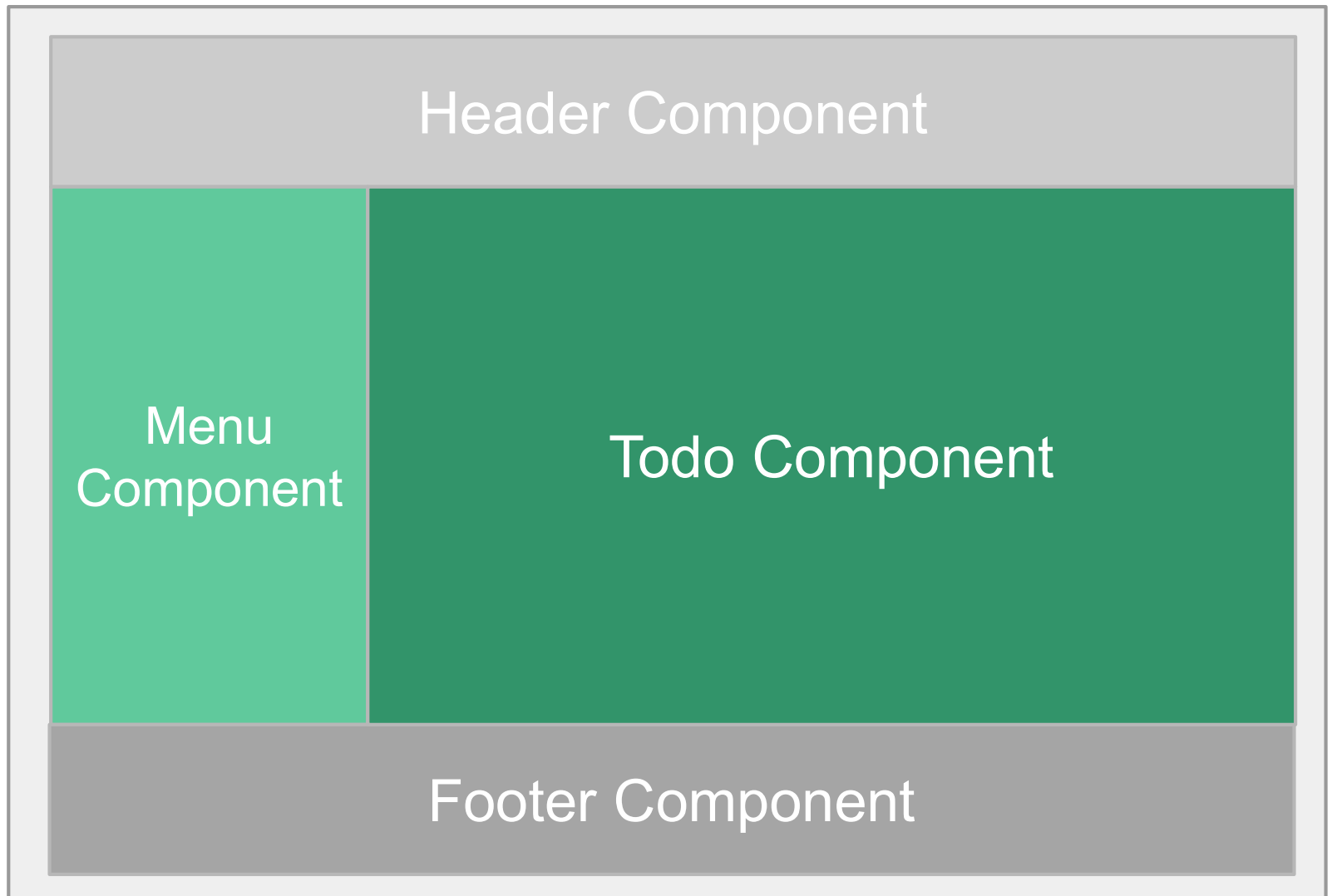
<https://universal.angular.io/>

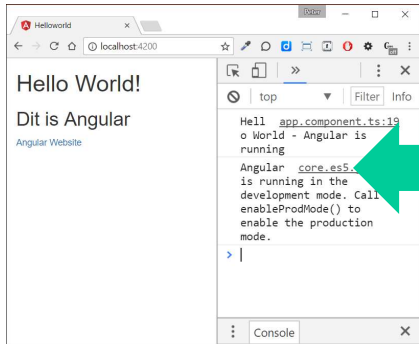
Angular 2 - components



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

Components – visual representation





main.ts / bootstrapper

ngModule / root module

AppComponent

Services

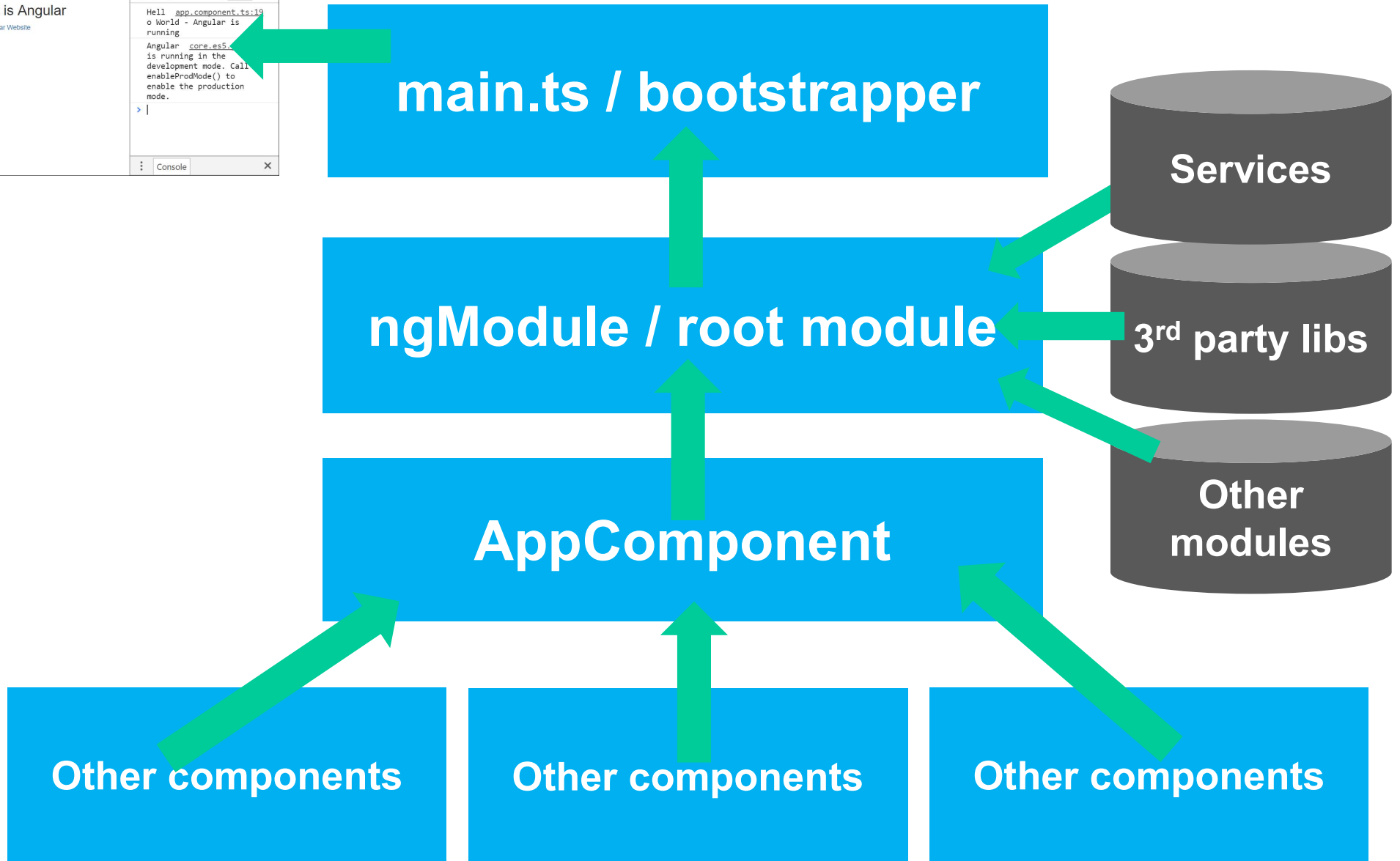
3rd party libs

Other modules

Other components

Other components

Other components

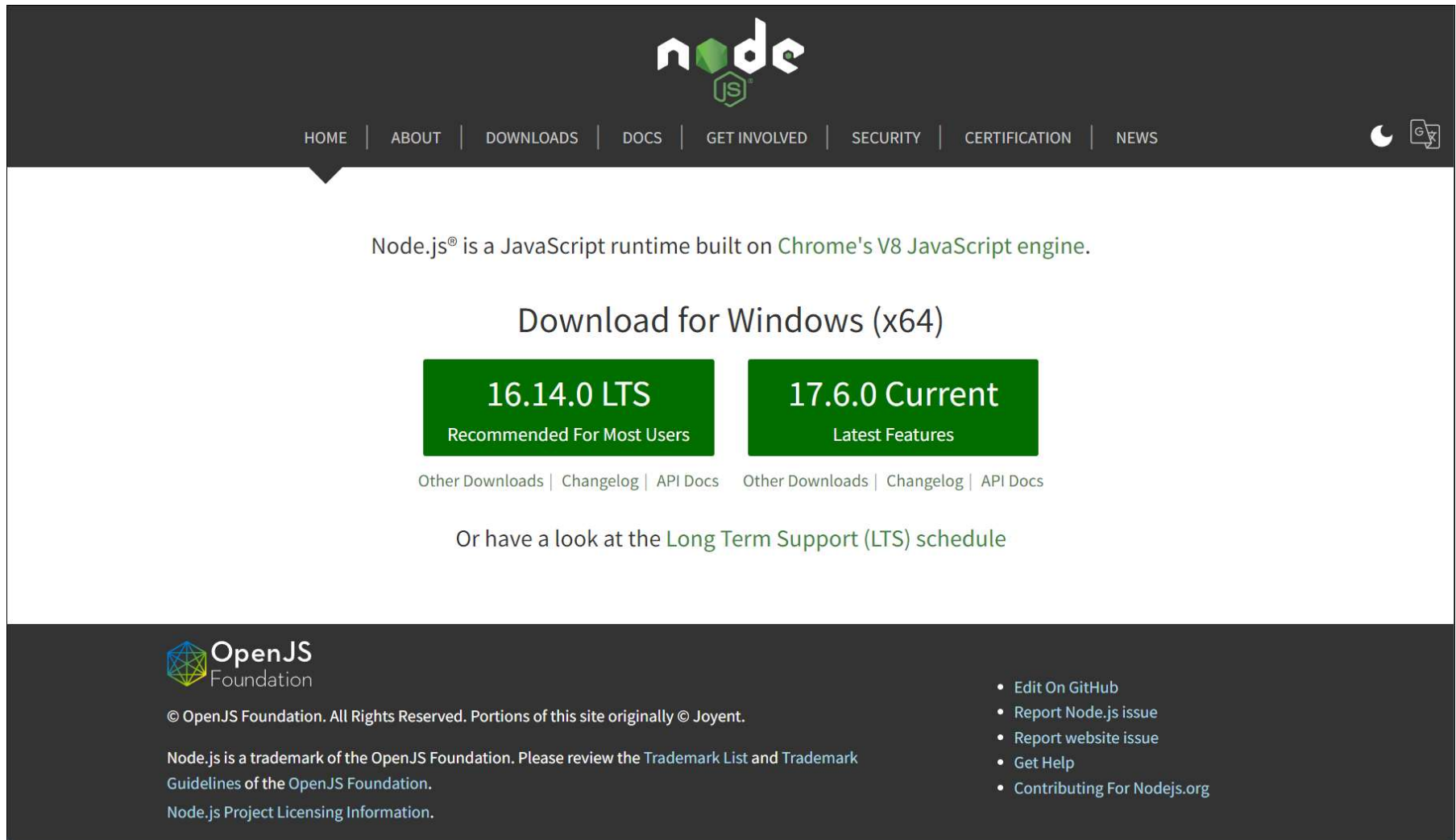




Let's write some code

Hello World in Angular

Angular development dependency: NodeJS 12.20+



node

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

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

Download for Windows (x64)

16.14.0 LTS	17.6.0 Current
Recommended For Most Users	Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#)

OpenJS
Foundation

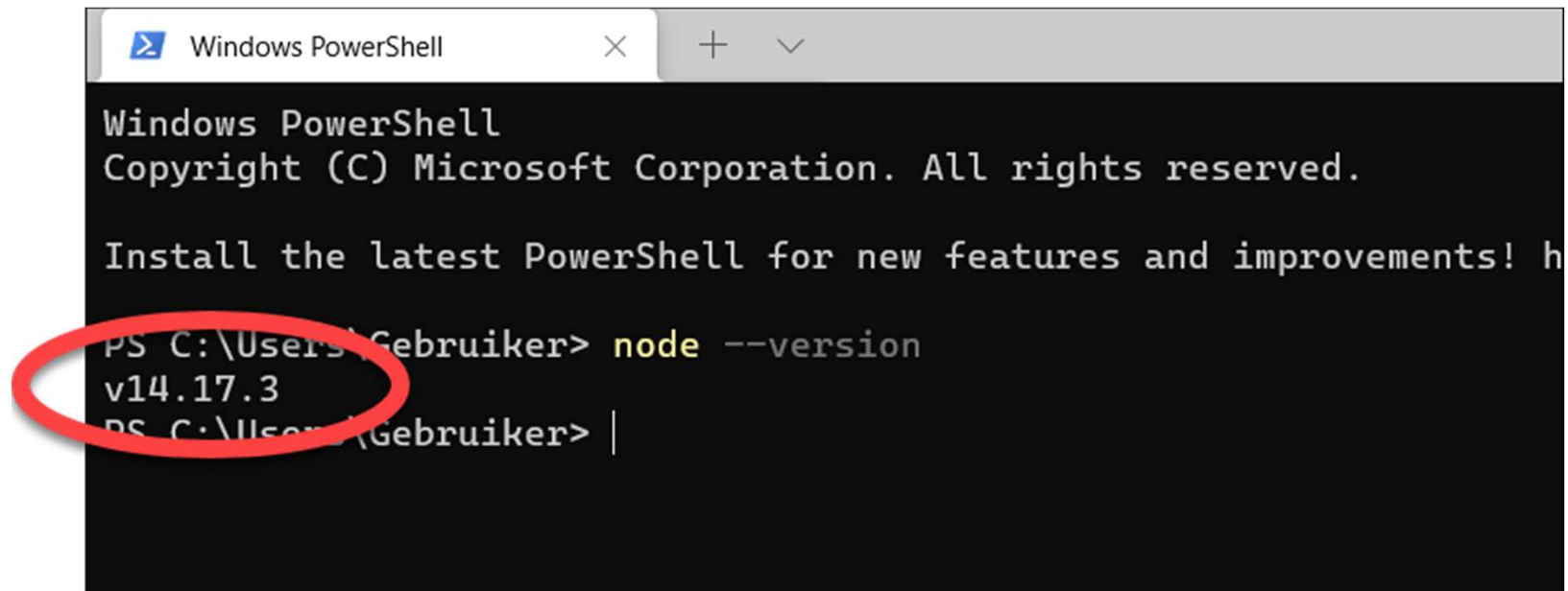
© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of the OpenJS Foundation. Please review the [Trademark List](#) and [Trademark Guidelines of the OpenJS Foundation](#).

[Node.js Project Licensing Information](#).

- [Edit On GitHub](#)
- [Report Node.js issue](#)
- [Report website issue](#)
- [Get Help](#)
- [Contributing For Nodejs.org](#)

Node – check your version



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PowerShellLatest

PS C:\Users\Gebruiker> node --version
v14.17.3
PS C:\Users\Gebruiker> |
```

Mini workshop

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

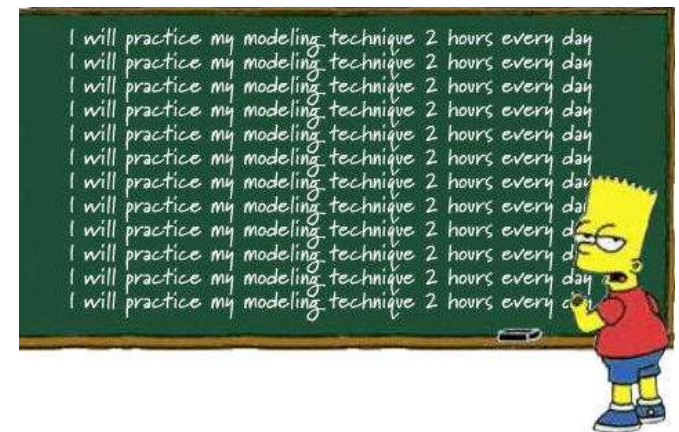
```
cd examples
```

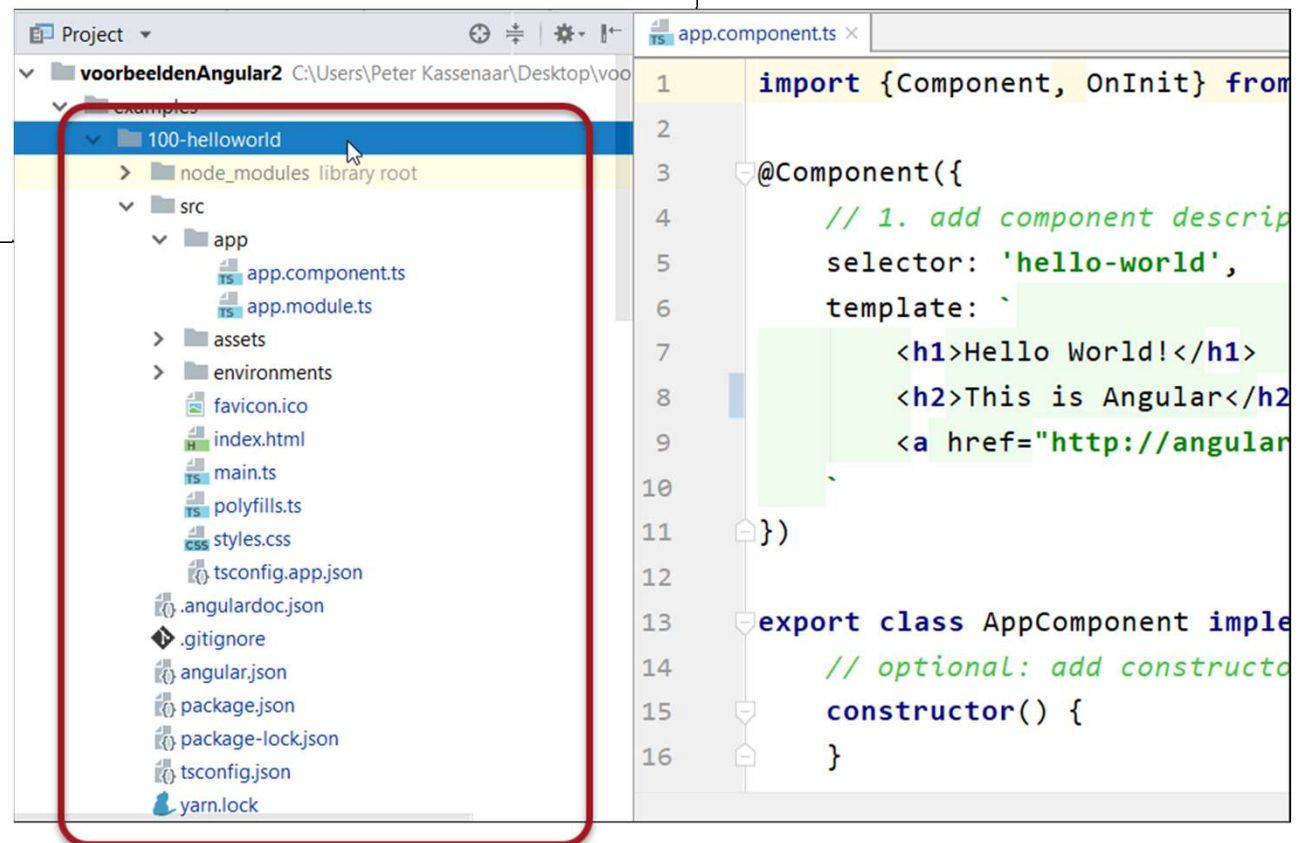
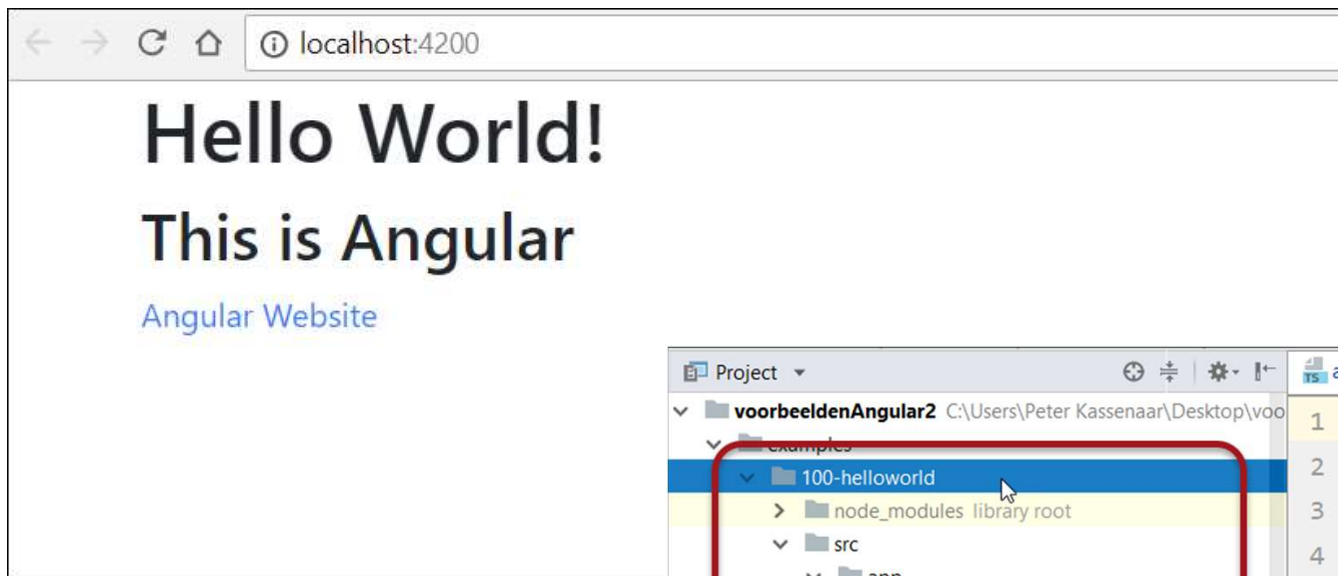
```
cd 100-helloworld
```

```
npm install
```

```
npm start
```

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





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 #1/3 - 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>"
}
```

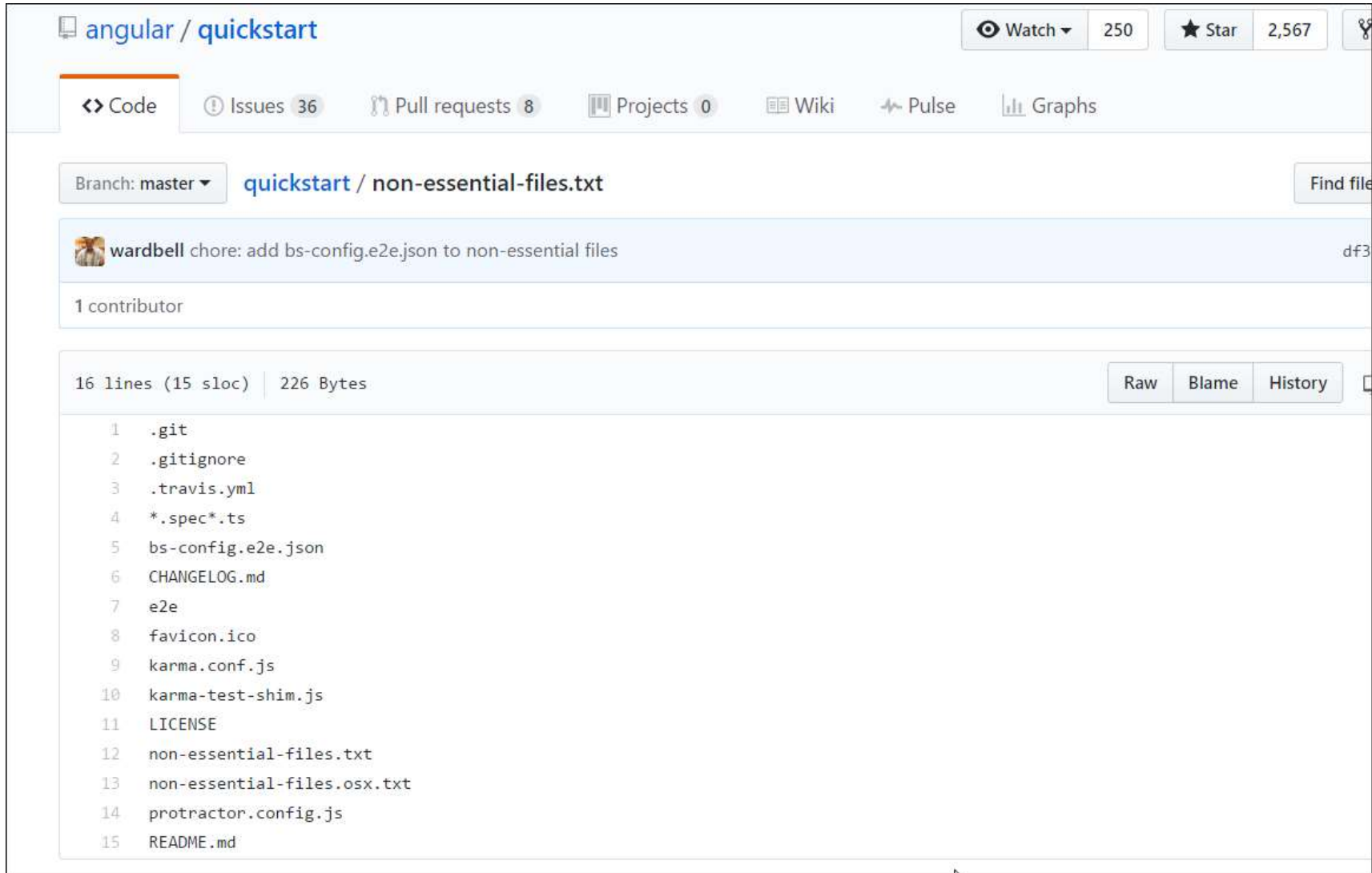
Boilerplate files #2/3 - tsconfig.json

```
{
  "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 #3/3 – 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 the GitHub interface for the 'angular/quickstart' repository. At the top, the repository name is 'angular / quickstart' with 250 watchers and 2,567 stars. Below this, navigation tabs include 'Code', 'Issues 36', 'Pull requests 8', 'Projects 0', 'Wiki', 'Pulse', and 'Graphs'. The current view is the 'Code' tab, showing the file 'quickstart / non-essential-files.txt' on the 'master' branch. A commit by 'wardbell' is shown with the message 'chore: add bs-config.e2e.json to non-essential files'. Below the commit, it says '1 contributor'. The file details show '16 lines (15 sloc)' and '226 Bytes'. On the right, there are buttons for 'Raw', 'Blame', and 'History'. The file content is a list of 15 files:

```
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



<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

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

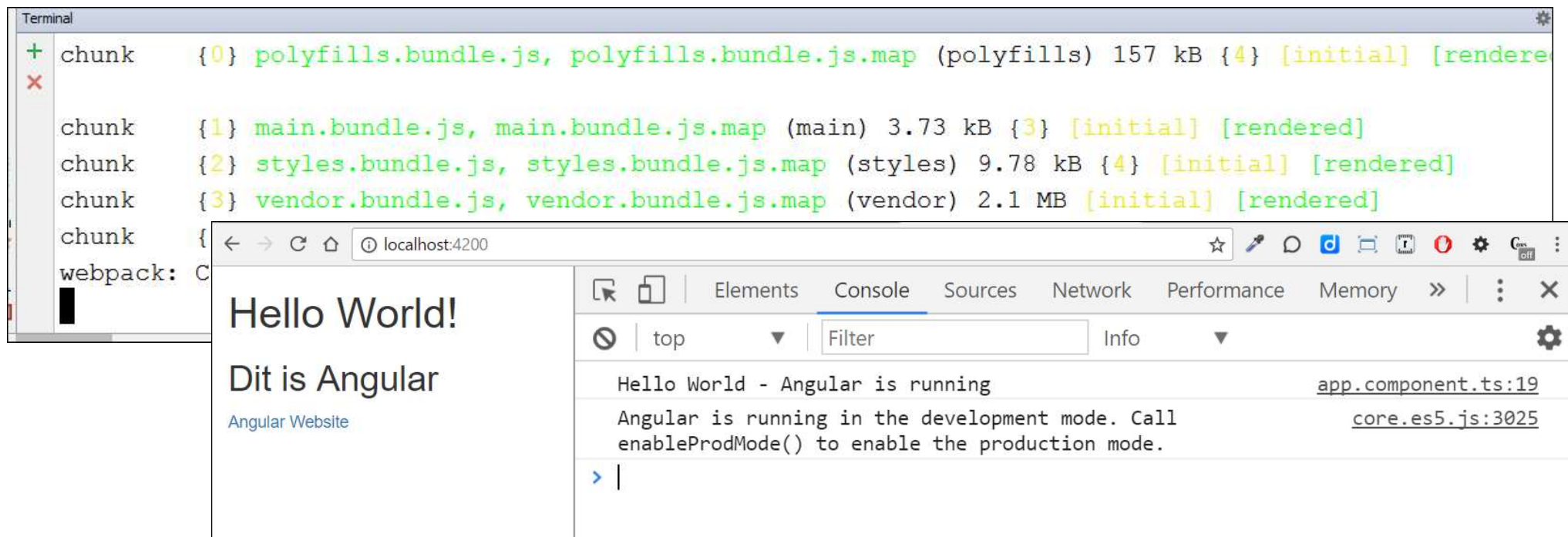
Verwijzing naar de root-component:

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

App draaien

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

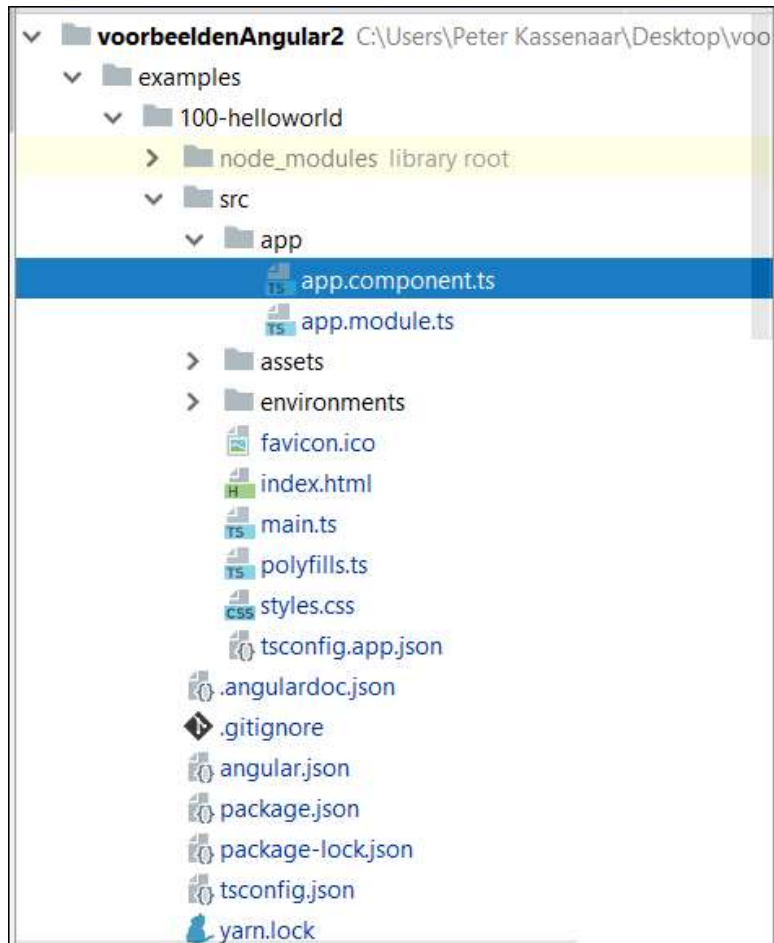
`ng serve` – start globale angular-cli instantie

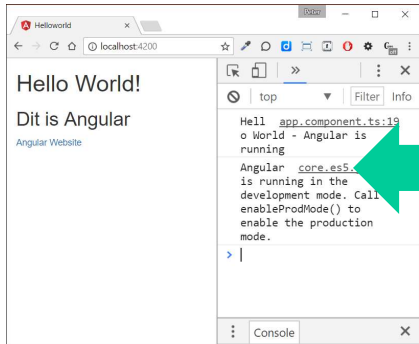


Daarna: wijzigingen aanbrengen in `app.component.ts`

– worden opgepikt door Live Reload

Basic Project Structure





main.ts / bootstrapper

ngModule / root module

AppComponent

Services

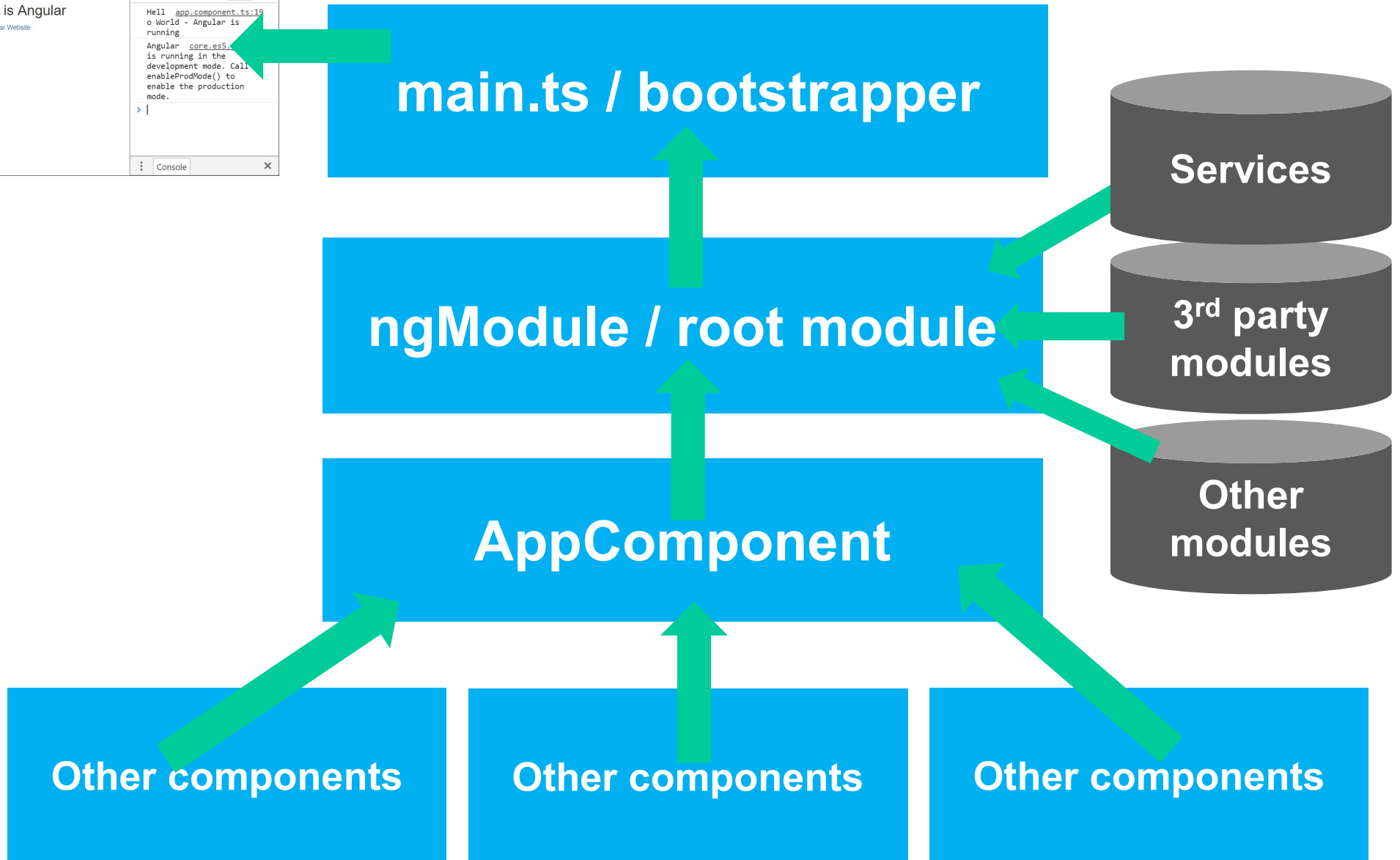
**3rd party
modules**

**Other
modules**

Other components

Other components

Other components



Assets

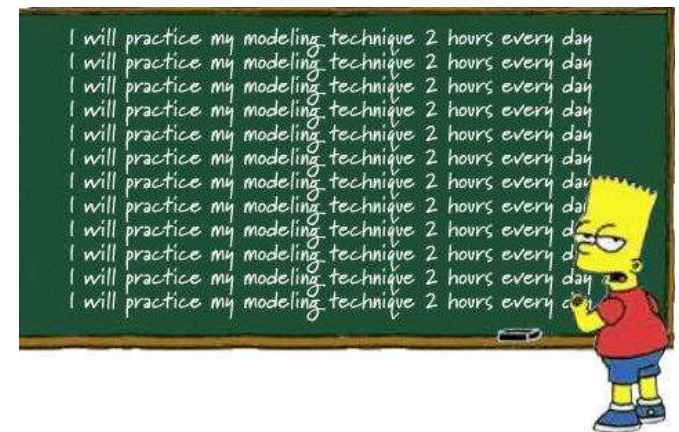
github.com/PeterKassenaar/voorbeeldenAngular2

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)

Workshop....





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/>

ANGULAR

FEATURESDOCSRESOURCESEVENTSBLOG

Search



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

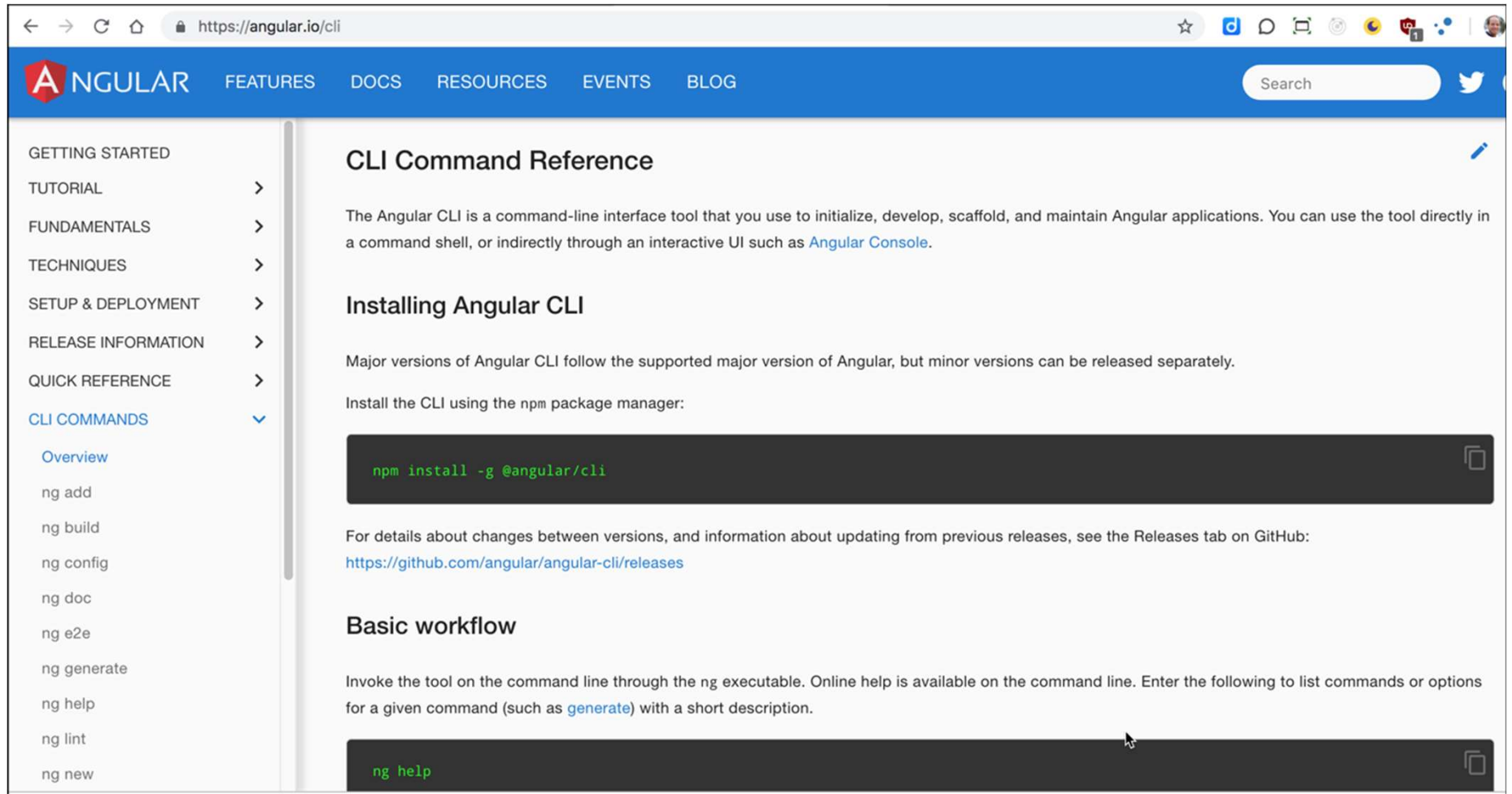
```
npm install -g @angular/cli
```


Achtergrondinformatie – Mike Brocchi



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

Documentatie - in de Angular Docs



The screenshot shows the Angular CLI documentation page on the Angular website. The browser address bar displays `https://angular.io/cli`. The page features a blue header with the Angular logo and navigation links: FEATURES, DOCS, RESOURCES, EVENTS, and BLOG. A search bar is located on the right side of the header. On the left, a sidebar lists various documentation sections, with 'CLI COMMANDS' expanded and 'Overview' selected. The main content area is titled 'CLI Command Reference' and includes an introduction to the Angular CLI, a section on 'Installing Angular CLI' with a code block for `npm install -g @angular/cli`, and a 'Basic workflow' section with a code block for `ng help`. The page also includes a link to the GitHub releases page for more details on updates.

← → ↻ 🏠 `https://angular.io/cli` ☆ [Icons] | [Profile]

ANGULAR FEATURES DOCS RESOURCES EVENTS BLOG Search [Twitter]

GETTING STARTED
TUTORIAL >
FUNDAMENTALS >
TECHNIQUES >
SETUP & DEPLOYMENT >
RELEASE INFORMATION >
QUICK REFERENCE >
CLI COMMANDS ▾
 Overview
 ng add
 ng build
 ng config
 ng doc
 ng e2e
 ng generate
 ng help
 ng lint
 ng new

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>



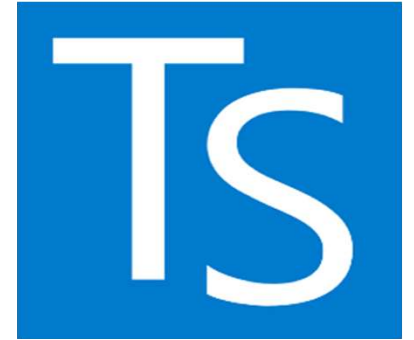
Angular 2 Code - Backend

Kort over TypeScript en ES6

Programmeertalen

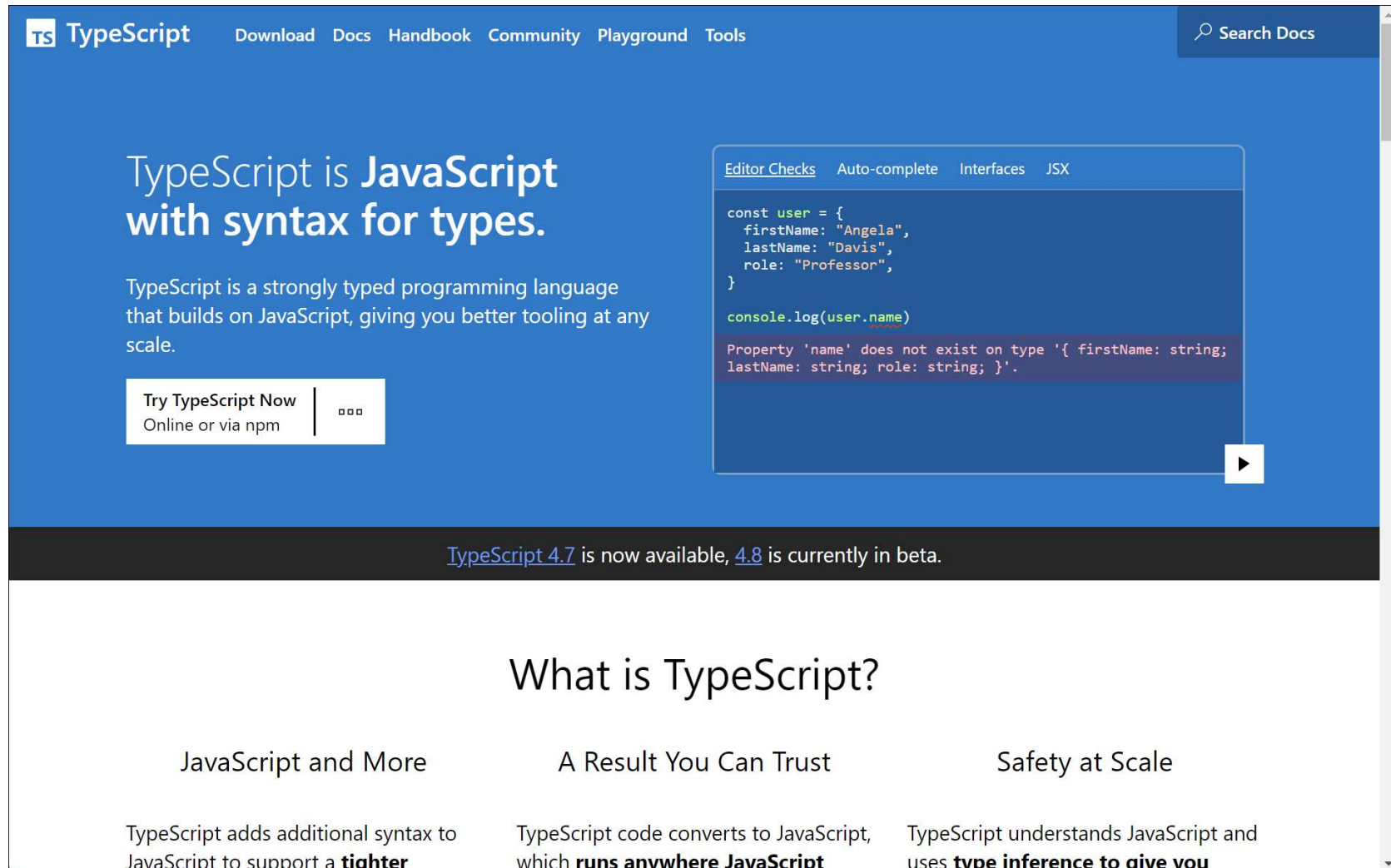


TypeScript



- **Beperkt** in deze module
- Voor zover nodig om Angular te gebruiken.
- Maar, **TypeScript is niet beperkt tot Angular!**
- Algemene **superset** van JavaScript
- Compileert plain JavaScript, dat browsers begrijpen
- Twee taken
 - 1. **Compiling/Transpiling**
 - 2. **Type Safety**

TypeScript website



The screenshot shows the TypeScript website homepage. The header is blue with the TypeScript logo and navigation links: Download, Docs, Handbook, Community, Playground, Tools. A search bar is in the top right. The main content area has a blue background with the text 'TypeScript is JavaScript with syntax for types.' and a description: 'TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.' Below this is a button 'Try TypeScript Now' with a subtext 'Online or via npm'. To the right is a code editor snippet showing a TypeScript object and a console log with a red error message: 'Property 'name' does not exist on type '{ firstName: string; lastName: string; role: string; }'.' Below the main content is a dark blue banner with the text 'TypeScript 4.7 is now available, 4.8 is currently in beta.' The footer is white and contains the heading 'What is TypeScript?' followed by three columns of text: 'JavaScript and More', 'A Result You Can Trust', and 'Safety at Scale'.

TS TypeScript Download Docs Handbook Community Playground Tools Search Docs

TypeScript is **JavaScript** with syntax for types.

TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.

Try TypeScript Now
Online or via npm

```
const user = {
  firstName: "Angela",
  lastName: "Davis",
  role: "Professor",
}

console.log(user.name)
```

Property 'name' does not exist on type '{ firstName: string; lastName: string; role: string; }'.

TypeScript 4.7 is now available, 4.8 is currently in beta.

What is TypeScript?

JavaScript and More	A Result You Can Trust	Safety at Scale
TypeScript adds additional syntax to JavaScript to support a tighter	TypeScript code converts to JavaScript, which runs anywhere JavaScript	TypeScript understands JavaScript and uses type inference to give you

www.typescriptlang.org



A Venn diagram consisting of three concentric circles. The outermost circle is dark teal and contains the text 'TypeScript'. Inside it is a medium teal circle containing the text 'ES6'. Inside the 'ES6' circle is a light teal circle containing the text 'ES5'. This visualizes that ES5 is a subset of ES6, and ES6 is a subset of TypeScript.

TypeScript

ES6

ES5

ES6 en TypeScript

De toekomst van JavaScript is ES6/ES2015

Major update van JavaScript als programmeertaal

Modules, classes en meer

Verplicht in Angular, optioneel in JavaScript, Vue, React, ...

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