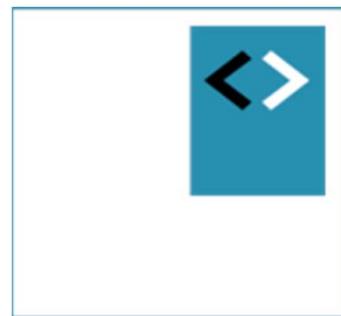




Angular Fundamentals

Module 1 – Core

Lendex powered by  NIBC



Peter Kassenaar
info@kassenaar.com

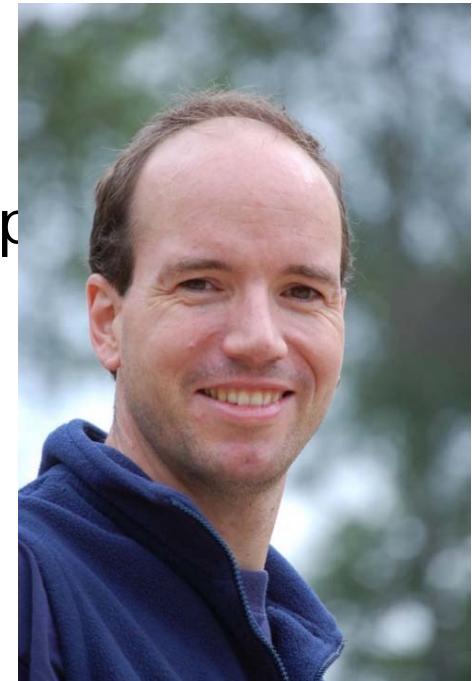
Peter Kassenaar

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

www.kassenaar.com/blog

info@kassenaar.com

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





Zutphen, NL

Angulartraining.nl
2018 dates now available!



Angulartraining.nl

Home Training Dates Information Contact

```
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/lendex

The screenshot shows the GitHub repository page for `PeterKassenaar / lendex`. The repository is public and has 1 star and 0 forks. It contains 1 branch and 0 tags. The main file listed is `README.md`, which contains the following content:

```
lendex
Slides and example code on the training Angular - October 2021
```

The repository has 1 commit by `PeterKassenaar` from 3 minutes ago. The commit details are:

- `Initial commit` (file: `.gitignore`)
- `Initial commit` (file: `README.md`)

The repository page also includes sections for About, Releases, and Packages, both of which currently have no content.

Footer links include: © 2021 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

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 3-4 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

18,19 – 21,22 oktober 2021 – Mo.-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

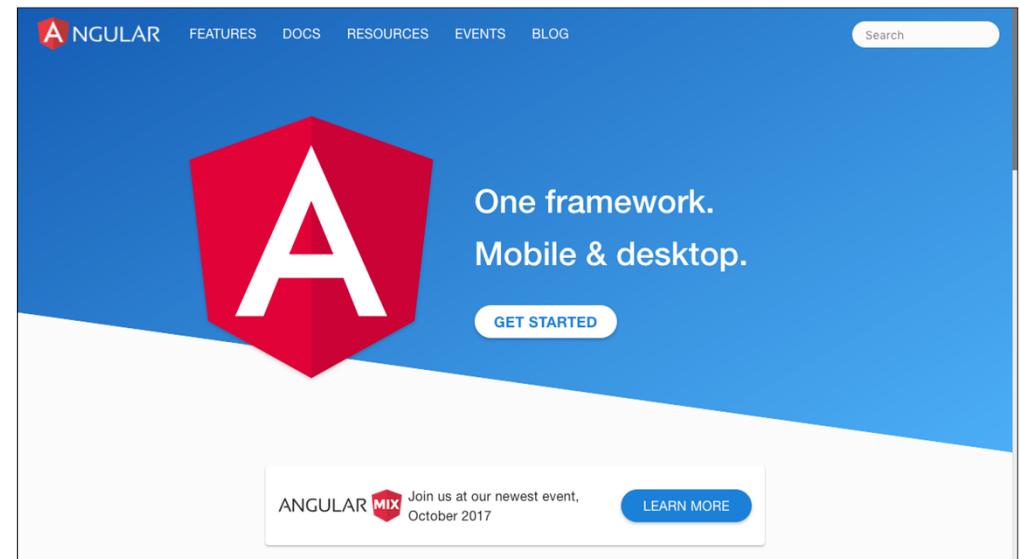
Friday: probably wrap up a little bit early

Agenda - Fundamentals, 2 days

- Introduction & short history – Why Angular?
- Key features of Angular 2 to 11
- 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?



AngularJS 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?

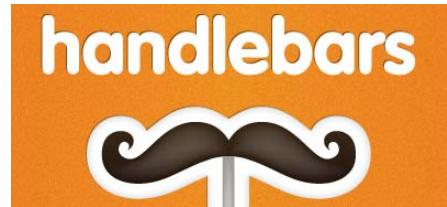
Rethorical question:

**“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

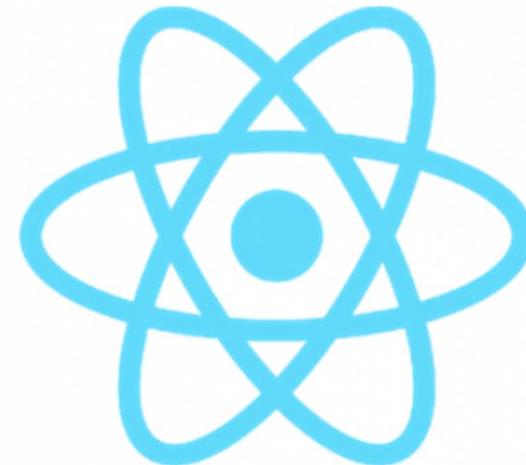


...

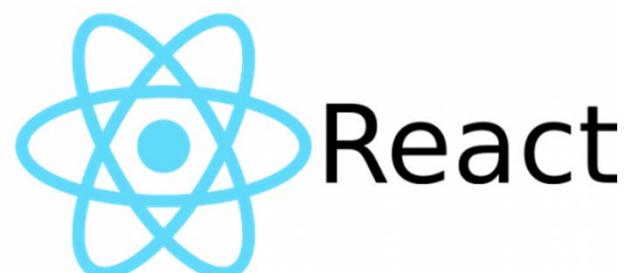
“The Frankenstein Framework”



Front-end Frameworks – the big four



Similarities

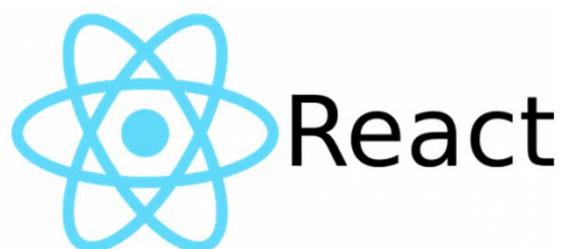


- Creating Single Page Applications
- Based on components
- Data binding, props, events, routing, state management, ...
- Huge ecosystem
- Huge community
- High adaptation rate

Differences (apart from syntax)



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

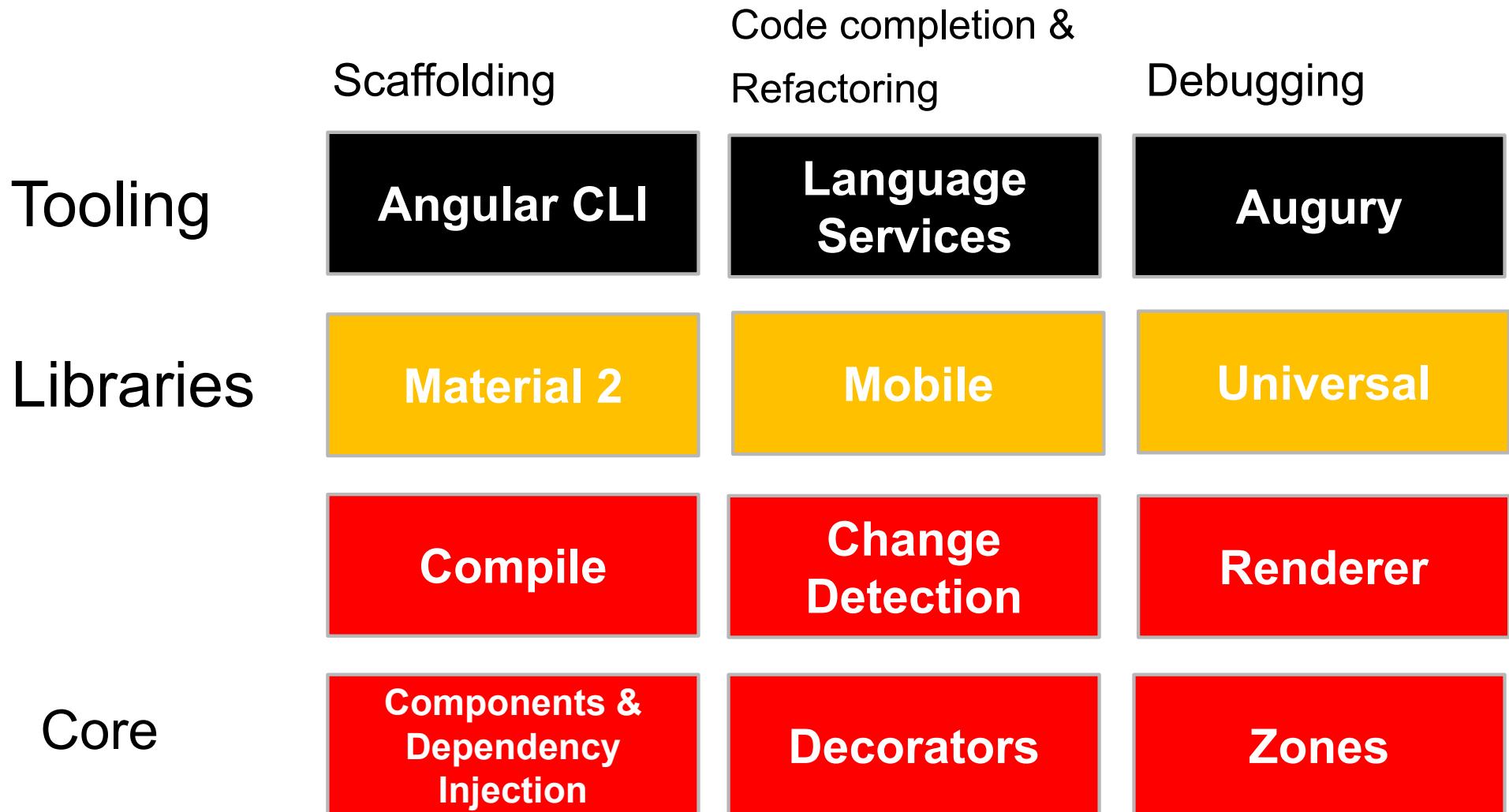


- Point of departure: **JavaScript**, JavaScript, JavaScript (JSX)
- Build-all-yourself / choice anxiety



Platform

Framework to Platform



Z he#216 R P

Vhuvh#2#
Dqjxodu#
Xqlyhuvo

R iilfh#
Dgg0lo

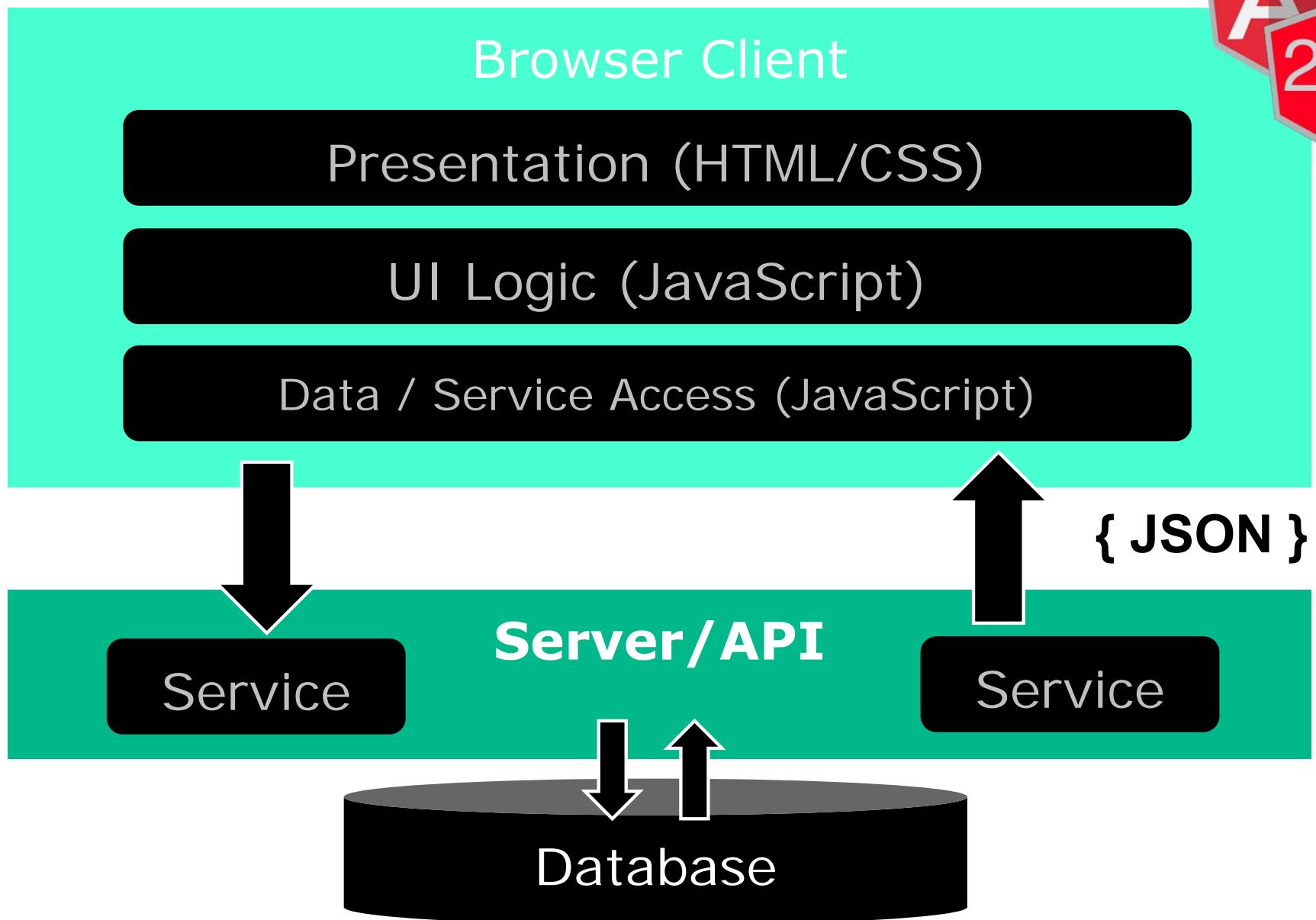
Dqjxodu#
P relh

Q dwlyh#
Uhqghulqj

j dh#
arp h#
whqvlrqv

Single Page Application

2010 – 20??



On Versioning Numbering



Planned – six months major release cycle





- Predictability
- Painless Updates
- Long Term Support

Angular Versions and -Long Time Support

→ <https://angular.io/guide/releases>

The screenshot shows the Angular website's navigation bar with links for FEATURES, DOCS, RESOURCES, EVENTS, and BLOG. Below the navigation is a sidebar with links to INTRODUCTION, GETTING STARTED, FUNDAMENTALS, TECHNIQUES, DEV WORKFLOW, CONFIGURATION, EXTENDING ANGULAR, TUTORIALS, and RELEASE INFORMATION. The RELEASE INFORMATION section is expanded, showing sub-links for Keeping Up-to-Date, Release Practices, Updating to Version 10, Deprecations, Angular Ivy, Upgrading from AngularJS, ANGULAR STYLE AND USAGE, CLI COMMAND REFERENCE, and API REFERENCE. The main content area discusses release cycles, support policies, and a table of supported Angular versions.

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

Angular versioning and releases

- Angular versioning
- Supported update paths
- Preview releases
- Release frequency**
- Support policy and schedule
- Deprecation practices
- Public API surface
- Angular Labs

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

The screenshot shows the Angular Update Guide interface. At the top, there's a blue header bar with the title "Angular Update Guide" and two icons on the right. Below the header, a large white form area contains the following fields:

- Select the options matching your project:**
- Angular Versions**: Two dropdown menus labeled "From: 8.0" and "To: 10.0".
- Warning:** We do not recommend moving across multiple major versions.
- App Complexity**: Three buttons: "Basic" (highlighted in grey), "Medium", and "Advanced".
- We'll show update information relevant to all Angular developers.**
- Other Dependencies**: Two checkboxes:
 - I use ngUpgrade to combine AngularJS & Angular
 - I use Angular Material
- Show me how to update!**: A blue button at the bottom of the form.

At the bottom of the page, there are two sections:

- Angular Update Guide | 8.0 → 10.0 for Basic Apps**
- Before Updating**

"It's just

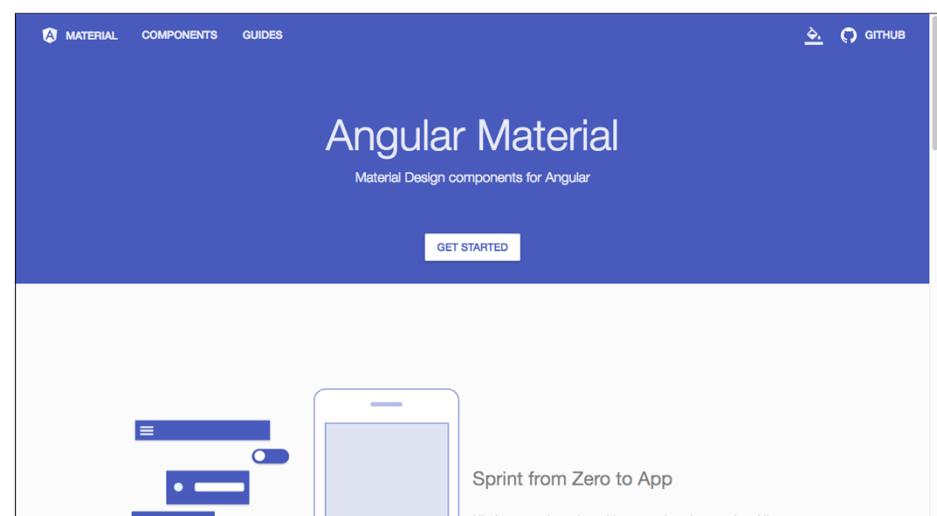
Angular

Angular as a Platform



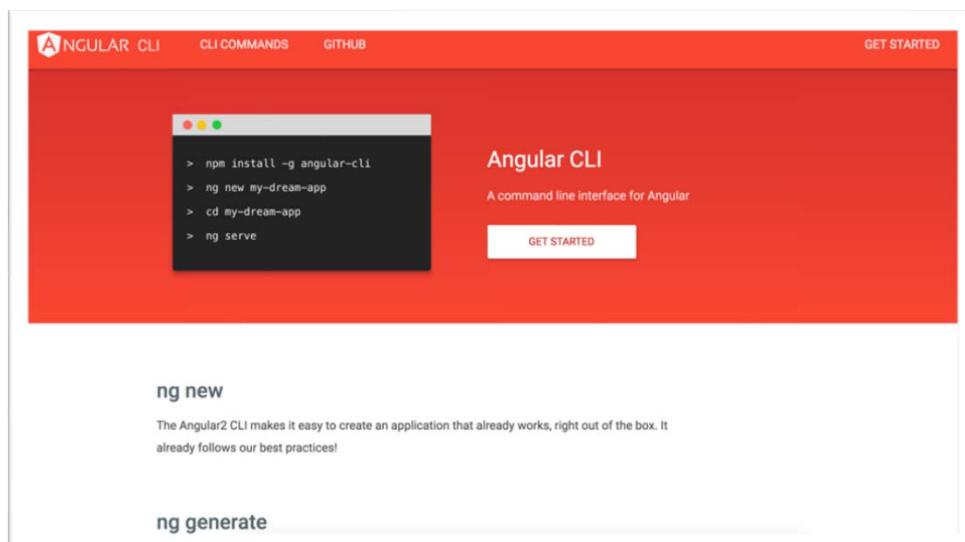
The Angular homepage features a large red hexagonal logo with a white 'A' on the left. To its right, the text 'One framework. Mobile & desktop.' is displayed. Below this is a 'GET STARTED' button. At the bottom left, there's a 'ANGULAR MIX' event announcement: 'Join us at our newest event, October 2017'. To the right of the event info is a 'LEARN MORE' button.

<https://angular.io/>



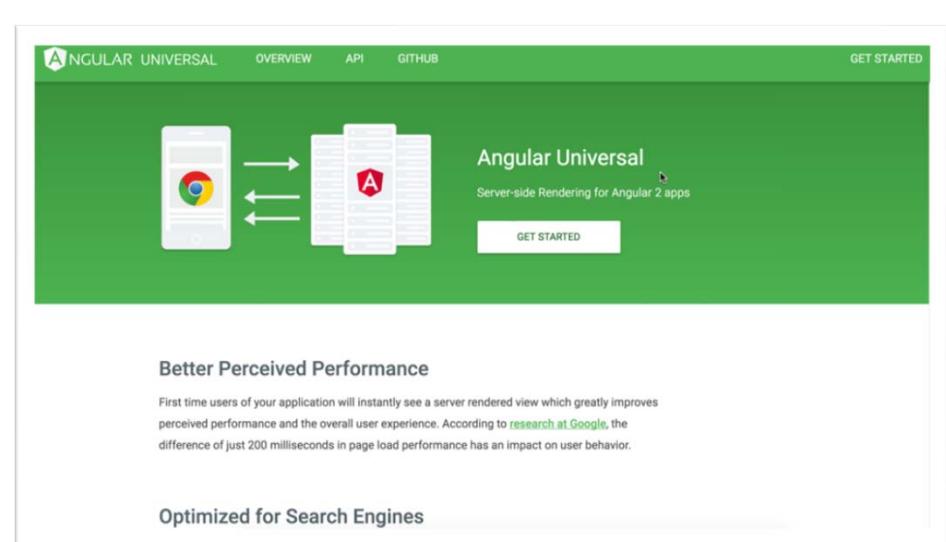
The Angular Material homepage has a blue header with navigation links: MATERIAL, COMPONENTS, and GUIDES. On the right, there are icons for GitHub and a user profile. The main title is 'Angular Material' with the subtitle 'Material Design components for Angular'. A 'GET STARTED' button is located in the top right. In the center, there's a graphic showing a smartphone and a laptop, with the text 'Sprint from Zero to App'. Below the graphic, it says 'Hit the ground running with comprehensive, modern UI'.

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



The Angular CLI homepage has a red header with links for 'CLI COMMANDS' and 'GITHUB'. It features a 'GET STARTED' button. Below the header is a terminal window showing command-line instructions: 'npm install -g angular-cli', 'ng new my-dream-app', 'cd my-dream-app', and 'ng serve'. The main content area contains sections for 'ng new' (describing the CLI's ability to create applications) and 'ng generate' (describing the CLI's ability to generate code). There's also a 'GET STARTED' button.

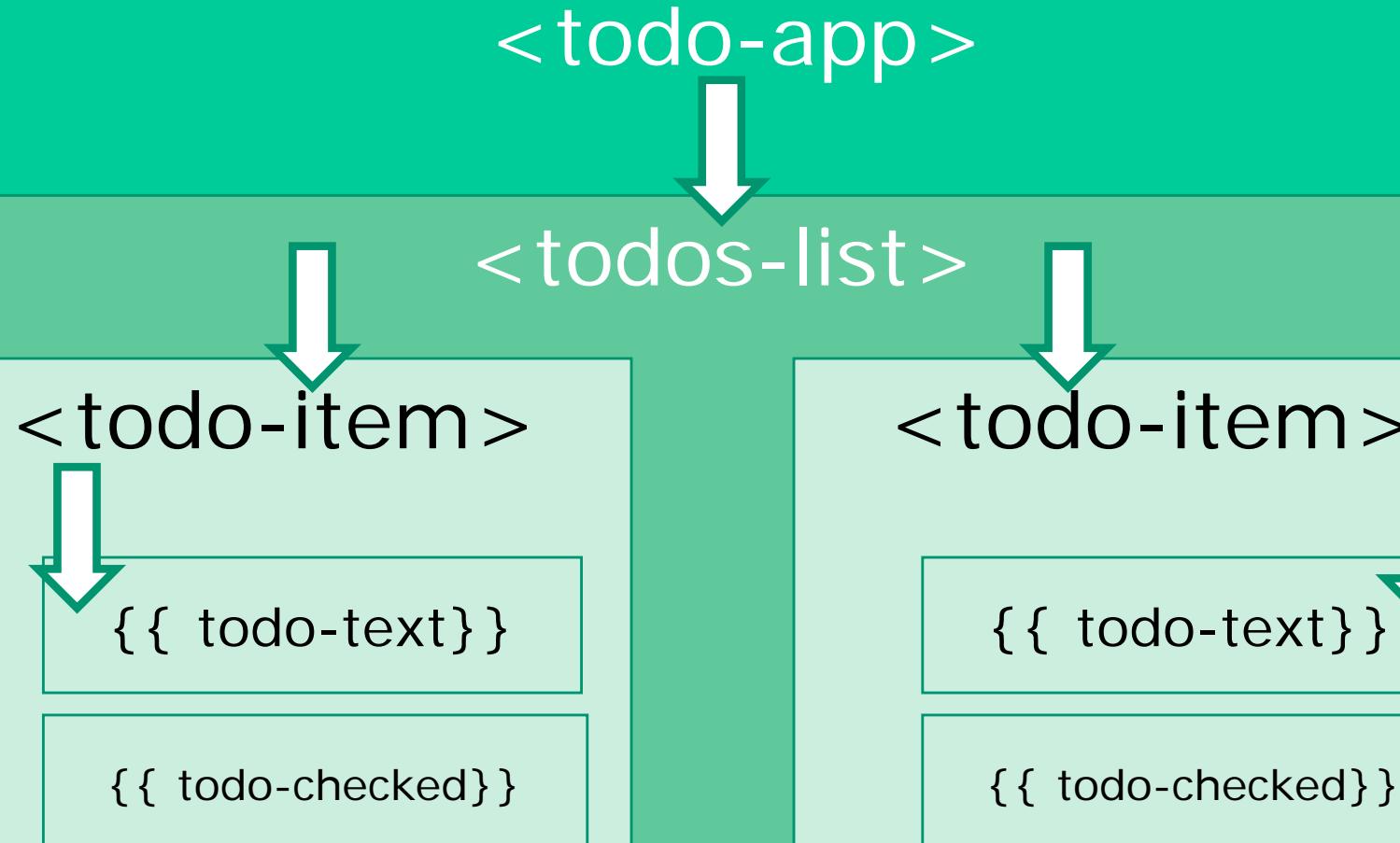
<https://cli.angular.io/>



The Angular Universal homepage has a green header with links for 'OVERVIEW', 'API', and 'GITHUB'. It features a 'GET STARTED' button. The central part of the page shows a diagram where a browser icon is connected to a server icon containing an 'A'. The text 'Angular Universal' and 'Server-side Rendering for Angular 2 apps' is displayed. Below this, there's a section titled 'Better Perceived Performance' with a detailed explanation. At the bottom, there's a 'GET STARTED' button and a section titled 'Optimized for Search Engines'.

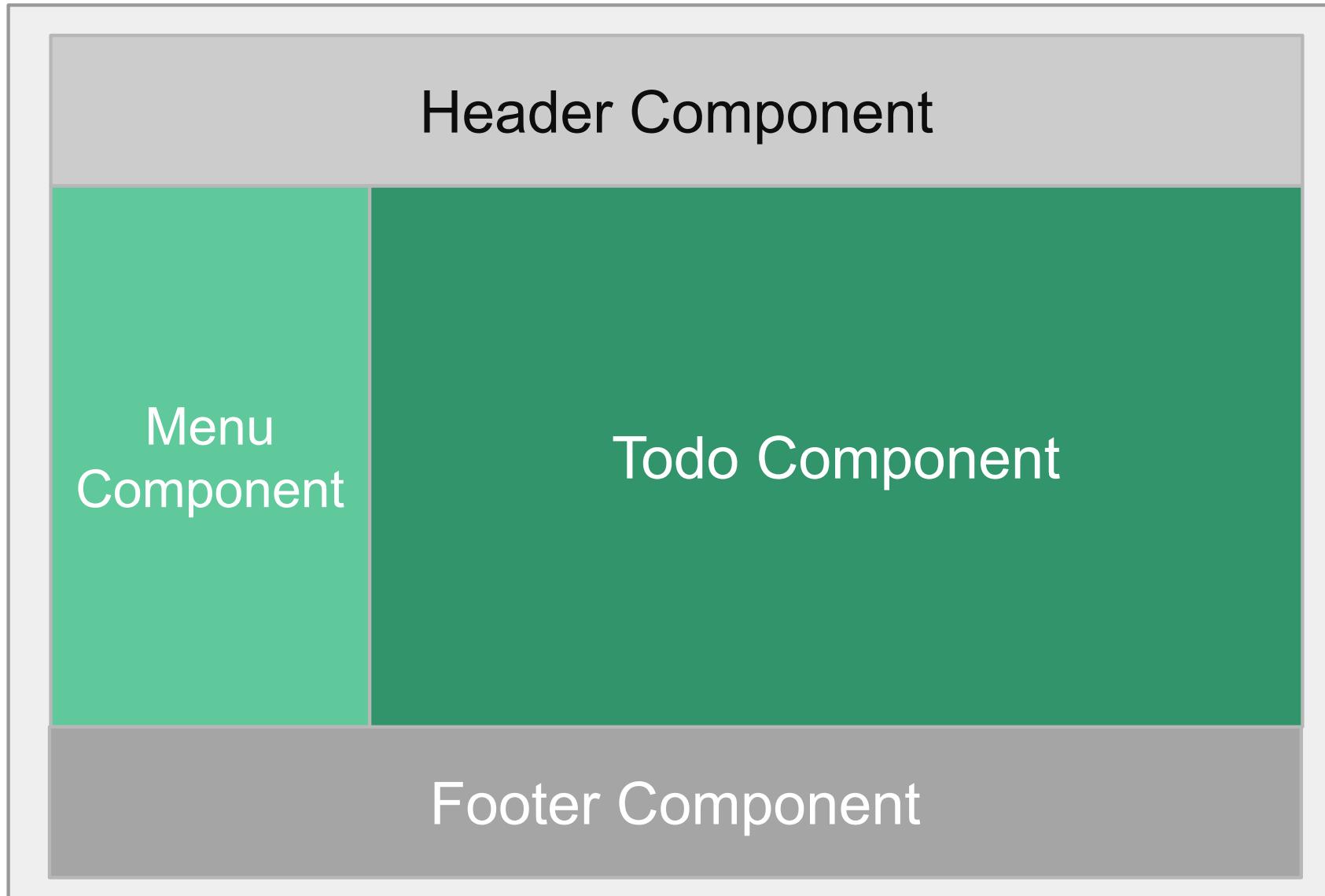
<https://universal.angular.io/>

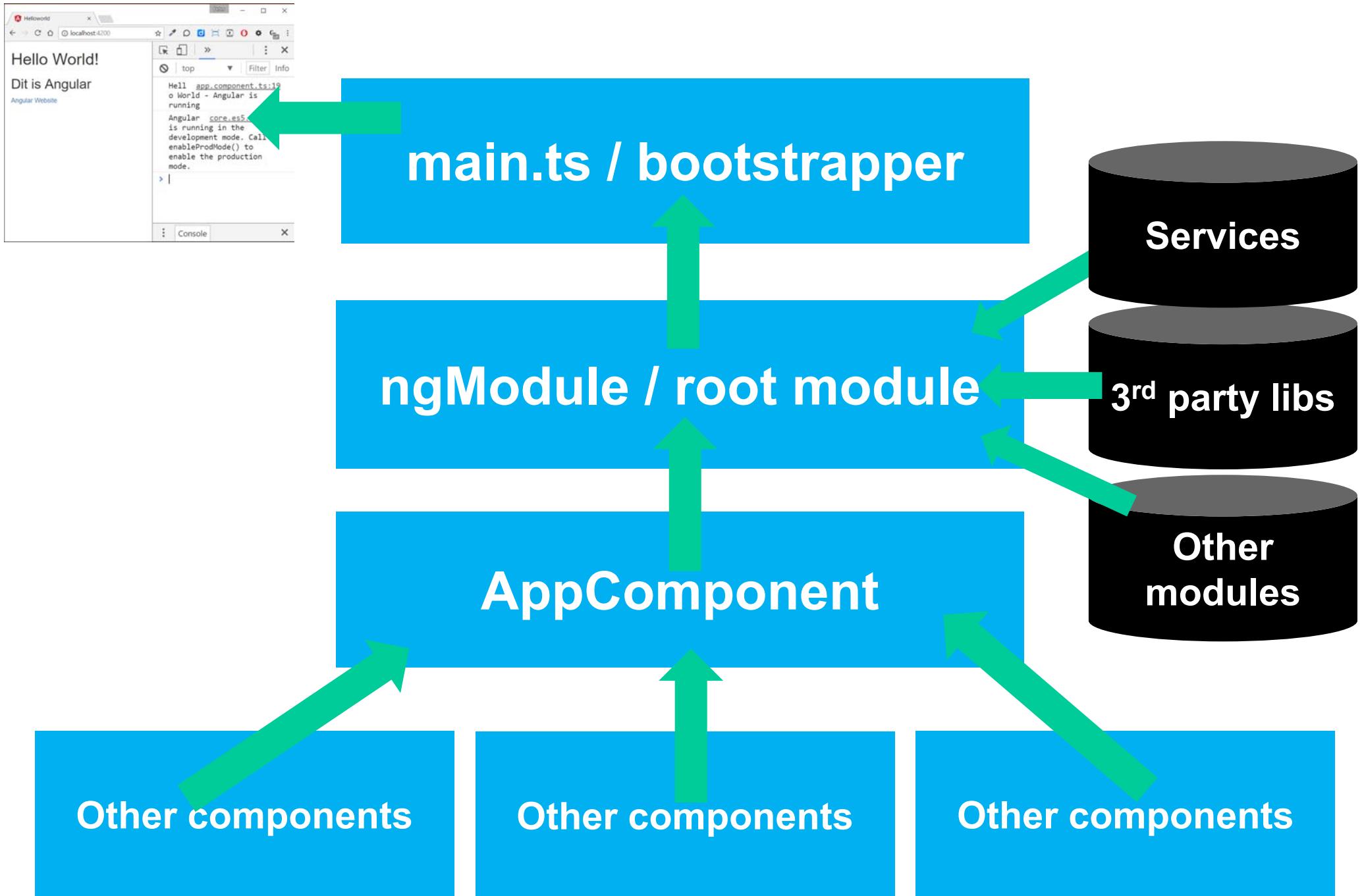
Angular 2 - components



*“An Angular-app is a
tree
of components”*

Components – visually



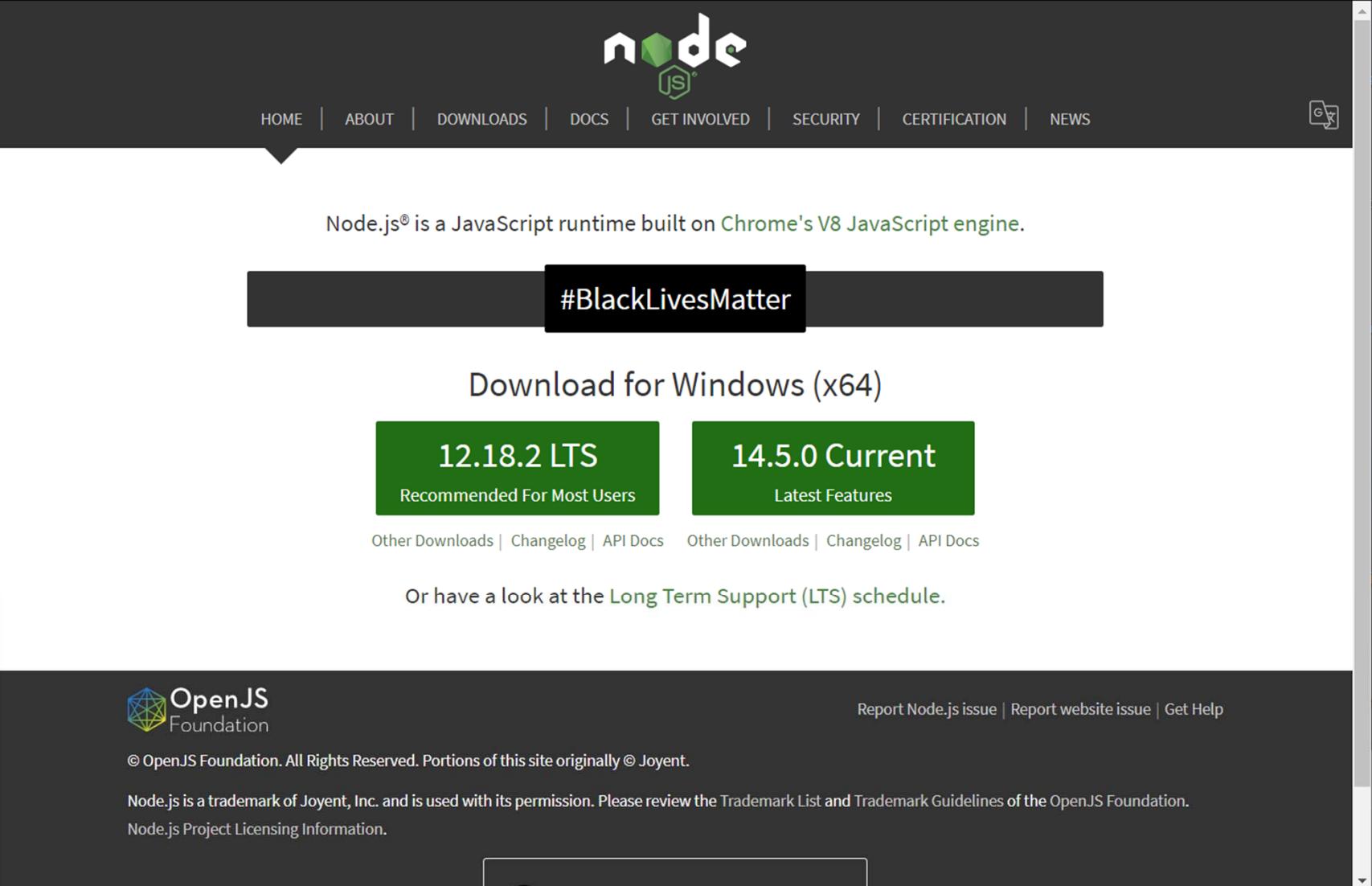




Let's look at some code

'Hello World' in Angular

Angular development dependency: NodeJS 12+



The screenshot shows the official Node.js website. At the top, there's a dark header bar with the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. A small 'Get' button is also visible. Below the header, a banner states: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." A prominent black bar features the "#BlackLivesMatter" hashtag. The main content area has a white background. It includes a heading "Download for Windows (x64)" followed by two green buttons: one for "12.18.2 LTS" (labeled "Recommended For Most Users") and another for "14.5.0 Current" (labeled "Latest Features"). Below these buttons are links for "Other Downloads | Changelog | API Docs" and "Report Node.js issue | Report website issue | Get Help". The footer contains the OpenJS Foundation logo, copyright information ("© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent."), trademark and licensing notices, and a "Report Node.js issue | Report website issue | Get Help" link.

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

#BlackLivesMatter

Download for Windows (x64)

12.18.2 LTS
Recommended For Most Users

14.5.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

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

OpenJS
Foundation

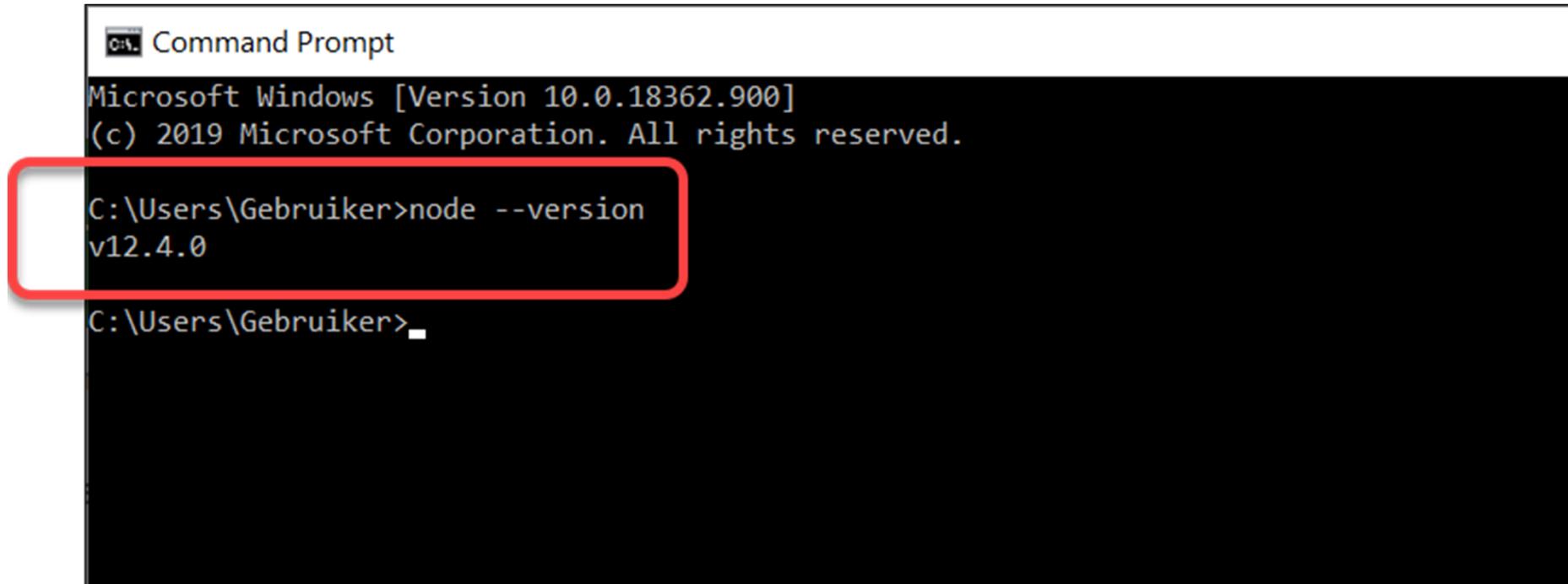
© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

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](#)

Report Node.js issue | Report website issue | Get Help

Node – check your version



A screenshot of a Microsoft Windows Command Prompt window. The window title is "Command Prompt". The system information shows "Microsoft Windows [Version 10.0.18362.900]" and "(c) 2019 Microsoft Corporation. All rights reserved.". In the command line area, the user has typed "C:\Users\Gebruiker>node --version" and the output is "v12.4.0". A red rectangular box highlights the command and its output. The prompt "C:\Users\Gebruiker>" is visible at the bottom.

```
Command Prompt
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

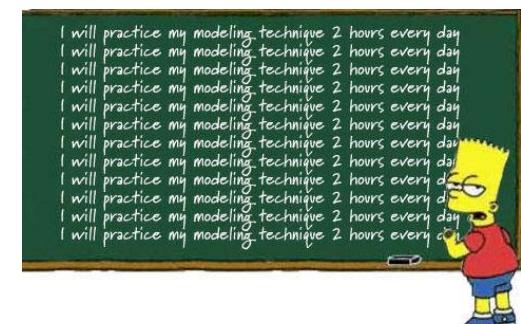
cd examples

cd 100-helloworld

npm install

npm start

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



The screenshot shows a comparison between a running Angular application and its development environment.

Top Left: A browser window displaying the Angular application. The title bar says "localhost:4200". The content area shows the text "Hello World!" in large bold black font, followed by "This is Angular" in dark blue font, and a link "Angular Website" in blue font.

Bottom Right: An IDE interface showing the project structure and the code for `app.component.ts`.

Project Structure:

- Project: voorbeeldenAngular2 (C:\Users\Peter Kassenaar\Desktop\voorbeeldenAngular2)
- Examples
- 100-helloworld (highlighted with a red box)
- node_modules (library root)
- src
- app
- 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

Code Editor: The file `app.component.ts` is open in the editor.

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

# Hello World!



## This is Angular

Angular Website
  `
})
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) - @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

The screenshot shows a blog post titled "Introducing Angular Modules - Root Module" by John Papa. The post discusses the `@NgModule` decorator and its purpose of organizing an application into cohesive blocks of functionality. The post is dated September 5, 2016, and includes social sharing icons for Twitter, Facebook, Google+, LinkedIn, and StumbleUpon. The sidebar features a bio for John Papa, who is a Google Developer Expert and Microsoft Regional Director, and links to his most recent posts.

HOME SPEAKING VIDEOS ARTICLES ABOUT CONTACT

Angular Modules

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, 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

```
<html>

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

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

Body of index.html

Element reference (selector) of root-component:

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

Run the app

npm start – run start script from package.json.

ng serve - start global angular-cli instance

The terminal window shows the following output:

```
+ 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} vendor.js, vendor.js.map (vendor) 1.1 MB [initial] [rendered]

webpack: C
```

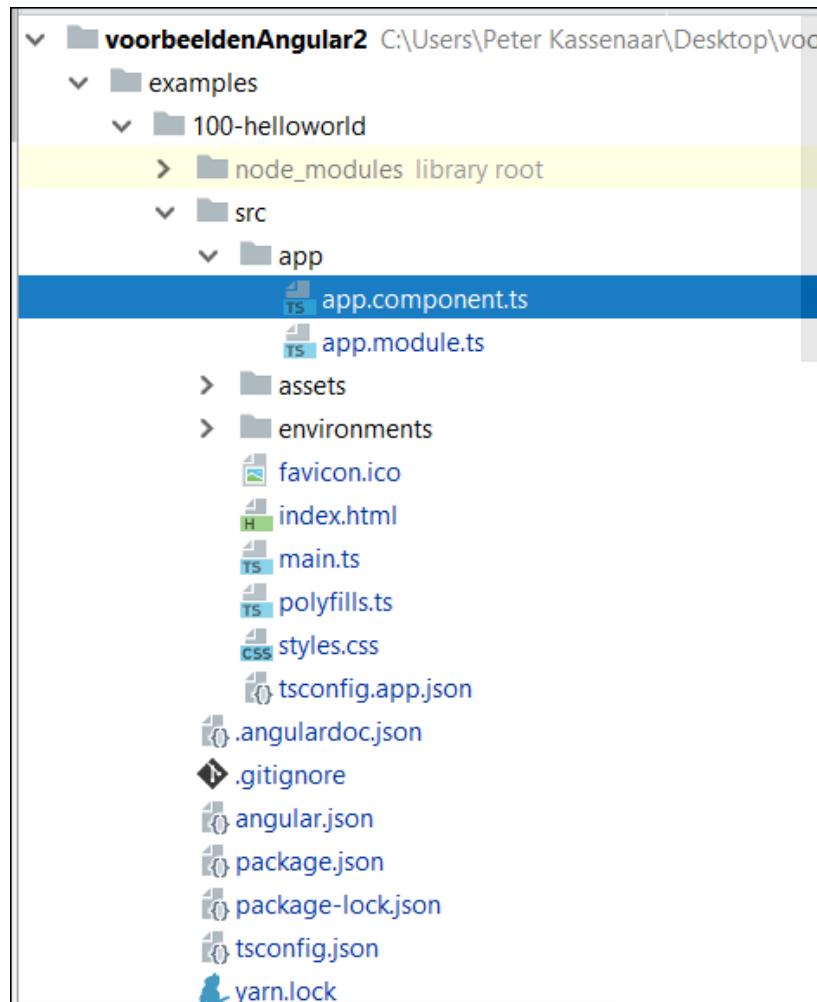
The browser screenshot shows the Angular application running at `localhost:4200`. The page content includes "Hello World!", "Dit is Angular", and "Angular Website". The developer tools console tab shows the following logs:

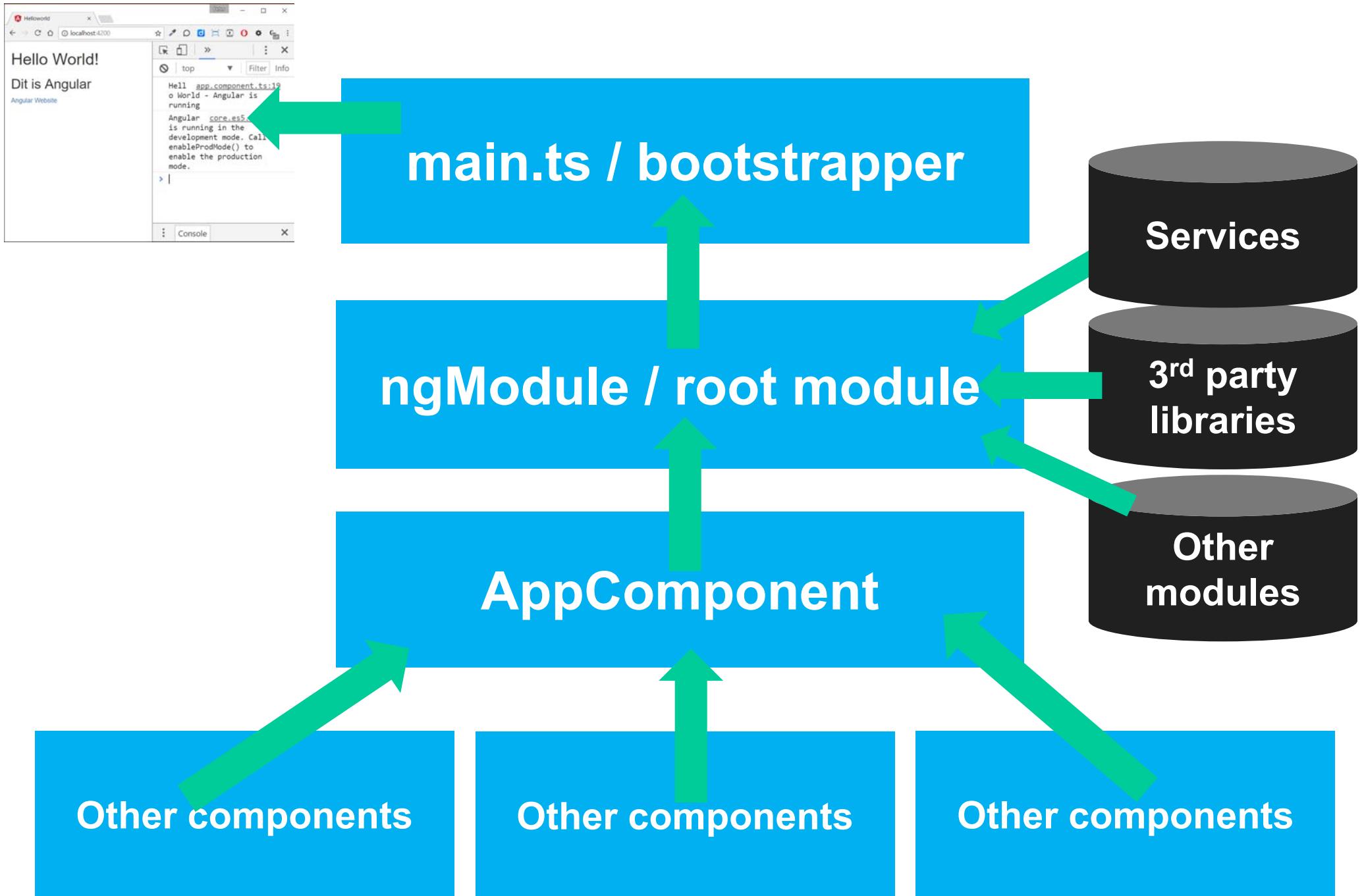
```
Hello World - Angular is running
Angular is running in the development mode. Call
enableProdMode() to enable the production mode.
```

After that: edit `app.component.ts`

– Automagically refreshed through Live Reload

So, a Basic Project Structure and Architecture





Checkpoint

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

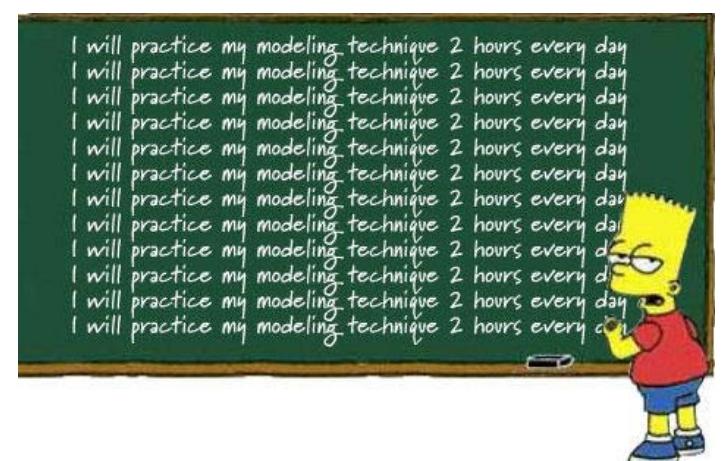
(At least) Five steps:

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



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
```

The screenshot shows the Angular CLI documentation site. The top navigation bar includes links for FEATURES, DOCS, RESOURCES, EVENTS, and BLOG, along with a search bar and social media icons. The left sidebar has a tree view of documentation categories, with 'Reference' expanded to show 'Conceptual Reference' and 'CLI Command Reference'. The main content area features a large title 'CLI Overview and Command Reference' and a brief introduction about the Angular CLI's purpose. Below this is a section titled 'Installing Angular CLI' with instructions on how to install it using npm. A code block shows the command: `npm install -g @angular/cli`. Further down, there's information about releases and a 'Basic workflow' section.

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

Background info



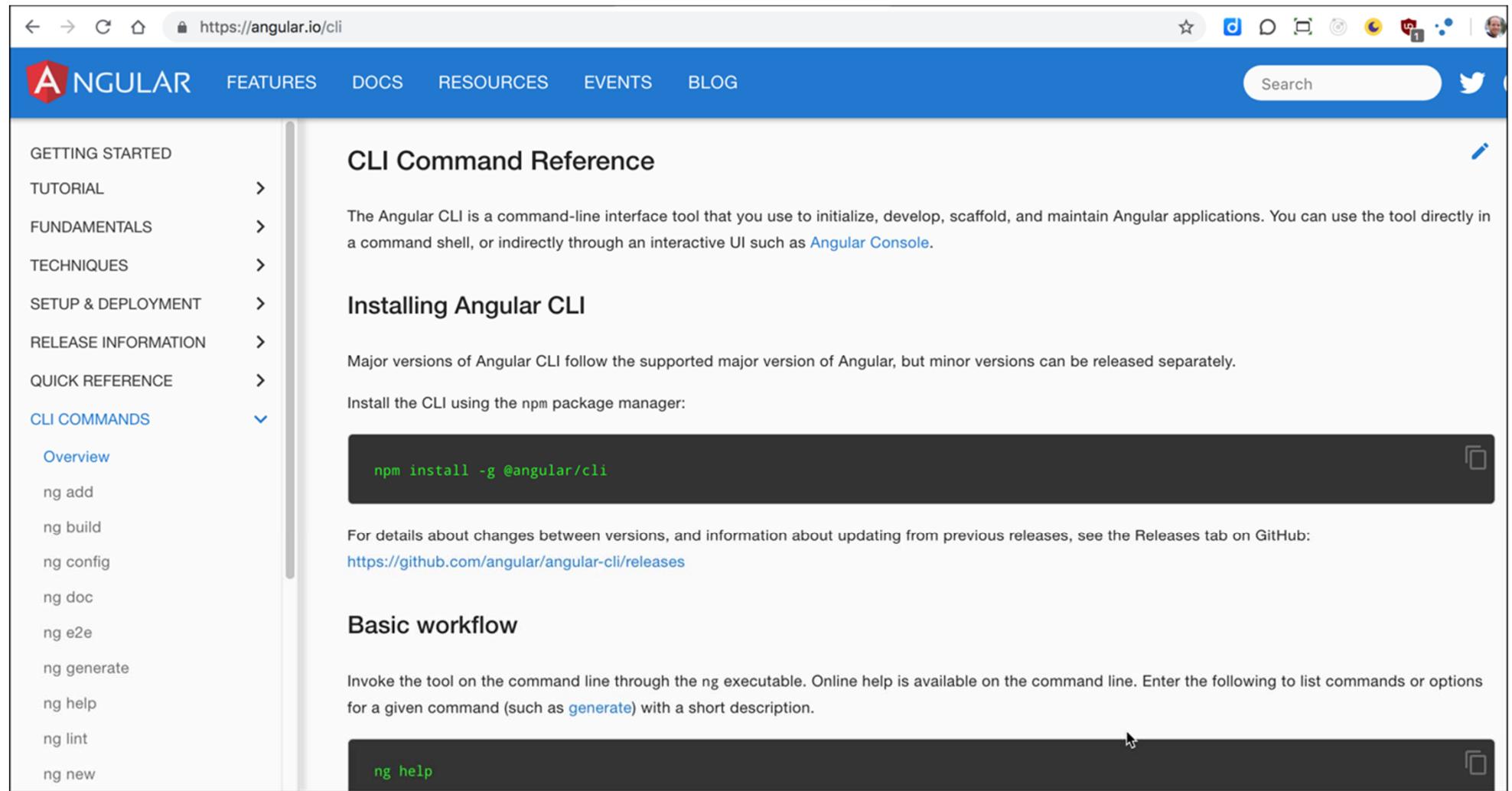
<https://www.youtube.com/watch?v=wHZe6gGI5RY>

Main commands

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

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

Documentation - in the Angular Docs



The screenshot shows a browser window 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 in the header.

The main content area has a sidebar on the left containing sections like GETTING STARTED, TUTORIAL, FUNDAMENTALS, TECHNIQUES, SETUP & DEPLOYMENT, RELEASE INFORMATION, QUICK REFERENCE, and CLI COMMANDS. Under CLI COMMANDS, there's an Overview and a list of commands: ng add, ng build, ng config, ng doc, ng e2e, ng generate, ng help, ng lint, and ng new.

The main content starts with a section titled "CLI Command Reference". It describes the Angular CLI as a command-line interface tool used for initializing, developing, scaffolding, and maintaining Angular applications. It can be used directly in a command shell or through an interactive UI like the Angular Console.

Below this is a section titled "Installing Angular CLI". It notes that major versions of the CLI follow the supported major version of Angular, while minor versions can be released separately. It provides instructions to install the CLI using npm:

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

For more details about changes between versions and information about previous releases, it points to the Releases tab on GitHub at <https://github.com/angular/angular-cli/releases>.

The page also includes a "Basic workflow" section with instructions on how to invoke the tool on the command line using the ng executable. It suggests entering commands like ng help to list commands or options for a given command like ng generate.

<https://angular.io/cli>



Angular Code - “Backend”

On TypeScript en ES6

Programming languages



w|shVfulsw

HV9

HV8

HV9#dgg#W|shVfulsw

Wkh#ixwkuh#ri#MydVfulsw#lv#IV92HV5348

P dmr#ksgdwh#urp #MydVfulsw#lv#surjup p lqj#dqjxdjh

P rgxdhv#folvhv#dgg#p ruh

Khosv#q#ghyhar slqj #Dqjxodut#ssv

W|shVfulsw#lv#d#W|shg#xshumhw#ri#IV9=

Dqqrwdwlrqv# #W|shv

Iqhuidfhv

Frp sldhu

TypeScript – tooling support

Types, Autocompletion, color coding.

Compile-time checking in editors.

Everything in TypeScript is
optional.

You can always use just JavaScript

.

DufkLhfwxuh#riff rxuFrp srqhqw#Fodw

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