



Global Knowledge®

Angular Fundamentals

Introduction + Core

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, frontend general.

www.kassenaar.com

info@kassenaar.com

VANDUUREN
MEDIA



ING



OHRA

euricom
A DIMENSION DATA COMPANY



s a n o m a

delta lloyd

zenito
BETERE ZEKERHEID
VOOR ONDERNEMERS

Atos



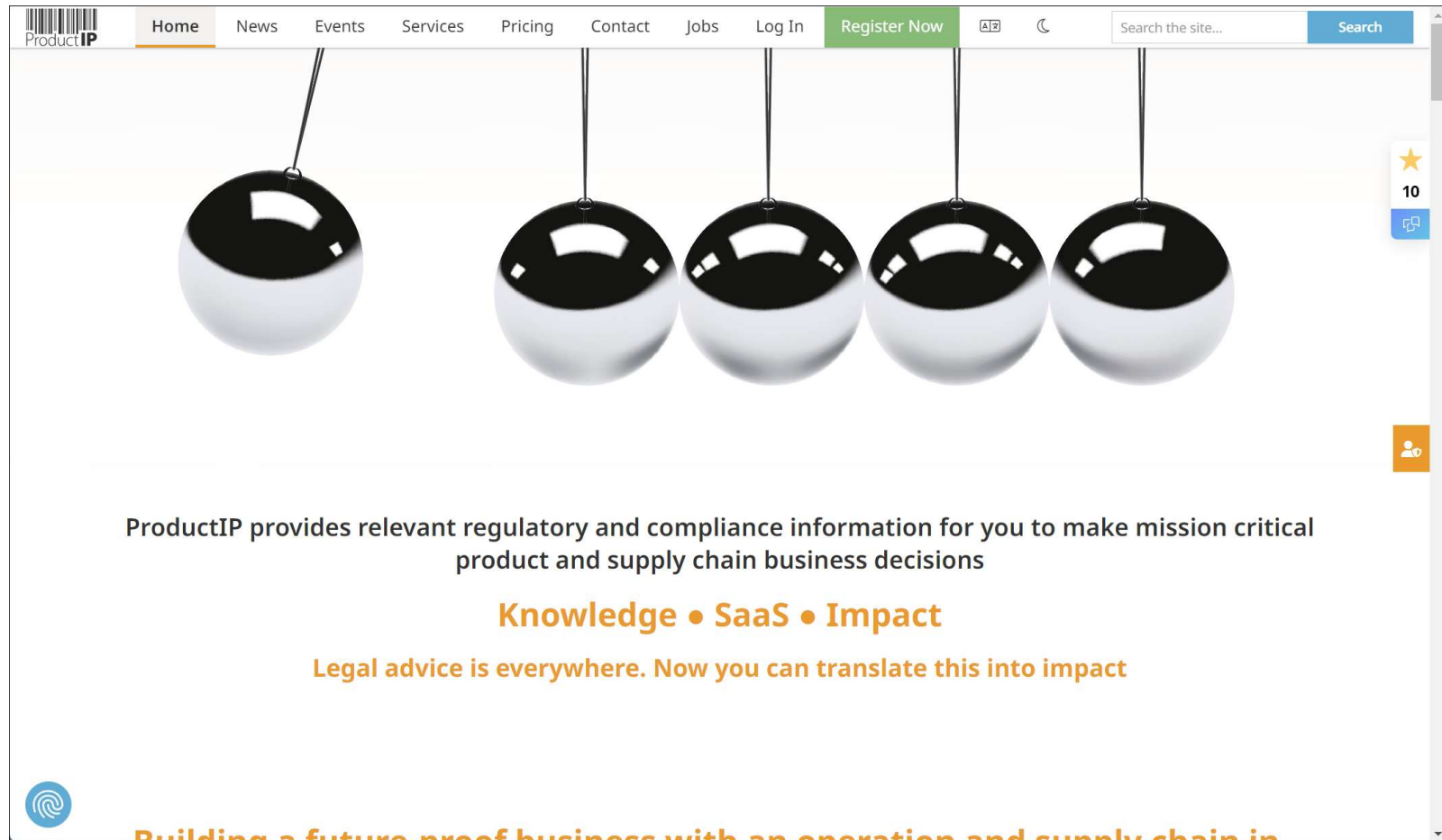
OBERON INTERACTIVE

woonbron

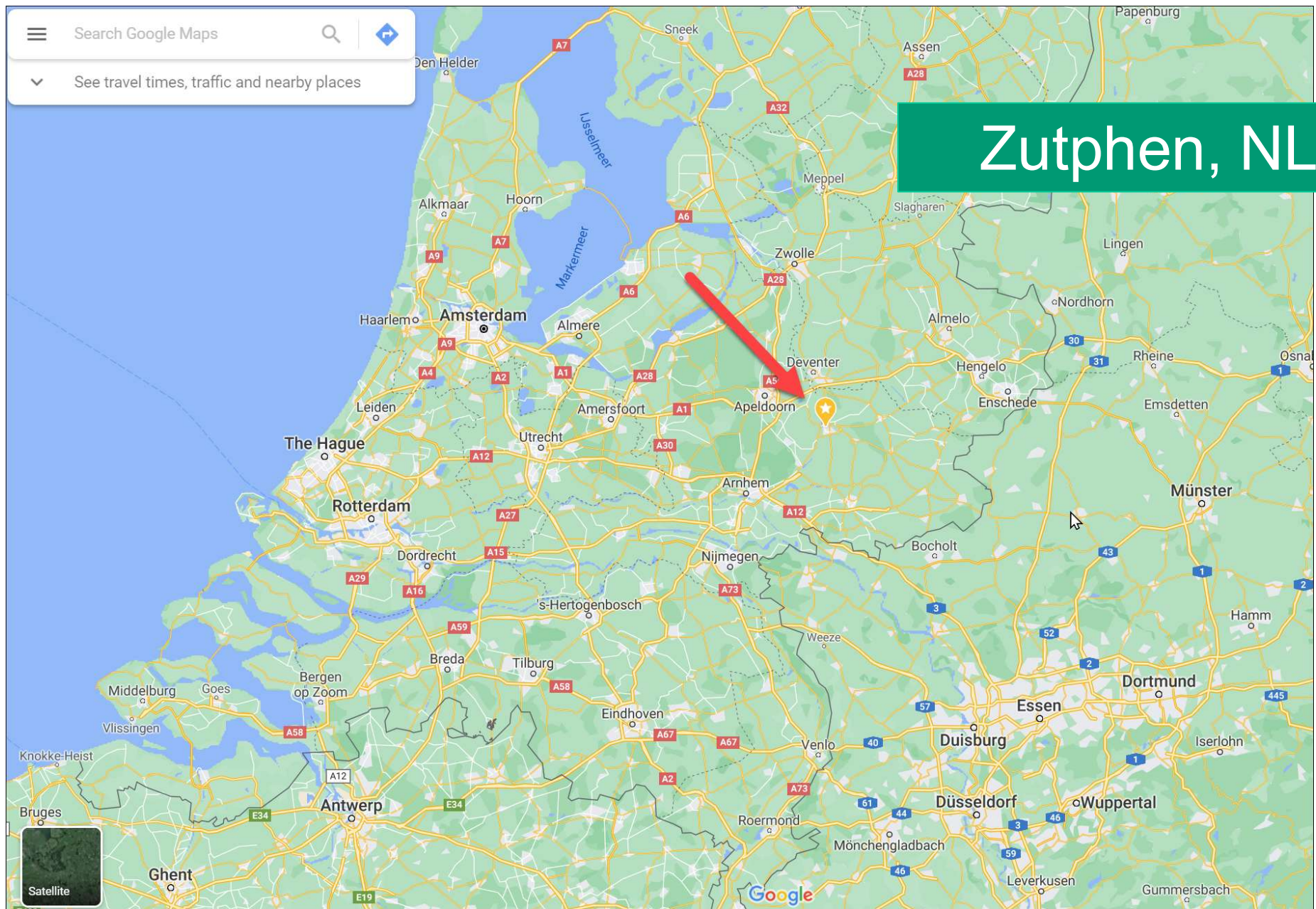


the eforum
FACTORY

Sr. frontend developer ProductIP, Ede, NL (50%)



www.productip.com

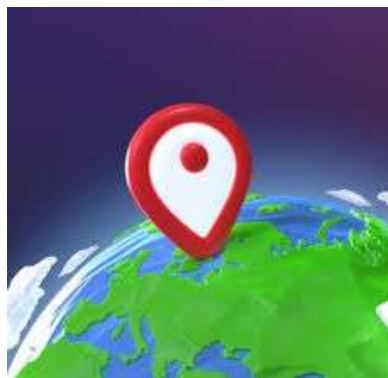




peterkassenaar



PeterKassenaar



PeterKassenaar



pkas06

github.com/PeterKassenaar/ilionx

PeterKassenaar / ilionx

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ilionx Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

File	Commit Message	Time
Angular	chore: updated README.md	2 minutes ago
JavaScript	chore: updated structure	4 minutes ago
.gitignore	Added slides and updated README.md	3 weeks ago
LICENSE	Initial commit	3 weeks ago
README.md	chore: updated README.md	2 minutes ago

README MIT license

ilionx

Slides and sample code on various trainings Ilionx, Young Professional Program, Fall 2024

- See the different folders in the repo for slides, example code and more. This repo will be deleted on Nov. 1, 2024. Make sure you have a copy before that date.

Questions? Feel free to reach out at info@kassenaar.com.

About

Slides and sample code on the training JavaScript Fundamentals, Ilionx, Fall 2024

Readme MIT license Activity 0 stars 1 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

About you



Introduce yourself shortly

Current knowledge, mobile apps, Angular apps?

Previous AngularJS 1.x- knowledge?

Other (web) languages?

Expectations of the training?

Specific or current projects?

Specific questions or techniques you want to learn about?

Agenda – globally

22, 23 April, 19 May – Mo. - Tue.

~ 9:00 start

~ 10:00, 11:00 short break

~ 12:00 lunch

~ 12:45 Afternoon session

~ 14:00, 15:00 short break

~ 16:00 End

When possible – finish a bit early
on last day



Goals of this training

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

but....

Goals

1. You *will* learn about the **structure and architecture** of Angular Apps.
From a small `hello-world` app to the largest enterprise applications.
2. You are familiar with the **main Angular concepts** of the framework.
You can always google or AI the code details yourself.
3. You will have **some hands-on experience** regarding creating apps and components, services, API's/backends, forms and Rabo specific components.
4. You will have a **general understanding** of the way modern web apps are created using Angular, TypeScript and build tools.

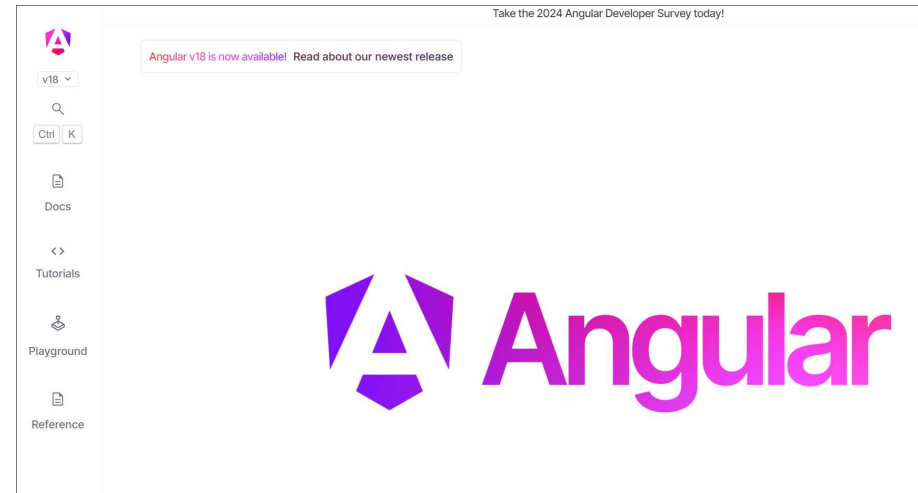
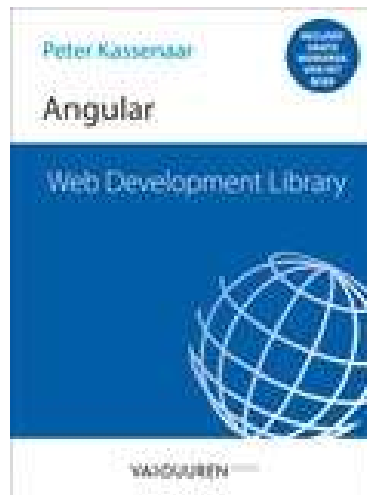
Agenda - Fundamentals

- **Introduction & short history** - why Angular, TypeScript?
- **Key Features** in Angular, -17, and 17+
- CLI, Hello World in Angular – Look at the **boilerplate-code**
- Angular in depth
 - Modules + Moduleless
 - **Components**
 - Data **binding**
 - **Dependency Injection** (DI) – more components
 - **Services** and http, **Observables** (RxJS), communicate backend
 - Intercomponent **communication**



Materials

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



angular.dev/

2 Richtlijnen

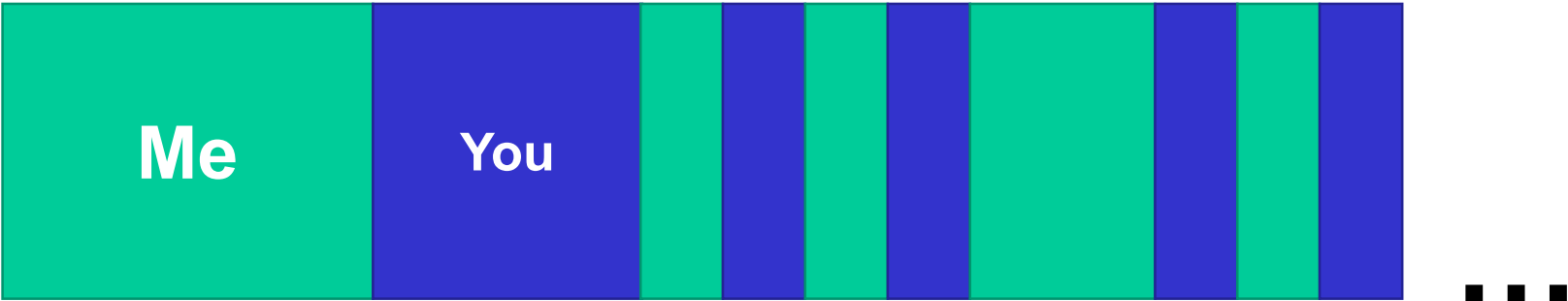
1. Workshops / Exercises

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

2. Example code – see online repo

- To support the exercises – ready made examples
- Work in progress – check Angular-site!
- <https://github.com/PeterKassenaar/voorbeeldenAngular2>

How I work...



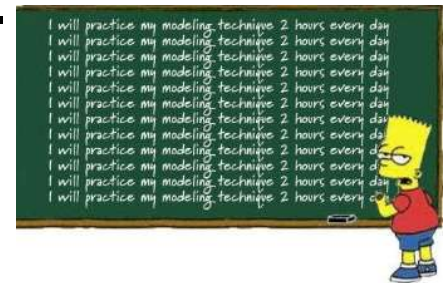
Advanced warning - First morning

Not so much code...

*Concepts, architecture,
structure*

On Workshops...

- ... Look like this slide 🙏
- Are executed **between** theoretical parts
- Are **NOT** written line-by-line. Think for yourself.
- Time during training **may be to short** to finish al workshops.
 - Finish **in your own time**
 - At least you are familiar with the **concepts** of the workshop
 - This **is a choice**. There is more to discuss! I let this prevail.
- The example code often has (but not always!) the/a 'solution' to the workshop. Use them! That's what they're for.
 - But of course: it would be nice if you do this yourself



Questions?



Angular vs. The Rest

Differences, similarities, new features

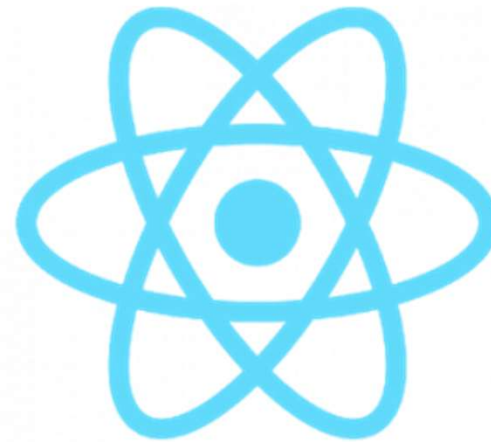


Taking one step back:

A look at front-end frameworks

What is a good choice, what is popular?

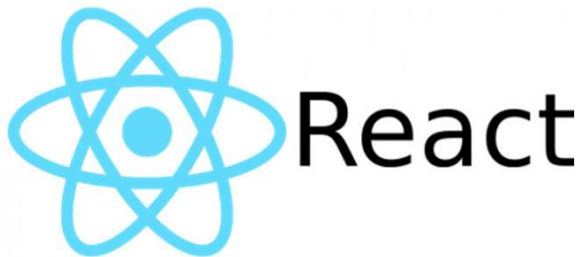
Front-end Frameworks – the big four



React



Similarities

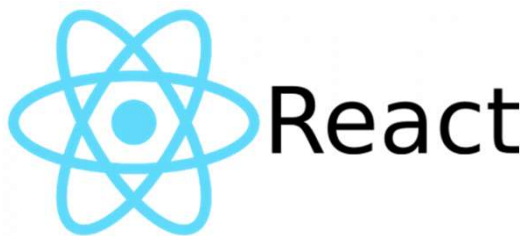


- Creating **Single Page Applications**
- Based on **components**
- Data **binding**, **props**, **events**, **routing**,
state management, ...
- Huge **ecosystem**
- Huge **community**
- High **adaptation** rate

Differences (apart from syntax)



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



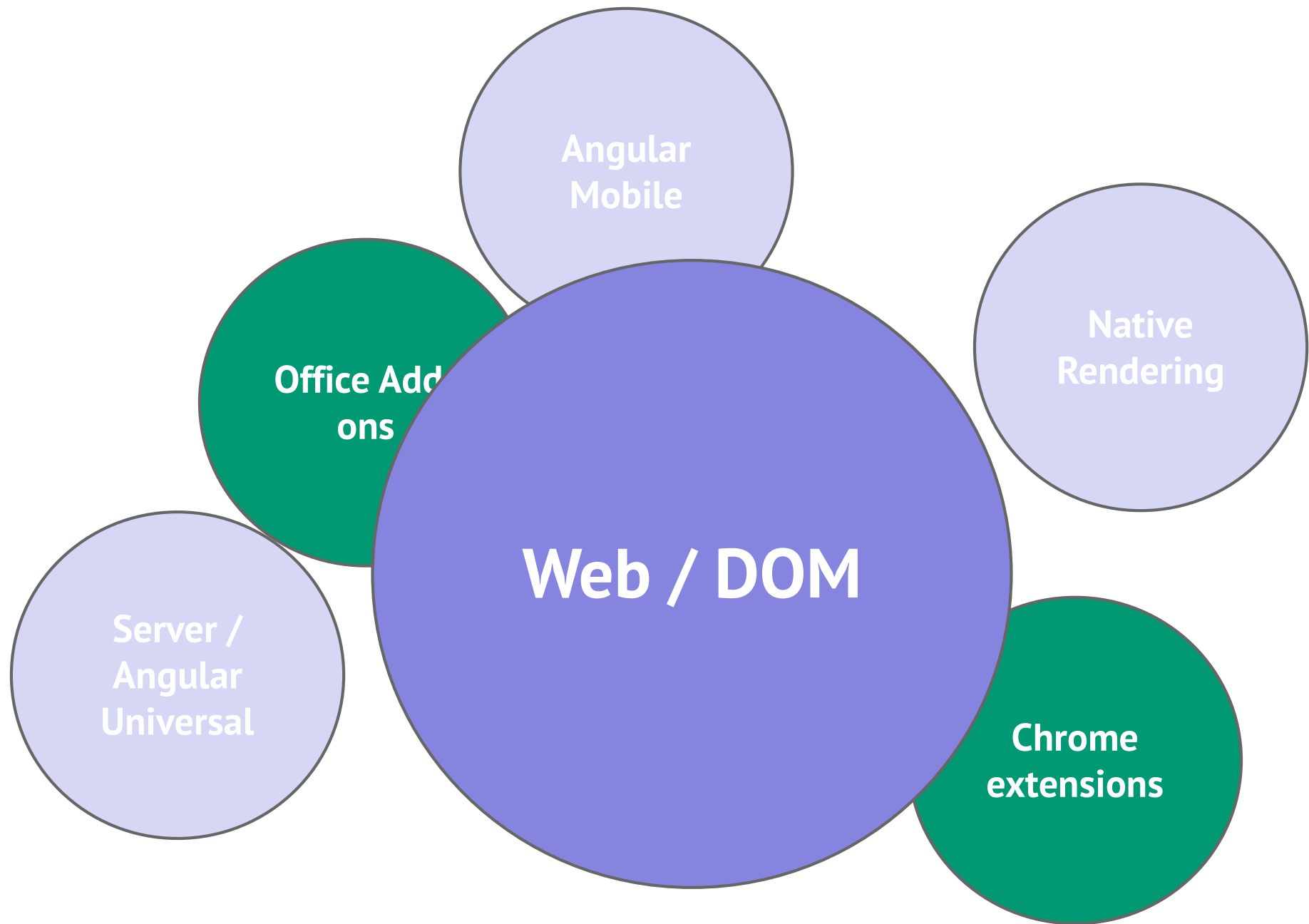
- Point of departure: **JavaScript**, JavaScript, JavaScript (JSX)
- Build-all-yourself / choice anxiety



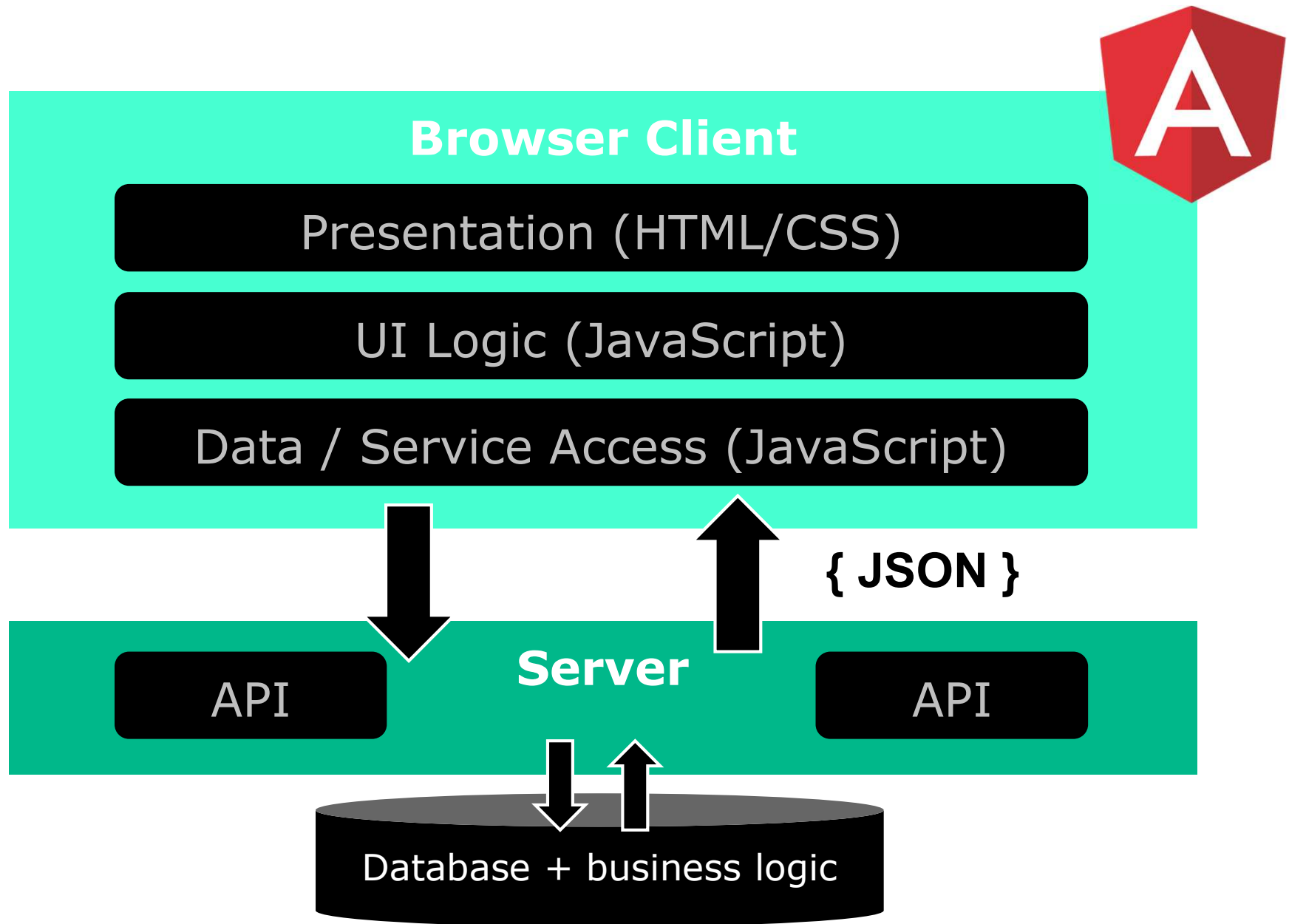
Platform

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





Time-based Releases

@IgorMinar, Nov. 9, 2018

Angular updates

- Important milestones
 - **Angular 2** – breaking change with Angular JS (1.x).
 - Syntax, structure, architecture
 - **Angular 6** – Introduction modern Angular CLI
 - **Angular 9** – Ivy Compiler
 - **Angular 14** – Standalone components
 - **Angular 17** – 'Modern' Angular
 - **Angular 19** – Current



angular.dev/update-guide

Update Guide

Select the options that match your update

Angular versions

From v. 17.0 To v. 18.0

Application complexity

✓ Basic Medium Advanced

Shows information for all Angular developers.

Other dependencies

- ☐ I use ngUpgrade to combine AngularJS & Angular
- ☐ I use Angular Material
- ☒ I use Windows

Show me how to update!

Angular Versions en -Long Time Support

→ <https://angular.dev/reference/releases>

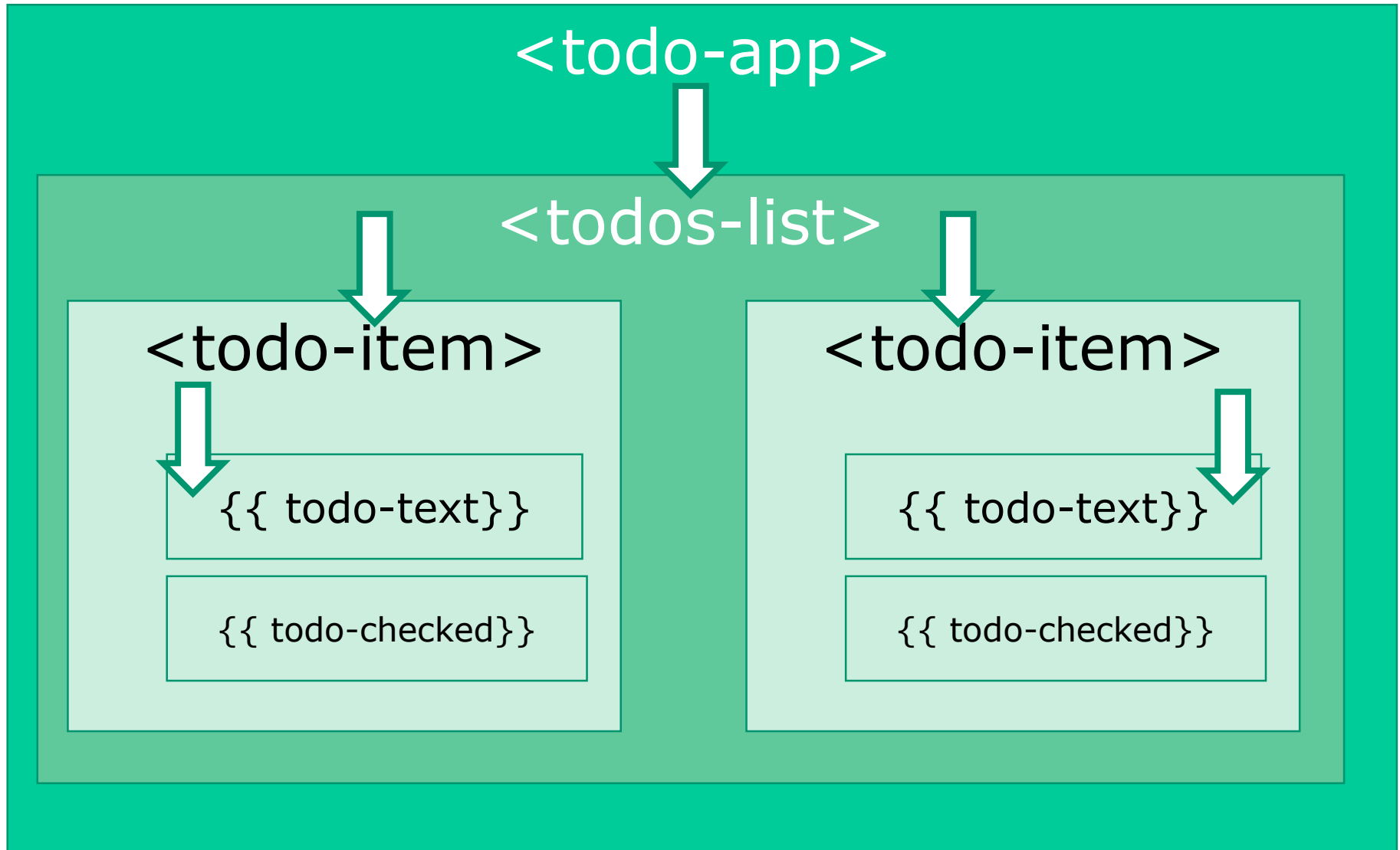
Release schedule

Version	Date
v19.1	Week of 2025-01-13
v19.2	Week of 2025-02-24
v20.0	Week of 2025-05-26

Support window

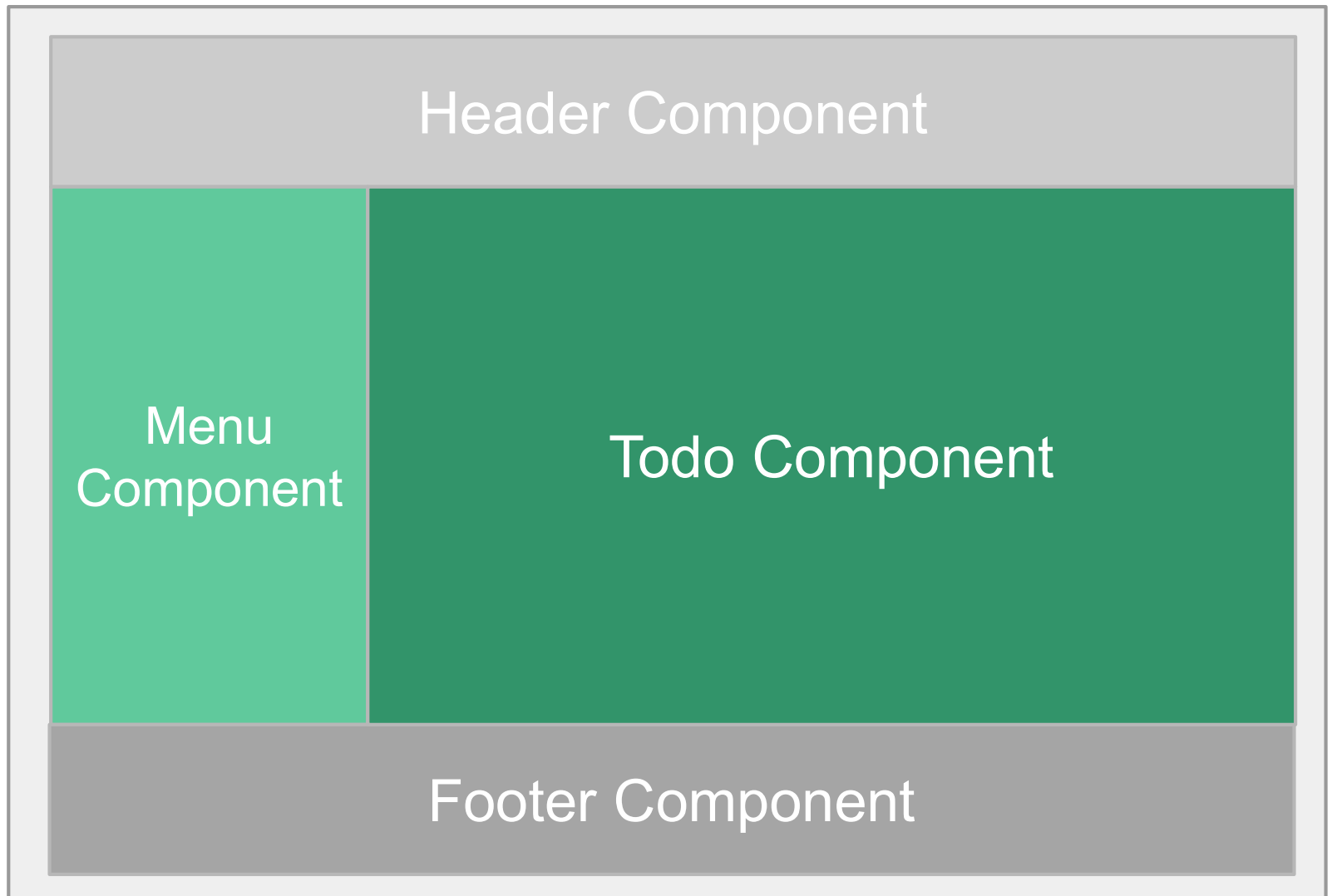
All major releases are typically supported for 18 months.

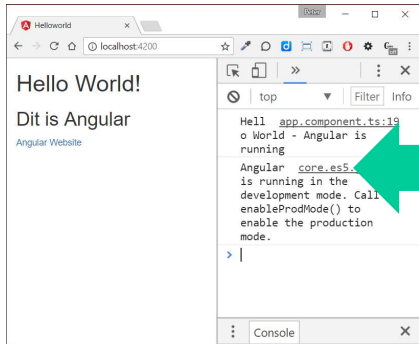
Angular – components



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

Components – visual representation





main.ts / bootstrapper

AppComponent

Services

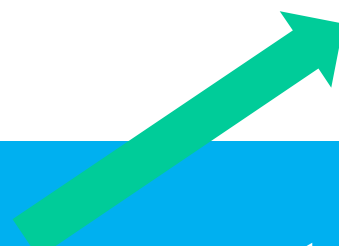
3rd party libs

Other modules

Other components

Other components

Other components

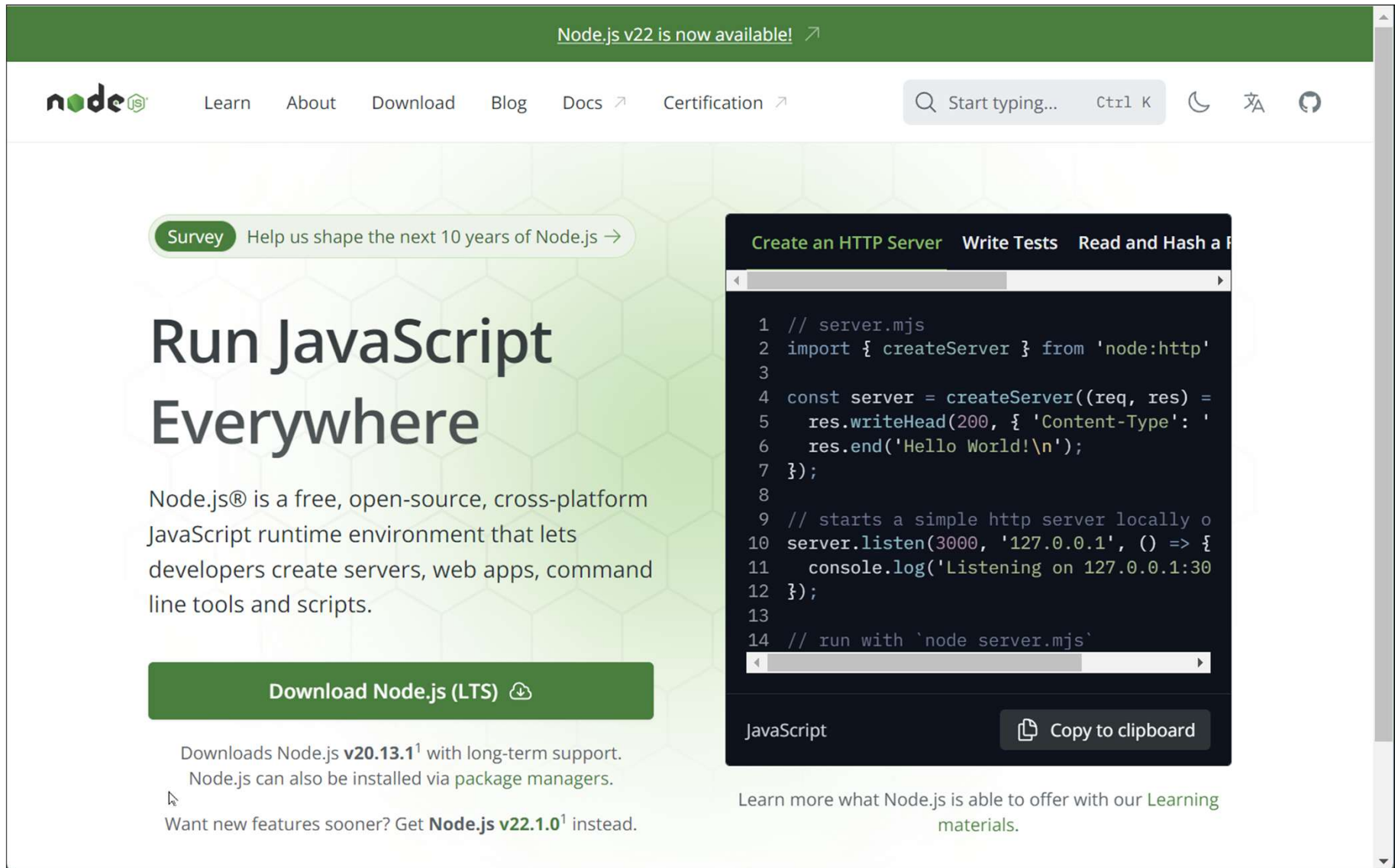




Let's write some code

Hello World in Angular

Angular development dependency: NodeJS 18.x +



The screenshot shows the Node.js website homepage. At the top, a green banner reads "Node.js v22 is now available!". Below this is a navigation bar with links: "node.js", "Learn", "About", "Download", "Blog", "Docs", and "Certification". A search bar on the right contains the text "Start typing..." and "Ctrl K". The main content area features a large heading "Run JavaScript Everywhere" and a subheading "Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts." Below this is a green button labeled "Download Node.js (LTS)". To the right of the button, text indicates that the download includes "Node.js v20.13.1" with long-term support and mentions that it can also be installed via package managers. A link to "Node.js v22.1.0" is provided for those wanting new features sooner. On the right side of the page, there is a code editor window titled "Create an HTTP Server" showing a JavaScript code snippet for creating a simple HTTP server. The code includes imports for 'node:http', a 'createServer' function, and a 'listen' method. A "Copy to clipboard" button is located below the code editor. At the bottom right, a link to "Learning materials" is provided.

Node.js v22 is now available! ↗

node.js Learn About Download Blog Docs ↗ Certification ↗

Search Start typing... Ctrl K

Survey Help us shape the next 10 years of Node.js →

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) ⬇

Downloads Node.js **v20.13.1**¹ with long-term support. Node.js can also be installed via package managers.

Want new features sooner? Get **Node.js v22.1.0**¹ instead.

Create an HTTP Server

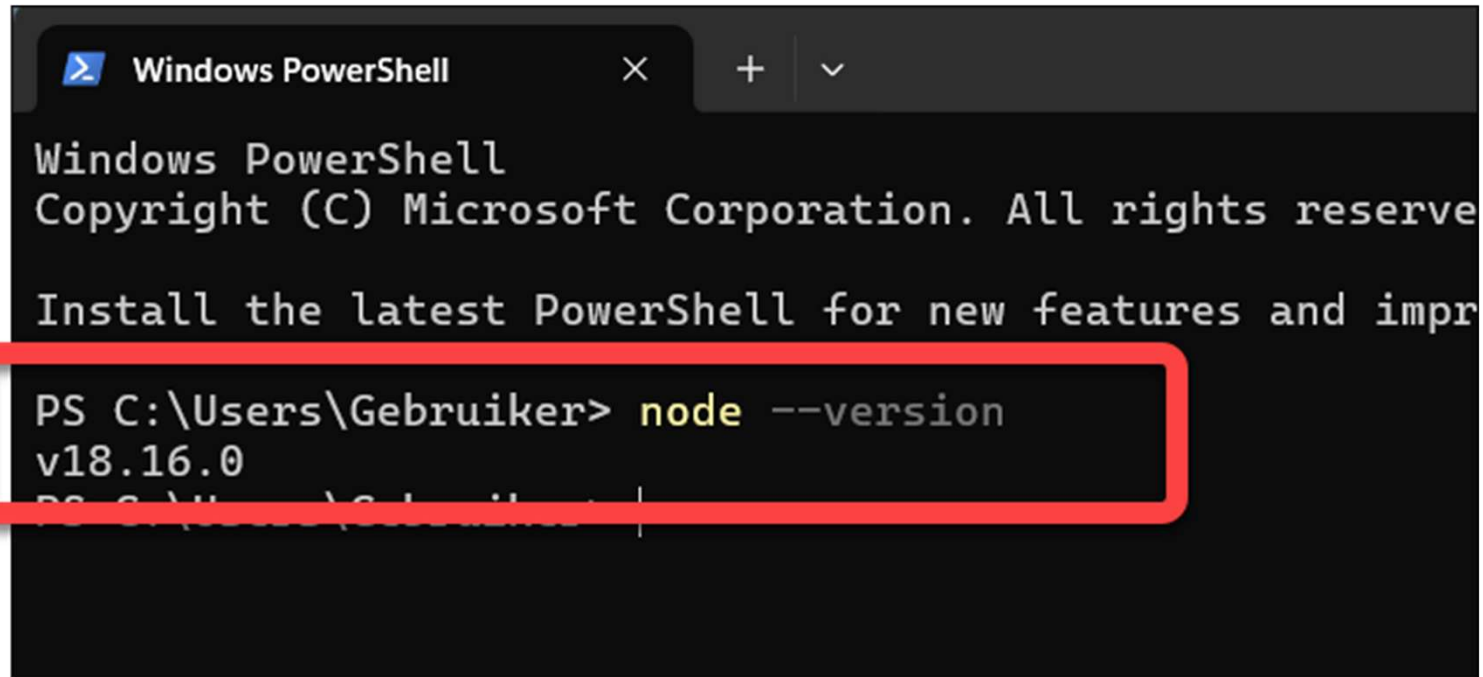
```
1 // server.mjs
2 import { createServer } from 'node:http'
3
4 const server = createServer((req, res) =
5   res.writeHead(200, { 'Content-Type': '
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally o
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:30
12 });
13
14 // run with `node server.mjs`
```

JavaScript Copy to clipboard

Learn more what Node.js is able to offer with our [Learning materials](#).

<https://nodejs.org/en>

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
v18.16.0
```

Mini workshop

- Download or clone
<https://github.com/PeterKassenaar/angular-fundamentals>
- When zipped: unpack the repository and cd into /Angular-fundamentals-main

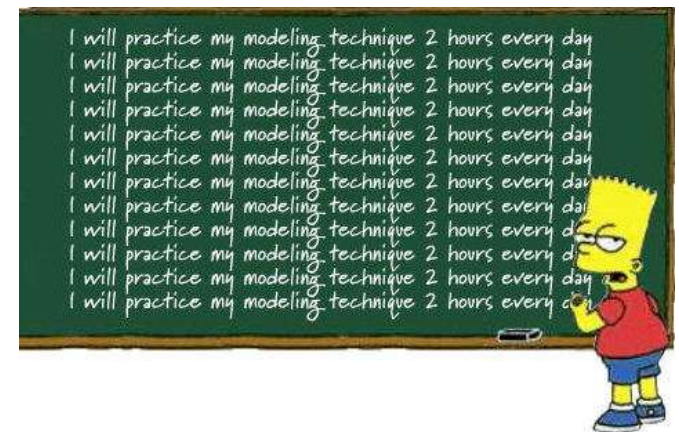
cd examples

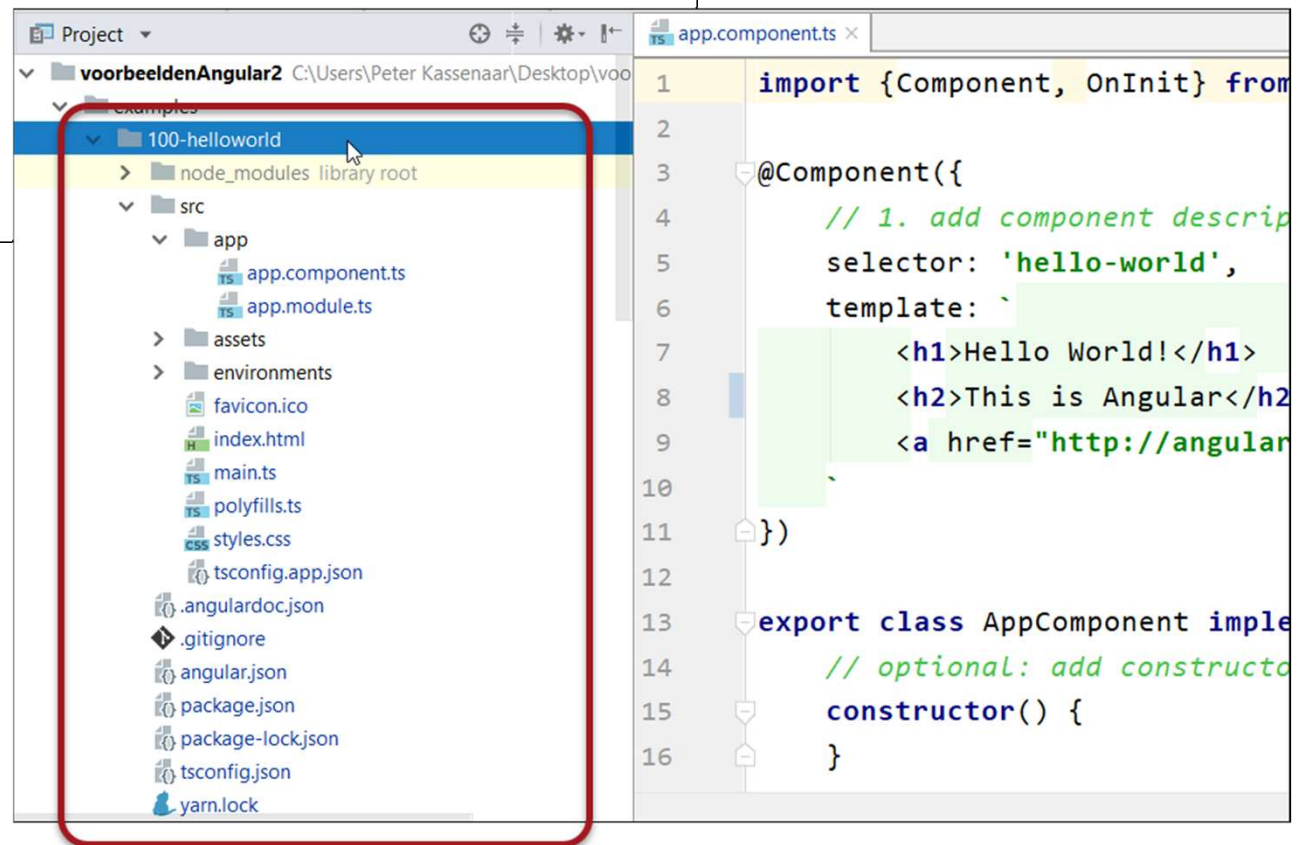
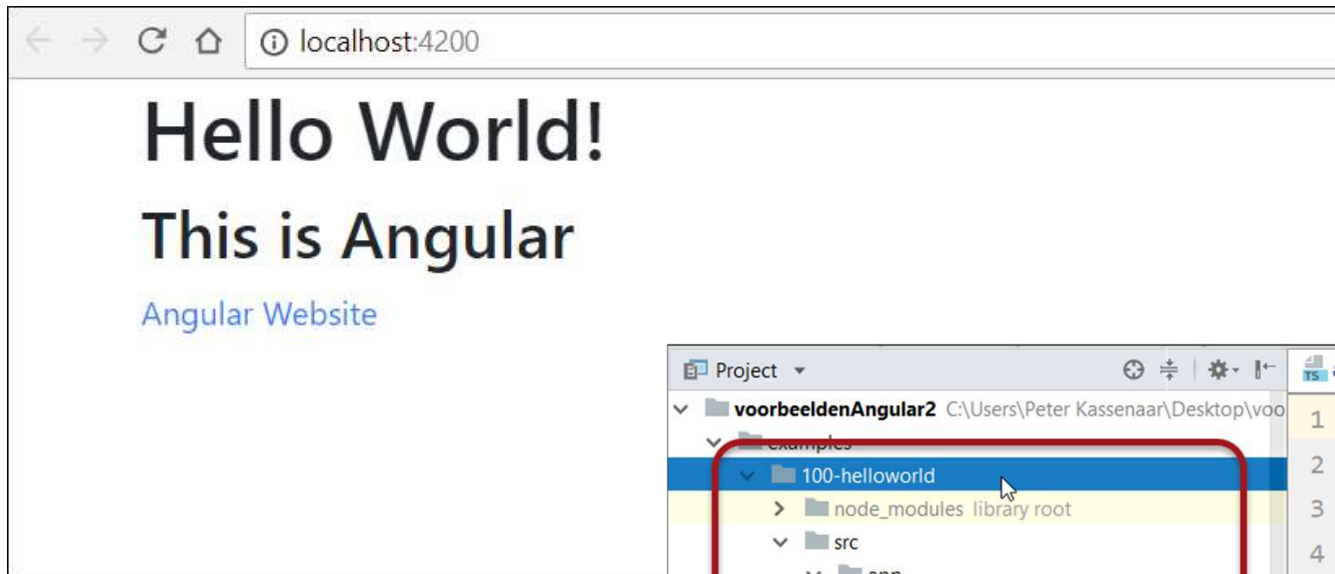
cd 100-helloworld

npm install

npm start

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





Alternative workshop

- Download and install Angular CLI

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

`ng new`

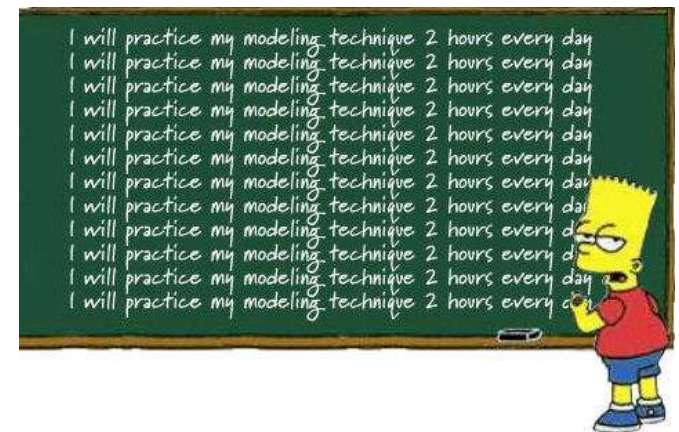
Answer questions / take defaults

`cd-to-directory`

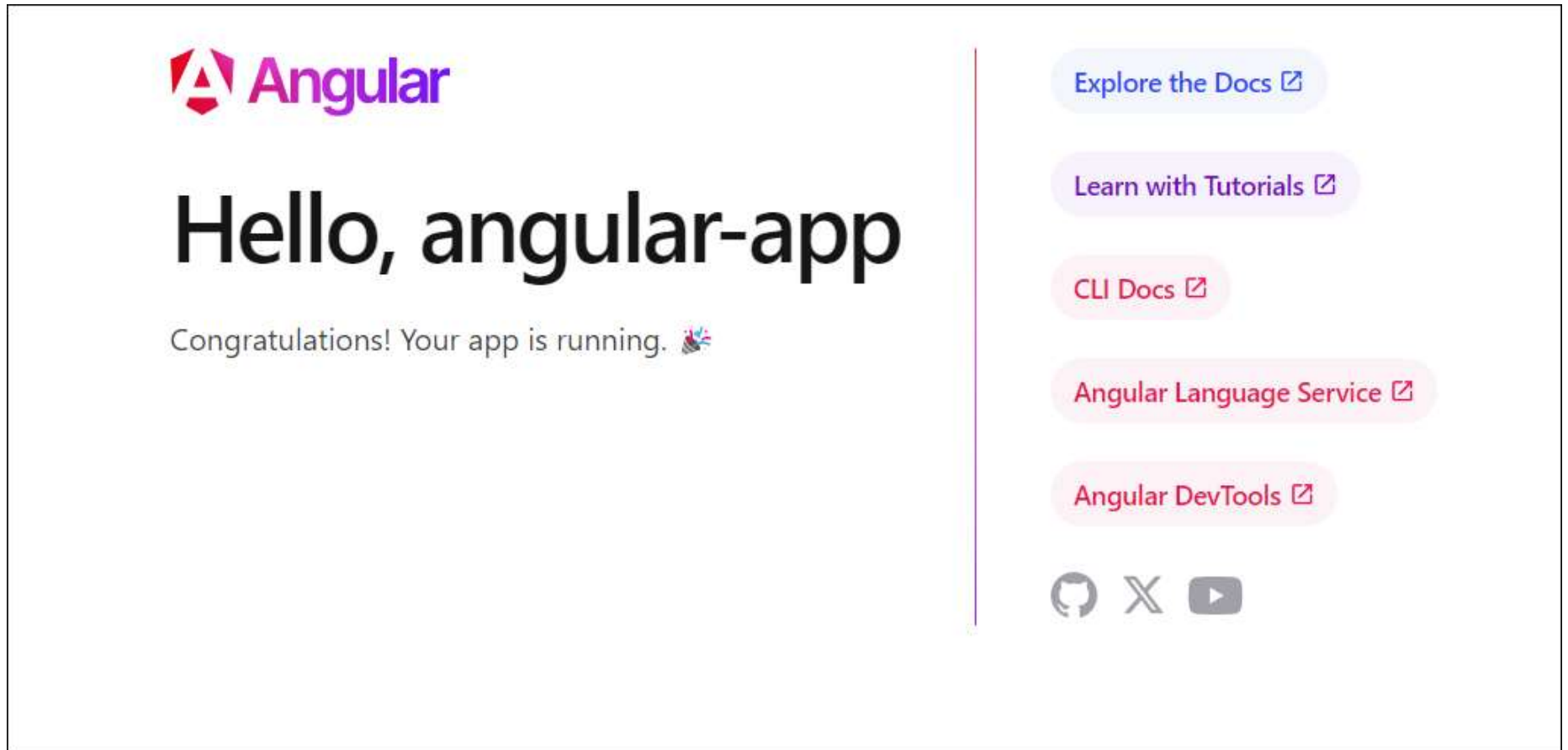
`npm start`

- Bring some updates to your app

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



Result default angular app w/ CLI





Structure of Angular apps

Important files that are present in *all* applications

Boilerplate code for Hello World

Steps

1. Set up environment, boilerplate & libraries
 - Important configuration files
2. Angular Component(s)
3. Angular Module(s): @ngModule() ➔ NOT required in V16+
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": "16.0.0",
    "@angular/common": "16.0.0",
    "@angular/compiler": "16.0.0",
    "@angular/core": "16.0.0",
    "@angular/forms": "16.0.0",
    "rxjs": "^7,8.0",
    "zone.js": "^0.14.3"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.6.0",
    "@angular/cli": "6.3.7",
    "typescript": "~5.4.2"
  },
  "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"               : [
      "es2022",
      "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",
            ...
          }
        }
      }
    }
  }
}
```

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 {
}
```

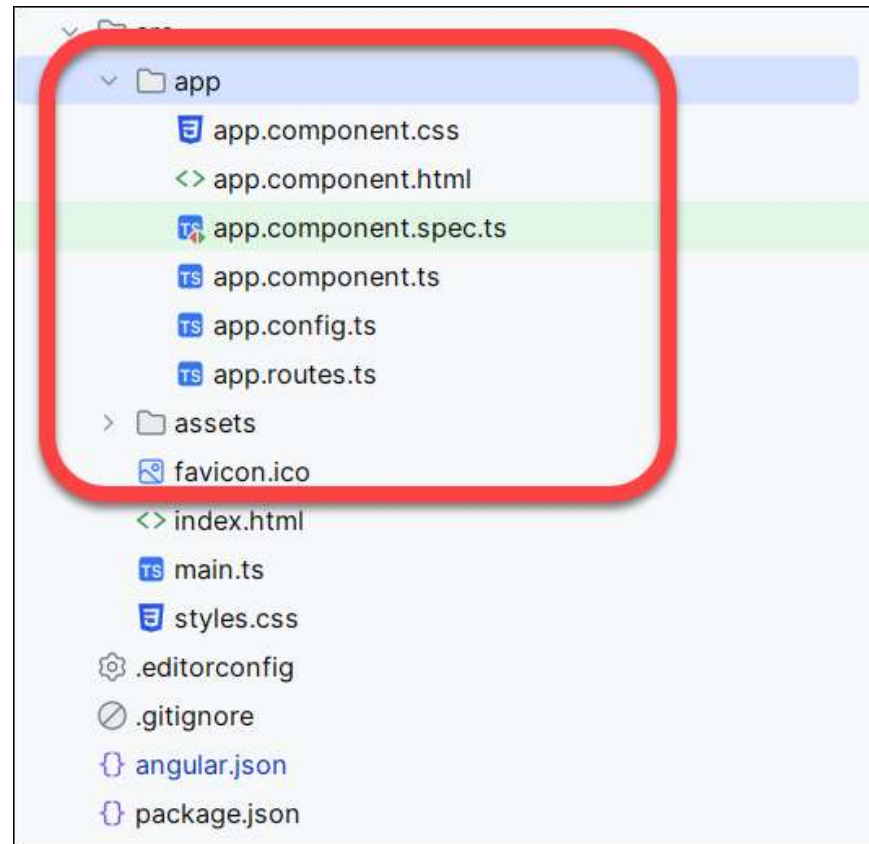
Root Module of the application

NOTE: NOT mandatory in newer applications

- As of Angular V16/17, the default choice is **NOT** to have Angular Modules anymore
- All applications are now composed out of **standalone components**

*However, you still need to know about
modules, as likely all existing
applications use them!*

New Angular structure



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 - HTML file - expanded at runtime by WebPack or Vite

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 of index.html

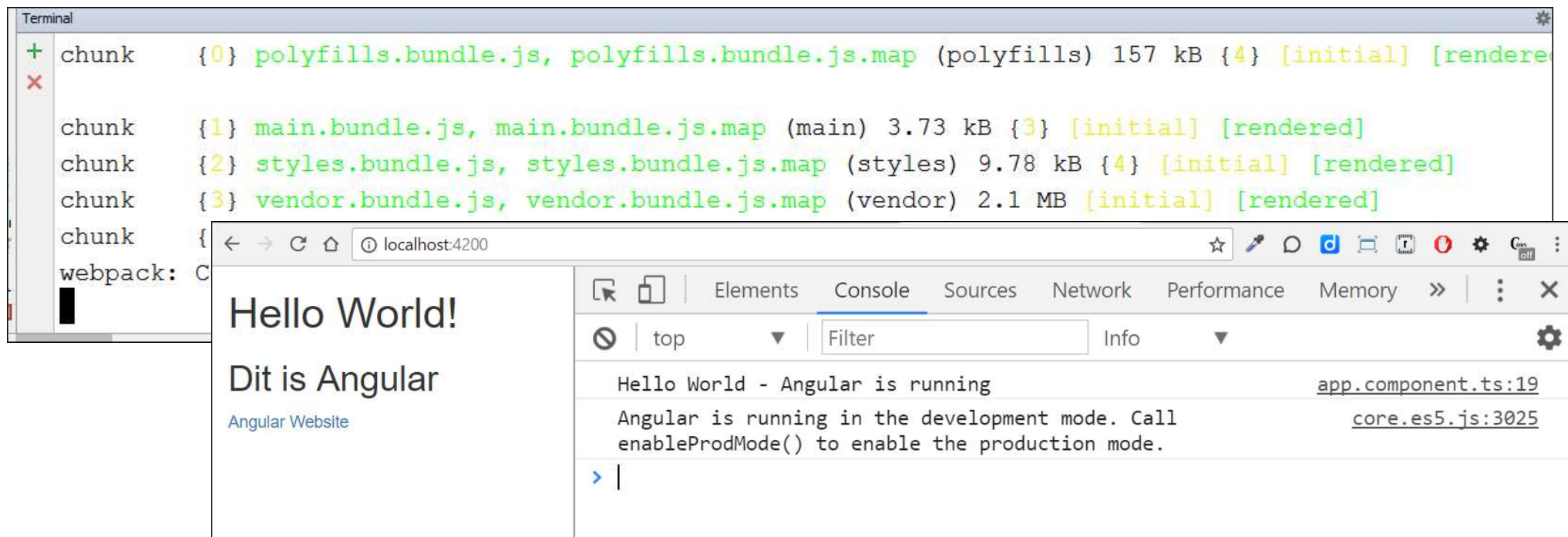
Element reference (`selector`) of root-component:

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

Run the app

`npm start` – run script start from `package.json`.

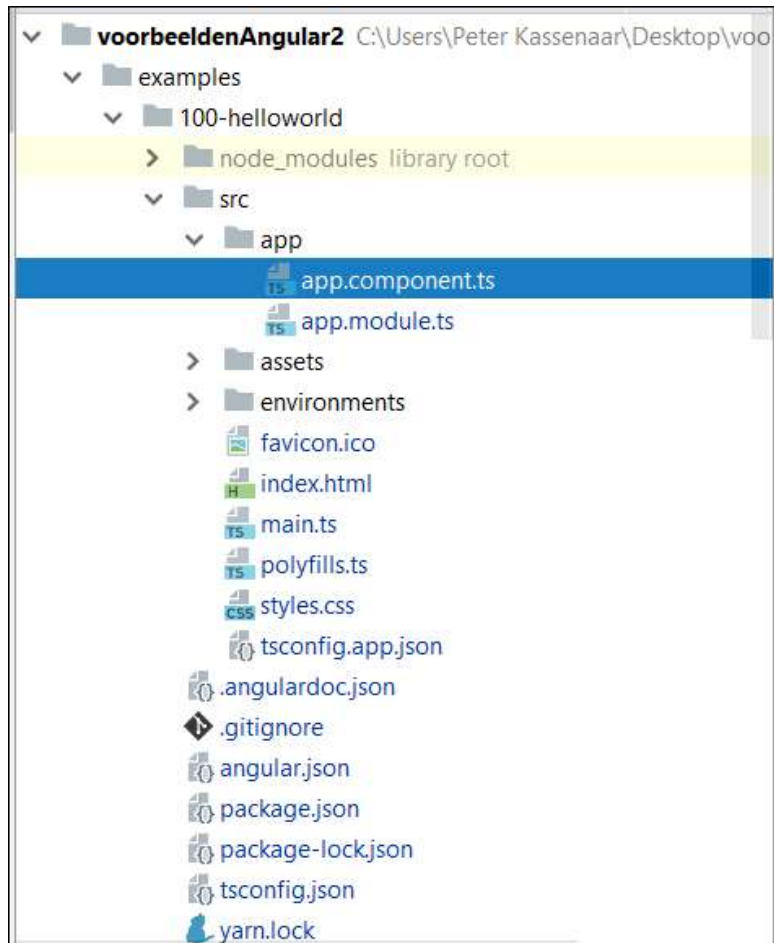
`ng serve` – start *global* angular-cli instance



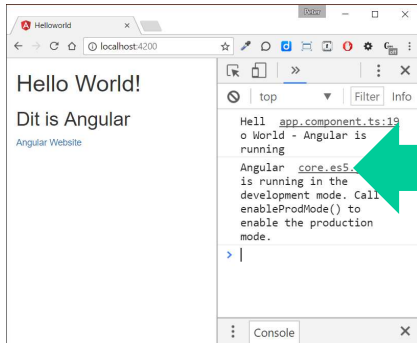
After that: edit `app.component.ts`

– Automagically refreshed through Live Reload

Basic Project Structure



Existing Angular apps



main.ts / bootstrapper

ngModule / root module

AppComponent

Services

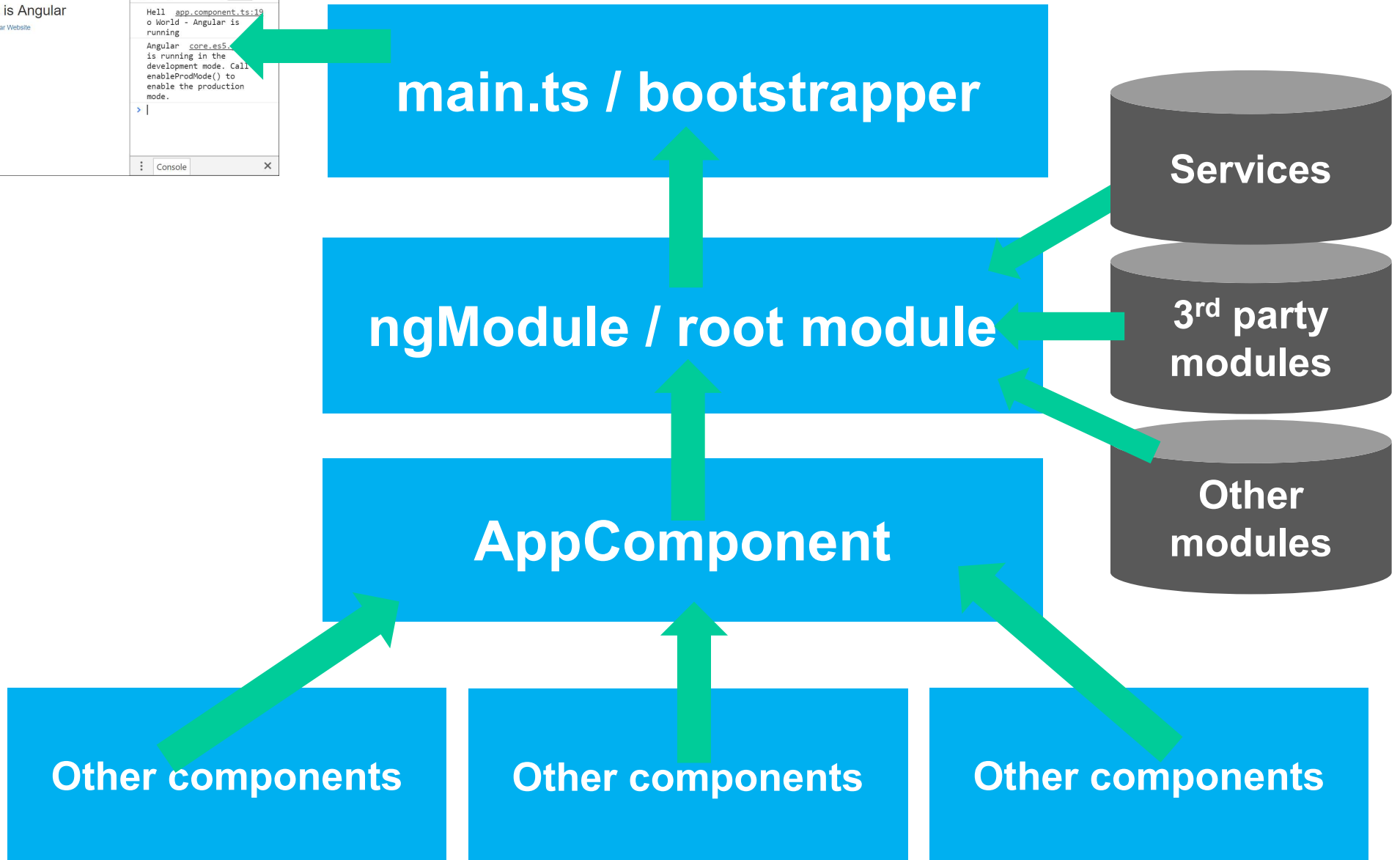
3rd party modules

Other modules

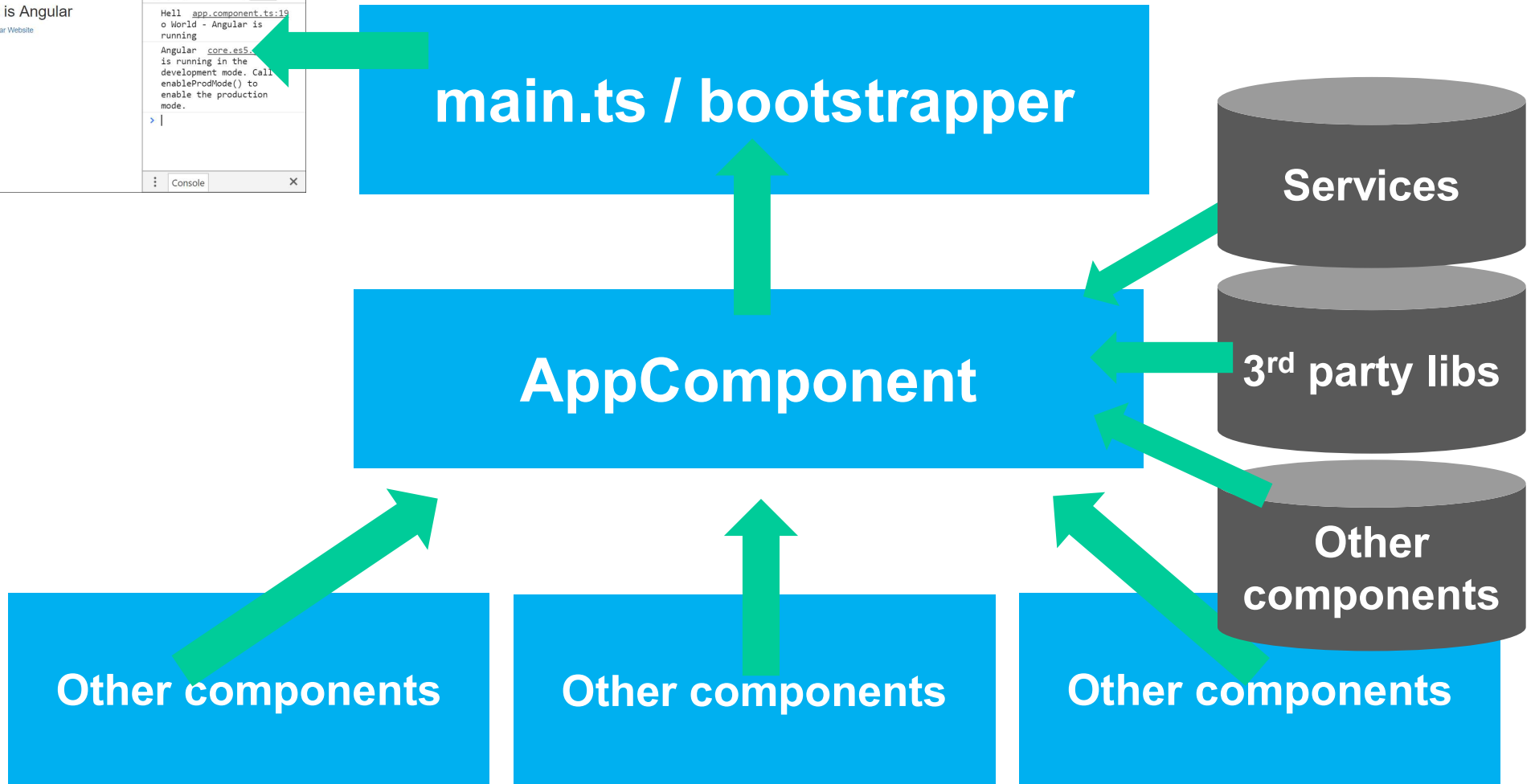
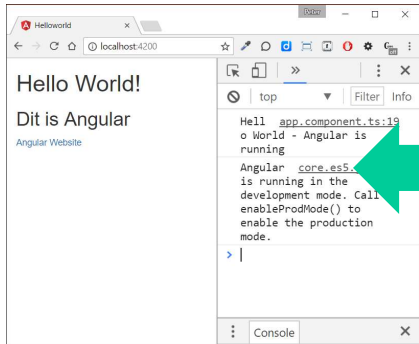
Other components

Other components

Other components



Modern Angular apps



(Note: no Module anymore, as this is now handled by standalone components)

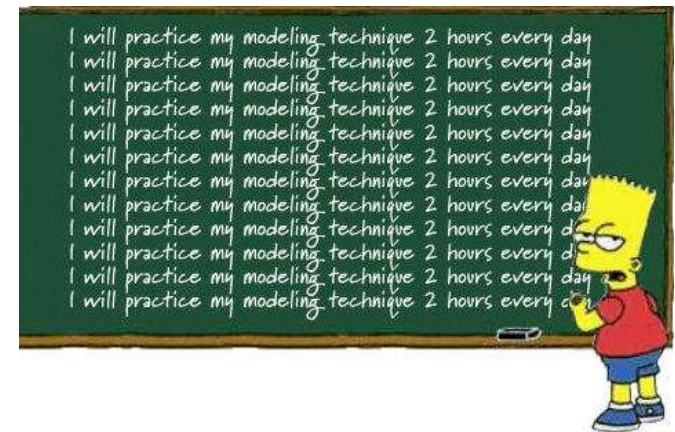
Assets

github.com/PeterKassenaar/angular-fundamentals

Example code

Workshop on Angular CLI

- Install Angular CLI from cli.angular.io
 - `npm install -g @angular/cli`
- Create a new project using `ng new <projectName>`
- Run the new project using `ng serve`
- Remove the default HTML in `app.component.html` – replace it with your own HTML
- Add at least one new component using `ng generate component <componentName>`
- Load this component in your `AppComponent`





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
 - ...

The Angular CLI

The Angular CLI is a command-line interface tool which allows you to scaffold, develop, test, deploy, and maintain Angular applications directly from a command shell.

Angular CLI is published on npm as the `@angular/cli` package and includes a binary named `ng`. Commands invoking `ng` are using the Angular CLI.

Try Angular without local setup

If you are new to Angular, you might want to start with [Try it now!](#), which introduces the essentials of Angular in the context of a ready-made basic online store app for you to examine and modify. This standalone tutorial takes advantage of the interactive [StackBlitz](#) environment for online development. You don't need to set up your local environment until you're ready.

Getting Started

Install Angular CLI to create and build your first app.

[Get Started](#)

Command Reference

Discover CLI commands to make you more productive with Angular.

[Learn More](#)

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

ng version

```
Windows PowerShell
Angular CLI

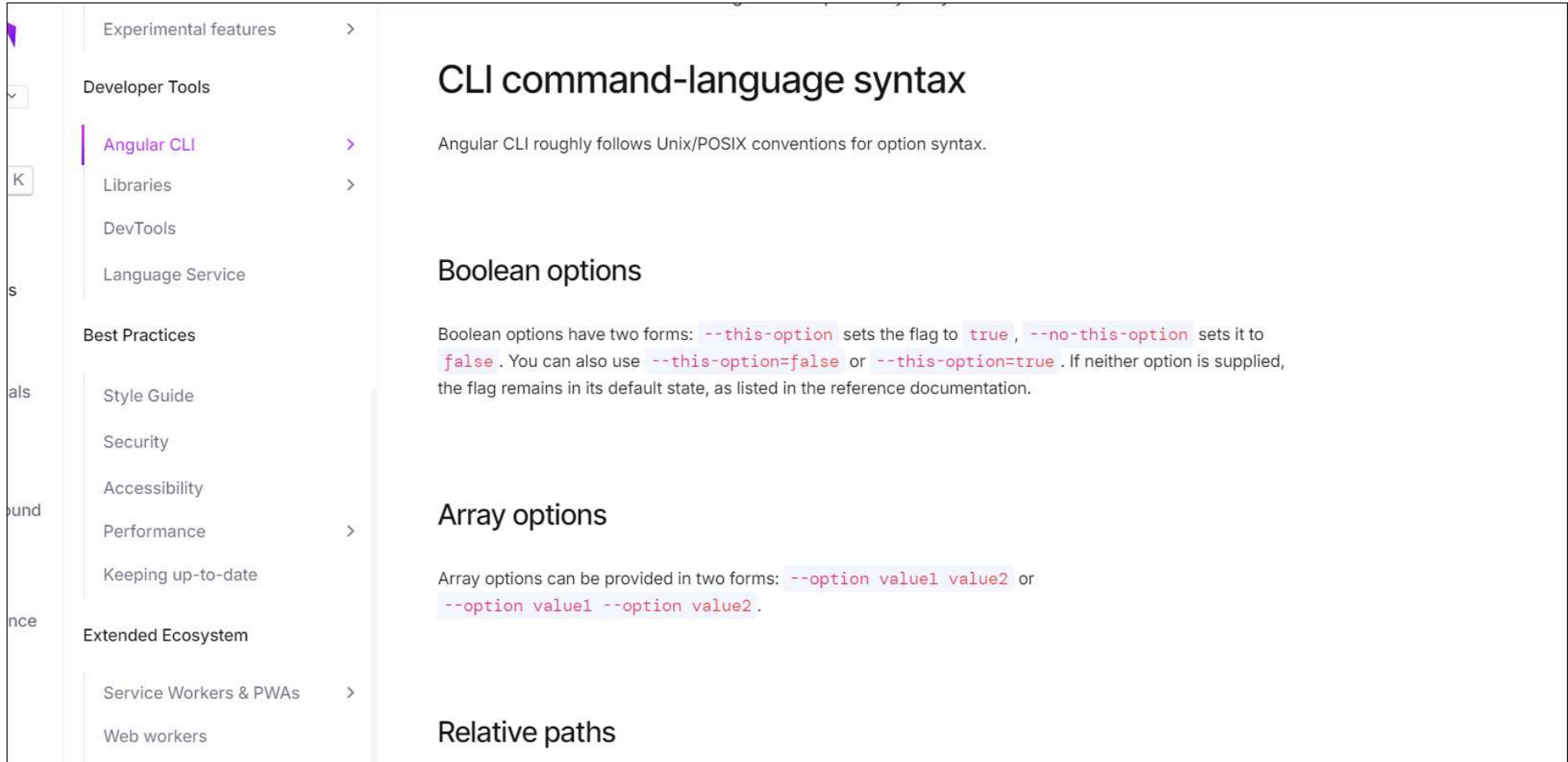
Angular CLI: 17.3.10
Node: 20.11.1
Package Manager: npm 10.2.4
OS: win32 x64

Angular: 17.3.12
... animations, common, compiler, compiler-cli, core, forms
... platform-browser, platform-browser-dynamic, router

Package                                Version
-----
@angular-devkit/architect              0.1703.10
@angular-devkit/build-angular          17.3.10
@angular-devkit/core                   17.3.10
@angular-devkit/schematics             17.3.10
@angular/cli                           17.3.10
@schematics/angular                   17.3.10
rxjs                                   7.8.1
typescript                             5.4.5
zone.js                                0.14.10

PS C:\Users\info\Desktop\angular-fundamentals\examples\100-helloworld> |
```

Documentation - in the Angular Docs



The screenshot shows the Angular CLI documentation page. The sidebar on the left contains the following sections:

- Experimental features >
- Developer Tools
 - Angular CLI >
 - Libraries >
 - DevTools
 - Language Service
- Best Practices
 - Style Guide
 - Security
 - Accessibility
 - Performance >
 - Keeping up-to-date
- Extended Ecosystem
 - Service Workers & PWAs >
 - Web workers

The main content area is titled "CLI command-language syntax" and includes the following sections:

- CLI command-language syntax**

Angular CLI roughly follows Unix/POSIX conventions for option syntax.
- Boolean options**

Boolean options have two forms: `--this-option` sets the flag to `true`, `--no-this-option` sets it to `false`. You can also use `--this-option=false` or `--this-option=true`. If neither option is supplied, the flag remains in its default state, as listed in the reference documentation.
- Array options**

Array options can be provided in two forms: `--option value1 value2` or `--option value1 --option value2`.
- Relative paths**

angular.dev/cli



Angular Code - "Backend"

Brief – on TypeScript and ES6

Programmeertalen

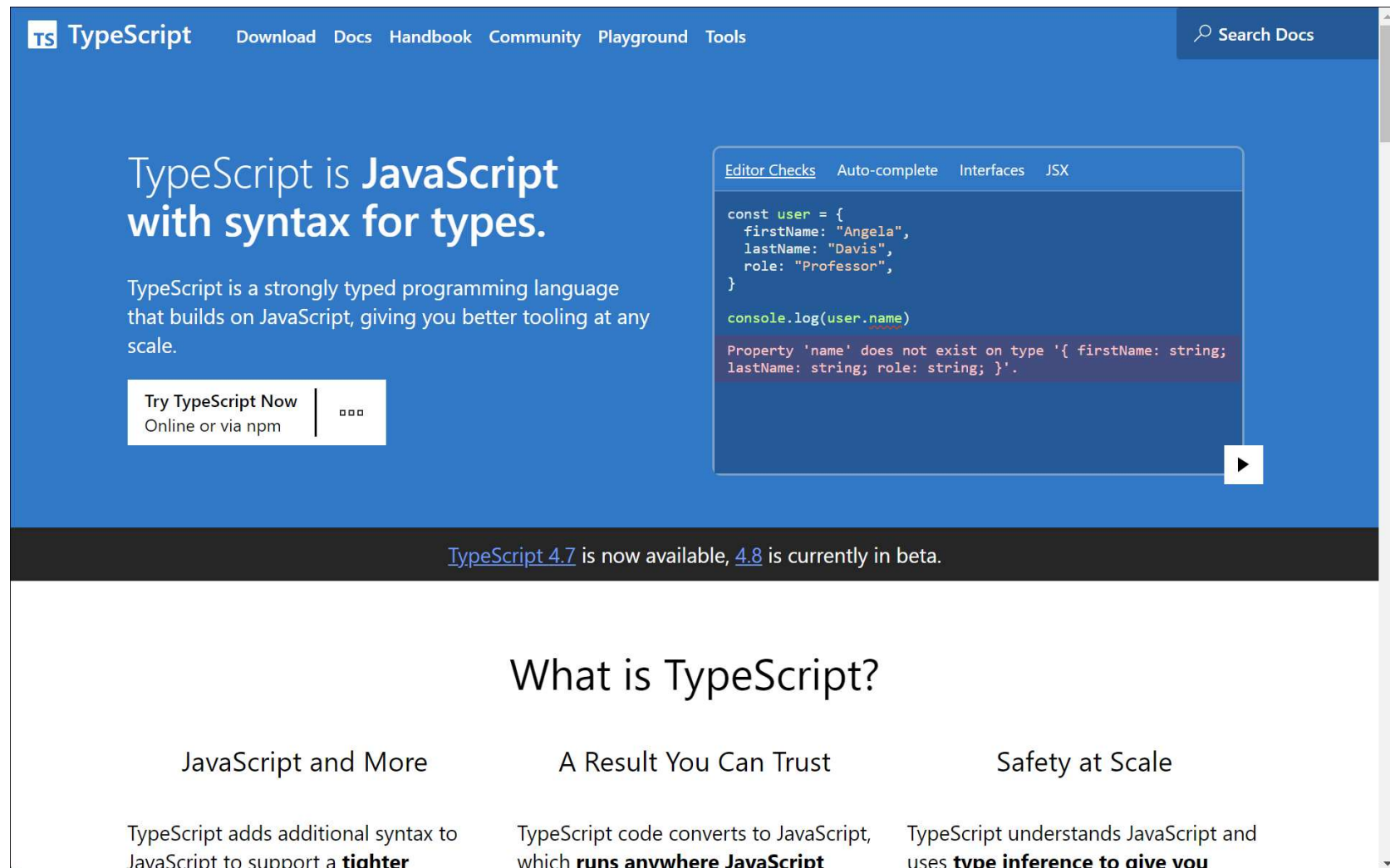


TypeScript



- **Limited** in this module
- As far as necessary to use Angular
- But, **TypeScript is not limited to Angular!**
- General **superset** of JavaScript
- ***Compiles to plain JavaScript***, that browsers execute
- Two tasks
 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: 'JavaScript and More', 'A Result You Can Trust', and 'Safety at Scale', each with a brief description of TypeScript's features.

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



TypeScript

ES2022

ES2020

ES2018

ES6 and TypeScript

'Modern JavaScript', since ES6/ES2015

Major update of JavaScript as a programming language

Modules, classes and more

Required in Angular, optional in JavaScript, Vue, React, ...

TypeScript is a typed superset of ES6:

Annotations & types

Interfaces

Compiler, ... and more

Architecture of your Component Class

imports

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

annotations

```
@Component({  
  selector: 'orders',  
  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