



Angular Fundamentals

Module - Inleiding



Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

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

www.kassenaar.com/blog

info@kassenaar.com

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



ING

OHRA

euricom
A DIMENSION DATA COMPANY

sanoma

delta lloyd

zenito
BETERE ZEKERHEID
VOOR ONDERNEMERS

Atos

woonbron
OBERON INTERACTIVE

ROC West-Brabant

the eforum
FACTORY

Angulartraining.nl

Home

Training

Dates

Information

Contact

2018 dates now available!



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/incentro

About you...



Voorkennis webdevelopment, (mobile/web-) apps?

Kennis AngularJS 1.x?

Voorkennis andere (web)talen?

Specifieke verwachtingen?

Concrete projecten?

Intake

Ik heb de deelnemers ook gevraagd "wat wil je leren?" Hierop werd meestal geantwoord "alles" maar de specifieke vragen die ik tot nu toe heb gehad van deelnemers zijn:

- wat zijn en hoe werken Services, observable,
- Hoe het precies zit met de componenten en databinding.
- Hoe werkt het concept van Services en dependency injection
- Hoe maak je een relatie tussen Directives, Services en Componenten.

Agenda -

- Middag
 - Theorie - Introductie & geschiedenis - waarom Angular (2/4/5)?
 - Hello World in Angular – inzicht in boilerplate-code
 - Components & Decorators
 - Data binding – 4 methoden
- Avond
 - Services en Dependency Injection (DI)
 - Observables (RxJS)
 - Services en HTTP, Live API's
 - Next Steps...

2 Richtlijnen

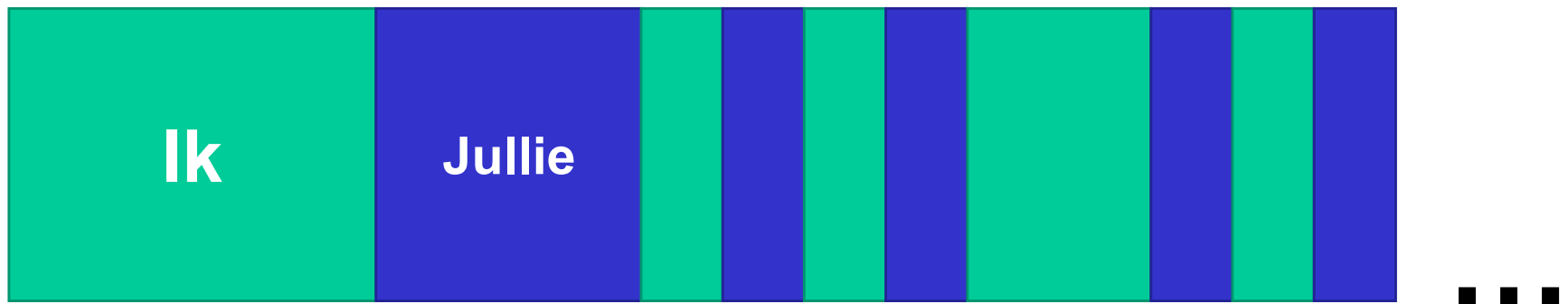
1. Oefeningen

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

2. Voorbeeldcode

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

Globale werkwijze



Vragen?



Angular 1 vs. Angular 2

Differences, similarities, new features



MV*

Framework



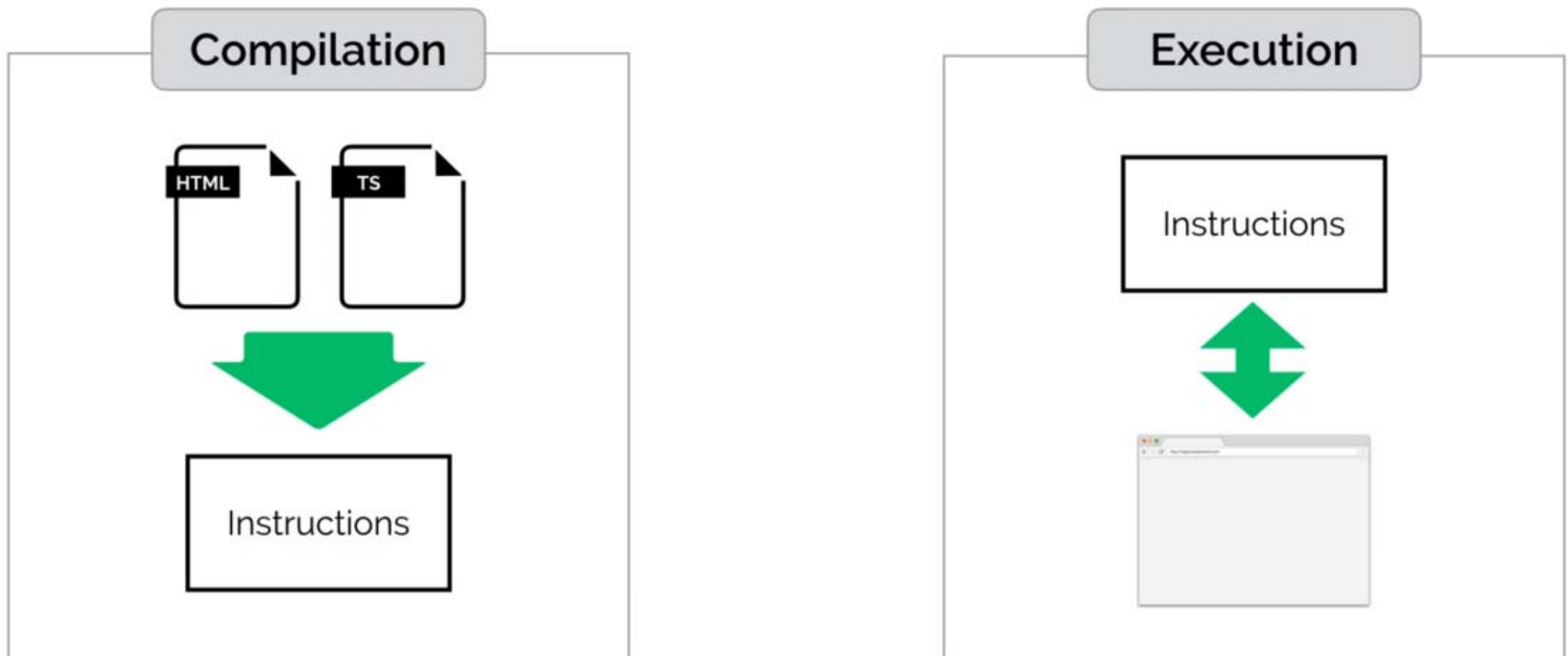
Platform

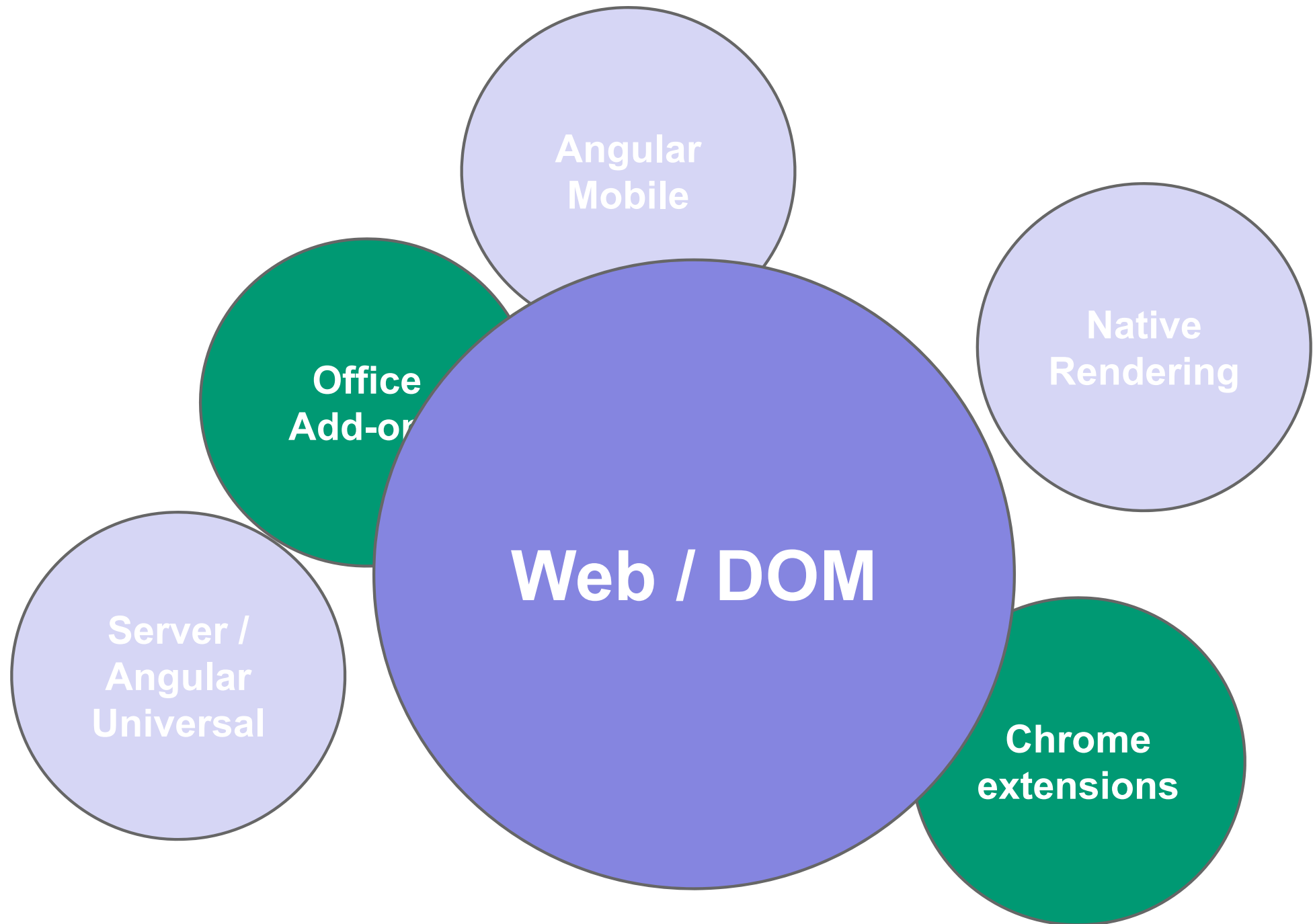
Framework to Platform

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

"Compile" Angular Applications?

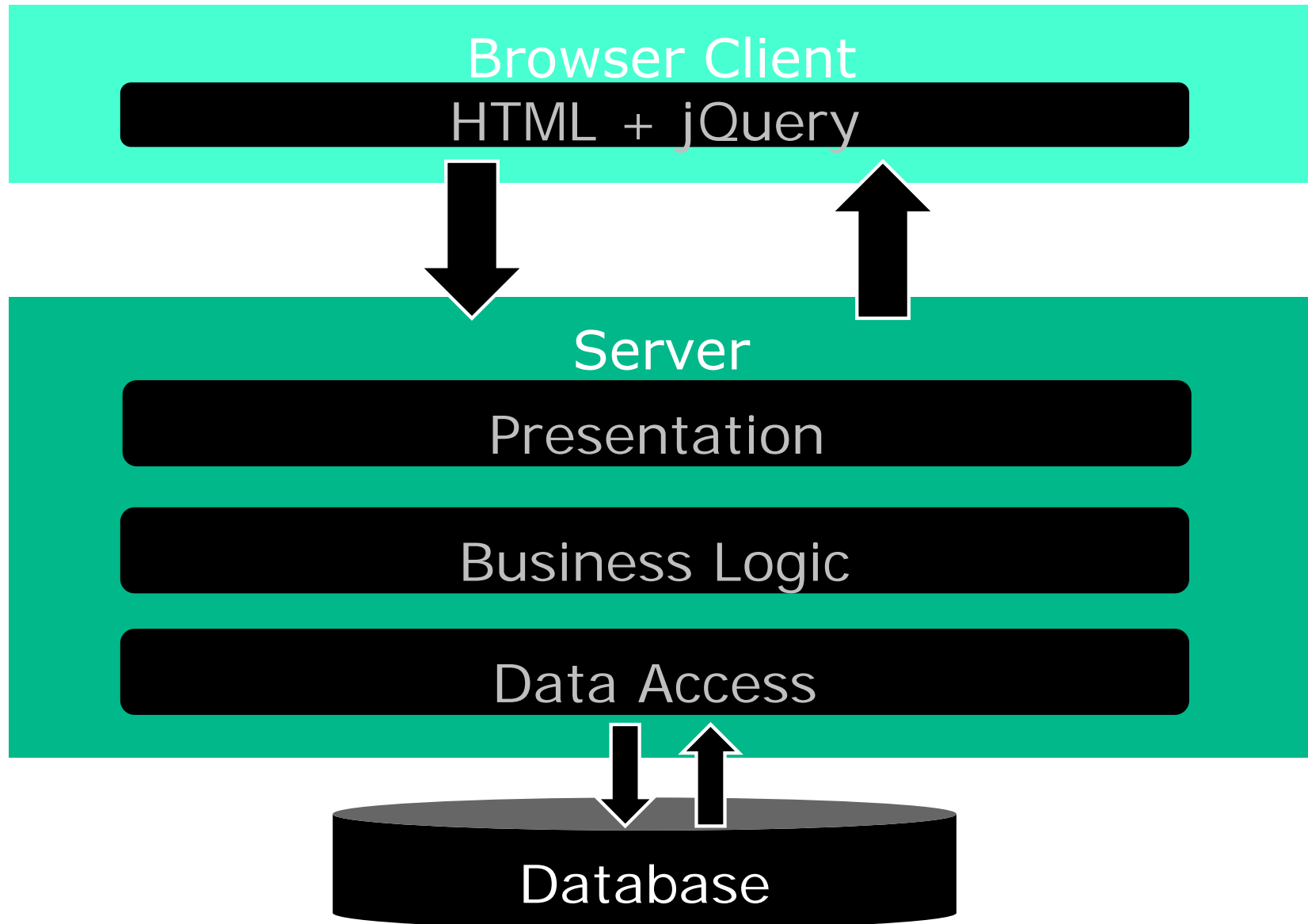
The Two Render Phases



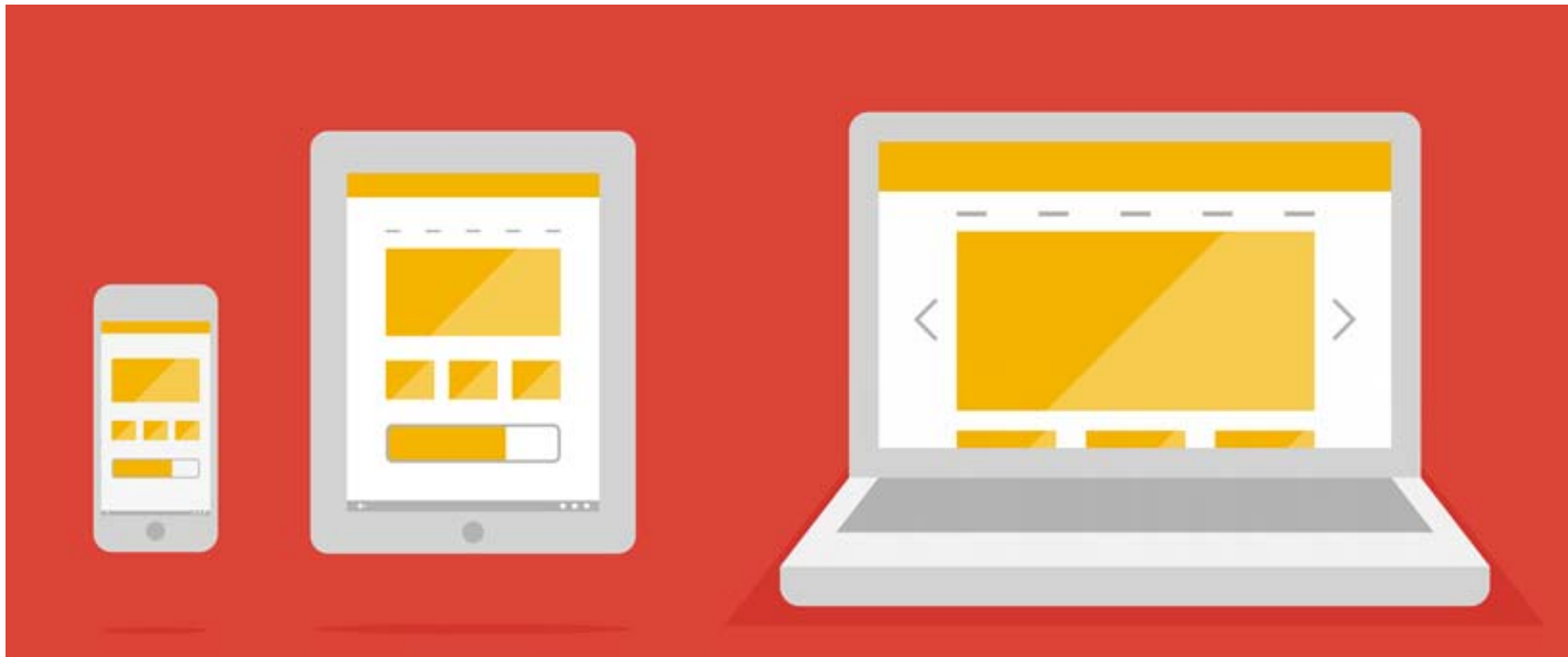


Conventional Web App

2000 - 2013

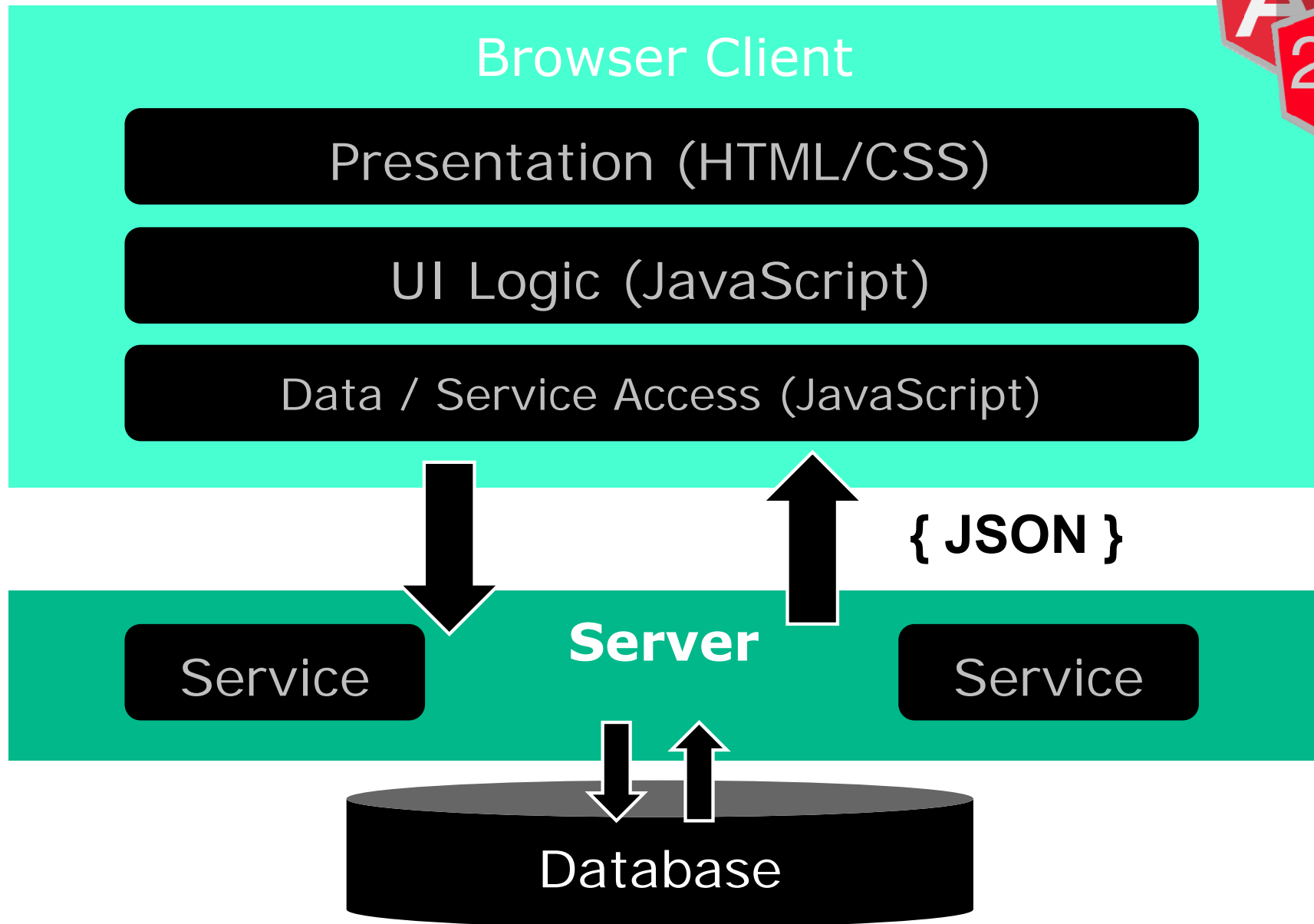


Maar: ca. 2010 -



Single Page Application

2010 – 20??





Summary

Angular 4

March 2017

Backwards Compatible w/ Angular 2



Predictable, Transparent & Incremental Evolution

Angular 5 - September / October 2017

Angular 6 - March 2018

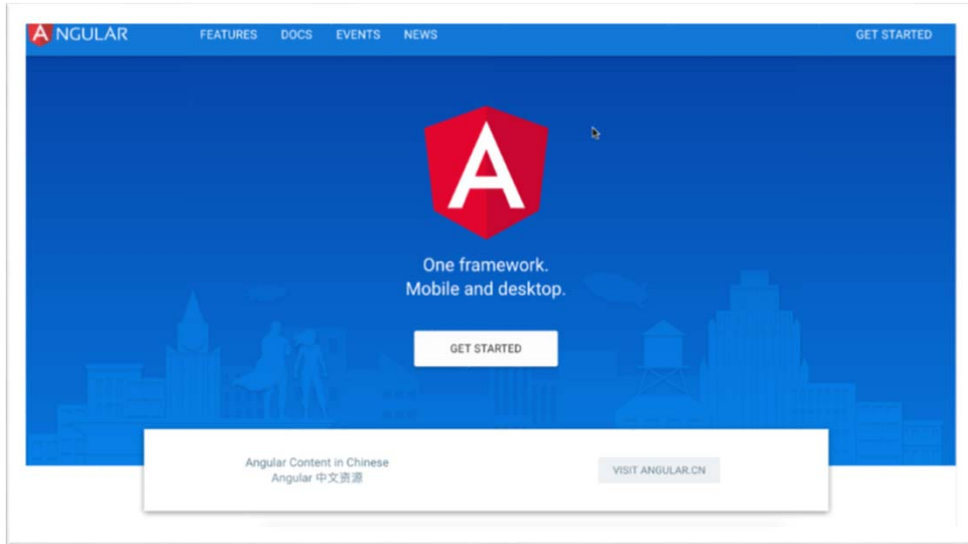
Angular 7 - September / October 2018

(tentative schedule)

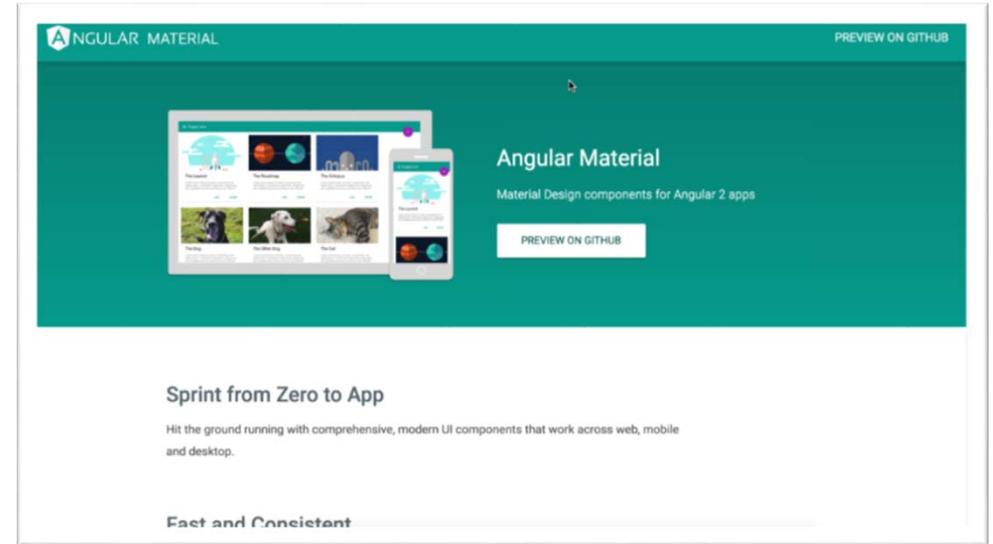
"It's just

Angular

Angular as a Platform



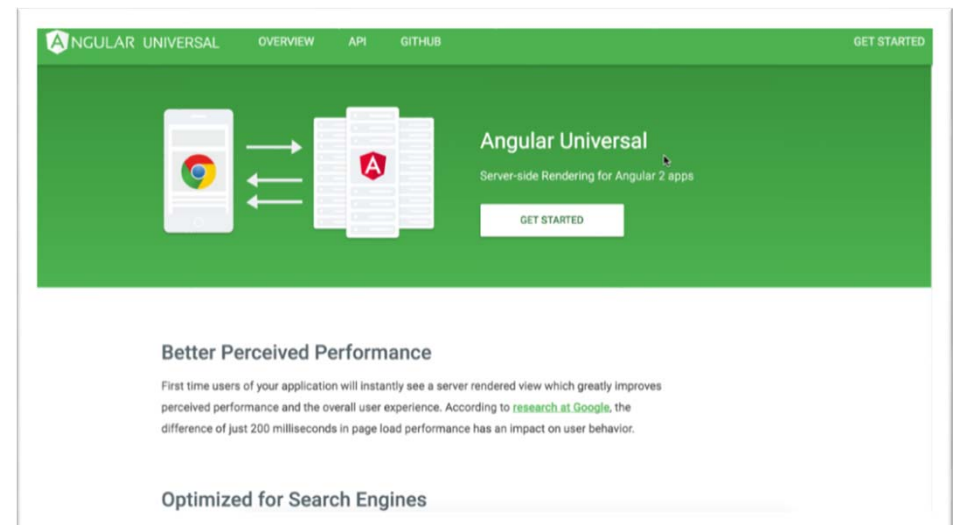
<https://angular.io/>



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

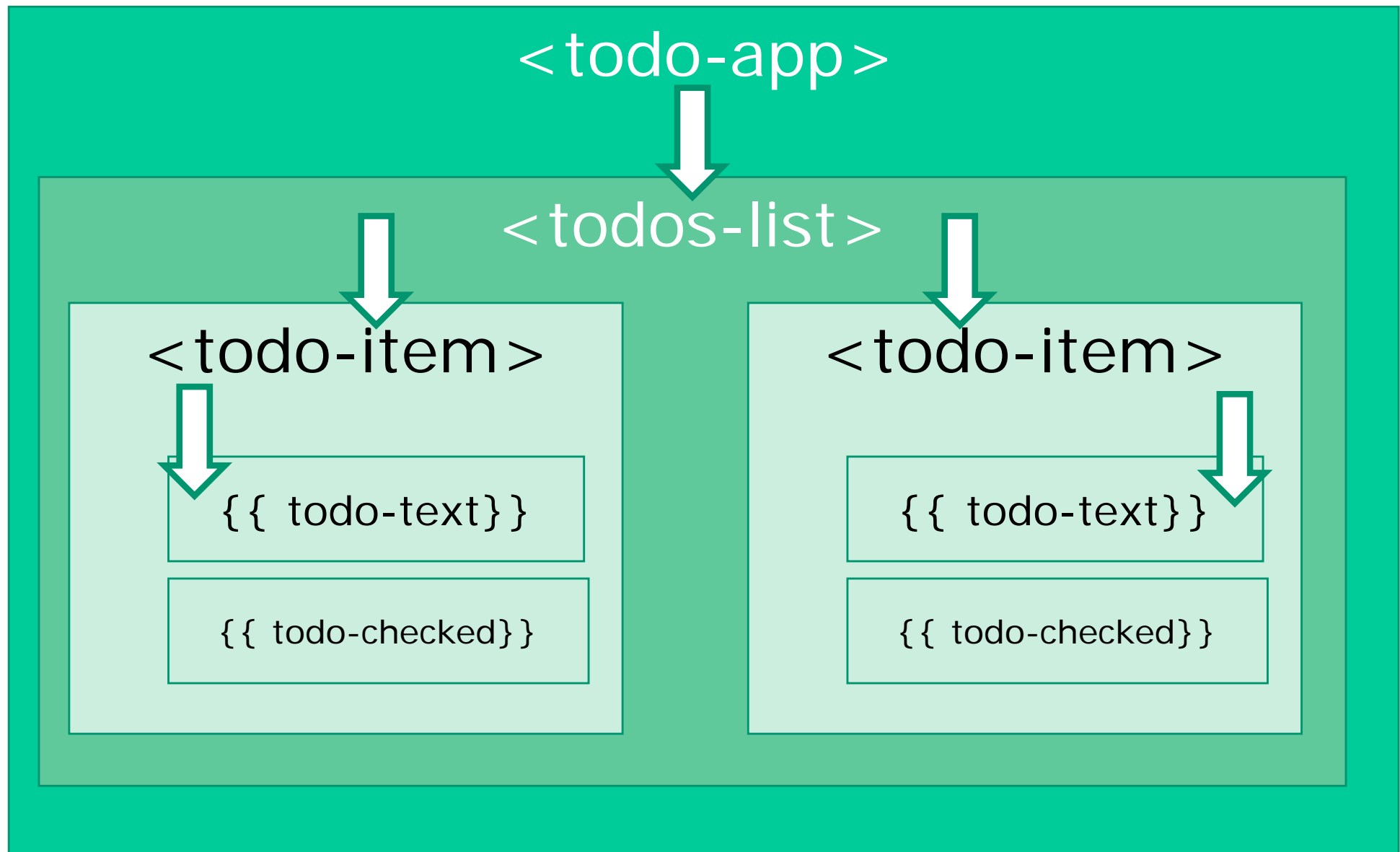


<https://cli.angular.io/>



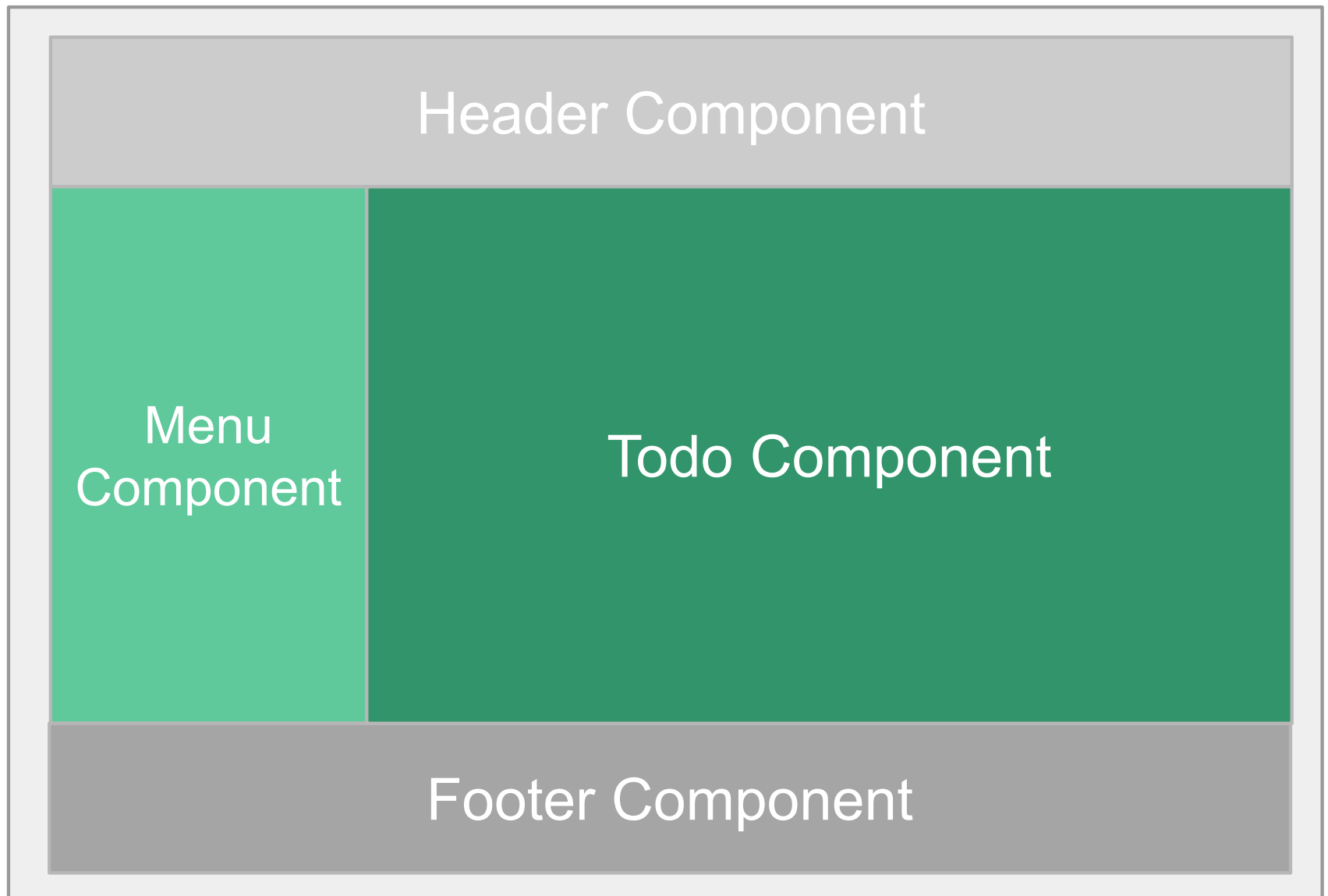
<https://universal.angular.io/>

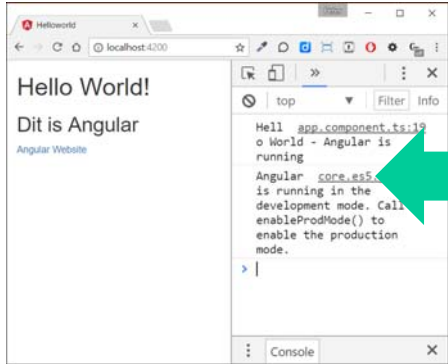
Angular 2 - components



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

Components – visual representation





main.ts / bootstrapper

ngModule / root module

AppComponent

Services

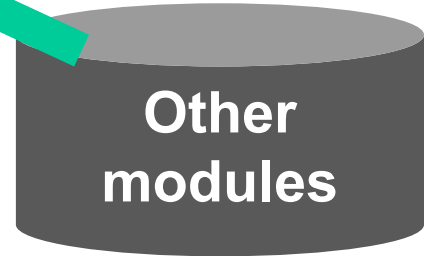
3rd party libs

Other modules

Other components

Other components

Other components



Angular 2

Components

Data binding

Services

TypeScript

Routing

Classes

Form

Event binding

Observables

Templates

Pipes

Views

...





Let's write some code

Hello World in Angular 2

Angular 1:

```
<script src="angular.min.js"></script>
```

Angular development dependency: NodeJS 6.9+



The screenshot shows the Node.js website homepage. At the top is the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the navigation bar is a large banner for Node.js Interactive. The banner features the Node.js logo and the text: "Join us for Node.js Interactive happening in Vancouver, Canada October 4 - 6, 2017". Below the banner, there is a paragraph describing Node.js: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world." Below this paragraph is a section titled "Download for Windows (x64)" with two buttons: "v6.10.3 LTS Recommended For Most Users" and "v7.10.0 Current Latest Features". At the bottom of the page, there are links for "Other Downloads", "Changelog", and "API Docs" for both versions.

node

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

node JS Interactive
NORTH AMERICA
Vancouver, Canada
October 4-6, 2017

Join us for Node.js Interactive
happening in Vancouver, Canada
October 4 - 6, 2017

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

Download for Windows (x64)

v6.10.3 LTS
Recommended For Most Users

v7.10.0 Current
Latest Features

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

Exercise

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

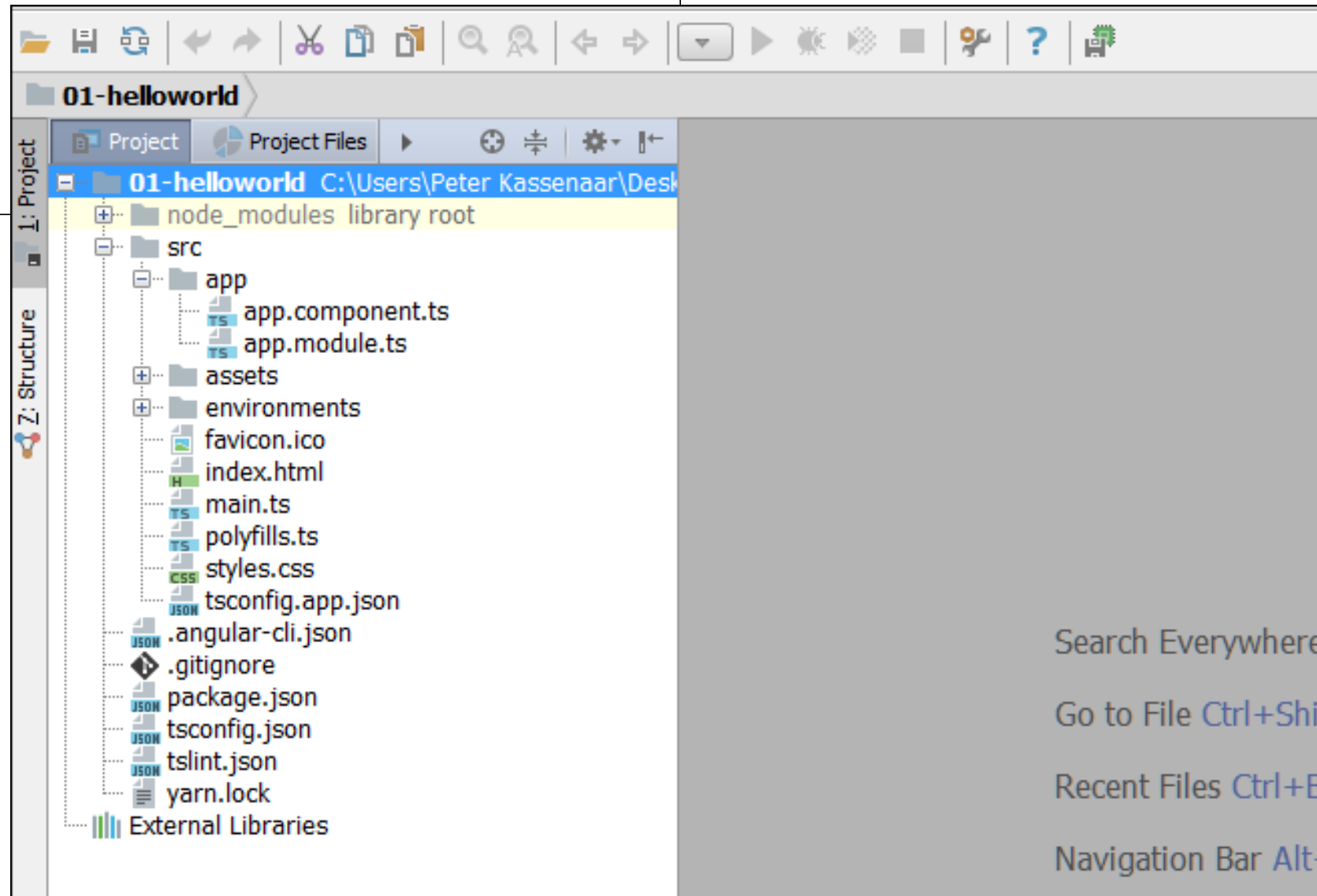
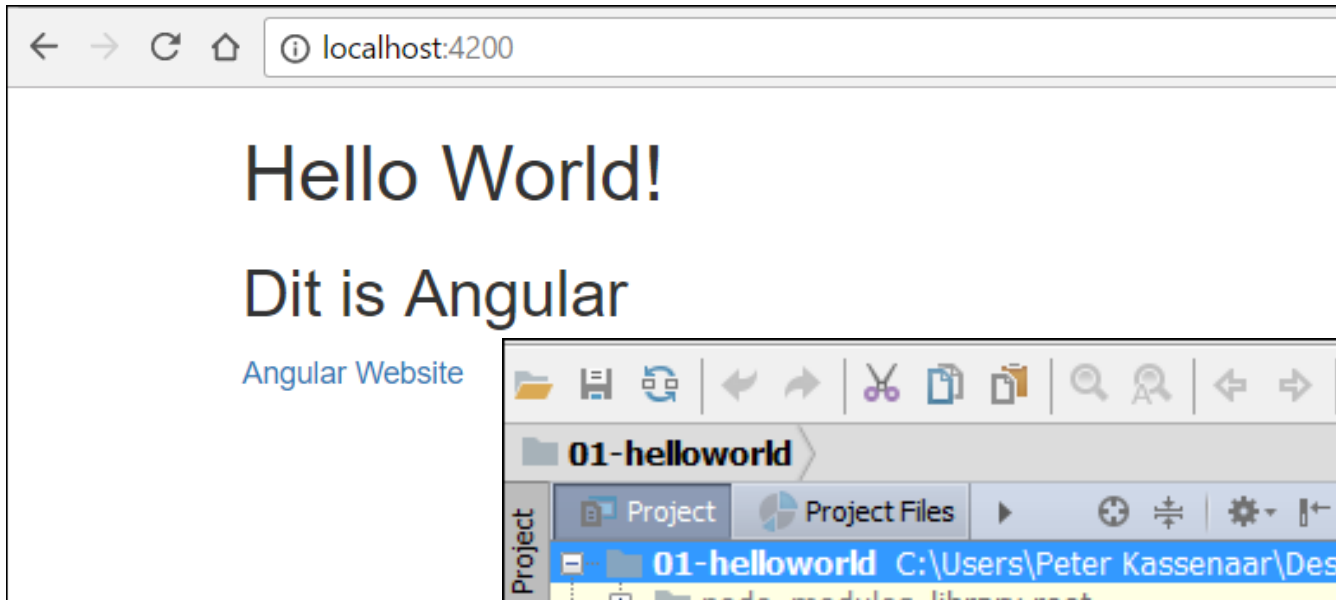
```
cd oefeningen
```

```
cd 100-helloworld
```

```
npm install (or yarn install)
```

```
npm start
```

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



Boilerplate code for Hello World

Steps

1. Set up environment, boilerplate & libraries
2. Write Angular Root Component
3. Write @ngModule Component
4. Bootstrap component
5. Write HTML-pagina (`index.html`)



Boilerplate files #1 - package.json

```
{
  "name"          : "hello-angular",
  "version"       : "1.0.0",
  "scripts"      : {
    "start"       : "tsc && concurrently \"npm run tsc:w\" \"npm run lite\" ",
    "lite"        : "lite-server",
    "postinstall" : "typings install",
    "tsc"         : "tsc",
    "tsc:w"       : "tsc -w",
    "typings"     : "typings"
  },
  "license"      : "ISC",
  "dependencies" : {
    "@angular/compiler" : "^5.2.0",
    "@angular/core"     : "^5.2.0",
    ...
    "rxjs"              : "5.5.0",
    "zone.js"           : "^0.6.12",
    "bootstrap"         : "^4.0.0"
  },
  "devDependencies": {
    "concurrently" : "^2.0.0",
    "lite-server"  : "^2.2.0",
    "typescript"   : "^2.4.0",
    "typings"      : "^0.8.1"
  },
  "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-cli.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "name": "helloworld"
  },
  "apps": [
    {
      "root": "src",
      "outDir": "dist",
      "assets": [
        "assets",
        "favicon.ico"
      ],
      "index": "index.html",
      "main": "main.ts",
      "prefix": "app",
      "styles": [
        "styles.css"
      ],
      "scripts": [],
      "environmentSource": "environments/environment.ts",
      ...
    }
  ].....
}
```

"Nice to have" - non-essential files

The screenshot shows the GitHub interface for the `angular / quickstart` repository. At the top, there are buttons for `Watch` (250), `Star` (2,567), and a link to `Issues` (36). Below this, there are tabs for `Code`, `Issues`, `Pull requests` (8), `Projects` (0), `Wiki`, `Pulse`, and `Graphs`. The current view is the `quickstart / non-essential-files.txt` file page. 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`. There are buttons for `Raw`, `Blame`, and `History`. The file content is a list of 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-cli.json`

Filename: `src/app/app.component.ts`

```
import {Component} from '@angular/core';  
  
@Component({  
  selector: 'hello-world',  
  template: '<h1>Hello Angular 2</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 {  
}
```

This is the entry point 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

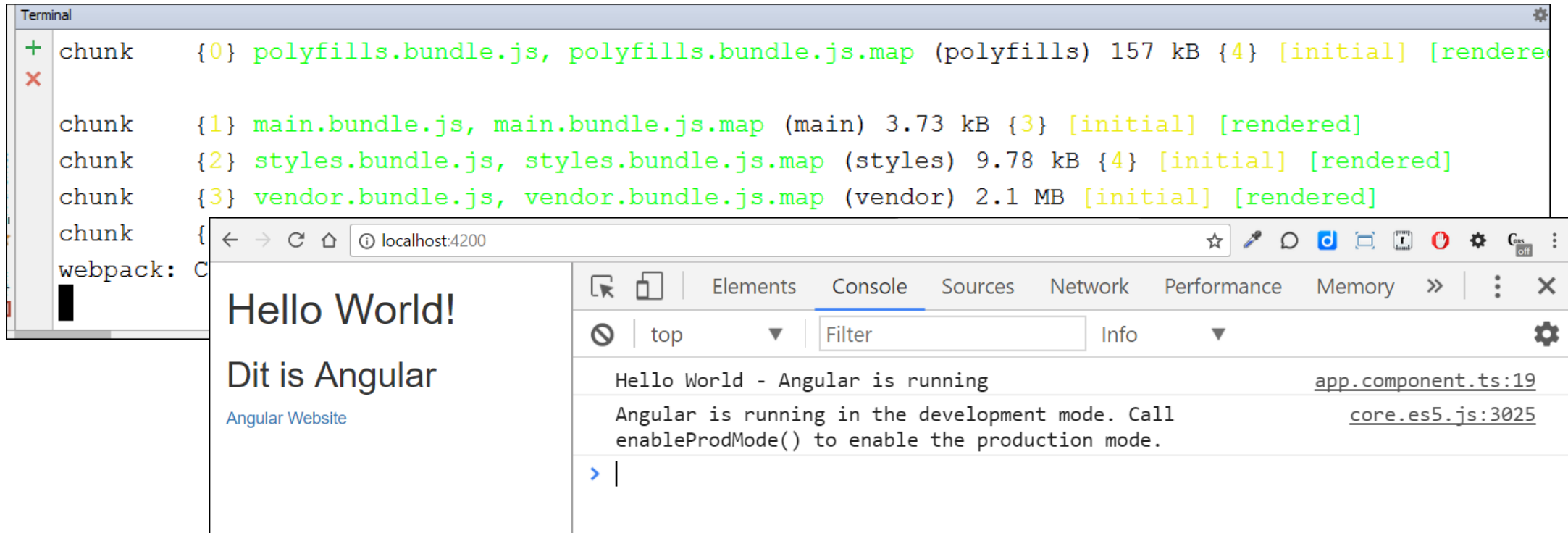
Verwijzing naar de root-component:

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

App draaien

`npm start` – draait de script opdracht start uit `package.json`.

`ng serve` - start globale angular-cli instantie

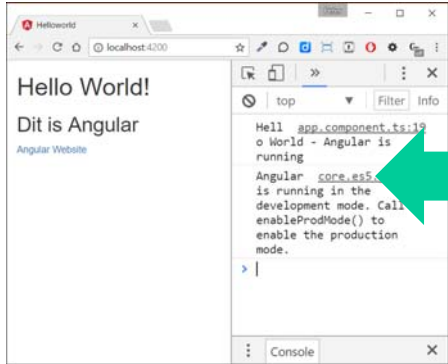


Daarna: wijzigingen aanbrengen in `app.component.ts`

– worden opgepikt door Live Reload

Structuur

01-helloworld		src > app > app.component.ts	
Project	Project Files		package
1: Project	01-helloworld C:\Users\Peter Kassenaar\Desktop	node_modules library root	Compile Ty
Z: Structure	src		1
	app		2
	app.component.ts		3
	app.module.ts		4
	assets		5
	environments		6
	favicon.ico		7
	index.html		8
	main.ts		9
	polyfills.ts		10
	styles.css		
	tsconfig.app.json		
	.angular-cli.json		
	.gitignore		
	package.json		
	tsconfig.json		
	tslint.json		
	yarn.lock		
	External Libraries		



main.ts / bootstrapper

ngModule / root module

AppComponent

Services

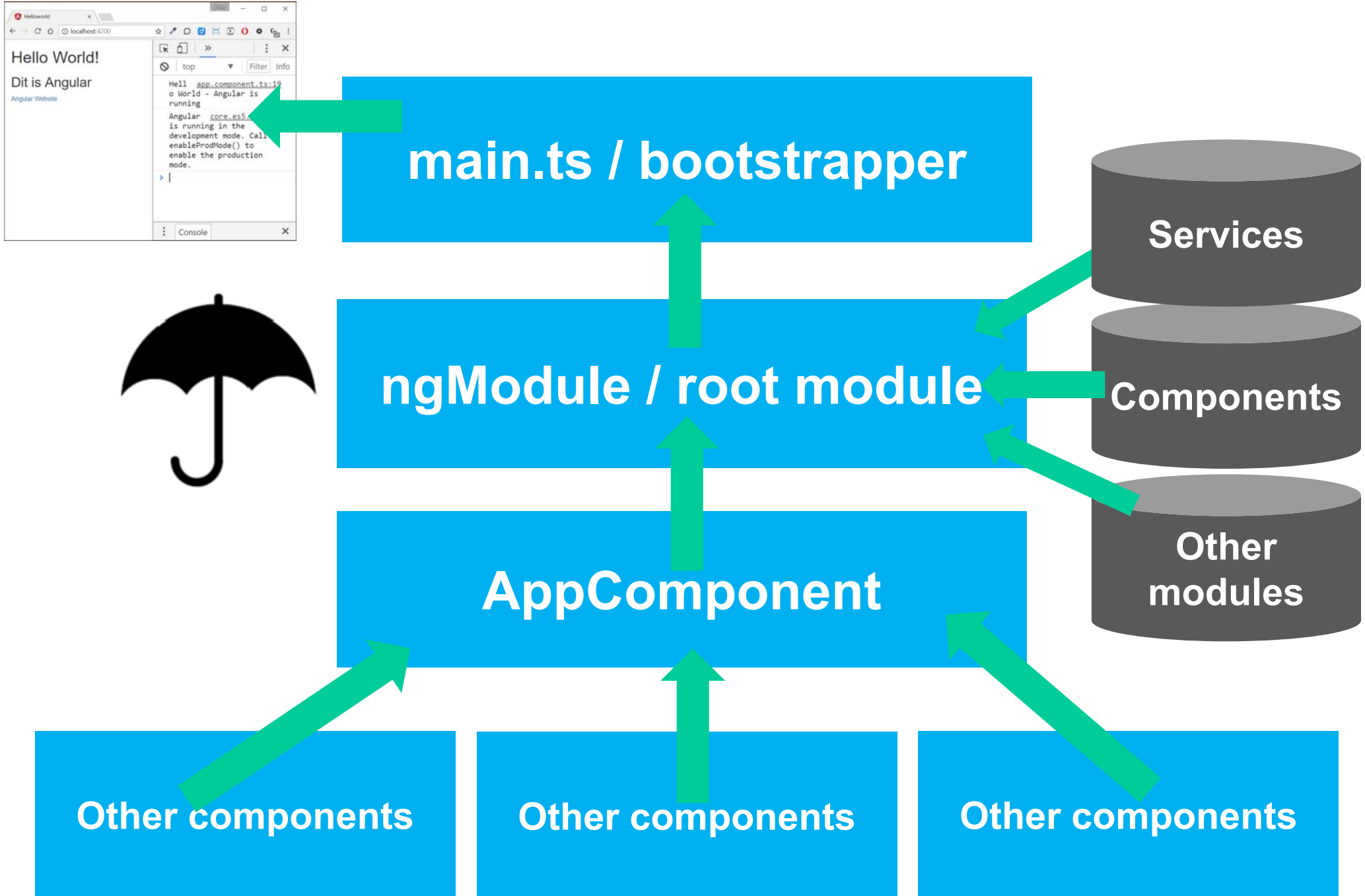
Components

Other modules

Other components

Other components

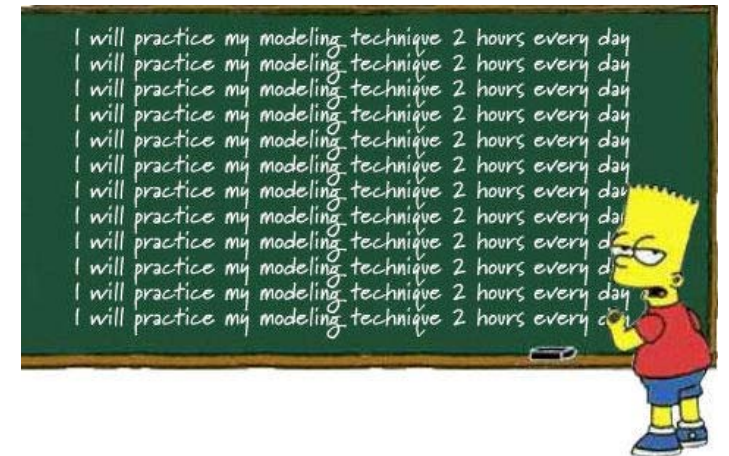
Other components



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....



Assets

github.com/PeterKassenaar/voorbeeldenAngular2

Oefeningen en meer voorbeeldcode



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/>

typings.json

chore(): add typings files to assist devs

4 months ago

README.md

Angular-CLI

chat

on

gitter

build

passing

dependencies

up to date

devDependencies

up to date

npm

v1.0.0-beta.5

Prototype of a CLI for Angular 2 applications based on the [ember-cli](#) project.

Note

This project is very much still a work in progress.

We still have a long way before getting out of our alpha stage. If you wish to collaborate while the project is still young, check out [our issue list](#).

Prerequisites

The generated project has dependencies that require **Node 4 or greater**.

Table of Contents

- [Installation](#)
- [Usage](#)
- [Generating a New Project](#)
- [Generating Components, Directives, Pipes and Services](#)
- [Generating a Route](#)
- [Creating a Build](#)
- [Running Unit Tests](#)
- [Running End-to-End Tests](#)
- [Deploying the App via GitHub Pages](#)
- [Linting and formatting code](#)
- [Support for offline applications](#)
- [Commands autocompletion](#)
- [CSS preprocessor integration](#)
- [3rd Party Library Installation](#)

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



```
> npm install -g angular-cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

Angular CLI

A command line interface for Angular

GET STARTED

ng new

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

ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.



CLI can...

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

NG
CONF

4:50 / 23:39

Learn Clingon - Mike Brocchi

The image shows a YouTube video player interface. The main content area displays a terminal window with the text 'CLI can...' and a command prompt showing the command '> ng generate component my-comp'. To the right of the terminal, there is a small inset video of a man in a green shirt standing on a stage with a circuit board background. Below the terminal, there is a large 'NG CONF' logo. At the bottom of the video player, there is a progress bar showing '4:50 / 23:39' and standard video controls. Below the video player, the title 'Learn Clingon - Mike Brocchi' is displayed.

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

Main commands

ng new

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

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



scotch.io
code on the rocks

LEARN

- Tutorials
- Bar Talk
- Community

Scotch School

OUR STUFF

- About
- Shop
- Join Us On Slack
- Hire Us

Search Scotch for Cool Stuff

Login / Sign Up Write a Post More

Advertise Here

Popular Scotchers

-  Chris Sevilleja
-  Nicholas Cerminara
-  Holly Lloyd
-  Ado Kukic
-  Ryan Chenkie
-  Ken Wheeler
-  Alex Sears
-  Chris Nwamba
-  Become a Scotcher!

Use the Angular CLI for Faster Projects

angular2 angularJS javascript

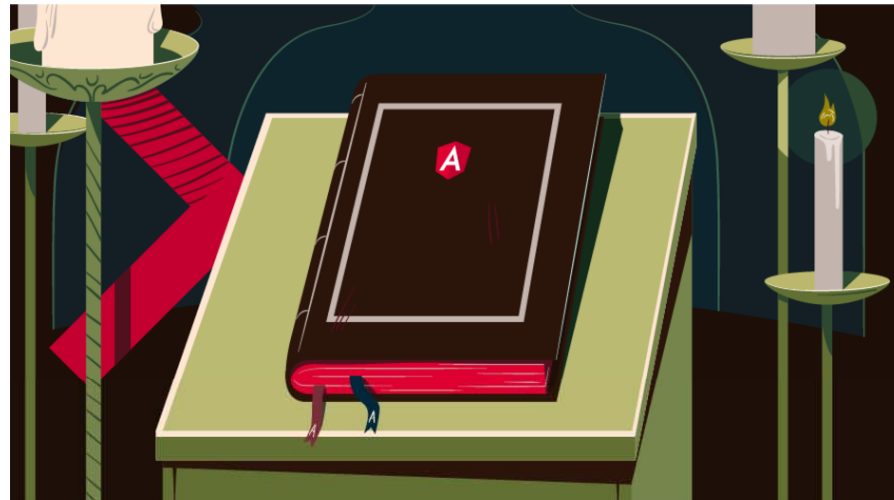
Use the Angular CLI For Faster Angular 2 Projects

Creating Angular 2 projects has never been faster than with the Angular CLI

<https://scotch.io/tutorials/use-the-angular-cli-for-faster-angular-2-projects>

The Ultimate Angular CLI Reference Guide

Related Topics:



2017.04.25: As of March 24, Angular CLI v1.0 was released. This article has been updated to reflect the latest changes. If you want to add the latest features of Angular CLI v1.0 to your existing Angular project that was generated with an earlier version of Angular CLI, check out the [Angular CLI v1.0 migration guide](#).

2017.02.17: As of February 9, 2017, the `ng deploy` command has been removed

SPONSORS



- JAVASCRIPT

An Introduction to Component Routing with Angular Router

Jurgen Van de Moere, 12 hours ago



<https://www.sitepoint.com/ultimate-angular-cli-reference/>



Angular 2 Code - Backend

Kort over TypeScript en ES6

Programmeertalen





A Venn diagram consisting of three concentric circles. The outermost circle is dark green and contains the text 'TypeScript'. Inside it is a medium green circle containing the text 'ES6'. Inside the 'ES6' circle is a light green 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

Helpt bij het ontwikkelen in Angular 2

TypeScript breidt ES6 verder uit

Annotaties & types

Interfaces

Compiler

TypeScript – tooling support

Types, Autocompletion.

Compile-time checking in editors.

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

Onderdelen van een Component Class

imports

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

annotations

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

class

```
export class OrdersComponent {  
  ...  
}
```

Checkpoint

- Angular 2 is een totaal ander framework dan Angular 1
- Component-based vs. Page-based
- Nieuwe syntaxis
- Nieuwe programmeertalen en andere nieuwe kenmerken
- Concepten komen – deels – overeen
- Voorlopig: veel boilerplate-code nodig voor een Quickstart
- Daarna: niet meer naar omkijken. Concentreren op de componenten