

# Angular Fundamentals

## day 2 - Short recap

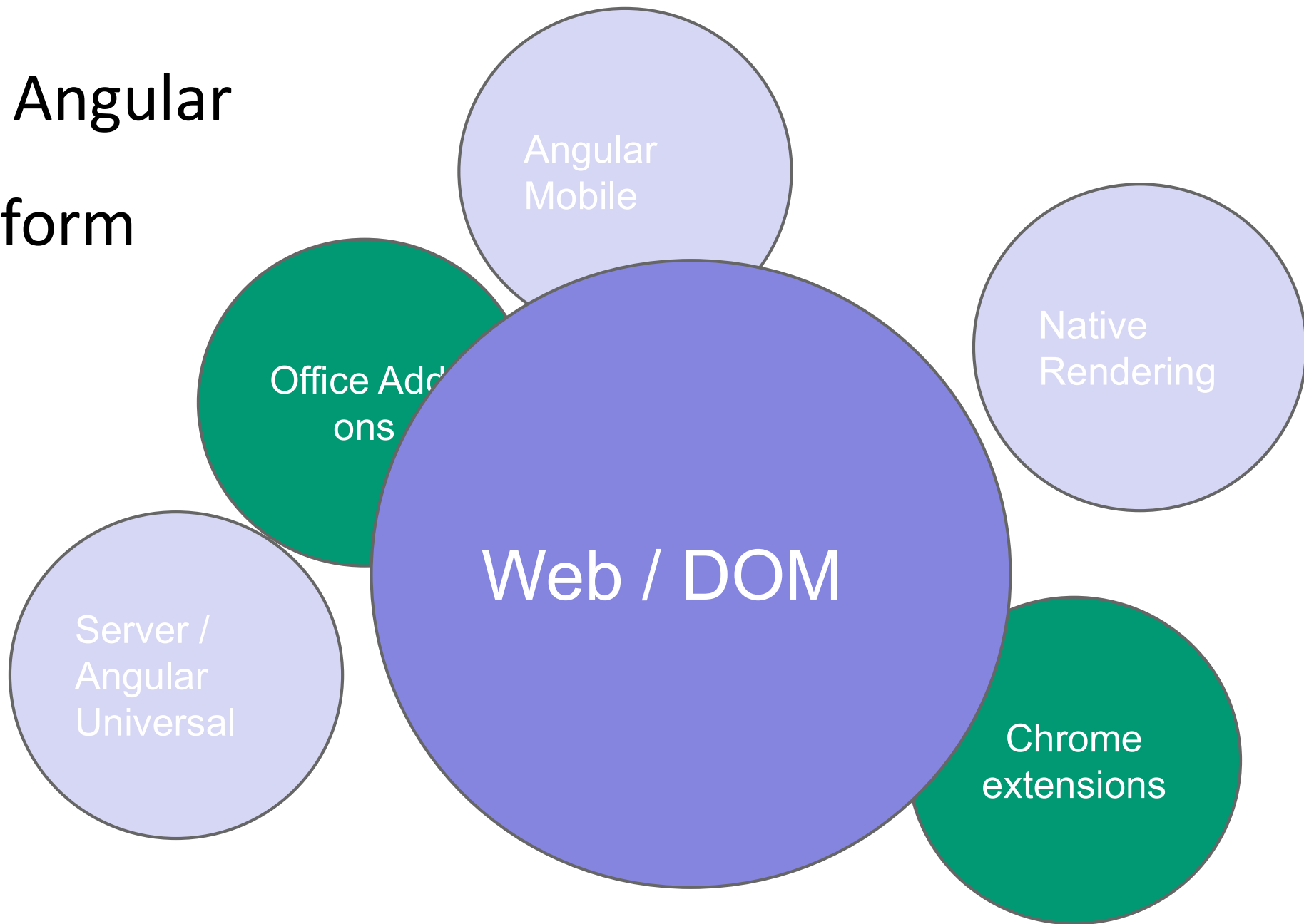


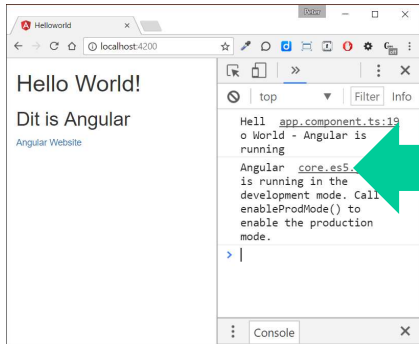
Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

# Angular Architecture – Framework to Platform

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

# The Angular Platform





main.ts / bootstrapper

ngModule / root module

AppComponent

Other components

Other components

Other components

Services

3<sup>rd</sup> party libs

Other modules



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

 freeCodeCamp(🔥)

Follow



HOMEDEV | LEARN TO CODE FOR FREE



Per Harald Borgen

Follow

Co-founder of Scrimba, the next-generation platform for teaching and learning code.  
<https://scrimba.com>.

Jul 16 · 8 min read

## Want to learn TypeScript? Here's our free 22-part course.



Project

FILES

- Index.html
- Index.js

DEPENDENCIES

### What & Why?

- Superset of JavaScript
- Built by Microsoft
- Compiles down to regular JavaScript
- Tends to be less error prone.

<https://medium.freecodecamp.org/want-to-learn-typescript-heres-our-free-22-part-course-21cd9bbb5ef5>

# Boilerplate code - steps

1. Set up environment, boilerplate & libraries

- `npm install`

2. Write Angular Components & Module for app

3. Bootstrap (`main.ts`)

4. Write HTML-pagina (`index.html`)

5. Run application

- `npm start`

# Components

- @Component decorator
- View
- Controller / Class

```
import { Component, OnInit } from '@angular/core';

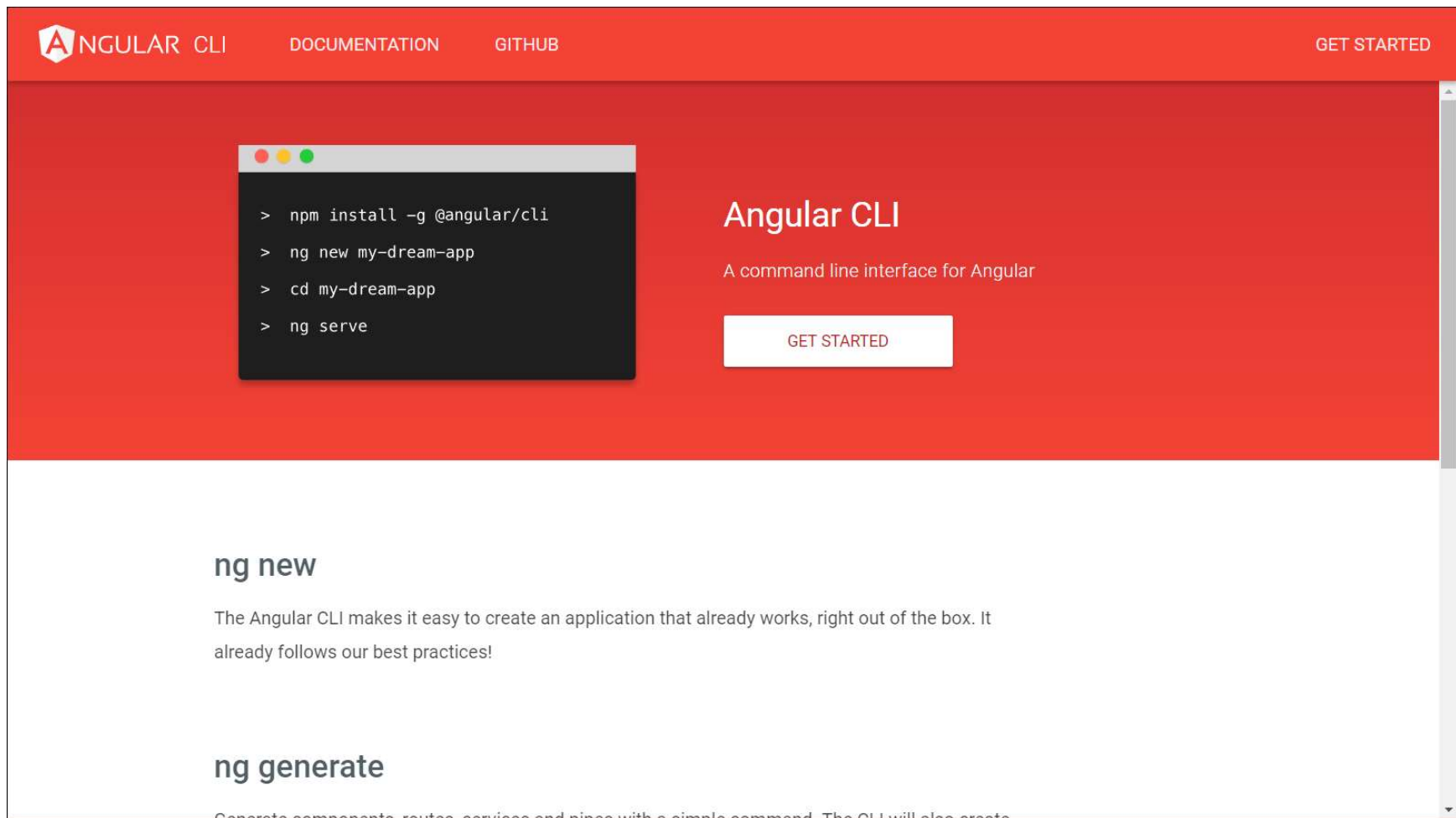
@Component({
  selector: 'app-customer',
  templateUrl: './customer.component.html',
  styleUrls: ['./customer.component.css']
})
export class CustomerComponent implements OnInit {

  constructor() {}

  ngOnInit() {}
}
```



# Angular CLI



# Ng serve

- Does multiple things – you *could* do this by hand:
  1. TypeScript compilation
  2. (SASS compilation)
  3. Fire up webserver on <http://localhost:4200>
  4. Fire up Live Reload
  5. Analyzing and bundling of application by firing up WebPack
  6. Watch for changes and recompile, rebundle, etc.

# Data binding

- New syntax in HTML-views/partials.
  1. Simple data binding `{{ ... }}`
  2. Event binding `(...)="..."`
  3. One-way data binding `[...]="..."`
  4. Two-way data binding `[(ngModel)] = "..."`

*Questions?*

# Today

- Data binding (continued)
  - Event- and Attribute binding
  - Two-way Data Binding
- Services
  - Static Services
  - RxJS / Observables
  - Live API's



# Tomorrow

- Tree of components
  - Component Communication
  - Building applications with multiple components
  - Data flow between components `@Input()` and `@Output()`
- Routing
  - Basic Routing
  - Routing Parameters
- Forms
  - Template Forms
  - Reactive Forms