



# Angular

BY Heba Samy

# Day 1 Agenda

- ▶ SPA
- ▶ Angular is a Modular
- ▶ Installation
- ▶ hello world
- ▶ Project Structure
- ▶ Components
- ▶ Structural Directives



# Angular Is ...



A JavaScript framework

For building client-side applications

Using HTML, CSS and JavaScript

# Why Angular?



**Expressive  
HTML**



**Powerful  
Data  
Binding**



**Modular  
By Design**



**Built-in  
Back-End  
Integration**



# Angular Libraries

@angular/core	2.3.0
@angular/compiler	2.3.0
@angular/http	2.3.0
@angular/router	<b>3.3.0</b>

# What Is TypeScript?



Open source language

Superset of JavaScript

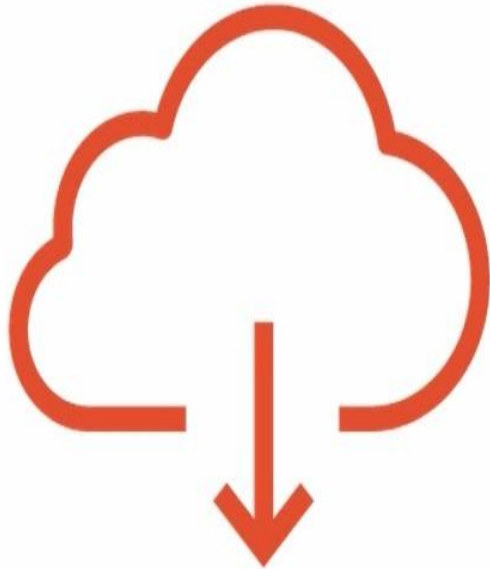
Transpiles to plain JavaScript

Strongly typed

- TypeScript type definition files (\*.d.ts)

**Class-based object-orientation**

npm

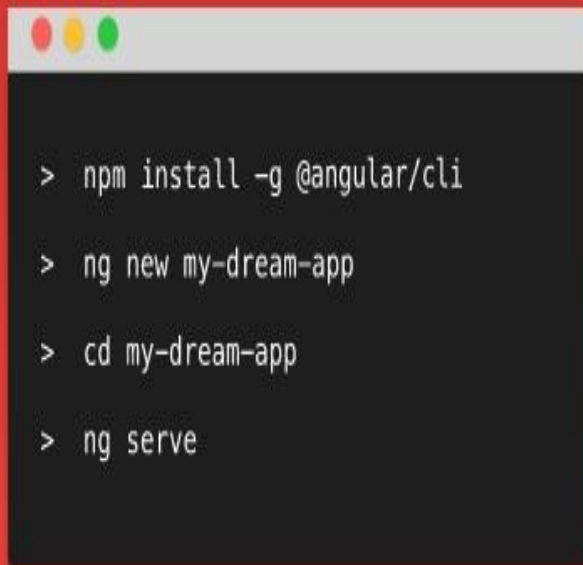


Node Package Manager

Command line utility

Installs libraries, packages, and applications

<https://www.npmjs.com/>

A terminal window with a dark background and a light gray title bar containing three colored window control buttons (red, yellow, green). The terminal displays four commands, each preceded by a prompt character '>'.

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

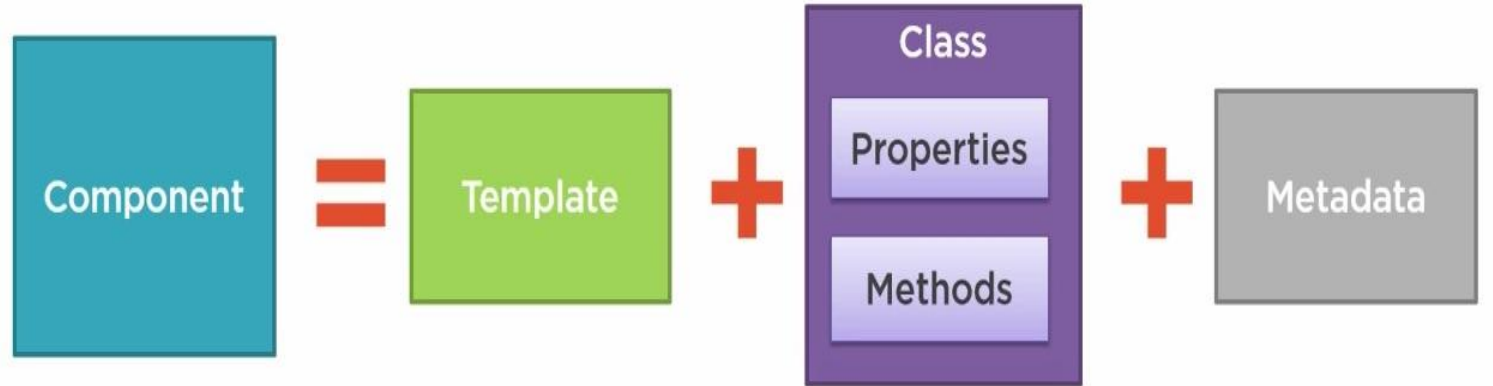
# Angular CLI

A command line interface for Angular

GET STARTED



# What Is a Component?



- View layout
- Created with HTML
- Includes binding and directives

- Code supporting the view
- Created with TypeScript
- Properties: data
- Methods: logic

- Extra data for Angular
- Defined with a decorator

# Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

class  
keyword

Class Name

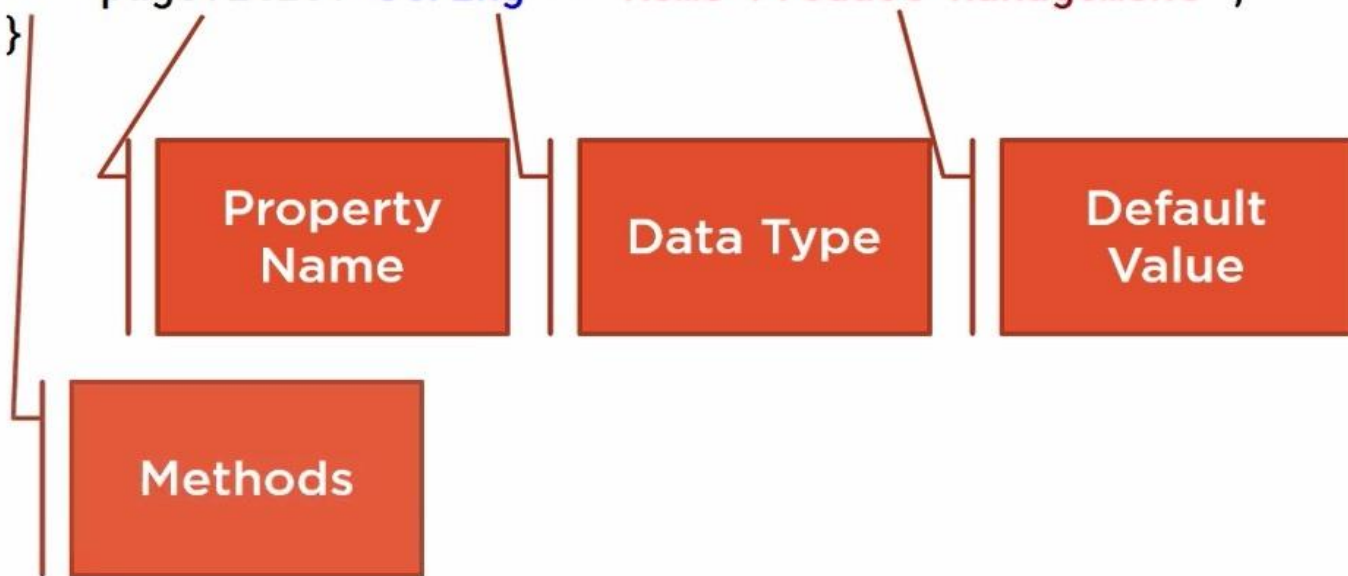
export  
keyword

Component Name  
when used in code

# Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```



# Defining the Metadata

app.component.ts

```
@Component({  
  selector: 'pm-app',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

Component  
decorator

Directive Name  
used in HTML

View Layout

Binding

# Angular Is Modular

@angular/  
core

@angular/  
animate

@angular/  
http

@angular/  
router

# Importing What We Need

app.component.ts

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

```
@Component({  
  selector: 'pm-app',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `
```

```
})  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

import keyword

Angular library  
module name

Member name

# Defining a Template in a Component

## Inline Template

```
template:  
"<h1>{{pageTitle}}</h1>"
```

## Inline Template

```
template: `  
  <div>  
    <h1>{{pageTitle}}</h1>  
    <div>  
      My First Component  
    </div>  
  </div>  
`
```

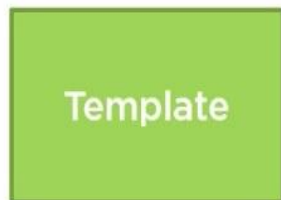
ES 2015  
Back Ticks

## Linked Template

```
templateUrl:  
'product-list.component.html'
```

# Binding

Coordinates communication between the component's class and its template and often involves passing data.





# Interpolation

## Template

```
<h1>{{pageTitle}}</h1>
{{'Title: ' + pageTitle}}
{{2*20+1}}
{{'Title: ' + getTitle()}}
```

## Class

```
export class AppComponent {
  pageTitle: string =
    'Acme Product Management';
  getTitle(): string {...};
}
```

# Property Binding

```
<img [src]='product.imageUrl'>
```

```
<img src={{product.imageUrl}}>
```

```
<img src='http://openclipart.org/{{product.imageUrl}}'>
```

# Event Binding


Template

```
<h1>{{pageTitle}}</h1>
```

```
<img [src]='product.imageUrl'>
```

```
<button (click)='toggleImage()'>
```

Class



```
export class ListComponent {  
  pageTitle: string = 'Product List';  
  products: any[] = [...];  
  toggleImage(): void {...}  
}
```

# Two-way Binding

Template

Class

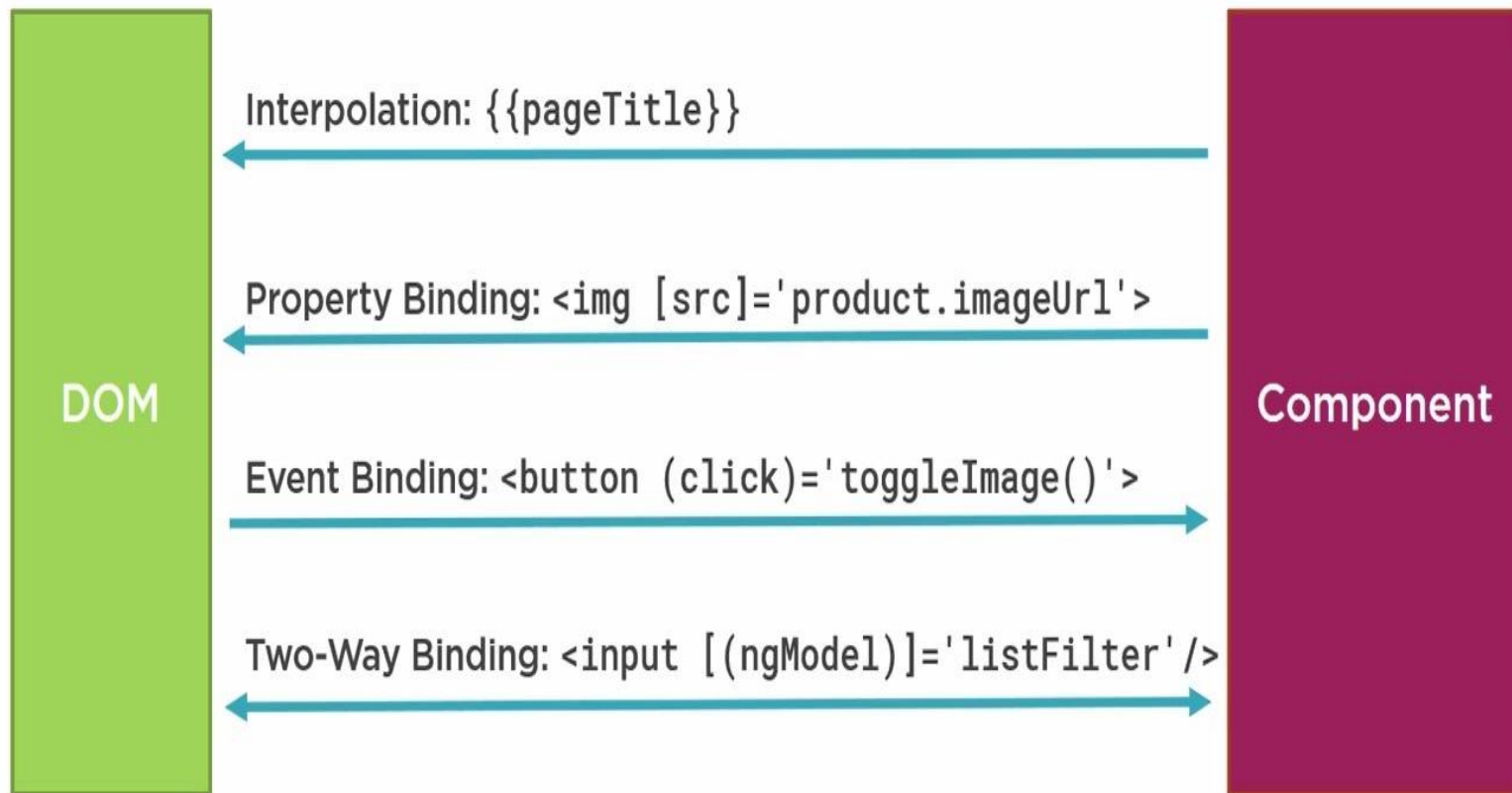
`<input [(ngModel)]='listFilter'>`

```
export class ListComponent {  
  listFilter: string = 'cart';  
}
```

`[( )]`

Banana in a Box

# Data Binding





# Directive

Custom HTML element or attribute used to power up and extend our HTML.

- Custom
- Built-In

# Using a Component as a Directive

## app.component.ts

```
@Component({
  selector: 'pm-app',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <pm-products></pm-products>
    </div>`
})
export class AppComponent { }
```

## product-list.component.ts

```
@Component({
  selector: 'pm-products',
  templateUrl:
    'app/products/product-list.component.html'
})
export class ProductListComponent { }
```



# Angular Built-in Directives

## Structural Directives

\*ngIf: If logic

\*ngFor: For loops



## \*ngIf Built-In Directive

```
<div class='table-responsive'>
  <table class='table' *ngIf='products && products.length'>
    <thead> ...
  </thead>
  <tbody> ...
</tbody>
</table>
</div>
```

## \*ngFor Built-In Directive

```
<tr *ngFor='let product of products'>
  <td></td>
  <td>{{ product.productName }}</td>
  <td>{{ product.productCode }}</td>
  <td>{{ product.releaseDate }}</td>
  <td>{{ product.price }}</td>
  <td>{{ product.starRating }}</td>
</tr>
```