

The background of the slide is a photograph of a modern, multi-story office building with a glass facade. The building is viewed from a low angle, looking up. The word "CYBAGE" is visible on the upper part of the building's facade. A semi-transparent white box with a blue square is overlaid on the right side of the image, containing the title and presenter information.

CYBAGE

Angular JS 4 (Day 2)

Presentation By: Asfiya Khan (Technical Trainer)

Document History

Version No.	Authored/ Modified by	Remarks/ Change History	Date <dd- mon-yy >
1.0	Asfiya Khan	First version of Angular 4	13 March 2018

Course Structure

Target audience	Trainee,SE,SSE
Level	1,2,3
Pre-requisites	Javascript,TypeScript,HTML,CSS
Training methods	Presentation , Demos, Hands-on
Evaluation	Multiple Choice Question

Agenda



Architecture
and
Components



Data Binding
and Pipes



Routing and
Navigation



Templates
,Interpolation
and Directives



Angular
Modules



Services and
Dependency
Injection



Ng-Forms



Retrieving
data using
HTTP

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'
```

Building the Component

product-list.component.ts

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

@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```

Using a Component as a Directive

app.component.ts

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

product-list.component.ts

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

Using a Component as a Directive

app.component.ts

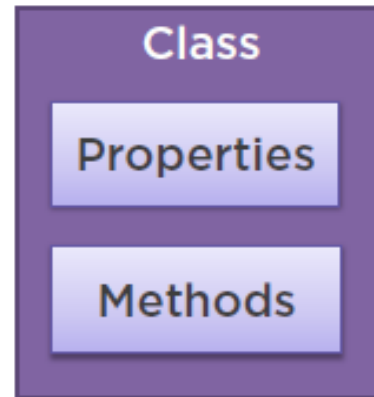
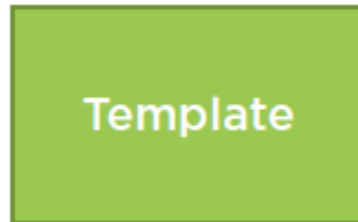
```
@Component({  
  selector: 'pm-root',  
  template: `  
1    <div><h1>{{pageTitle}}</h1>  
      <pm-products></pm-products>  
    </div>`  
})  
export class AppComponent { }
```

product-list.component.ts

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


Binding

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



Interpolation

- One Way Data Binding
 - From component class property to element property.
- Define with double curly braces
 - Contains a template expression
 - No quote needed.

Interpolation

Template

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

Class

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

Directive

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

- Custom
- Built-In

Custom Directives

app.component.ts

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

product-list.component.ts

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

Angular Built-in Directives

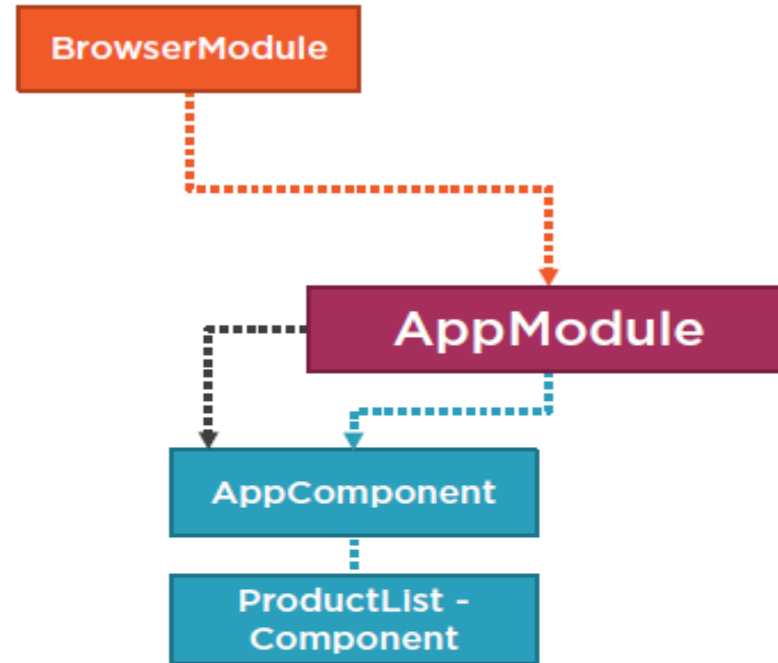
Structural Directives

***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>
```



- Imports
- Exports
- Declarations
- Providers
- Bootstrap

*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>
```

**Template
input variable**

for...of vs for...in

for...of

- Iterates over iterable objects, such as an array.
- Result: di, boo, punkeye

```
let nicknames= ['di', 'boo', 'punkeye'];  
  
for (let nickname of nicknames) {  
    console.log(nickname);  
}
```

for...in

- Iterates over the properties of an object.
- Result: 0, 1, 2

```
let nicknames= ['di', 'boo', 'punkeye'];  
  
for (let nickname in nicknames) {  
    console.log(nickname);  
}
```



*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>
```

DEMO

Property Binding

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

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

Element
Property

binding

Template
Expression

```
product.imageUrl}}'>
```

Event Binding

Template

Class

```
<h1>{{pageTitle}}</h1>  
<img [src]='product.imageUrl'>  
<button (click)='toggleImage()'>
```

```
export class ListComponent {
```

<https://developer.mozilla.org/en-US/docs/Web/Events>

()

Target Event

''

Template
Statement



Two-way Binding

Template

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

Class

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

`[()]`

Banana in a Box

Transforming Data with Pipes

**Transform
bound
properties
before
display**

Built-in pipes

- date
- number, decimal, percent, currency
- json, slice
- etc

**Custom
pipes**



Pipe Examples

```
{{ product.productCode | lowercase }}
```

```
<img [src]='product.imageUrl'  
      [title]='product.productName | uppercase'>
```

```
{{ product.price | currency | lowercase }}
```

```
{{ product.price | currency:'USD':true:'1.2-2' }}
```

Data Binding

DOM

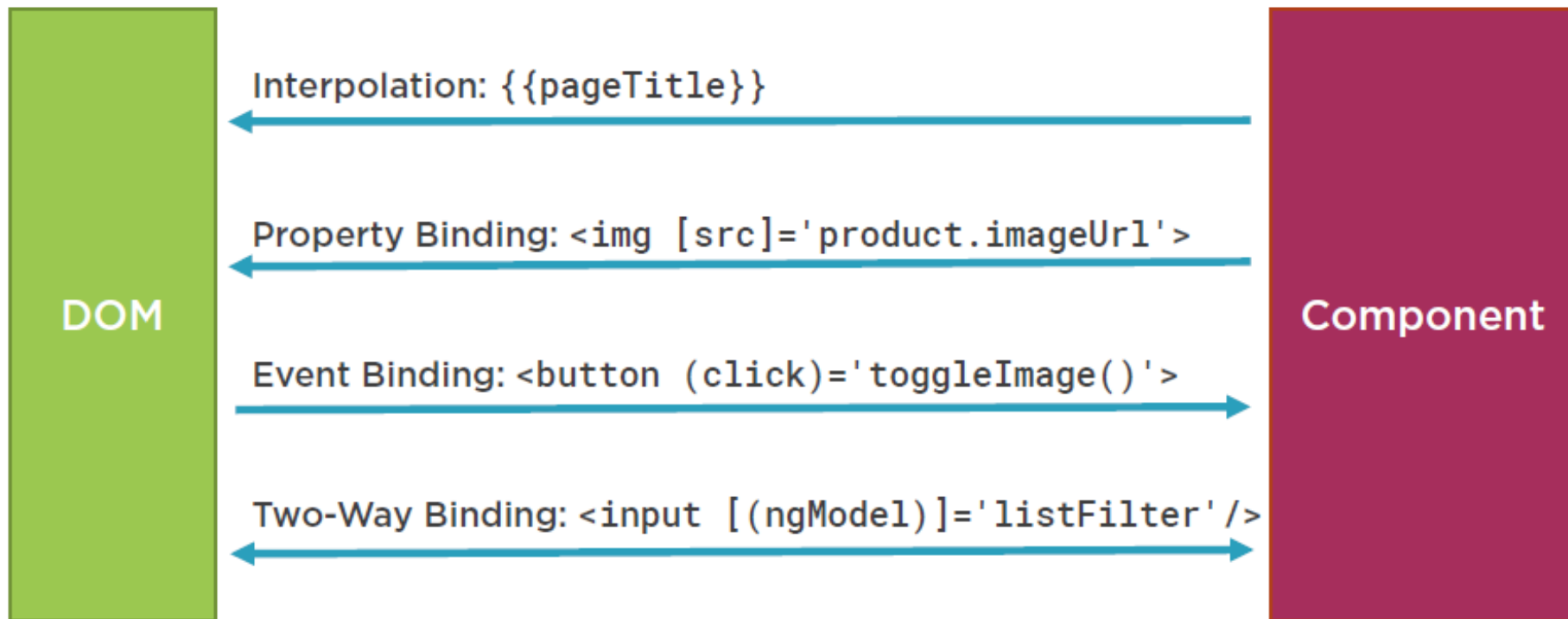
```
<pm-app>
  <div>
    <h1>Acme Product Management</h1>
    <pm-products>
      <div class="panel panel-primary">
        <div class="panel-heading">
          Product List
        </div>
        <div class="panel-body">
          ::before
          <div class="row">...</div>
          <div class="row">...</div>
          <div class="table-responsive">
            <!--template bindings={}-->
            <table class="table">
              <thead>...</thead>
              <tbody>...</tbody>
            </table>
          </div>
          ::after
        </div>
      </div>
    </pm-products>
  </div>
</pm-app>
```

product-list.component.ts

```
@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  listFilter: string = 'cart';
  products: any[] = [...];
  toggleImage(): void {...}
}
```



Data Binding





Thank You!