



R a i s i n g t h e b a r

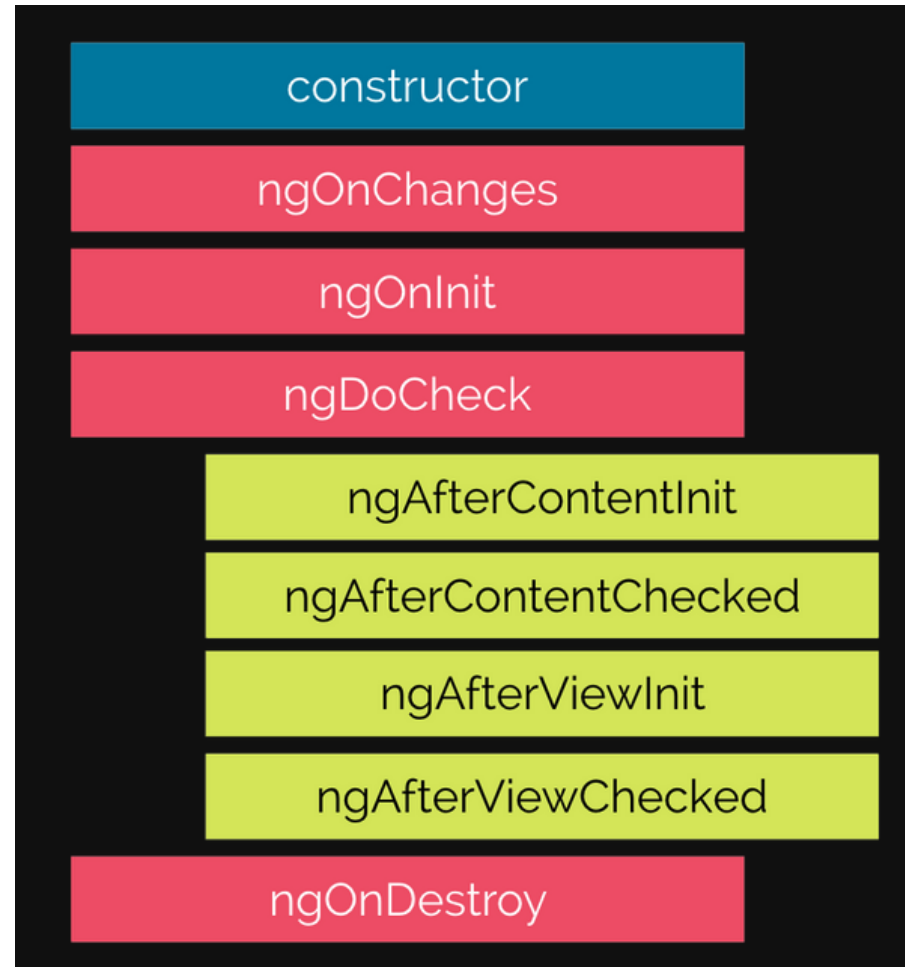
Angular Component & Template

Mục tiêu

- Vòng đời của component
- Tạo được các component tùy biến.
- Truyền được dữ liệu vào component.
- Sử dụng được các directive trong template.
- Sử dụng được style cho template.

Angular Component

Vòng đời của Component



Tạo Component

- Để tạo component trong Angular sử dụng câu lệnh:
ng generate component tên-component
Hoặc có thể viết nhanh là
ng g c tên-component
- Sau khi sử dụng câu lệnh trên sẽ được tạo ra 4 file:
 - tên-component.component.ts
 - tên-component.component.html
 - tên-component.component.css
 - tên-component.component.spec.ts

Angular built-in directives

Chuẩn bị

- Generate component mới:
 - ng generate component rating-bar

Structure directives - ngIf

```
<div *ngIf="condition">...</div>
```

```
<ng-template [ngIf]="condition">
```

```
  <div>...</div>
```

```
</ng-template>
```


Structure directives - ngIf

```
<div *ngIf="condition; then thenBlock else elseBlock"></div>  
<ng-template #thenBlock>...</ng-template>  
<ng-template #elseBlock>...</ng-template>
```

```
<div *ngIf="condition as value; else elseBlock">  
  {{value}}  
</div>  
<ng-template #elseBlock>...</ng-template>
```

Structure directives - ngForOf

```
<div *ngFor="let item of items;
      index as i; trackBy:
      trackByFn">...</div>
```

```
<ng-template ngFor let-item[ngForOf]="items" let-
i="index" [ngForTrackBy]="trackByFn">
  <div>...</div>
</ng-template>
```

Attribute directives – class/ngClass

```
<some-element [class.class-name]="expression"></some-element>
```

```
<some-element [ngClass]=" 'first second' ">...</some-element>
```

```
<some-element [ngClass]=" ['first', 'second'] ">...</some-element>
```

```
<some-element [ngClass]=" { 'first': true, 'second': true, 'third':  
false } ">...</some-element>
```

```
<some-element [ngClass]=" stringExp|arrayExp|objExp ">...</some-element>
```

```
<some-element [ngClass]=" { 'class1 class2 class3' : true } ">
```

```
...
```

```
</some-element>
```

Attribute directives – style/ngStyle

<some-element [style.style-property]="expression"></some-element>

<some-element [style.styleProperty]="expression"></some-element>

<some-element [style.style-property.unit]="expression"></some-element>

<some-element [style.styleProperty.unit]="expression"></some-element>

<some-element [ngStyle]="{'font-style': styleExp}>...</some-element>

<some-element [ngStyle]="{'max-width.px': widthExp}>...</some-element>

<some-element [ngStyle]="objExp">...</some-element>

Angular Pipes

Chuẩn bị

- Generate component mới:
 - nggenerate component pipe-demo

Built-in Pipes

- AsyncPipe
- CurrencyPipe
- DatePipe
- DecimalPipe
- JsonPipe
- LowerCasePipe
- PercentPipe
- SlicePipe
- TitleCasePipe
- UpperCasePipe
- ...

Sử dụng Pipe

~~pipe-demo.component.html~~

```
<h2>Pipes</h2>
```

```
<p>
```

```
Today: {{ today | date:'fullDate' }}
```

```
</p>
```

```
<p>
```

```
Username: {{user.name | titlecase}}
```

```
</p>
```

~~pipe-demo.component.ts~~

```
today = new Date();
```

```
user = {
```

```
    name: 'bob mike'
```

```
};
```

Pipes

Today: Tuesday, September 4, 2018

Username: Bob Mike

Angular component style

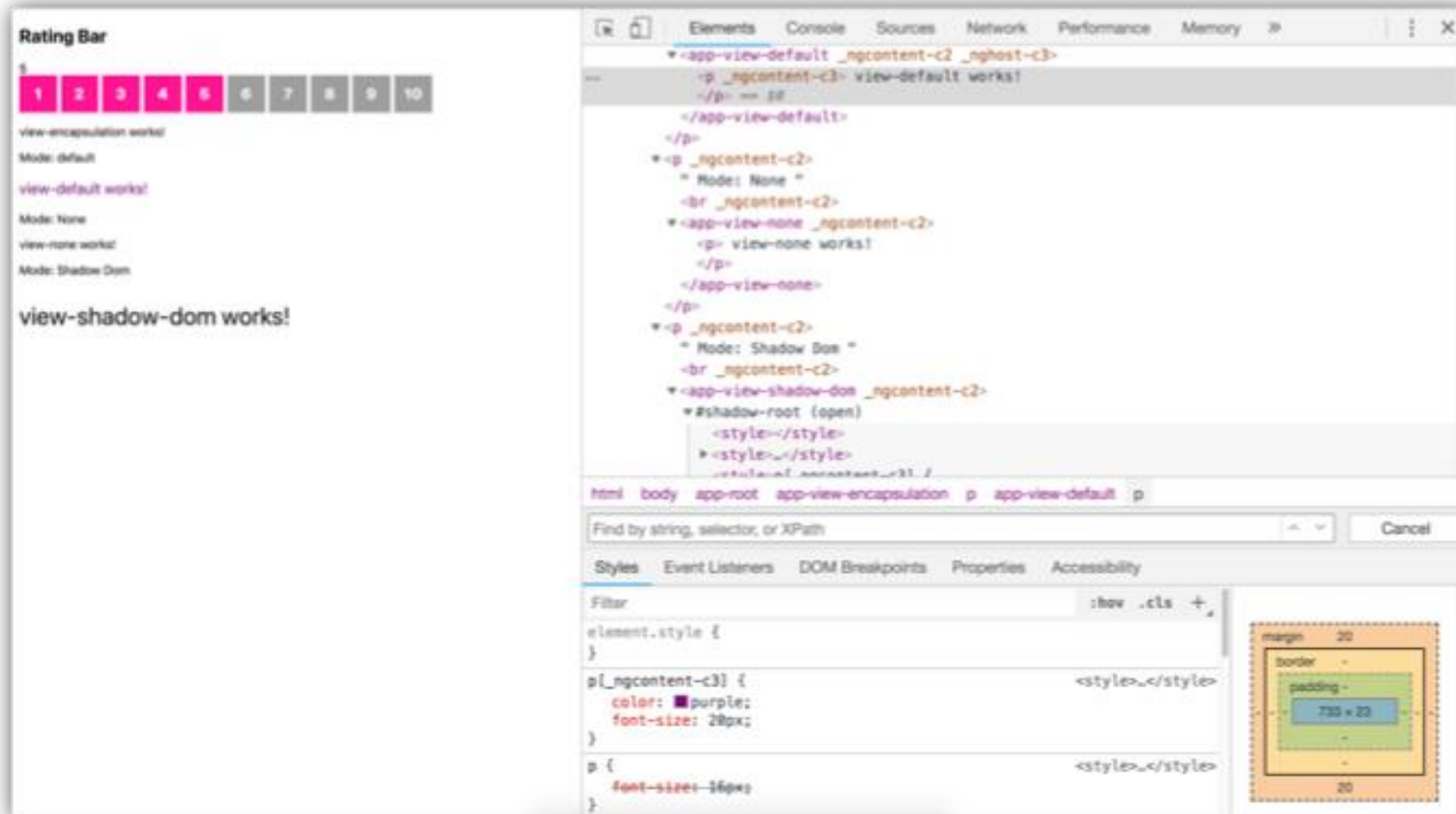
Chuẩn bị

- Generate component mới:
 - generate component view-encapsulation
 - ng generate component view-default
 - ng generate component view-shadow-dom
 - ng generate component view-none

Scope style

- Global style (styles.scss/styles.css/etc): ảnh hưởng chung đến toàn bộ app.
- Component style – View Encapsulation:
 - Emulated – default: đóng gói thành scope cho component đó, không bị leak style dẫn đến ảnh hưởng tới component khác.
 - None: không đóng gói gì cả, style trong component sẽ trở thành global style và ảnh hưởng chung đến toàn bộ app.
 - ShadowDom (Angular v6.1+)/Native (deprecated từ Angular v6.1+): sử dụng cơ chế Shadow Dom để đóng gói, không bị leak style sang component khác.

Kết quả khi apply scope style





R a i s i n g t h e b a r