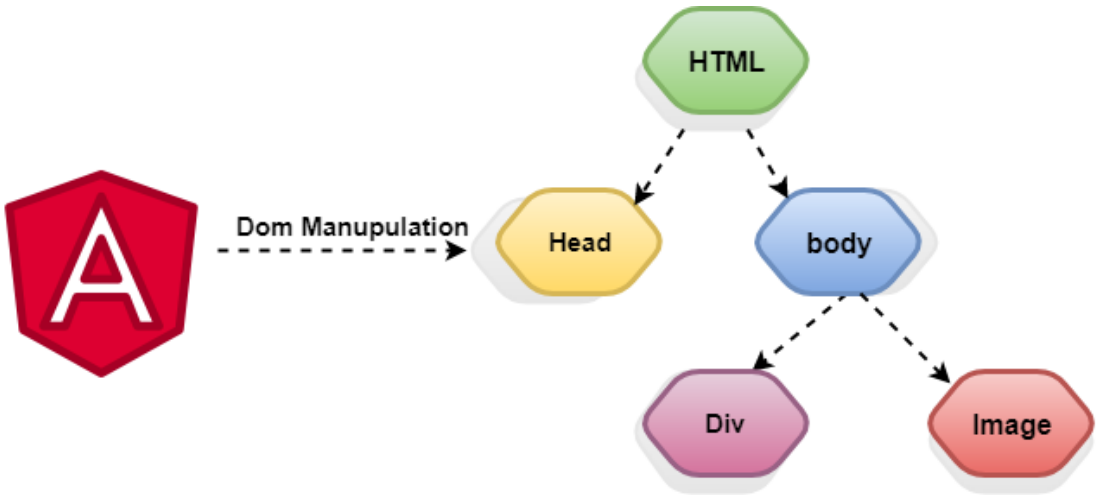
Angular 8 Directives

July 21, 2019

Angular 8 Directives

Directives are instructions in the **DOM** (Document Object Model). It specifies how to place our business logic in Angular. The directive is markers on a DOM element that tell Angular to attach a specified behavior to that DOM element or even transform the DOM element and its children. Mostly directives in Angular starts with ng- where **ng** stands for **Angular,** and it extends the HTML.

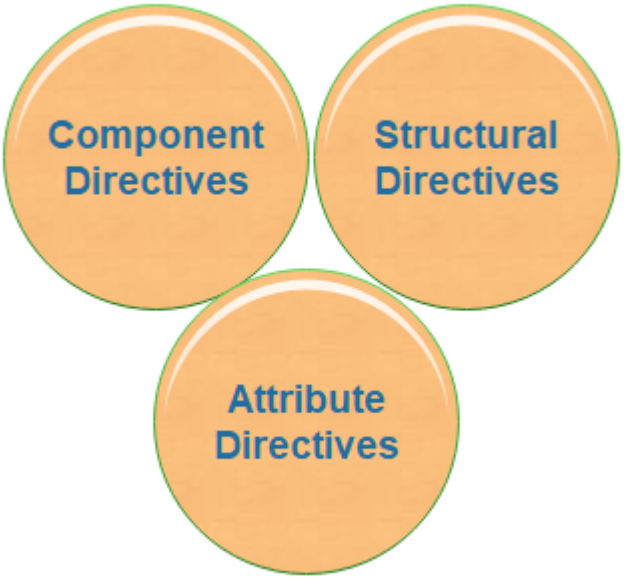


There are three kinds of directives:

1. Component Directives

2. Structural Directives

3. Attribute Directives



Component Directives

Components are the most common of the directives. It contains the details of how the component should be processed, instantiated, and used at runtime. The component comprises meta-data.

Structural Directives

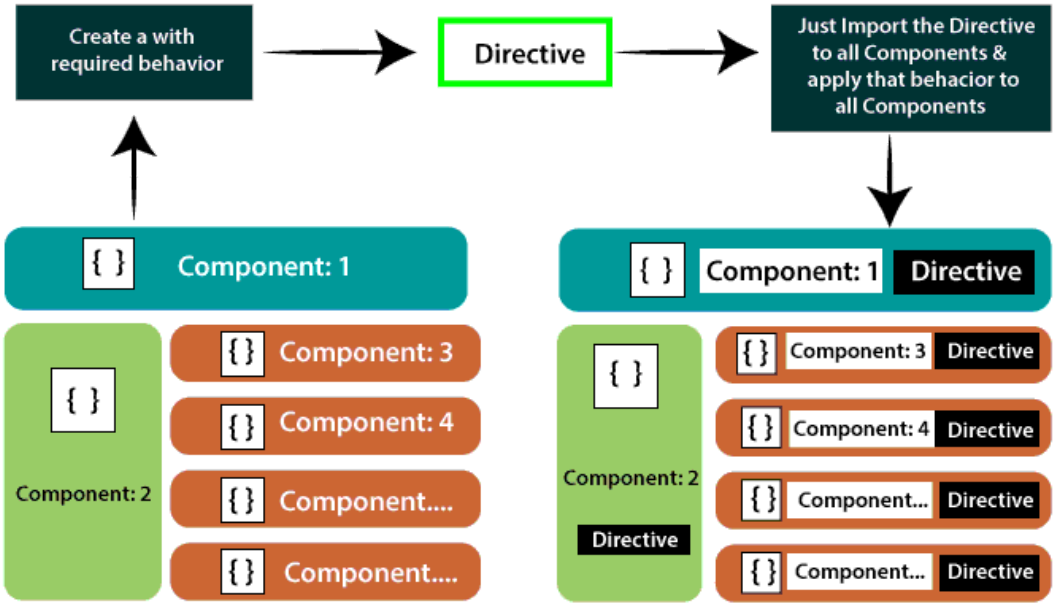
Structural Directivesaredone in the elements section. These directives are used to manipulate and change the structure of the DOM elements. Structural directives have a star (*) sign before the directive. **Like as,* ngIf**, ***ngFor, and *ngSwitch directive.**

- ***ngIf Directive:** The *ngIf allows us to Add/Remove DOM Element.
- ***ngSwitch Directive:** The *ngSwitch will enable us to Add/Remove DOM element. It is same as the switch statement of C#.
- ***ngFor Directive:** The *ngFor directive is used to repeat a part of HTML template once per each item from an iterable list (Collection).

Attributes Directives

Itdeals with changing the look and behavior of the DOM element. For example: **ngClass**, **ngStyle** etc.

- **NgClass Directive:** The ngClass Directive is used to add or remove CSS classes to an element.
- **NgStyle Directive:** The ngStyle Directive facilitates you to modify the style of an HTML element using the expression. We can also use the ngStyle Directive to change the style of our HTML element dynamically.



Difference between Structural Directive and Attribute Directive

Structural Directive	Attribute Directive
Structural directives are applied to <template> elements and used to add or remove content (stamp and template).	**The component has their view (HTML and styles) because of that, there can be one component on a host element, but multiple directives.**

How to Create Custom Directives?

We can create our custom directives to use in Angular components with the help of the command line. The command which is used to develop the Directive using the command line is as follows-

```
ng g directive name of the Directive
e.g
ng g directive change text
```

It is seen in the command line as given in the below code-

```
C:\projectA7\angular7-app>ng g directive change text
CREATE src/app/change-text.directive.spec.ts (241 bytes)
CREATE src /app/change-text.directive.ts (149 bytes)
UPDATE src/app/app.module.ts (565 bytes)
```

The above files, i.e., **change-text directive.spec.ts** and **change-text.directive.ts** created and the **app.module.ts** is updated.

app.module.ts

import { BrowserModule } from '@angular/platform-browser';

```
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NewcmpComponent } from '. /new-cmp.component';
import { ChangeTextDirective } from '. /change-text.directive';
@NgModule({
declarations: [
AppComponent,
NewCmpComponent,
ChangeTextDirective
],
Imports: [
BrowserModule,
AppRoutingModule
],
providers: [ ],
bootstrap: [AppComponent]
}]
export class AppModule { }
```

The ChangeTextDirective class has been included in the declarations in the above file. The class is also imported from the file given below-

Change-text.directive

```
 Import {Directive}  from '@angular/core';
@Directive ({
Selector; '[changeText]'
})
Export class ChangeTextDirective {
Constructor ( ) { }
}
```

The above file has a directive and it also has a selector property. Whatever we define in the selector, the same has to match in the view, where we assign the custom directive.

In the app.component.html view, add the directive as follows-

```
 <!–The content below is only a placeholder and replaced.-->
<div style= "text-align:center">
<h1>Welcome to {{title}}. </h1>
</div>
<div style = "text-align:center">
<span changeText > Welcome to {{ title}}.</span>
</div>
```

We will write the changes in **change-text.directive.ts** file as-

change-text.directive.ts

```
 import { Directive, elementRef} from
‘@angular/core’;
@Directive({
Selector: ‘[changeText]’’
})
Export class ChangeTextDirective {
Constructor(Element: ElementRef) {
Console.log(Element);
Element.nativeElement.innerText= “Text is changed by changeText Directive.”;
}
}
```

In the above file, there is a class known as **ChangeTextDirective**, and also a constructor.

← Previous

Next →

Angular 8

Routing in Angular 8

Architecture of Hibernate

Latest Tutorials

Angular 8 Tutorial

AI Tutorial