

Angular -8

Angular Directives Part-1

Presentation BY RAJ PRUDHVI

Contact us: <https://training.uplatz.com>

Email: info@uplatz.com

Phone: +44 7836 212635



Angular Directives

- The Angular directives helps us to manipulate the DOM.
- You can change the appearance, behavior, or layout of a DOM element using the directives.
- They help you to extend HTML.
- The Angular directives are classified into three categories based on how they behave
 - ✓ Component Directives
 - ✓ Structural Directives
 - ✓ Attribute Directives

Component Directives

- Components are special directives in Angular. They are the directive with a template (view)
- The @Component decorator identifies the class immediately below it as a component class, and specifies its metadata.

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

- ❑ **selector**: Tells the template tag which specifies the beginning and end of the component.
- ❑ **templateURL**: Consists of the template used for the component.
- ❑ **styleUrls**: It is of array type which consists of all the style format files used for the template.

Structural Directives

- Structural directives can change the DOM layout by adding and removing DOM elements
- They shape or reshape the DOM's structure, typically by adding, removing, or manipulating elements.
- As with other directives, you apply a structural directive to a host element. The directive then does whatever it's supposed to do with that host element and its descendants.
- Structural directives are easy to recognize. An asterisk (*) precedes the directive attribute name
 - ❖ *ngIf
 - ❖ *ngSwitch
 - ❖ *ngFor

*ngIf Structural Directive

- The *ngIf is an Angular Structural Directive, which allows us to add/remove DOM Element based on some condition.

```
<p *ngIf="condition">  
  Welcome to Uplatz  
</p>
```

- If the expression evaluates to false then the Angular removes the entire element from the DOM. If true it will insert the element into the DOM.

Hidden attribute Vs ngIf

- ngIf does not hide the DOM element. It removes the entire element along with its subtree from the DOM. It also removes the corresponding state freeing up the resources attached to the element.
- hidden attribute does not remove the element from the DOM. But just hides it.
- The difference between **[hidden]='false'** and ***ngIf='false'** is that the first method simply hides the element. The second method with ngIf removes the element completely from the DOM.

ngIf else

- The ngIf allows us to define optional else block using the ng-template

```
<div *ngIf="condition; else elseBlock">  
content to render, when the condition is true  
</div>
```

```
<ng-template #elseBlock>  
content to render, when the condition is false  
</ng-template>
```

- The expression starts with a condition followed by a semicolon.

ngIf then else

- We can also define then else block using the ng-template

```
<div *ngIf="condition; then thenBlock else elseBlock">  
  This content is not shown  
</div>
```

```
<ng-template #thenBlock>  
  content to render when the condition is true.  
</ng-template>
```

```
<ng-template #elseBlock>  
  content to render when condition is false.  
</ng-template>
```


*ngFor Structural Directive

- The ngFor is an Angular structural directive, which repeats a portion of the HTML template once per each item from an iterable list (Collection).

Syntax of ngFor

```
<html-element *ngFor="let <item> of <items>;">  
  <html-Template></html-Template>  
</html-element>
```



Thank You

Contact us: <https://training.uplatz.com>

Email: info@uplatz.com

Phone: +44 7836 212635



