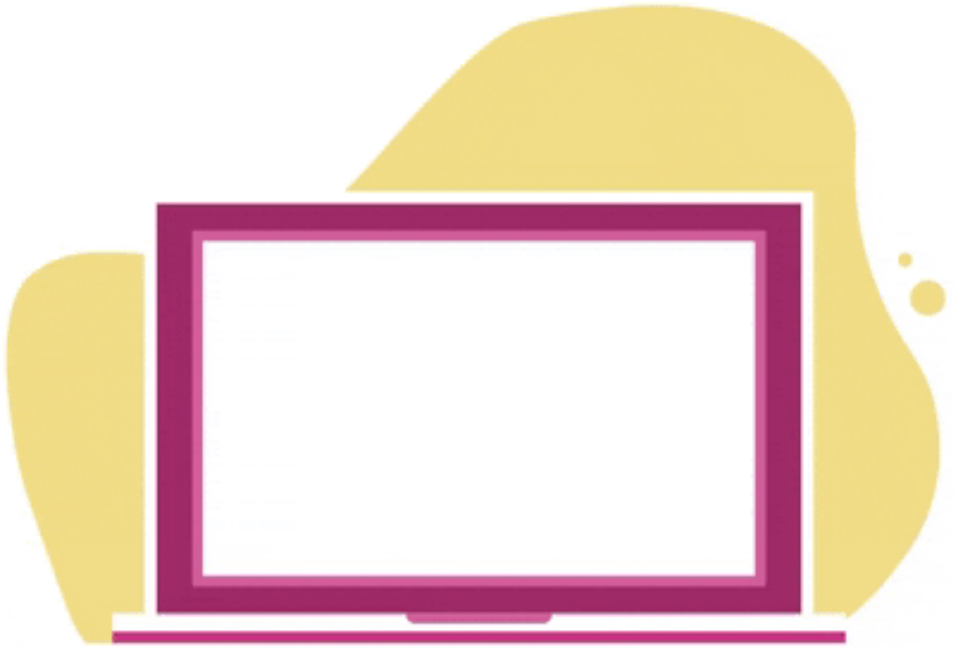


Displaying Data & Event Handling

Aitrich

Contents

- View
- Databinding
- Directives

View

A view refers to the user interface (UI) components that are rendered and displayed to the user.

A view in Angular is typically composed of HTML templates combined with Angular directives and data bindings.

Views are associated with components. A component is responsible for defining the behavior and structure of a specific section of the UI.

The template in Angular is an HTML file that describes the structure of the view. It contains HTML markup combined with Angular-specific syntax, such as directives and data bindings.

DataBinding

Databinding is a feature of angular allows connection between the components data and the view.

It display and manipulate data in the UI dynamically.

Two types of databinding

- One-way databinding
- Two-way data binding

One-way Data Binding & Two-way databinding

One-way databinding:

In one-way data binding, data flows only in one direction

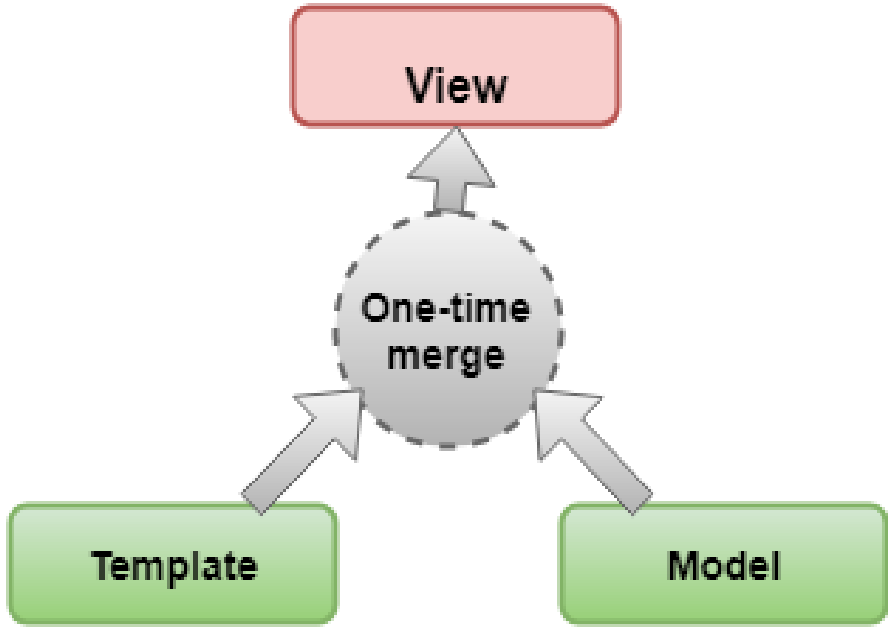
from the models to the views.

Two-way data binding :

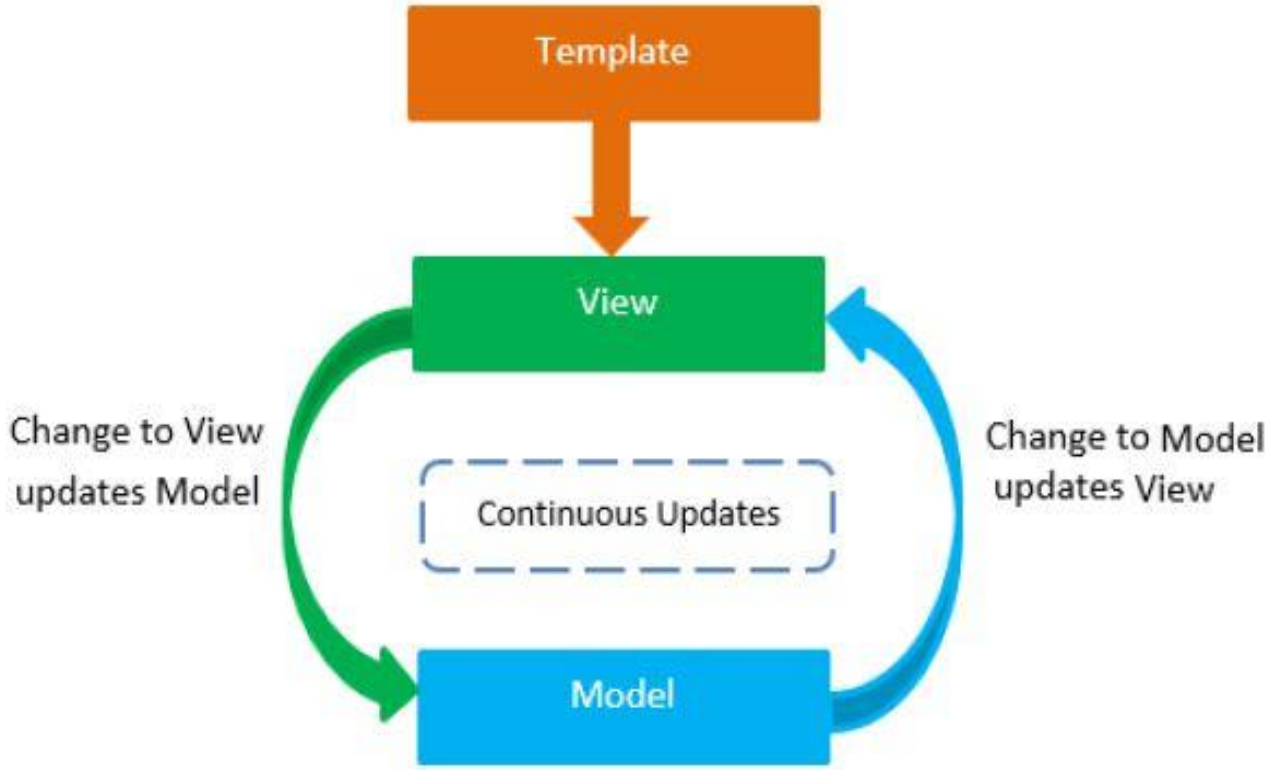
Two-way data binding allows to sharing of data between two directions From the template to view and view to the template

Aitrich

Flow Diagrams

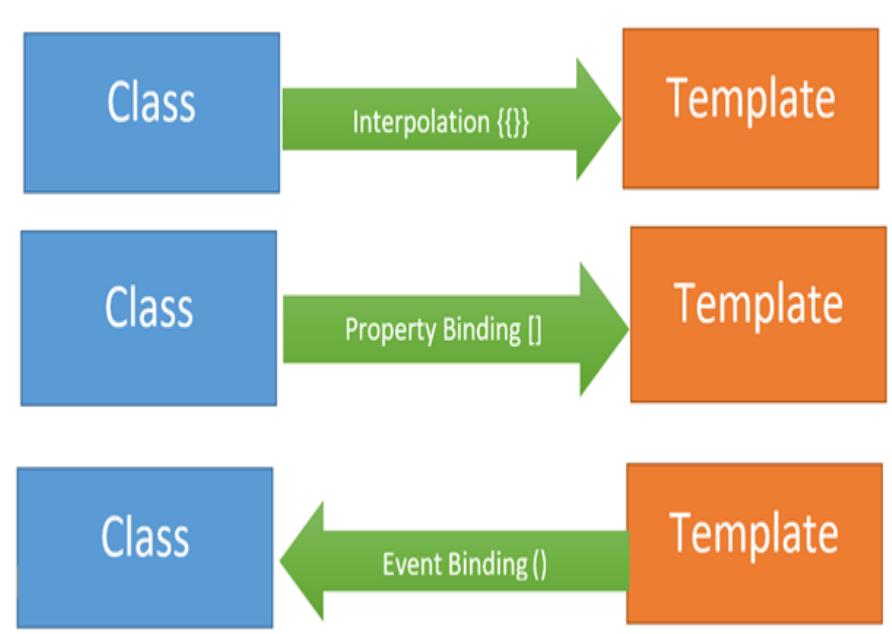


One-way data binding



Two-Way Data Binding

One-way data binding



One-way data binding is achieved through:

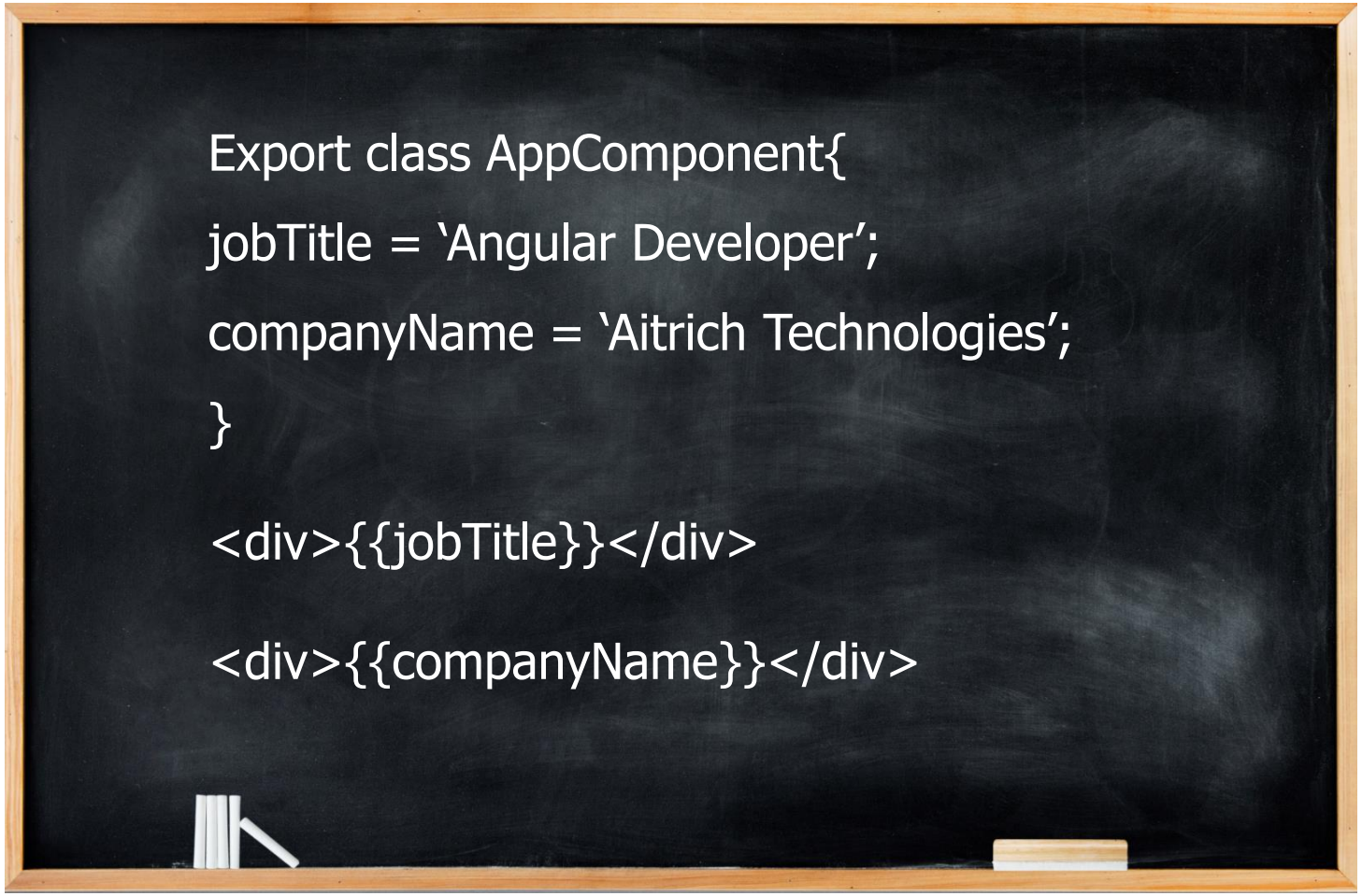
- Interpolation or String Interpolation
- Property binding
- Event binding

Interpolation Binding

Interpolation provides the data-binding from the component to the View.

The Template expression (double curly braces {{ }}) used for interpolation in Angular

Example



Aitrich

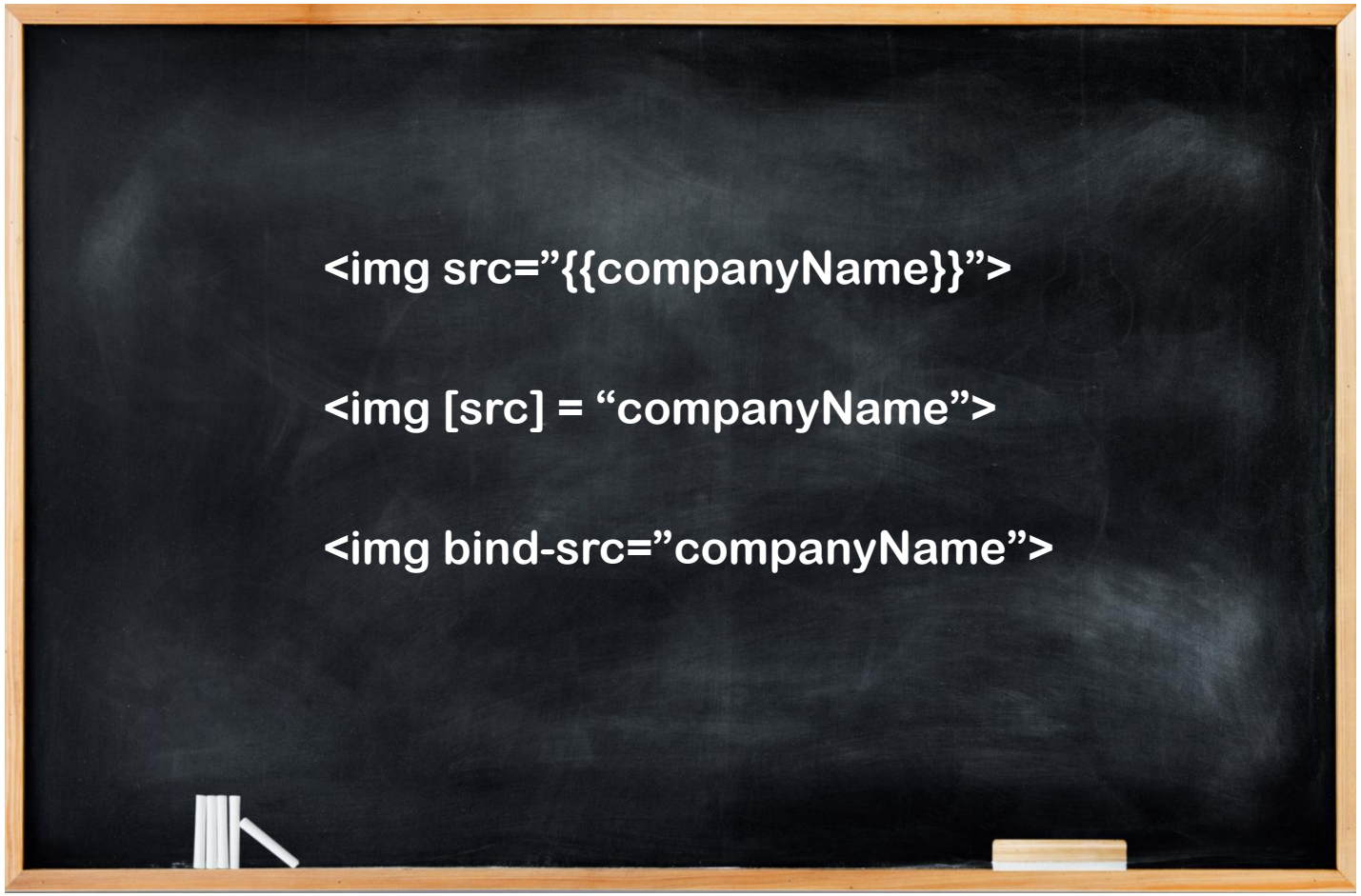
Property Binding

- Property binding binds the value of a component's property to a property of an HTML element.
- Changes in the component's property will update the corresponding property in the HTML element.
- property binding uses [ ]. The Binding source is enclosed in quotes

Example

```
Export class JobsComponent{

divBackgroundColor1: string = "rgb(202, 169, 232)";

}

<div class="inline-div" [style.background-
color]="divBackgroundColor1" >

All Skill

</div>
```

There are 3 ways to define a property binding in Angular



Event Binding

Handle user interactions, such as clicks, mouse movements, key presses etc.

Event binding bind methods or functions in your component to specific events triggered by HTML elements.

Whenever you wish to capture an event from the view, you Do so by wrapping the event in question with ( ) parenthesis.

Data Flow From Template to Class

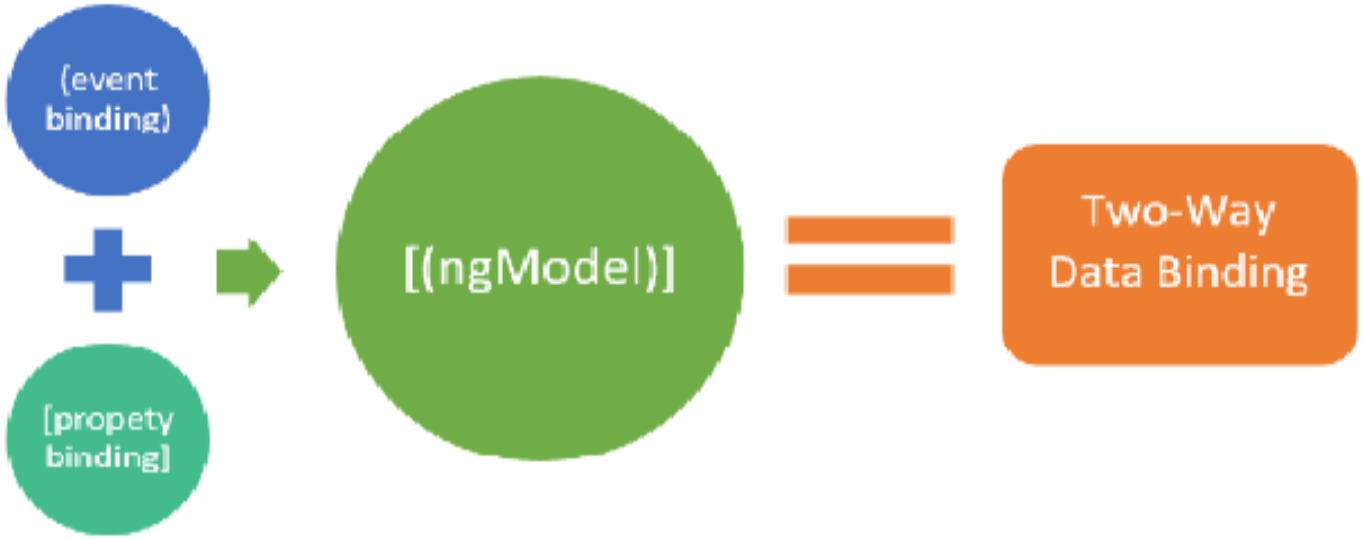
Example

```
<div  [style.background-color]="divBackgroundColor1"
(mouseover)="changeBackground()" >
Programmer
</div>

changeBackground() {

      this.divBackgroundColor1 = .blue;

    }
```

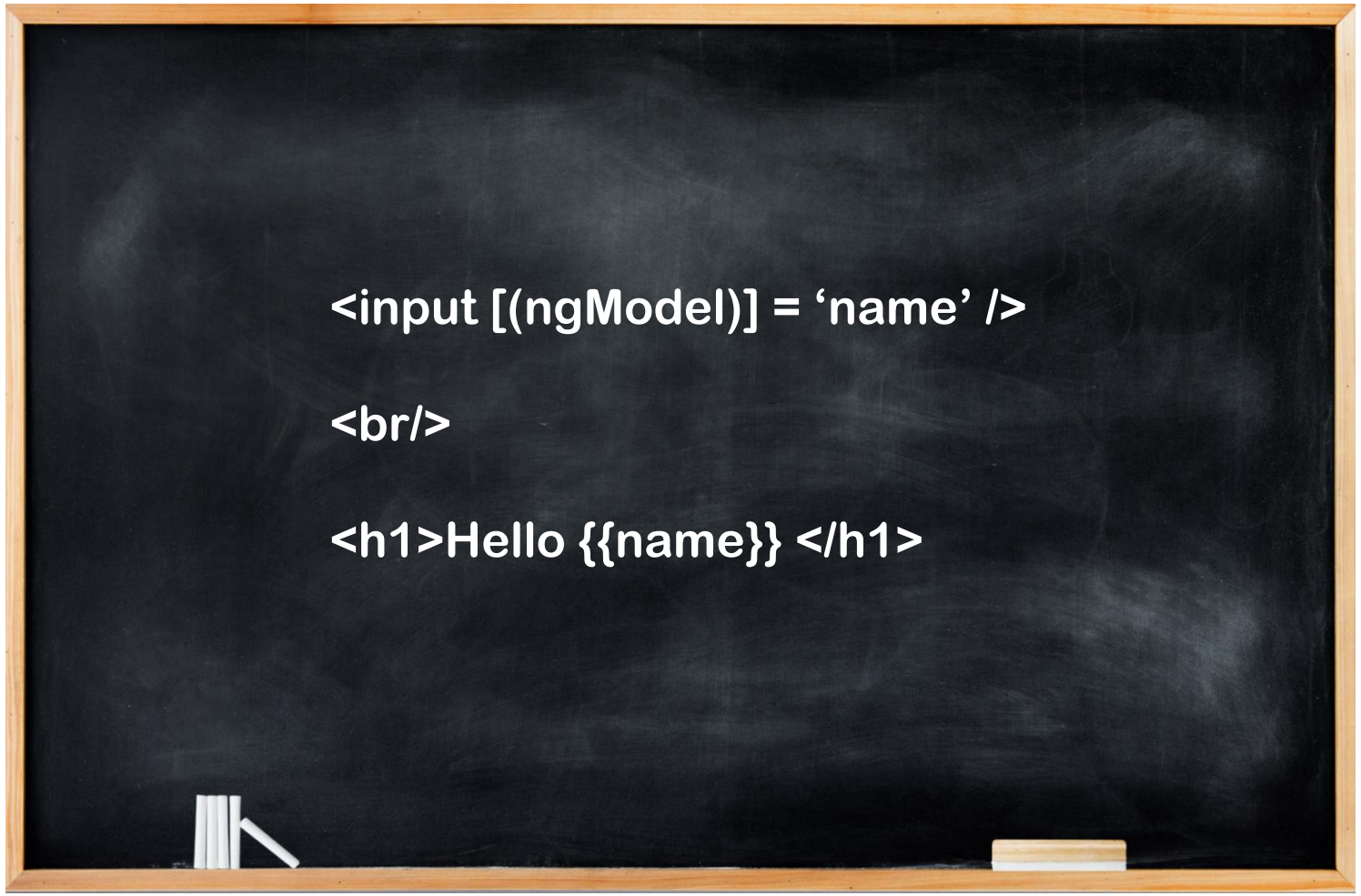
Two-way Databinding

- Angular allows two-way data binding that will allow application to share data in two directions. From the components to the templates and template to components.
- Two-way data binding will perform two things setting of the element property and listening to the element change events.
- The syntax of two way binding is – [( )].

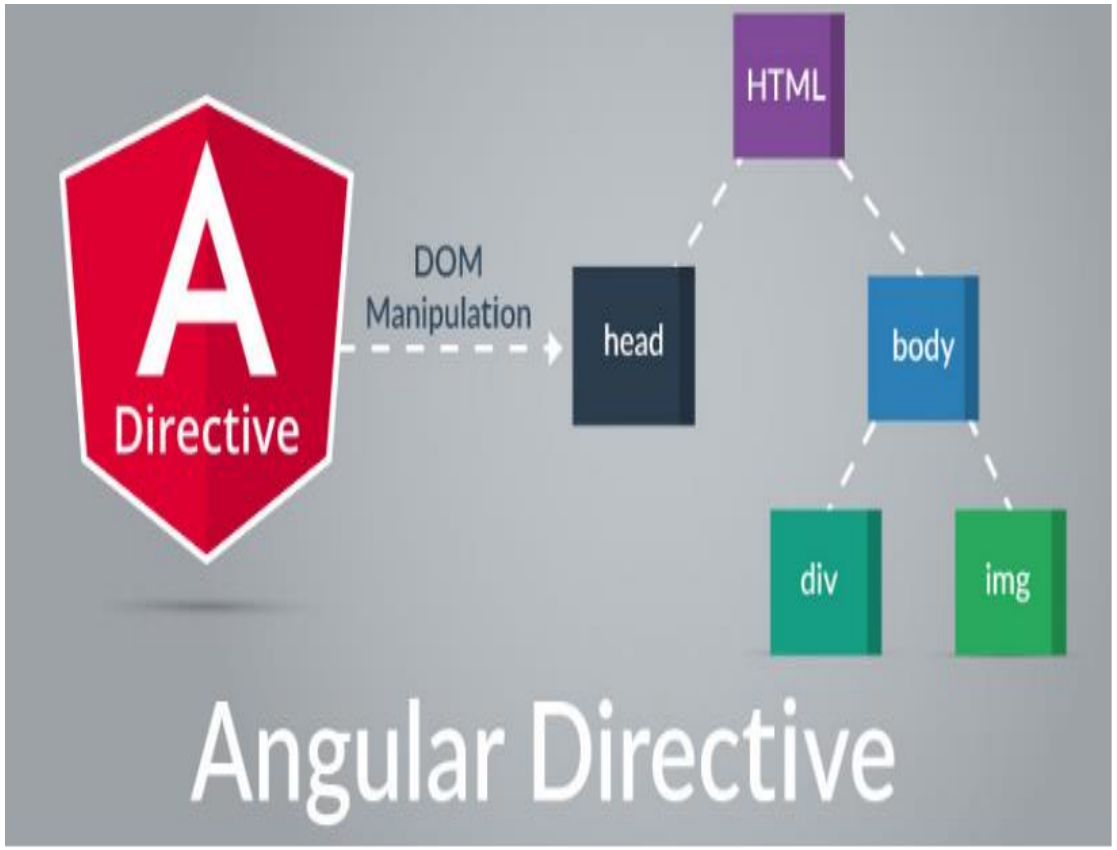
Flow Diagram



Example



Directives



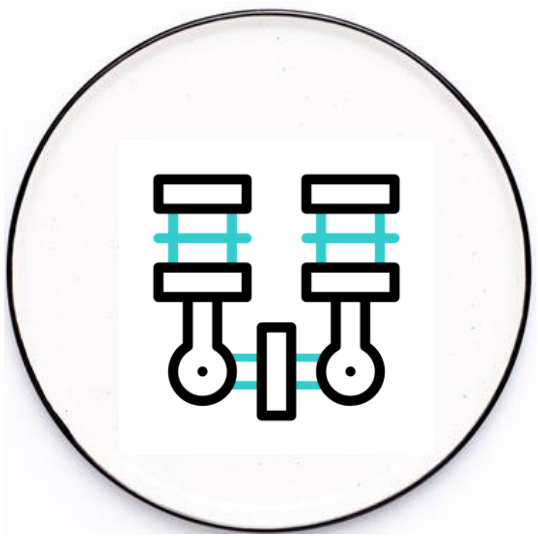
Directives are used to manipulate the DOM.

Directives are directions to the DOM.

Directives can change the appearance,

Bahavior or layout of a DOM Element

Types of Directives



Component Directives

Structural Directives

Attribute Directives

Component Directive

Component directives are used in the main class

They contain the detail of how the component should be processed, instantiated and used at runtime.

Component directives provide a way to build reusable and modular pieces of functionality with their own template, styles, and behaviour.

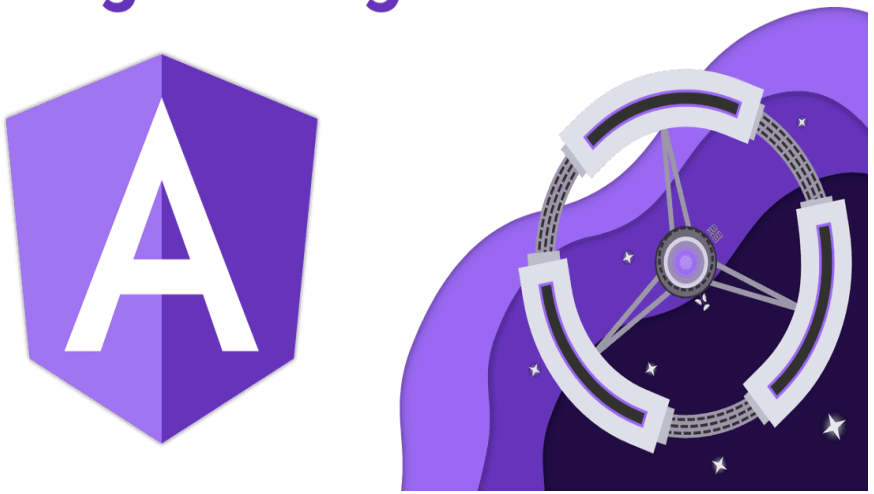
They are a key building block in Angular Applications, allowing for component-based architecture and code organization.

Example : <app-component> </app-component>

Structural Directive

- Structural directives can change the DOM layout by adding and removing DOM elements.
- Rather than adding new DOM elements, Structural directives modify the existing DOM and do not have templates.
- All structural Directives are preceded by Asterisk symbol(*)
- Commonly used structural directives are *ngIf, *ngFor, *ngSwitch.

NgFor



The ngFor is an Angular structural directive, which repeats a portion of HTML template once per each item from an itethatrable list.

```
<ul>
<li *ngFor=”let job of jobList”>
{{job}}
</i>
</ul>
```

*ngIf

The ngIf is another Angular Structural Directive, which allows us to Add/Remove DOM Elements.

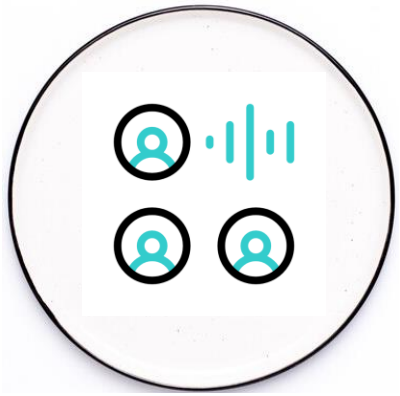
```
<p *ngIf ="condition">
        condition is true and ngIf is true.
</p>
<p *ngIf ="!condition">
        condition is false and ngIf is false.
</p>
```

Aitrich

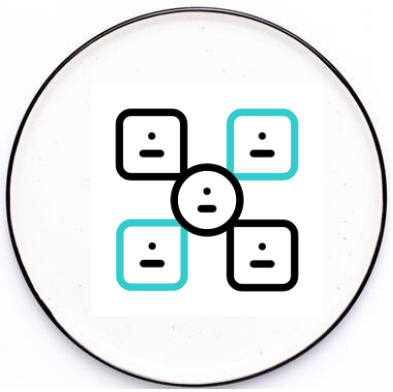
Attribute Directive

An Attribute or style directive can change the appearance or behavior of an element.

ngStyle : Dynamically applies CSS styles to an element based on expressions in the component.



ngClass : Dynamically adds or removes CSS classes to an element based on conditions in the component.



ngModel : It is a built-in Angular directive that provides two-way data binding between a form input control and a property in the component.

Aitrich

QUESTIONS

Aitrich