

Angular 4

Data Binding

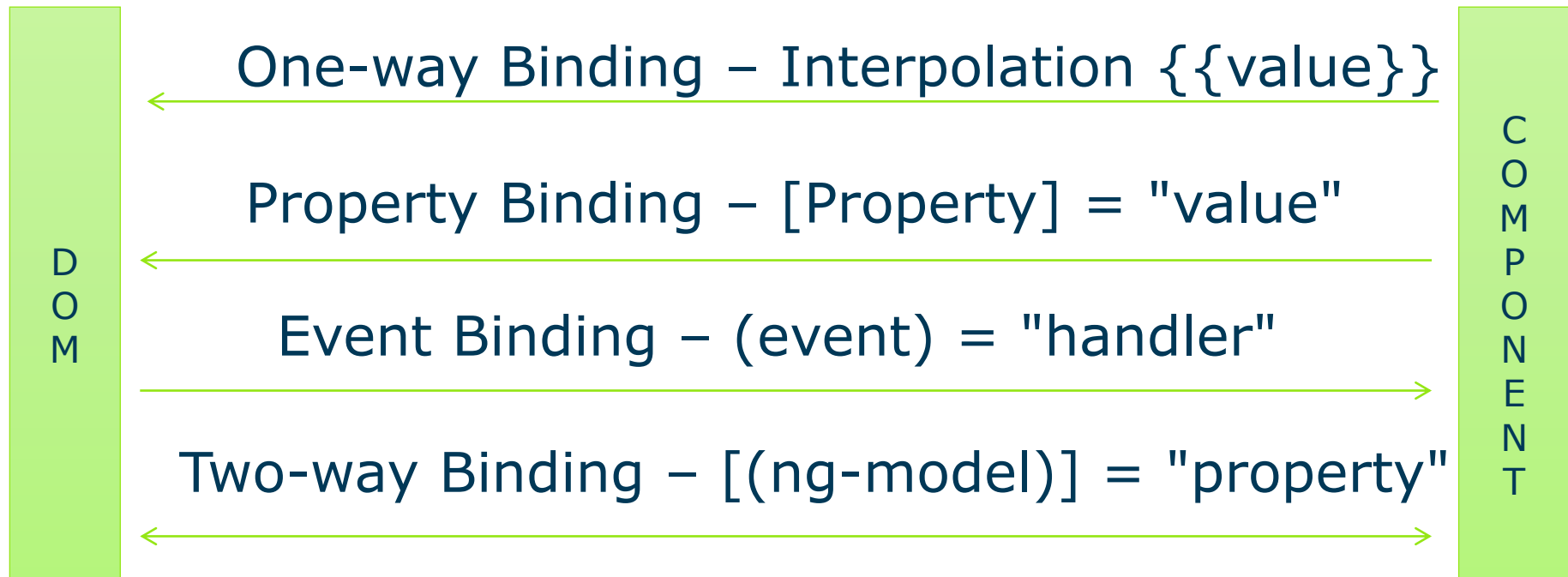
Databinding



Angular supports data binding, a mechanism for coordinating parts of a template with parts of a component.

Binding markups are added to the template HTML to tell Angular how to connect both sides.

There are four forms of data binding syntax. Each form has a direction to the DOM, from the DOM, or in both directions.





Interpolation(One-Way binding)

Interpolation performs one way binding from the class property to the template given in a double curly braces.

Using Interpolation operations like concatenation, simple calculations, calling method on a class can be performed.

Interpolation is used to insert the interpolated strings into the text between HTML elements.

The syntax between the interpolation curly braces is called as template expression.

Angular evaluates that expression using the component as the context.

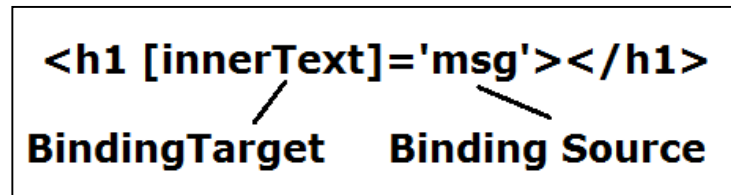
- Angular looks to the component to obtain property values or to call methods.
- Angular then converts the result of the template expression to a string and assigned that string to an element or directive property

```
<h2>{{greet}}</h2>    {{'Hi' + greet}}    {{'Hi' +  
                        getMessage()}}    {{2+1}}
```



Property Binding

Property binding allows us to set a property of an element to the value of a template expression.



Binding target is always enclosed in square brackets and the Binding source is always enclosed in quotes and specifies the template expression

Like interpolation property binding is one way from the source class property to the target element property

Property binding effectively allows to control the template DOM from a component class.

The general guideline is to prefer property binding all for interpolation. However to include the template expression as part of a larger expression then use interpolation

Demo



PropertyBinding



Event Binding

Event binding is used to send information from the element property to the component class property to respond for user events.

- For instance to perform an operation when the user clicks a button a component listens for user actions using event binding.

| |
|---|
| <pre><button (click)='sayHi()'>SayHi</button></pre> <div><div>TargetEvent</div><div>TemplateStatement</div></div> |
|---|

The name of the bound event is enclosed in parentheses identifying it as the target event.

Template statement is often the name of a component class method enclosed in quotes.

If the defined event occurs; the template statement is executed calling the specified method in the component

Demo



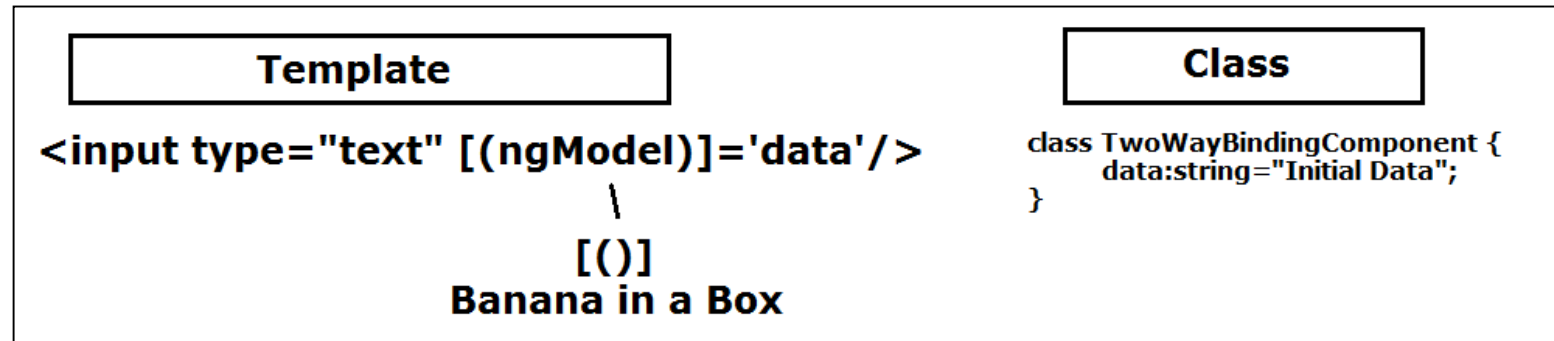
EventBinding
GetUserInput



Two-way Binding

To display a component class property in the template and update that property when the user makes a change with user entry HTML elements like input element two-way binding is required.

In Angular ***ngModel*** directive is used to specify the two way binding.



ngModel in square brackets is used to indicate property binding from the class property to the input element

Parentheses to indicate event binding to send the notification of the user entered data back to the class property

Demo

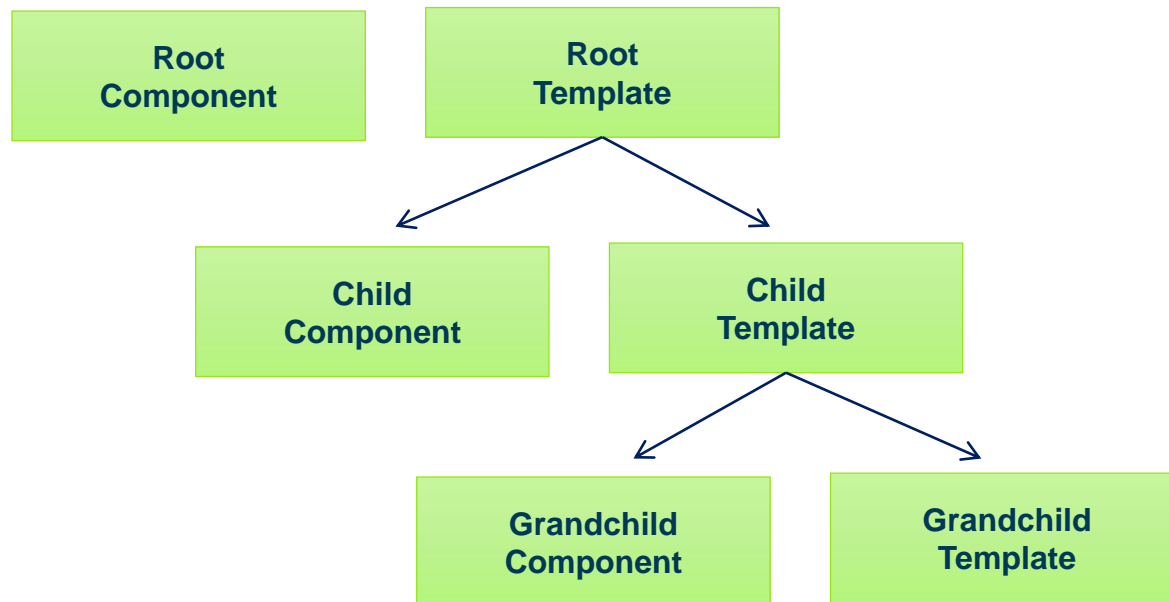


TwoWayBinding



Nested Components

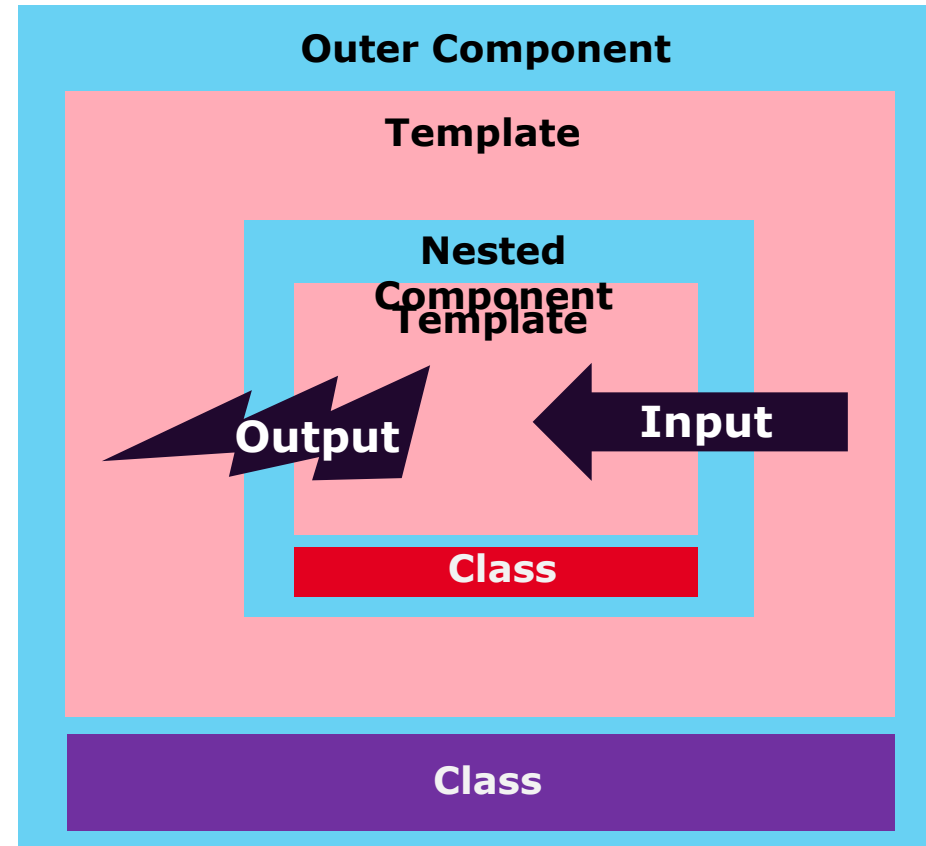
- Components have templates which may contain other components.
- Outer component is referred as the container or parent component
- Inner component is referred as the nested or child component.





Using @Input and @Output

- Nested component receives information from its container using input properties(@Input) and outputs information back to its container by raising events(@Output).
- Input, Output & EventEmitter need to be imported in Nested component from @angular/core module.
- emit() method is used to trigger the event by emitting data from inner component to outer component which can be accessed via \$event



Demo



Building-NestedComponents



Summary

Angular provides 4 forms of data binding syntax

- One-way Binding – Interpolation {{value}}
- Property Binding – [Property] = "value"
- Event Binding – (event) = "handler"
- Two-way Binding – [(ng-model)] = "property"

Nested component receives information from its container using input properties(@Input) and outputs information back to its container by raising events(@Output).

When Parent component render, it recursively render its children component
emit() method is used to trigger the event by emitting data from inner component to outer component which can be accessed via *\$event*