

1. What is an expression in AngularJS?

Ans: Expressions are used to bind application data to html. Expressions are written inside double flower braces {{expression}}. Expressions behaves in same way as ng-bind directives. AngularJS application expressions are pure javascript expressions and outputs the data where they are used.

2. What is Interpolation?

Ans: Interpolation uses the {{expression}} to render the bound value to the component template, Double curly braces {{ }} contain the template expression which allow us to read primitive or objects values from component properties. From Angular1.X we are using formally {{ }} for interpolation.

3. What is the purpose of the Property Binding / one-way binding?

Ans: Property Binding is passing the data from your component class and setting the value of a given element in the view. It's also called as one-way binding because the data will be passed from the component class to the view in one-way directional.

4. How many ways we can do one-way binding in Angular?

Ans: We can do one-way binding in four ways:-

- i. Using double curly braces {{expression}} interpolation.
ex: <input type="text" value="{{myname}}" />
- ii. By wrapping the element property with square braces [prop]
ex:<input type="text" [value]="myname" />
- iii. By adding bind- before the element property ex: bind-value, bind-src
ex: <input type="text" bind-value="myname" />
- iv. Using ngModel [ngModel], where ngModel is an angular directive.
ex:<input type="text" [ngModel]="myname" />

5. What is Event Binding?

Ans: Event Binding is also a one-way binding where it is exactly opposite to the Property Binding where it sends information from view to component class based on the user actions. To raise any event either it may be mouse events, click events, keyboard events etc. need to wrap the event property with parenthesis ()

6. Give some examples of commonly used events?

Ans:

- i. Click
- ii. Dblclick
- iii. Mouseenter
- iv. Mouseup
- v. Mousedown

7. What is Two-way binding?

Ans: In two-way binding the component class and its view can communicate in two-way directional, i.e. data can be passed from component class to view and vice versa.

In Angular 1.X ng-model is the directive, which is used for two-way communication, but now from Angular 2.X \$watch and \$digest cycles were no more used to watch the changes that occur in the models. Now we are using ngModel.

```
ex:<p>Hello {{myname}}</p>
    <input type="text" [(ngModel)] ="myname" />
```

8. How can we will access the element without using ngModel?

Ans: Angular >= 2.X has a feature of “template reference variables”, with this we can directly access the element from the DOM, it is declared by preceding an identifier with hash/pound (#).

ex: <input type="text" #txtName (keyup)="0" />

9. What is **@Input** decorator and what is the use of this decorator?

Ans : @Input decorator is used to define an input property to achieve component property binding.

@Input is one way communication from parent to child component.

10. What is @Output decorator ? Why @Output decorator?

Ans: @Output decorator is used to define output property to achieve custom event binding. This is also one way communication child to parent component.