

1. What decorator we used in the linking an internal property to an input property on host element.
- a) @ElementRef
  - b) @HostListener
  - c) @HostBinding
  - d) @Renderer

Ans:c

2. NgIf and NgFor are the\_\_\_\_\_.
- a) Components directive
  - b) Structural directive
  - c) Attribute directive

Ans: b

3. ngClass is used to add and remove CSS classes on an HTML element.
- a) Yes
  - b) No

Ans:a

4. what decorator we used in the Call back event handler methods to listen the events that are raised on the host element.
- a) @ElementRef
  - b) @HostListener
  - c) @HostBinding
  - d) @Renderer

Ans: b

5. ngStyle is the\_\_\_\_\_.
- a) Components directive
  - b) Structural directive
  - c) Attribute directive
  - d) None of the above

Ans: c

6. You can't define \_\_\_\_\_ in directive.
- a) attribute
  - b) pipe
  - c) expression
  - d) none of the above

Ans:b

7. ngClass is added into our component element and its value is given as ?
- a) Object
  - b) reference

- c) class
- d) value

Ans:c

8. NgFor and NgIf are \_\_\_\_\_
- a) Structural Directives
  - b) Components
  - c) Attribute directives

Ans:a

9. import { Component } from '@angular/core'

```
@Component({  
  
  selector: 'data-binding',  
  
  templateUrl: './directives.html'  
  
})  
  
export class AppComponent {  
  
  hide = false;  
  
  myname = "Surya";  
  
}  
  
(in directives.html)  
  
<p *ngIf="hide ;else template1">{{myname}}</p>  
  
<ng-template #template1>  
  Hello {{myname}}  
</ng-template>
```

Output will be?

- a) Surya
- b) Hello Surya
- c) Hello
- d) error

Ans: a

10. HostListener syntax

- a) \$HostLisener("event")methodName( ) { //your code }
- b) @HostListener("event") methodName( ) { //your code }

Ans: b

Deccansoft