

Questions

1. ****What is the primary purpose of Angular decorators?****

- a) To enhance the appearance of the UI
- b) To provide metadata to Angular classes
- c) To manage state in Angular applications
- d) To handle HTTP requests

2. ****Which decorator is used to define a component in Angular?****

- a) @Injectable
- b) @Directive
- c) @Component
- d) @Pipe

3. ****What is the purpose of the @Injectable decorator?****

- a) To mark a class as a component
- b) To mark a class as a service that can be injected
- c) To define a directive

- d) To create a custom pipe

4. ****Which of the following is NOT a parameter of the @Component decorator?****

- a) selector
- b) templateUrl
- c) styleUrls
- d) useFactory

5. ****How do you apply the @Directive decorator?****

- a) On a service class
- b) On a component class
- c) On a class that modifies the DOM
- d) On a module

6. ****What is the function of the @NgModule decorator?****

- a) To declare a service
- b) To declare a module and its dependencies
- c) To create a component
- d) To handle forms

7. ****The @Input decorator is used for:****

- a) Sending data from parent to child components
- b) Emitting events from child to parent components
- c) Defining a service
- d) Creating a pipe

8. ****Which decorator would you use to make a property in a child component accessible from the parent component?****

- a) @Output
- b) @HostBinding
- c) @Input
- d) @HostListener

9. ****What is the purpose of the @Output decorator?****

- a) To mark an output property for event binding
- b) To define an input property for data binding
- c) To create a custom directive
- d) To inject a service

10. ****How is the @HostListener decorator used?****

- a) To listen to events on a host element
- b) To bind a property to a host element
- c) To declare a module
- d) To create a component

11. ****The @Pipe decorator is used to:****

- a) Create a service
- b) Define a component
- c) Transform data in templates
- d) Manage forms

12. ****Which decorator is used to add metadata to a property that binds to a host element's attribute?****

- a) @HostListener
- b) @HostBinding
- c) @Input
- d) @Output

13. ****What is the purpose of the @Self decorator?****

- a) To inject a service provided in the current injector
- b) To inject a service from a parent injector
- c) To declare a component
- d) To define a module

14. ****The @SkipSelf decorator is used to:****

- a) Inject a service from a parent injector, skipping the current one
- b) Skip a module
- c) Declare a directive
- d) Handle HTTP requests

15. ****Which decorator would you use to declare a service in Angular?****

- a) @Injectable
- b) @Component
- c) @Directive
- d) @Pipe

16. ****The @Optional decorator is used for:****

- a) Creating an optional input property
- b) Marking a dependency as optional
- c) Declaring a module
- d) Binding an output property

17. ****Which decorator helps in listening to events in Angular?****

- a) @Output
- b) @HostBinding
- c) @HostListener
- d) @Directive

18. ****The @Attribute decorator is used to:****

- a) Inject a specific attribute value from the host element
- b) Bind a property to a host element
- c) Create a service
- d) Handle HTTP requests

19. ****What is the function of the @ContentChild decorator?****

- a) To reference a single child element in a component's content
- b) To reference multiple child elements
- c) To declare a service
- d) To handle form inputs

20. ****Which decorator is used to reference multiple projected content elements in a component?****

- a) @ContentChild
- b) @ContentChildren
- c) @ViewChild
- d) @ViewChildren

Answers

1. ****b) To provide metadata to Angular classes****
2. ****c) @Component****
3. ****b) To mark a class as a service that can be injected****
4. ****d) useFactory****
5. ****c) On a class that modifies the DOM****

6. ****b)** To declare a module and its dependencies******
7. ****a)** Sending data from parent to child components******
8. ****c)** **@Input********
9. ****a)** To mark an output property for event binding******
10. ****a)** To listen to events on a host element******
11. ****c)** Transform data in templates******
12. ****b)** **@HostBinding********
13. ****a)** To inject a service provided in the current injector******
14. ****a)** Inject a service from a parent injector, skipping the current one******
15. ****a)** **@Injectable********
16. ****b)** Marking a dependency as optional******
17. ****c)** **@HostListener********
18. ****a)** Inject a specific attribute value from the host element******
19. ****a)** To reference a single child element in a component's content******
20. ****b)** **@ContentChildren********