



Angular Interview Questions

Practice for Angular interviews by solving TestDome questions. Our interview questions are used by more than 5,000 companies and 450,000 individual test takers.

Jobseekers: Certify Your Knowledge

Companies: Use Our Tests for Screening

Take a Certification Test

Buy a Pack Of Candidates

Try to solve 5 Angular interview questions below. Hints can help you find answers to questions you are having trouble with.

Used by



1. Component

Easy  COMPONENTS

Which of the following statements about components in Angular are correct?

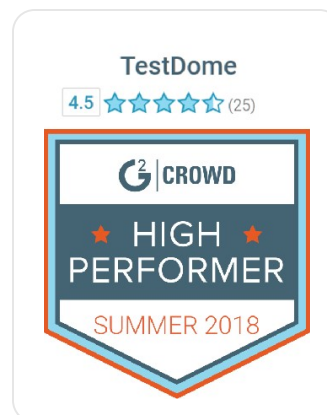
(Select all acceptable answers.)

- ☐ The properties of a component's children are available in the component's constructor.
- ☒ When a component depends on a service, the injector can be used to configure dependency injection.
- ☒ A component defines its input parameters with the @Input decorator.
- ☐ A component is a type of directive and as such, should be defined by the Directive class decorator.
- ☐ A component selector must reference the class name in lowercase hyphenated format.
- ☒ A component's ngOnDestroy method is called just before Angular destroys the component.

[Report an issue](#)

Your score is 100%, perfect!

What Others Say



Chat with us!

2. Welcome

Easy  COMPONENTS

Consider the following component:

```
import { Component, Input } from
  '@angular/core';

@Component({
  selector: 'welcome',
  template: `<h1>Welcome to {{name}}!</h1>`,
  styles: [`h1 { font-family: Lato; }`]
})
export class WelcomeComponent {
  @Input() name: string;
}
```

Select the statements about its use (in another components template or module) that are correct.

(Select all acceptable answers.)

- ☒ `<welcome name="TestDome"></welcome>` will display: "Welcome to TestDome!".
- ☐ `<welcome></welcome>` will display nothing.
- ☒ `@NgModule({ declarations: [WelcomeComponent] }) export class WelcomeModule {}` declares that the welcome component belongs to the welcome module.
- ☐ `<hello name="{{ name }}"></hello>` will display: "Welcome to name!".

[Report an issue](#)

Your score is 100%, perfect!

3. Animal Noise

Easy  EXPRESSIONS

Consider the following component, which can be used to model an animal and the noise it makes.

```
import {Component, Input, Output} from
  '@angular/core';

@Component({
  selector: 'animal-noise',
  template: `
```

Simple, straight-forward technical testing

TestDome is simple, provides a reasonable (though not extensive) battery of tests to choose from, and doesn't take the candidate an inordinate amount of time. It also simulates working pressure with the time limits.

Jan Opperman,
Grindrod Bank

On the TestDome Blog

How to Screen Software Developers

Screening programmers is different than screening other employees, and we learned that the hard way.

First, you will almost always want to hire an already good programmer. Why not hire an average one and then train them up? Years of experience and training don't guarantee...

[Read more](#)

[Chat with us!](#)

```

        <span>{{animal}}</span>
        <button (click)="makeNoise()">Make
noise</button>
    `
})
export class AnimalNoise {
    @Input('animal') animal: string;
    @Input('noise') noise: string;

    makeNoise() {
        alert(`${this.noise}`);
    }
}

```

Select the statements about the *AnimalNoise* component that are correct.

(Select all acceptable answers.)

- ☐ Component, Input and Output are all required imports for this component.
- ☐ Both animal and noise inputs must be provided when including the AnimalNoise component in a template.
- ☒ The 'animal' parameter of the @Input('animal') declaration does not alter the interface of the component.
- ☒ When included in a components template, the AnimalNoise component creates a span containing the interpolated animals name and a button bound to *makeNoise()*.
- ☐ The AnimalNoise component can be included in another template using the <AnimalNoise> tag.

[Report an issue](#)

Your score is 100%, perfect!

4. Router Module

Hard 

HTTP ROUTING

Consider the following application module:

```

import { NgModule }           from
 '@angular/core';
import { RouterModule, Routes } from
 '@angular/router';
import { HomeComponent }      from
 './home.component';
import { ItemDetailComponent } from

```

[Chat with us!](#)

```

    './item-detail.component';
    const routes: Routes = [
      { path: '', redirectTo: '/home',
        pathMatch: 'full' },
      { path: 'home', component:
        HomeComponent },
      { path: 'detail/:id', component:
        ItemDetailComponent, outlet: 'route1' }
    ];
    @NgModule({
      imports: [ RouterModule.forRoot(routes)
    ],
      exports: [ RouterModule ]
    })
    export class AppRoutingModule {}

```

Which of the following statements about the routers behavior are correct?

(Select all acceptable answers.)

- ☐ The home component can only be accessed via the root URL redirect.
- ☐ The id parameter is optional when a call is made to the /detail/ URL.
- ☒ Optional parameters can be passed to any component via the query parameters of the ActivatedRoute.
- ☒ /detail/100 will use the <router-outlet name='route1'> to determine the position of the view.
- ☐ Route that redirects to *HomeComponent* would activate on same routes if the *pathMatch: 'full'* property was removed.

[Report an issue](#)

Hint 1: The router module is used to map a client's URL to the appropriate HTML page and position.

Your score is 100%, perfect!

5. Address Template

[Easy](#) COMPONENTS

Consider the following components:

```

export class Address {
  street: String;

```

[Chat with us!](#)

```

    city: String;
    zipCode: String;
}

@Component({
  selector: 'app-address',
  templateUrl: './address.component.html',
  styleUrls: ['./address.component.css']
})
export class AddressComponent implements OnInit {

  @Input() address: Address;
  constructor() { }

  ngOnInit() {
  }
}

@Component({
  selector: 'app-address-list',
  templateUrl: './address-list.component.html',
  styleUrls: ['./address-list.component.css']
})
export class AddressListComponent implements
OnInit {

  @Input() addresses: Address[];
  constructor() { }

  ngOnInit() {
  }
}

```

Below is the file address-list.component.html. Fill in the blanks, so that the template renders one *li* element per address and it correctly one-way binds each *address* to the *AddressComponent* field.

```

<ul>
  <li *ngFor [?]="let address of [?]">

```

Chat with us!

```
<app-address [address] ="address"></app-  
address>  
</li>  
</ul>
```

[Report an issue](#)

Your score is 100%, perfect!

If you feel ready, take one of our timed public Angular Interview

Questions tests:



Angular Online Test ([Easy](#) / [Hard](#))

Angular and TypeScript Online Test ([Easy](#) / [Hard](#))



HTML/CSS, Angular, TypeScript, and Node.js Online
Test ([Easy](#) / [Hard](#))

HTML/CSS, Angular, TypeScript, Node.js, and SQL
Online Test ([Easy](#) / [Hard](#))

Not exactly what you are looking for? Go to our [For Jobseekers](#)
section.

Jobseekers: Certify Your Knowledge

Take a Certification Test

Companies: Use Our Tests for Screening

Buy a Pack Of Candidates

EXPLORE [Home](#) [Tour](#) [Tests](#) [Questions](#) [Pricing](#) [For Jobseekers](#) [Sign in](#) [Back to Top](#)

LEGAL [Terms of Use](#) [Privacy Policy](#) [Contact Us](#)

© TestDome

[Chat with us!](#)