Angular Interview Practice (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Table of Contents

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn’t just about memorization, it’s about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you’re preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You’ll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it’s correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It’s ideal for learners who want to:

- **Practice answering questions under realistic conditions,**
- **Improve accuracy and speed,**
- **Review explanations to strengthen weak areas, and**
- **Approach the exam with greater confidence.**

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn’t about getting every question right the first time, it’s about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let’s begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here’s how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Don’t worry about getting everything right, your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations, and take breaks to retain information better.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don’t just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning.

7. Use Other Tools

Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There’s no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly — adapt the tips above to fit your pace and learning style. You've got this!

Questions

1. **Which decorator is used to create a custom pipe in Angular?**
   - **A. @Component**
   - **B. @Injectable**
   - **C. @Pipe**
   - **D. @Directive**

2. **What does the DOM property 'checked' indicate in a checkbox element?**
   - **A. It displays the default state**
   - **B. It shows if the checkbox is currently checked**
   - **C. It represents the initial value when the page loads**
   - **D. It indicates the total number of checkboxes**

3. **Which best describes the component structure in Angular?**
   - **A. All components are global**
   - **B. Encapsulated into reusable components**
   - **C. Fixed component structure**
   - **D. Components are only for UI display**

4. **What is the role of the angular.json file in an Angular project?**
   - **A. It contains environment variables for the project**
   - **B. It stores the routing configuration of the application**
   - **C. It contains configuration settings for the Angular CLI**
   - **D. It holds documentation for project dependencies**

5. **When accessing a boolean attribute, how is it represented in the DOM?**
   - **A. As a numeric value**
   - **B. As a true or false property**
   - **C. As a string value**
   - **D. As an object**

6. How can you enable routing in a new Angular project during its creation?
   - A. By using the --routing flag with ng generate
   - B. By using the --routing flag when creating the project
   - C. By updating the angular.json file manually
   - D. By creating a router module after project creation

7. What does the ng generate command facilitate in an Angular application?
   - A. It creates Angular features such as components, directives, services, modules, and pipes
   - B. It updates existing features in the Angular application
   - C. It removes features from the Angular application
   - D. It lists all the generated features in the application

8. What is the root node of an HTML document's DOM?
   - A. <body> element
   - B. <html> element
   - C. <head> element
   - D. <div> element

9. What is a suggested use case for Virtual Scroll in Angular applications?
   - A. For small lists of items
   - B. For lists that require real-time updates
   - C. For applications with large datasets
   - D. For sorted lists of items

10. What method is used to check if an HTML attribute exists?
   - A. hasAttribute()
   - B. getAttribute()
   - C. checkAttribute()
   - D. existsAttribute()

Answers

1. **C**
2. **B**
3. **B**
4. **C**
5. **B**
6. **B**
7. **A**
8. **B**
9. **C**
10. **A**

Explanations

1. Which decorator is used to create a custom pipe in Angular?

A. @Component

B. @Injectable

<u>C. @Pipe</u>

D. @Directive

To create a custom pipe in Angular, the appropriate decorator to use is the @Pipe decorator. This decorator allows you to define the metadata for the pipe, including its name, which is how the pipe can be referenced within templates. By using @Pipe, you enable Angular to understand the functionality of your custom transformation logic that the pipe provides, ultimately allowing you to manipulate data displayed in your templates effectively. When defining a pipe, you'll typically implement the PipeTransform interface, which requires you to define the transform method that contains the logic for how input values should be transformed. This structure is pivotal for ensuring that Angular can use your custom pipe seamlessly during data binding in Angular's template syntax. While other decorators like @Component and @Directive serve their respective purposes in defining components and directives, they do not provide the specific functionality associated with transforming data, which is the central role of a pipe. @Injectable is used for services and dependencies, further distinguishing it from the pipe functionality. Thus, @Pipe is uniquely positioned to facilitate custom data transformation within Angular applications.

2. What does the DOM property 'checked' indicate in a checkbox element?

A. It displays the default state

<u>B. It shows if the checkbox is currently checked</u>

C. It represents the initial value when the page loads

D. It indicates the total number of checkboxes

The 'checked' property of a checkbox element in the Document Object Model (DOM) is crucial for determining the current state of the checkbox at any given time. This property returns a Boolean value—true if the checkbox is selected (checked) and false if it is not. It is essential in forms and interactive applications, as it enables developers to understand if the user has interacted with the checkbox. Using the 'checked' property allows scripts to assess the current status of the checkbox, making it possible to implement conditional logic based on whether the checkbox is checked or not, enhancing the overall user experience. Since checkboxes can be toggled by user interaction, having a property that indicates their current checked state accurately reflects the user's choice at that moment. The other options refer to different aspects of checkboxes that do not directly relate to the current user interaction with the element. For instance, the default state is not indicative of the current condition after user interaction. The initial value upon page load may differ from the current state, and the total number of checkboxes does not correlate with the state of a single checkbox. Thus, the 'checked' property specifically provides real-time feedback on whether the checkbox is selected, making the assertion about it showing if the checkbox is currently

3. Which best describes the component structure in Angular?

A. All components are global

B. Encapsulated into reusable components

C. Fixed component structure

D. Components are only for UI display

In Angular, the component structure is best described as encapsulated into reusable components. This means that each component is designed to be an independent, self-contained unit of functionality that can be reused throughout the application. Components are the building blocks of Angular applications, allowing developers to break down complex UIs into smaller, maintainable pieces. Each component has its own template, styles, and logic, which promotes a clear separation of concerns and facilitates better organization of the codebase. This encapsulation also enhances reusability; once a component is created, it can be utilized in different parts of the application without needing to rewrite the functionality. Additionally, Angular provides mechanisms such as inputs and outputs to allow communication between components, fostering modularity and flexibility in the overall application architecture. The other choices do not accurately represent the fundamental principles of Angular's component structure. Components being global would reduce modularity and lead to challenges with maintainability. A fixed component structure contradicts the dynamic and flexible nature of Angular's architecture, which encourages developers to create components that can be easily adapted and reused. Lastly, while components do often handle UI display, they are not limited to this role; they also encapsulate the logic and data manipulation associated with their specific concerns. Therefore, highlighting

4. What is the role of the angular.json file in an Angular project?

A. It contains environment variables for the project

B. It stores the routing configuration of the application

C. It contains configuration settings for the Angular CLI

D. It holds documentation for project dependencies

The angular.json file plays a crucial role in an Angular project as it contains configuration settings for the Angular CLI. This includes details such as the project's name, the root folder, the source directory, build options, and various settings related to the development and production builds. It manages project settings across multiple environments and facilitates the specification of various build targets, such as configurations for building and serving the application. Having a central configuration file like angular.json helps streamline the development process by allowing developers to configure and customize various aspects of their project easily, all from a single location. This centralization is particularly useful for handling complex projects with multiple applications and libraries under the same workspace, as it organizes and simplifies the management of various configurations.

5. When accessing a boolean attribute, how is it represented in the DOM?

A. As a numeric value

B. As a true or false property

C. As a string value

D. As an object

In the DOM, boolean attributes are represented as properties that can either evaluate to true or false. When a boolean attribute is present on an element, it is considered true, and when it is absent, it is false. This means that the browser treats these attributes as properties rather than text or objects. For example, when using the `disabled` attribute on an input element, if the attribute is present, the value of the `disabled` property will be true. If it is not present, the property will report as false. This characteristic allows developers to easily check the state of the attribute using JavaScript, enhancing dynamic behavior in applications. The other representations listed—numeric values, string values, and objects—do not accurately reflect how boolean attributes function within the DOM. Boolean attributes do not have a numeric or string representation; they simply toggle between the presence (true) and absence (false) in the DOM structure.

6. How can you enable routing in a new Angular project during its creation?

A. By using the --routing flag with ng generate

B. By using the --routing flag when creating the project

C. By updating the angular.json file manually

D. By creating a router module after project creation

Enabling routing in a new Angular project is achieved by utilizing the --routing flag during the project creation process. This flag informs the Angular CLI to set up the necessary files and configurations to support routing right from the beginning. When you create a new project with this option, Angular automatically generates a routing module that can be used to define the routes for your application. This setup saves time and ensures that the routing configuration adheres to Angular's best practices. This approach stands out because it integrates routing seamlessly into the project architecture from the start, allowing developers to focus on building features rather than spending time setting up routing afterward. The automatically generated routing module can be easily modified to define various application routes, manage lazy loading, and handle parameters, making it a foundational aspect for larger applications. Other methods of enabling routing, such as updating the angular.json file manually or creating a router module after the project's initial setup, require additional steps and do not provide the same level of initial convenience and integration that using the --routing flag offers during project creation.

7. What does the ng generate command facilitate in an Angular application?

A. It creates Angular features such as components, directives, services, modules, and pipes

B. It updates existing features in the Angular application

C. It removes features from the Angular application

D. It lists all the generated features in the application

The ng generate command is a powerful tool in Angular that streamlines the process of scaffolding new features within Angular applications. When used, it can create essential building blocks such as components, directives, services, modules, and pipes, allowing developers to quickly set up the necessary structure for their application without manually creating files and writing boilerplate code.   This command significantly enhances productivity and consistency in Angular development. For example, when generating a new component, ng generate automatically creates the component TypeScript file, the HTML template, the CSS styles, and the necessary test files, establishing a comprehensive setup with just a single command. The other options focus on functionalities that the ng generate command does not perform. It does not update existing features, remove components, or list features; those tasks are handled by other commands or processes in the Angular CLI. Thus, the primary purpose of ng generate is to facilitate the creation of new features efficiently.

8. What is the root node of an HTML document's DOM?

A. <body> element

B. <html> element

C. <head> element

D. <div> element

The root node of an HTML document's DOM is the <html> element. This element serves as the top-level container for the entire HTML structure of a web page. All other elements, including <head> and <body>, are considered descendants of the <html> element. As the root, it establishes the document as an HTML document and provides the necessary context for the browser to interpret and render the subsequent elements. When a browser parses an HTML document, it creates a tree-like structure known as the Document Object Model (DOM). The <html> element is at the very top of this hierarchy, and it encompasses both the <head> and <body> elements. Within the <head>, metadata about the document is specified, such as title, links to stylesheets, and scripts. The <body> then contains the content that is displayed to the user. Other elements, like <div> or <body>, while important parts of the structure and functionality of an HTML document, do not serve as the root node. The <head> element is also pivotal, but it is a child of the <html> element, further illustrating that the <html> element is indeed the root of the DOM structure.

9. What is a suggested use case for Virtual Scroll in Angular applications?

A. For small lists of items

B. For lists that require real-time updates

C. For applications with large datasets

D. For sorted lists of items

Virtual Scroll is an optimal feature in Angular applications designed to efficiently render large datasets. When dealing with extensive lists, rendering every item in the DOM can lead to performance issues, such as increased load times and sluggish interactions. Virtual Scroll addresses this by only rendering the items that are currently viewable in the viewport, which significantly reduces the number of DOM elements created and maintained. In cases where applications feature large datasets, adopting Virtual Scroll allows for a seamless user experience as it enhances performance by minimizing memory footprint and improving rendering times. This is particularly important in user interfaces that need to maintain responsiveness even when dealing with many items, such as product listings, chat messages, or any scenario where a substantial amount of data is displayed. Using Virtual Scroll for small lists, real-time updates, or sorted lists does not take full advantage of its capabilities. Small lists typically do not require optimization since they can be rendered efficiently without performance concerns. Similarly, while Virtual Scroll can help with real-time data display by updating the visible items efficiently, it is not specifically designed for real-time scenarios. Lastly, sorted lists can be efficiently managed without the necessity for Virtual Scroll, especially if their size remains manageable. Therefore, the use of Virtual Scroll is most beneficial in applications requiring the handling of large

10. What method is used to check if an HTML attribute exists?

A. hasAttribute()

B. getAttribute()

C. checkAttribute()

D. existsAttribute()

The method used to check if an HTML attribute exists is indeed hasAttribute(). This method is part of the Element interface in the DOM (Document Object Model) and allows you to determine whether a specified attribute is present on a given element. When you use hasAttribute() and pass the name of the attribute as an argument, it returns true if the attribute is found on the element, and false if it is not. This is particularly useful for conditionally performing actions based on whether certain attributes are set or not, thus enhancing your ability to interact dynamically with the DOM. In contrast, getAttribute() retrieves the value of an attribute if it exists, but it does not check for the presence of the attribute itself. checkAttribute() and existsAttribute() are not standard methods defined in the DOM, which is why they are not correct for this question.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we’d love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://angularinterview.examzify.com

We wish you the very best on your exam journey. You've got this!