Vue.js Interview Questions for Beginners

1. What exactly is Vue.js?

**Ans:** Vue.js is a framework for creating user interfaces that are built on JavaScript. It's a framework that's progressive. Vue.js is responsible for the view layer of an application. Vue.js may be used to create single-page applications. Vue is a framework that handles higher-level issues like fast DOM rendering, reactive state management, tools, client-side routing, and server-side rendering.

2. What exactly is the Prop component?

**Ans:**Data is passed from the parent component to the child component using Component Prop. A component can have as many props as it wants. The prop is a custom attribute that becomes a property on the component instance when a value is supplied.

From parent to a child component, a component Prop forms a one-way down binding. When the parent's property changes, it affects the child, but not the other way around.

|  |
| --- |
| ***Are you looking forward to becoming a Javascript Developer? Check out the***[***Javascript Online Training***](https://mindmajix.com/javascript-training)***and get certified. Enroll Now!!*** |

3. In Vue.js, what is reactivity?

**Ans:** **Reactivity** – When you change the data value, the page is updated to reflect the changes. Data properties and computed properties in Vue are reactive. Vue.js services are not reactive. Visit Reactivity in Vue.js for additional information.

4. Describe the lifecycle hooks in Vue.js?

**Ans:** Lifecycle hooks are functions that every Vue instance runs through. Each vue has eight lifecycle hooks. An instance of js

* **Before Create -**This is the first lifecycle hook that is called when a Vue instance has been created.
* **Created -**It's called immediately after the 'Before creates' hook, but the Vue instance has already set initial properties, data, etc.
* **Before mount -**invoked right before the instance is mounted on the DOM. The template has been completed at this time.
* **Mounted -** This is the name given to the template once it has been filled with data and is fully functional.
* **Before Updated -**When any changes to data require the DOM to be changed, this method is called before the update.
* **Updated -**It's called after the DOM has been updated.
* **Before destroy -** It's a location where you can clean up your resources before terminating the Vue instance.
* **Destroyed -**When all Vue instances have been destroyed, this method is called. When you call the destruct method on an object in code, it will be activated.

5. Describe the distinction between v-show and v-if?

**Ans:**The items are shown or hidden using v-show and v-if. However, there are some distinctions. The v-if directive is used to render a block conditionally. It features lazy behaviour, which means that if the initial condition is false, the block will not be rendered until it changes. During a condition change, v-if destroys and recreates the elements.

When your condition does not frequently change at runtime, use the v-if directive because it has a lower initial render cost but a higher toggling cost.

The v-show directive can also be used to render a block conditionally. The element is always rendered by v-show. Instead of removing the element or block from the DOM, it simply sets the CSS show attribute.  
It has a high initial rendering cost but a low toggle cost, so use the v-show directive if you need to toggle frequently.

[[](https://bit.ly/3if9dmk)](https://bit.ly/3if9dmk" \o "Subscribe MindMajix YouTube Channel" \t "_blank)

6. Explain how to use $parent?

**Ans:**In the child component, $parent is used to retrieve the parent component instance. It's more challenging to test and debug the programme, and you can't figure out where the mutation comes from. Vue also has a $child function that returns a child component instance, similar to $parent.

7. Describe the component of a single file?

**Ans:**SFCs, or Single File Components, are similar to other components in that they are self-contained in their own files. The following are some of the benefits of SFCs.

* **Global Definitions -**This refers to the fact that all SFCs are registered globally and given unique names.
* **Robust Templates -**Instead of using a single template property like other components, you can easily create template code in SFCs.
* **CSS Support –**SFCs allow you to apply CSS to the component itself.
* **Support for preprocessors –** preprocessors like babel, SAAS, and others can be simply used in SFCs.

8. What are the most essential elements of the State Management Pattern?

**Ans:** The primary components of state management are state, view, and actions.

The components are described in full below.

* Our app is driven by the state, which is the source of truth.
* The view is nothing more than a declarative state mapping.
* The actions are the various ways in which the state could change due to user input from the view.

9. How do you verify if a mutation occurs?

**Ans:** It will be easy to test mutations because they are simply functions that rely only on their arguments. You must keep mutations contained within your store.

The mutations should also be exported as a named export, apart from the default export.

***[Also Read:***[***JavaScript Frameworks***](https://mindmajix.com/javascript-frameworks)***]***

10. What exactly is the map State helper?

**Ans:**Creating a computed property whenever we wish to use the store's state property or getter in a Vuex application will be repetitive and lengthy, especially if a component requires many state properties.

11. Is using variables for mutation types required?

**Ans:** No, it isn't required. However, you may have noticed that state management frameworks like Flux and Redux employ constants for mutation types. This is just a preference, but it's beneficial to use tooling like linters, and putting all constants in a single file lets your collaborators see what modifications are allowed across the entire application at a glance.

12. Is it possible to use styled components in Vue.js?

**Ans:**[ReactJS](https://mindmajix.com/react-js-tutorial) applications typically employ styled components. VueJS styled-components library is available under the styled component library if you wish to use it for VueJS apps. VueJS Styled Component is a javascript module that helps you style your Vue.js projects.

13. What are the benefits of using Vue CLI?

**Ans:** The primary features of Vue CLI are listed below.

* @vue/cli scaffolding for interactive projects,
* an extensive collection of official plugins that integrate the top frontend tools.
* Vue.js projects may be created and managed using an entire graphical user interface.
* Rapid prototyping with zero configuration using @vue/CLI and @vue/CLI-service-global
* Built on top of webpack and expandable via plugins, @vue/cli-service is a runtime dependency.

14. What is instant prototyping, and how does it work?

**Ans:** With the vue serve(similar to vue create) and vue build commands in the Vue CLI, instant prototyping is known as fast prototyping with just a single \*.vue file.

***[Related Article:***[***JavaScript Interview Questions***](https://mindmajix.com/javascript-interview-questions)***]***

15. In VueJS, what is the purpose of nextTick?

**Ans:**After the data has been set and the DOM changed, the nextTick method is simply a convenient mechanism to perform a function.

Vue.js Interview Questions for Professionals

16. Why is it necessary for the component data to be a function?

**Ans:** Instead of just supplying the object, the component data must be a function. This is because each instance requires a copy of the returned data object. Otherwise, data changes in one component instance will affect data in all other instances.

17. In vue router, what are navigation guards?

**Ans:**Vue-navigation router's guards protect navigations by diverting or cancelling them.

The three possible techniques to plug into router navigations are listed below.

* Global
* Per-route
* In-component

18. What are mixins?

**Ans:** Mixins are a versatile approach to sharing reusable Vue component functionality. Any component choices can be included in a mixin object. When a component employs a mixin, all of the mixin's options are "mixed" into the component's own.

19. What are the advantages of utilising VueJS Mixins?

**Ans:**Mixins are used by developers in Vue.js to flexibly distribute reusable features. Mixins can have any number of component options. Mixins can be kept in one location so that all of the components that use them benefit at the same time.

20. List types of Directive available in Vue.js?

**Ans:**The following directives are supported in Vue js:

* General Directives
* Literal Directives
* Empty Directives
* Custom Directives

21. What is vue-loader?

**Ans:** Vue-loader is a webpack loader plugin that allows us to develop single-file components in the vue format. Template, script, and style are the three sections of a single component file. Webpack can extract and process each section using distinct loader modules like the SASS or SCSS loaders, thanks to the vue-loader module. As a result of the configuration, we can easily author apps using.vue files.

Static assets can also be regarded as module dependencies with the vue-loader module, and webpack loaders can be used to process them. It also enables hot reloading when developing.

22. What is Virtual DOM?

**Ans:**The Document Object Model (DOM) is an interface that allows languages like JavaScript to interact with and change HTML documents. Nodes in a tree represent elements, and the interface lets us manipulate them. This interface, however, comes with a cost, and the page will be slowed down by a considerable number of persistent DOM operations.

Vue overcomes this problem by storing a virtual representation of the document's structure in memory, where a virtual node (vnode) represents nodes in a document's structure. When manipulation is required, the computations and operations are executed in memory on the Virtual DOM rather than on the real DOM. This is faster, and it allows the Virtual DOM algorithm to determine the most efficient method of updating the actual DOM structure.

This is then applied to the actual DOM tree after it has been computed. This improves performance, which is why Virtual DOM-based frameworks like Vue and React have become so popular.

23. What is vue plugin?

**Ans:**Developers can create and add global level functionality to Vue via a Vue plugin. This can be used to add globally accessible methods and properties to an application and assets, options, mixins, and other custom APIs. VueFire, a Vue plugin that adds Firebase specific methods and binding to the entire application, is an example of a Vue plugin.

24. What is the best way to maintain state in an extensive vue application, and why is it recommended?

**Ans:**Managing state becomes tough when an application increases in size, both in terms of features, codebase and employing an eternal network of downstream props and upstream events is not sensible. It becomes vital to transfer state management to a central management system in such a situation. Vuex is the official state management library and a suggested paradigm for saving states centrally in the Vue ecosystem.

Vuex allows for the preservation of a central state. Vuex is used by components as a reactive data store, which updates as the state changes. The same and central state store can be used by multiple and unrelated components.

25. How can the performance of Vue.js be improved?

**Ans:** Your Vue application may become delayed for various reasons:

* You're relying on too many third-party libraries that are either too large or misused.
* You have a large number of unoptimised and compressed photos.
* You don't use lazy loading to load your images.
* To reduce redundant code, you aren't reusing functionalities across your programme.
* You have a lot of pages and want to load them all at once.

26. What are some of Vue's most crucial directives?

**Ans:**

* Based on the specified phrase, v-if inserts or removes DOM elements.
* Only when the expression adjacent to v-if resolves to false does v-else display content.
* V-show is similar to v-if, but instead of rendering all elements to the DOM, it uses the CSS display property to show and hide them.
* We can use v-for to iterate over items in an array or object.
* For two-way data bindings, the v-model is employed.
* The element is given an event listener by v-on.

27. Vue has which lifecycles and hooks?

**Ans:** When a Vue component instance is created, it goes through several setup procedures. It must set up data observation, compile the template, mount the instance to the DOM, and update the DOM when data changes, for example. It also runs lifecycle hooks along the route, which allow us to run custom code at specific points in the process.

* Before the component is added to the DOM, we can use creation hooks to conduct operations. During server-side rendering, these hooks are also activated. The created hook is ideal for triggering HTTP queries and populating any initial data required by the component.
* The most commonly used hooks are mounting hooks (before mount, mounted), which allow us to access the component before and after the first render. The mounted hook is a great place to add third-party libraries or access the DOM.
* When a component's reactive property changes or something else causes it to re-render, updating hooks (before the update, updated) are called. The DOM and model are in sync in the Updated hook, whereas the model is updated in the beforeUpdate hook, but not the DOM.
* Destruction hooks (beforeDestroy, destroyed) let us do things like cleanup or transmit analytics when a component is destroyed. We have access to the Vue instance in the beforeDestroy hook and can emit events.

28. Why should arrow functions not be utilised in Vue instances when writing lifecycle hooks?

**Ans:** Because arrow functions are not self-contained, they cannot define a 'this' of their own. The arrow functions are confined to the context of their parent function.

The keyword 'this' does not bind to the Vue instance when the arrow function (=>) is used in the Vue app, resulting in problems.

As a result, instead of using the arrow function, you strongly recommend utilising the regular function declaration.

29. What exactly are watchers?

**Ans:**Instead of utilising computed properties, watchers are a more generic way to react to data changes.

When asynchronous or expensive actions must be performed in response to changing data, they should be employed.

30. What makes Vue superior to Angular?

**Ans:** Because of its vast built-in functionalities and community support, [Angular](https://mindmajix.com/angularjs-tutorial) is an obvious choice for developing enterprise-based applications. When it comes to creating user interfaces and tackling complicated problems, Vue is a technically sound framework.