By OnlineInterviewQuestions.com

Vue js Interview Questions

About Vue.js

Vue.js is an open-source JavaScript framework for building user interfaces & single-page web applications. Vue.js is Inspired by the MVVM (Model-View-ViewModel) pattern. It's designed to help developers build better interactive interfaces while avoiding a steep learning curve, which means it's a great choice for beginners and experienced web designers alike.

Vue js Interview Questions and Answers

Finally, Practice here the top **Vue js Interview Questions** for the best preparation of the Vue js Interview. These Interview Questions are very popular and are asked many times in Vue js Interviews. So, practice these questions to check your final preparation for your interview. apart from this, you can also download here **Vue js Interview Questions PDF**, completely free.

Q1. What is Vue.js?

Vue js is progressive javascript script used to create dynamic user interfaces.Vue js is very easy to learn.In order to work with Vue js you just need to add few dynamic features to a website.You don't need to install any thing to use Vue js just need add Vue js library in your project.

Q2. List some features of Vue.js.

Vue js comes with following features

- Templates
- Reactivity
- Components
- Transitions
- Routing

Q3. Explain Life cycle of Vue Instance.

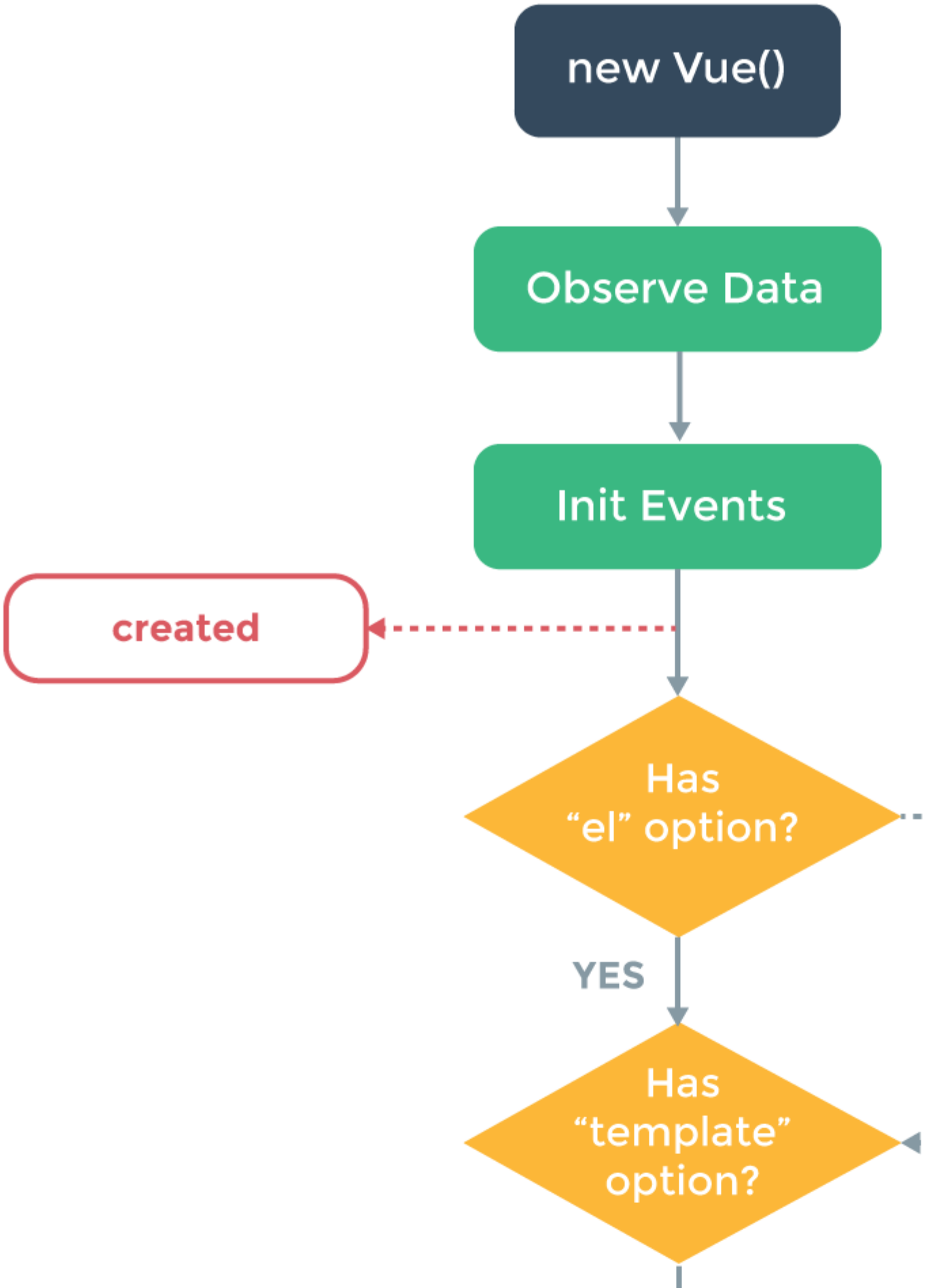
The Life cycle of each Vue instance goes through a series of initialization steps when it is created.
– for example, it needs to set up data observation, compile the template, and create the necessary data bindings. Along the way, it will also invoke some lifecycle hooks, which give us the opportunity to execute custom logic. For example, the created hook is called after the instance is created:

```
new Vue({
  data: {
    a: 1
  },
  created: function () {
    // `this` points to the vm instance
    console.log('a is: ' + this.a)
  }
})
// => "a is: 1"
```

There are also other hooks which will be called at different stages of the instance’s lifecycle, for example compiled, ready and destroyed. All lifecycle hooks are called with their this context pointing to the Vue instance invoking it. Some users may have been wondering where the concept of “controllers” lives in the Vue.js world, and the answer is: there are no controllers in Vue.js. Your custom logic for a component would be split among these lifecycle hooks.

Lifecycle Diagram

Below diagram shows complete life cycle of Vue Instance


new Vue()
Observe Data
Init Events
created
Has
“el” option?
YES
Has
“template”
option?

Source: https://v1.vuejs.org/guide/instance.html#Instance-Lifecycle

Also, Read **React js Interview questions**

Q4. How to create an instance of Vue js.

You can create Vue instance with the Vue function:

```
var vm = new Vue({
  // options
})
```

Q5. Explain the differences between one-way data flow and two-way data binding?

In one-way data flow the view(UI) part of application does not updates automatically when data Model is change we need to write some custom code to make it updated every time a data model is changed.
In Vue js **v-bind** is used for one-way data flow or binding.

In two-way data binding the view(UI) part of application automatically updates when data Model is changed.
In Vue.js **v-model** directive is used for two way data binding.

Q6. How to create Two-Way Bindings in Vue.js?

v-model directive is used to create Two-Way Bindings in Vue js.In Two-Way Bindings data or model is bind with DOM and Dom is binded back to model.
In below example you can see how Two-Way Bindings is implemented.

```
<div id="app">
  {{message}}
  <input v-model="message">
</div>
<script type="text/javascript">
  var message = 'Vue.js is rad';
  new Vue({ el: '#app', data: { message } });
</script>
```

Q7. What are filters in VUE.js?

In Vue js filters are used to transform the output that are going to rendered on browser.
A Vue.js filter is essentially a function that takes a value, processes it, and then returns the processed value. In the markup it is denoted by a single pipe (|) and can be followed by one or more arguments:

```
<element directive="expression | filterId [args...]"></element>
```

In Vue 2.0, there are no built-in filters are availables, however you are free to create your own filters.

Q8. How to create a custom filter in Vue.js?

Vue.filter() method is used to create and register a custom filter in Vue js. Vue.filter() method takes two parameters a filterId that is usnique name to filter that you going to create and a filter function that takes a value as the argument and returns the transformed value.

```
Vue.filter('reverse', function (value) {
  return value.split('').reverse().join('')
})
```

Q9. What are Components in Vue.js? How to register a component inside other component

Vue Components are one of most powerful features of Vue js.In Vue components are custom elements that help you extend basic HTML elements to encapsulate reusable code.
Following is the way to register a Vue component inside other component

```
export default {
  el: '#your-element'
  components: {
      'your-component'
  }
}
```

Q10. What are Directives in VUE.js, List some of them you used?

The concept of directive in Vue js is drastically simpler than that in Angular. Vue.js directives provides a way to extend HTML with new attributes and tags. Vue.js has a set of built-in directives which offers extended functionality to your applications.You can also write your custom directives in Vue.js .
Below are list of commonly used directives in Vue.js

- v-show
- v-if
- v-model
- v-else
- v-on

Q11. List type of Directive are available in Vuejs.

In Vue js following types of directives are available

- General Directives

- Literal Directives
- Empty Directives
- Custom Directives

Q12. What is VUE-resource, how can you install Vue Resource ?

VUE-resource is a plugin for vue.js that provides services for making web requests and handle responses using a XMLHttpRequest or JSONP
You can install it via yarn or NPM.

```
$ yarn add vue-resource
$ npm install vue-resource
```

Q13. How to create Constants in Vue js.

To create constant **const** keyword is used. In Vue.js we suggest creating a separate file for defining your constants.
Example:

Creating a Constant in Vue js.

```
export const SITE_URL = 'https://www.onlineinterviewquestions.com';
```

Importing a Constant in Vue js.

```
import {SITE_URL} from './path/to/constants.js';
```

Q14. What is virtual dom in Vuejs?

Virtual DOM in Vue is a JavaScript object that represents the Document Object Model (DOM). The application updates the Virtual DOM instead of the DOM directly. So, it minimizes the updating cost of the real DOM as it is computationally expensive. Virtual DOM offers the ability to control the timing at which the Virtual DOM is rendered. Virtual DOM will just maintain the state of the data without re-rendering until you choose it. Virtual DOM also offers the ability to optimize the performance of your web applications by minimizing the number of times the DOM has to be updated.

Q15. Why we need Vue.js mixins?

Mixins in Vue JS are a chunk of defined logic that is stored in a particular way. It can be re-used over and over to add functionality to your Vue instances and components. It is important that we need Vue JS because,

1. You can easily adhere to the DRY principle with mixins. It ensures that you do not repeat yourself.

2. You get a lot of flexibility with mixins. Mixin contains options for Vue components.
3. Mixins are safe and they do not affect changes outside their defined scope.
4. Mixins in Vue JS are a great platform for code reusability.

Q16. What is Vuex?

VueX is a state management pattern and library for the application using Vue JS. it acts as a centralized store for all the different components in your Vue JS application. It has rules to ensure that the state can be only mutated in a predictable fashion. It can be integrated with the official devtool extension of Vue to provide additional features. Vuex mainly helps in dealing with shared state management with the cost of more concepts and boilerplate.

Q17. What are filters in Vuejs?

Filters in Vue JS helps in applying common text formatting. It is used in two places, mustache interpolations, and v-bind expressions. It mainly filters the data on the DOM level. So you get data that is still intact in the storage but is represented in the custom specified manner. It enhances the presentation of the view layer. The filters are also reusable. You can declare a filter globally and use it on any desirable component. It gives you the power to format your data at the view level.

Q18. How to create a component in Vue js?

Components in Vue JS are a single, independent unit of an interface. They have their own state, markup, and style.

A Vue component can be defined in four ways.

- The first is new Vue({ /*options */ }).
- The second is Vue.component('component-name', { /* options */ }).
- The third way is by using the local components.
- The fourth is in the .vue files or Single File Components.

The first two ways are the standard ways to use Vue when building an application that is not a SPA (Single Page Application). The Single File Components are uses in the Single Page Application.

Q19. How to import js file in the Vue component?

There are **two ways to import a JavaScript library to the Vue Component**.

The first is to import a local JavaScript library. Here, you can import the JavaScript library by using the 'import' keyword inside the script tag of your Vue file.

```
import * as mykey from '../assets/js/mykey.js';
```

The second way is to include your external JavaScript file into the mounted hook of your Vue component.

Q20. How to call rest API from Vue js?

We can use various HTTP libraries to call REST Api's from Vue JS. One of the popular libraries is Axios. It simple to use and lightweight. To include it in your project, execute the following command.

```
npm install axios --save
```

Implementing GET method using Axios in Vue JS

```
axios({ method: "GET", "URL": "https://httpbin.org/ip" }).then(result => {
                this.ip = result.data.origin;
            }, error => {
                console.error(error);
            });
```

We can send an HTTP request using Axios with a promise. If the request is successful, we'll get the result.

Q21. How do you implement two-way data binding in a Vue.js component?

To implement **two-way data binding in a Vue.js** component, you can use the "v-model" directive. This directive binds a value to a form input element, and updates the value whenever the input element's value changes.

Here is an example of how you might use the "v-model" directive in a Vue.js component:

```
<template>
  <div>
   <input v-model="message" type="text">
   <p>{{ message }}</p>
  </div>
</template>
<script>
export default {
  data() {
    return {
      message: 'Hello world!'
    }
  }
}
</script>
```

In this example, the value of the "message" data property is bound to the value of the input element using the "v-model" directive. When the user types in the input element, the value of the "message" data property is updated, and this change is reflected in the text displayed by the p element.

Q22. How to implement computed properties in Vue.js?

Computed properties are properties in a Vue.js component that are calculated based on other properties in the component. They are like methods, but they are cached based on their dependencies, and will only re-evaluate when one of their dependencies changes. This can be more efficient than calling a method multiple times, especially if the method's result is expensive to compute.

Example:

```
<template>
 <div>
    <p>{{ reversedMessage }} </p>
  </div>
</template>
<script>
export default {
  data() {
    return {
      message: 'Hello world!'
    }
  },
  computed: {
    reversedMessage() {
      return this.message.split('').reverse().join('')
    }
  }
}
</script>
```