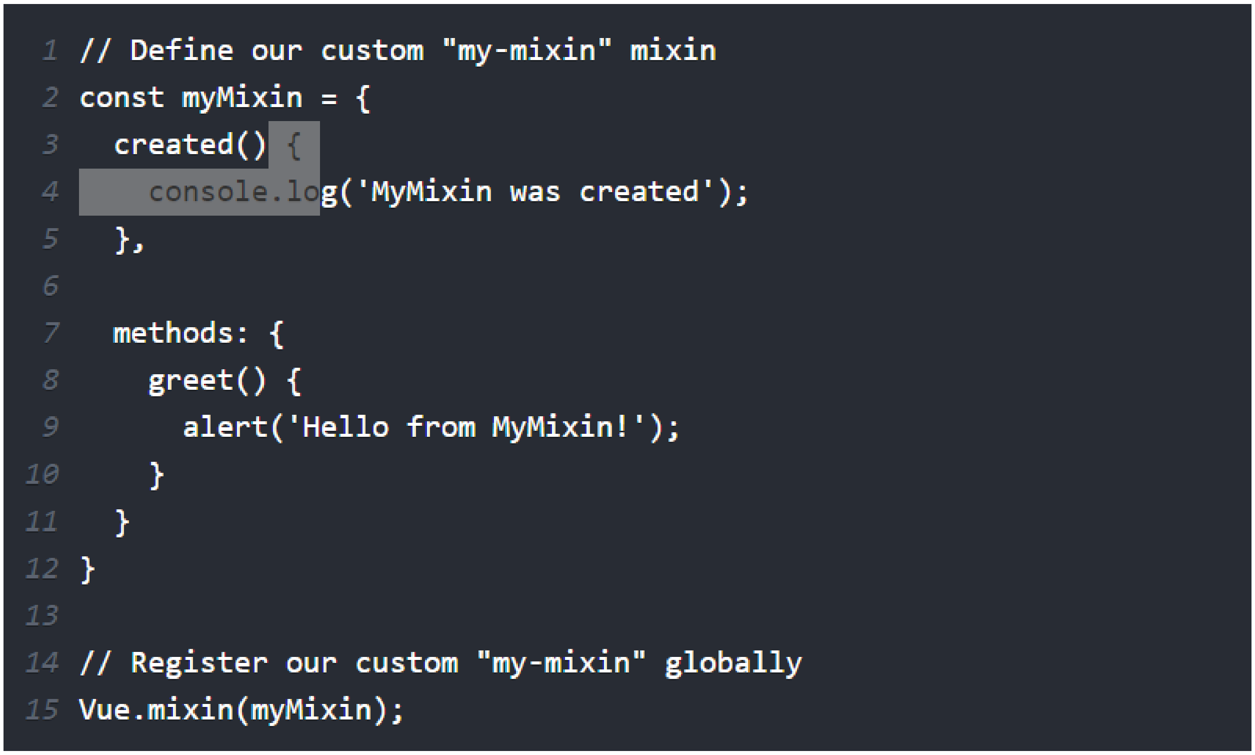
Mixins in Vue.js are a way to share reusable functionality between components. A mixin is essentially an object containing any component options that you want to reuse across multiple components. When a component uses a mixin, all of its properties and methods are merged with the component's own properties and methods.

To define a new mixin in Vue.js, you can use the Vue.mixin() method:



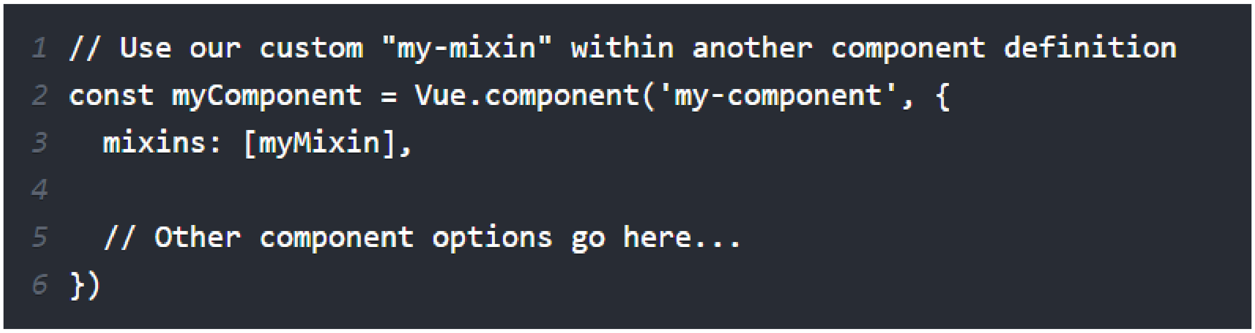
```
// Define our custom "my-mixin" mixin
const myMixin = {
  created() {
    console.log('MyMixin was created');
  },

  methods: {
    greet() {
      alert('Hello from MyMixin!');
    }
  }
}

// Register our custom "my-mixin" globally
Vue.mixin(myMixin);
```

In this example, we've defined a new myMixin object that includes a created hook and a greet method. We then register this mixin globally using the Vue.mixin() method so that it can be used by any component throughout our application.

To use the newly registered mixin within your components, simply include it as an array value for the mixins property when defining your component:



```
// Use our custom "my-mixin" within another component definition
const myComponent = Vue.component('my-component', {
  mixins: [myMixin],

  // Other component options go here...
})
```

In this example, we've included our previously registered myMixin as part of the mixins array when defining our new myComponent. This means that all of its properties and methods will be merged with those defined directly on the component itself.

Using mixins allows you to easily share code between different components without having to repeat yourself or write complex inheritance chains. However, it's important to use them thoughtfully since they can also lead to unexpected behavior if not used properly.