

What I'm most
excited for in

Vue 3



I'm @chrisvfritz



I'm @chrisvfritz



Introduction — Vue.js

https://vuejs.org/v2/guide/

Vue.js

Learn Ecosystem Team Support Vue Translations

Guide 2.x

Essentials

- Installation
- Introduction**
 - [What is Vue.js?](#)
 - [Getting Started](#)
 - [Declarative Rendering](#)
 - [Conditionals and Loops](#)
 - [Handling User Input](#)
 - [Composing with Components](#)
 - [Relation to Custom Elements](#)
 - [Ready for More?](#)
- [The Vue Instance](#)
- [Template Syntax](#)
- [Computed Properties and Watchers](#)
- [Class and Style Bindings](#)
- [Conditional Rendering](#)
- [List Rendering](#)
- [Event Handling](#)
- [Form Input Bindings](#)
- [Components](#)

Transitions & Animation

- [Enter/Leave & List Transitions](#)

Introduction

What is Vue.js?

Vue (pronounced /vju:/, like **view**) is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with [modern tooling](#) and [supporting libraries](#).

If you'd like to learn more about Vue before diving in, we [created a video](#) walking through the core principles and a sample project.

If you are an experienced frontend developer and want to know how Vue compares to other libraries/frameworks, check out the [Comparison with Other Frameworks](#).

Getting Started

! The official guide assumes intermediate level knowledge of HTML, CSS, and JavaScript. If you are totally new to frontend development, it might not be the best idea to jump right into a framework as your first step - grasp the basics then come back! Prior experience with other frameworks helps, but is not required.

A photograph of a diverse group of people sitting in rows, possibly at a conference or event. They are looking towards the right side of the frame, where a man in a plaid shirt is gesturing with his hands while speaking. The lighting is warm and focused on the speaker.

Didn't Evan
already talk about
Vue 3 yesterday?

Vue 3

CHARDER

Deeler

FASTER

STRONGER

Vue 3

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input
      v-bind="$attrs"
      @input="$emit('update', $event.target.value)"
      v-on="$listeners"
    />
  </label>
</template>
```

Trends in **Vue 3**

Simpler

More Explicit

Trends in **Vue 3**

Simpler

Less code

Fewer concepts

Fewer foot guns

Trends in **Vue 3**



Clear relationships

Explicit

Expressive code





WARNING:

This list is
neither complete
nor final.

All demoed
features are
subject to change
or removal before
the final release
of Vue 3.

What you've
heard already

Goodbye reactivity caveats!

Goodbye reactivity caveats!

```
// Setting an array item by index  
Vue.set(this.myArray, index, newValue)
```

```
// Adding a new property to an object  
Vue.set(this.myObject, key, value)
```

```
// Deleting a property from an object  
Vue.delete(this.myObject, key)
```

Goodbye reactivity caveats!

in **Vue 3**

```
// Setting an array item by index  
this.myArray[index] = newValue
```

```
// Adding a new property to an object  
this.myObject[key] = value
```

```
// Deleting a property from an object  
delete this.myObject[key]
```

Multiple root nodes

Multiple root nodes

```
<template>
  <div></div>
  <div></div>
</template>
```

(Emitted value instead of an instance of Error)

Error compiling template:

```
<div></div>
<div></div>
```

- Component template should contain exactly one root element. If you are using v-if on multiple elements, use v-else-if to chain them instead.

Multiple root nodes in **Vue 3**

```
<template>
  <div></div>
  <div></div>
</template>
```



Breaking
news!

Simpler transparent wrappers

Simpler transparent wrappers

```
<BaseInput  
  v-model="searchText"  
  placeholder="Search"  
  @keyup.enter="search"  
/>
```

Simpler transparent wrappers

```
<BaseInput  
  v-model="searchText"  
  placeholder="Search"  
  @keyup.enter="search"  
/>
```

====

```
<input  
  v-model="searchText"  
  placeholder="Search"  
  @keyup.enter="search"  
/>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input
      v-bind="$attrs"
      @input="$emit('update', $event.target.value)"
      v-on="$listeners"
    />
  </label>
</template>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input
      v-bind="$attrs"
      @input="$emit('update', $event.target.value)"
      v-on="$listeners"
    />
  </label>
</template>
```

```
<BaseInput
  v-model="searchText"
  placeholder="Search"
  @keyup.enter="search"
/>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input
      v-bind="$attrs"
      @input="$emit('update', $event.target.value)"
      v-on="$listeners"
    />
  </label>
</template>
```

```
<BaseInput
  v-model="searchText"
  placeholder="Search"
  @keyup.enter="search"
/>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input
      v-bind="$attrs"
      @input="$emit('update', $event.target.value)"
      v-on="$listeners"
    />
  </label>
</template>
```

```
<BaseInput
  v-model="searchText"
  placeholder="Search"
  @keyup.enter="search"
/>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>

??

<template>
<label>
  {{ label }}
  <input
    v-bind="$attrs"
    @input="$emit('update', $event.target.value)"
    v-on="$listeners"
  />
</label>
</template>
```

Simpler transparent wrappers

```
<script>
export default {
  inheritAttrs: false,
  model: {
    event: 'update'
  },
  props: ['label']
}
</script>
```

...drumroll...

```
<template>
<label>
  {{ label }}
  <input
    v-bind="$attrs"
    @input="$emit('update', $event.target.value)"
    v-on="$listeners"
  />
</label>
</template>
```

Simpler transparent wrappers in **Vue 3**

```
<script>
export default {
  props: ['label']
}
</script>

<template>
  <label>
    {{ label }}
    <input v-bind="$attrs" />
  </label>
</template>
```

```
<BaseInput
  v-model="searchText" ✓
  placeholder="Search" ✓
  @keyup.enter="search" ✓
/>
```

No more automatic attribute inheritance

Just `v-bind="$attrs"`

(`$attrs` includes all non-prop attributes)

No more `inheritAttrs: false`

v-on will
compile to attributes

(e.g. @keyup compiles to on-keyup)

Just v-bind="\$attrs"
(\$attrs includes all non-emitted listeners)

No more \$listeners

No more .native modifier for v-on

v-model will
compile to attributes

(model-value and on-model-update)

Just v-bind="\$attrs"

No more overriding the native input event

No more model option

Smarter v-model on components

v-model magic

v-model on an element Just Works™

```
import { isTextInputType } from 'web/util/element'
import { looseEqual, looseIndexOf } from 'shared/util'
import { mergeVNodeHook } from 'core/vdom/helpers/index'
import { warn, isIE9, isIE, isEdge } from 'core/util/index'

/* istanbul ignore if */
if (isIE9) {
  // http://www.matts411.com/post/internet-explorer-9-oninput/
  document.addEventListener('selectionchange', () => {
    const el = document.activeElement
    if (el && el.vmodel) {
      trigger(el, 'input')
    }
  })
}

const directive = {
  inserted (el, binding, vnode, oldVnode) {
    if (vnode.tag === 'select') {
      // #6903
    }
  }
}
```

★ v-model magic ★

v-model on an element Just Works™

```
import { isTextInputType } from 'web/util/element'
import { looseEqual, looseIndexOf } from 'shared/util'
import { trigger } from 'shared/pers/index'
import { isIE9, isIE, isEdge } from 'core/util/index'

/* istanbul ignore if */
if (isIE9) {
  // http://www.matts411.com/post/internet-explorer-9-oninput/
  document.addEventListener('selectionchange', () => {
    const el = document.activeElement
    if (el && el.vmodel) {
      trigger(el, 'input')
    }
  })
}

const directive = {
  inserted (el, binding, vnode, oldVnode) {
    if (vnode.tag === 'select') {
      // #6903
    }
  }
}
```

Current v-model on an element

```
<select v-model="selectedChoice">
  <option disabled />
  <option v-for="choice in choices">
    {{ choice }}
  </option>
</select>
```

Current v-model

```
<select v-model="selectedChoice">  
  <option disabled />  
  <option v-for="choice in choices">  
    {{ choice }}  
  </option>  
</select>
```



...to a component

```
<BaseSelect  
  v-model="selectedChoice"  
  :choices="choices"  
/>
```



Current v-model on a component

```
<template>
  <select @change="$emit('input', $event.target.value)">
    <option
      disabled
      :selected="!choices.includes(value)"
    />
    <option
      v-for="choice in choices"
      :selected="choice === value"
    >
      {{ choice }}
    </option>
  </select>
</template>
```

Current v-model on a component

```
<template>
  <select @change="$emit('input', $event.target.value)">
    <option
      disabled
      :selected="!choices.includes(value)"
    />
    <option
      v-for="choice in choices"
      :selected="choice === value"
    >
      {{ choice }}
    </option>
  </select>
</template>
```

Current v-model on a component

```
<template>
  <select @change="$emit('input', $event.target.value)">
    <option
      disabled
      :selected="!choices.includes(value)"
    />
    <option
      v-for="choice in choices"
      :selected="choice === value"
    >
      {{ choice }}
    </option>
  </select>
</template>
```

Current v-model on a component

```
<template>
  <select @change="$emit('input', $event.target.value)">
    <option
      disabled
      :selected="!choices.includes(value)"
    />
    <option
      v-for="choice in choices"
      :selected="choice === value"
    >
      {{ choice }}
    </option>
  </select>
</template>
```

Smarter v-model in **Vue 3**

```
<select v-model="selectedChoice">  
  <option disabled />  
  <option v-for="choice in choices">  
    {{ choice }}  
  </option>  
</select>
```



...to a component

```
<BaseSelect  
  v-model="selectedChoice"  
  :choices="choices"  
/>
```

...drumroll...

Smarter v-model in **Vue 3**

```
<template>
  <select v-bind="$attrs">
    <option disabled />
    <option v-for="choice in choices">
      {{ choice }}
    </option>
  </select>
</template>
```

Smarter v-model on a component in **Vue 3**

```
<template>
  <select v-bind="$attrs">
    <option disabled />
    <option v-for="choice in choices">
      {{ choice }}
    </option>
  </select>
</template>
```

Simpler render functions

Simpler render functions

JSX

```
<template>
  <BaseInput
    v-model="searchText"
    :class="$style.searchInput"
    placeholder="Search"
  />
</template>
```



```
render(h) {
  return (
    <BaseInput
      vModel={this.searchText}
      class={this.$style.searchInput}
      placeholder="Search"
    />
  )
}
```



Simpler render functions

JSX

```
render(h) {  
  return (  
    <BaseInput  
      vModel={this.searchText}  
      class={this.$style.searchInput}  
      placeholder="Search"  
    />  
  )  
}
```

without JSX

```
render(h) {  
  return h(BaseInput, {  
    props: {  
      value: this.searchText  
    },  
    on: {  
      update: newValue => {  
        this.searchText = newValue  
      }  
    },  
    class: this.$style.searchInput,  
    attrs: {  
      placeholder: 'Search'  
    }  
  })  
}
```



Simpler render functions in **Vue 3**

JSX

```
render(h) {  
  return (  
    <BaseInput  
      vModel={this.searchText}  
      class={this.$style.searchInput}  
      placeholder="Search"  
    />  
  )  
}
```

without JSX

```
render(h) {  
  return h(BaseInput, {  
    modelValue: this.searchText,  
    onModelUpdate: newValue => {  
      this.searchText = newValue  
    },  
    class: this.$style.searchInput,  
    placeholder: 'Search'  
  })  
}
```

Simpler render functions in **Vue 3**

JSX

```
render(h) {  
  return (  
    <BaseInput  
      vModel={this.searchText}  
      class={this.$style.searchInput}  
      placeholder="Search"  
    />  
  )  
}
```

without JSX

```
render(h) {  
  return h(BaseInput, {  
    modelValue: this.searchText,  
    onModelUpdate: newValue => {  
      this.searchText = newValue  
    },  
    class: this.$style.searchInput,  
    placeholder: 'Search'  
  })  
}
```



Trends in **Vue 3**

Simpler

More Explicit

Moving to
Vue 3

If I'm not already using Vue,
should I wait for **Vue 3**?



If I'm not already using Vue,
should I wait for **Vue 3**?

No

Will **Vue 3** support Internet Explorer?

Yes!

...but with the same
reactivity caveats as Vue 2

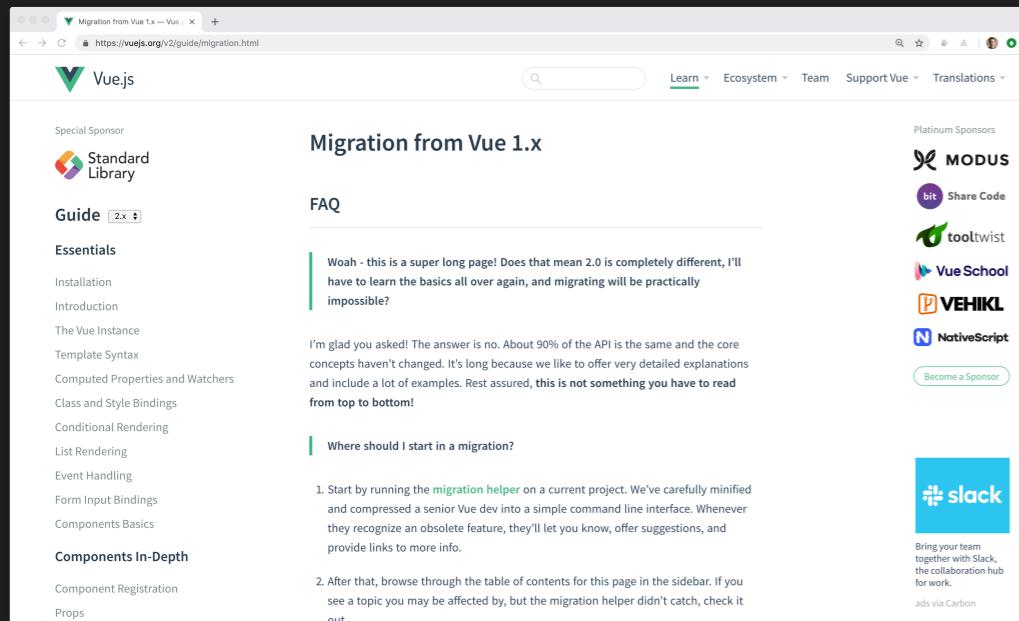
Will **Vue 3** break everything?



Will **Vue 3** break everything?

No

Will Vue 3 break everything?



Thorough migration guide

Will Vue 3 break everything?

The screenshot shows the 'Migration from Vue 1.x' page on the Vue.js website. The page has a sidebar with navigation links like 'Installation', 'Introduction', 'The Vue Instance', 'Template Syntax', etc. The main content area contains a large section titled 'FAQ' with two questions: 'Woah - this is a super long page! Does that mean 2.0 is completely different, I'll have to learn the basics all over again, and migrating will be practically impossible?' and 'Where should I start in a migration?'. Below these are two numbered steps: 1. Start by running the migration helper on a current project. 2. After that, browse through the table of contents for this page in the sidebar. The page also features a 'Platinum Sponsors' section with logos for MODUS, Share Code, tooltwist, Vue School, VEHIKL, and NativeScript, along with a 'Become a Sponsor' button.

Thorough migration guide



Next-gen migration helper

Will Vue 3 break everything?

The screenshot shows the 'Migration from Vue 1.x' page on the Vue.js website. The page has a sidebar with 'Essentials' and 'Components In-Depth' sections. The main content includes a 'FAQ' section with a note about the migration being a 'super long page'. It also contains a 'slack' integration card.

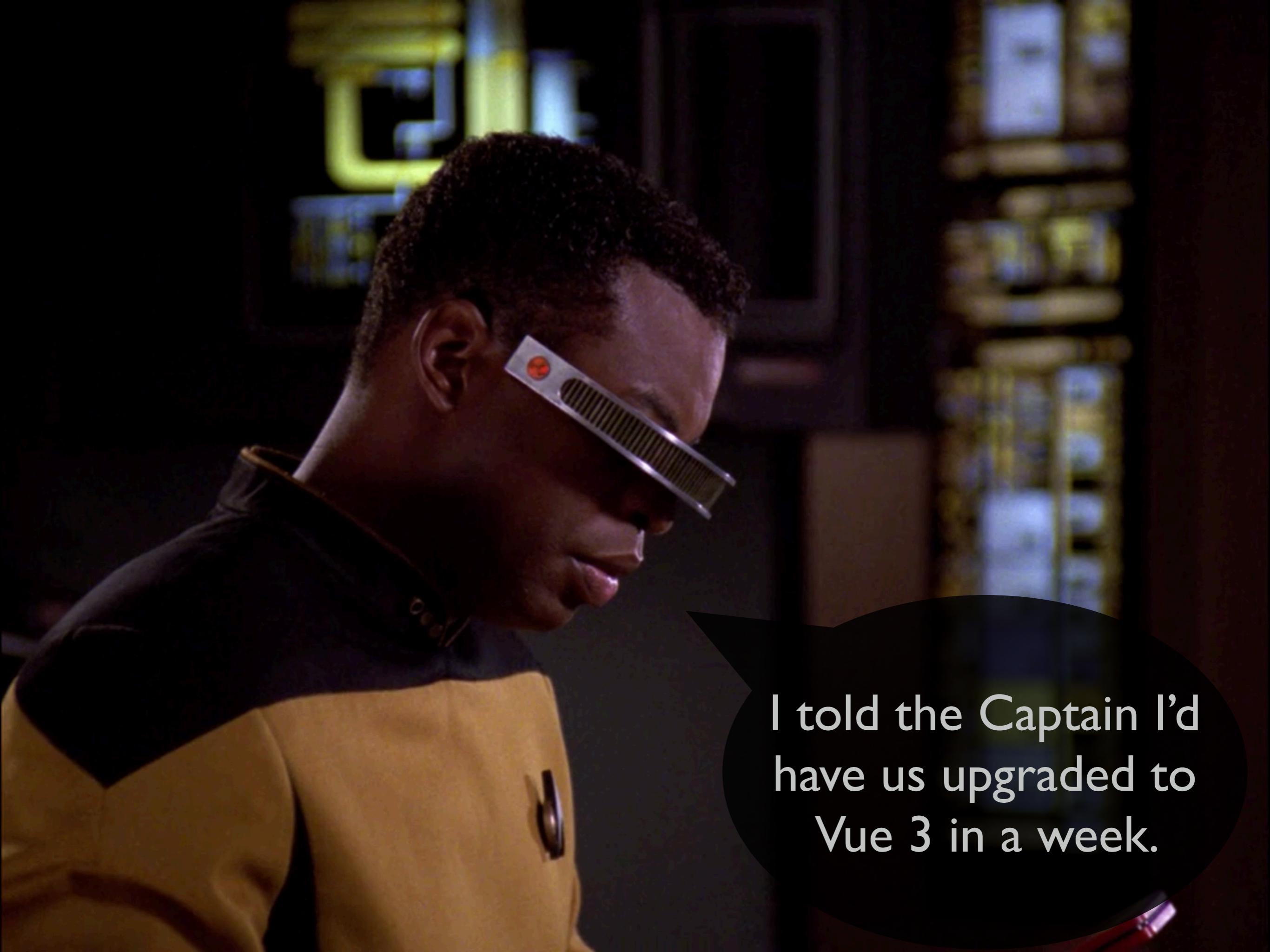
Thorough migration guide

The screenshot shows a terminal window titled 'vuejs/vue/examples dev ✘' running the command 'vue-migration-helper'. It lists four major breaking changes:

1. Replace `transition` attribute with either a `<transition>` or `<transition-group>` wrapper component
2. Replace `.v-leave` with `.v-leave-active` (if it's left over from Vue 1.x)
3. Delete `debounce="300"` and wrap the expensive operation(s) you want to throttle with lodash's `debounce` or `throttle` functions (see the link below for an example)
4. Replace `filtered value in v-html="input | marked"` with a computed property

Each item includes a line number, file path, reason, and a link to the migration guide for more info.

Next-gen migration helper

A close-up shot of a Star Trek character's head and shoulders. The character has dark skin and curly hair. A silver tricorder device is attached to their left ear, with a small red light glowing on it. They are wearing a dark, collared shirt. In the background, there are glowing blue and yellow lights, suggesting a futuristic or alien environment.

I told the Captain I'd
have us upgraded to
Vue 3 in a week.



How long will
it really take?

What if I'm stuck with **Vue 2**?

We're backporting as
many features as possible

How can I take part?

github.com/vuejs/rfcs

That's all!

Support me?



Chris Fritz is creating tools and resources for Vue developers

Overview Posts Become a patron

40 patrons

\$848 per month

+ Follow Share

If you're here, thank you! Whether or not you decide to donate, you're taking the initiative, researching how to support the work you love. Open source needs people like you. 🙏

Now, you're probably familiar with some of my work on the [Vue core docs](#), [style guide](#), [ESLint plugin](#), [migration guide](#) and [helper](#), my [enterprise boilerplate](#), among many other projects and even behind-the-scenes work you never see, such as helping to organize global Vue events and collaborating with other communities to improve

TIERS

Making It Happen
\$5 or more per month

This is your way of saying, "You just keep doing you, Chris." When you see me at events,

[patreon.com/chrisvuefritz](https://www.patreon.com/chrisvuefritz)

Thank you Vuesionaries!

John Leider from **Vuetify**

Rolf Haug from **Vue School**

github.com/chrisvfritz/vue-3-trends

@chrisvfritz

Thanks!

