

PEERS CONFERENCE 2019

VUE.JS 101

WIFI

Network: NOBICTenants

Password: Wireless@Tenants

GETTING SETUP

Repo: <https://github.com/bencodezen/vuejs-101-workshop>

Survey: <https://bencodezen.typeform.com/to/T1KAsV>

A little bit about me...





BEN HONG

Senior Frontend Engineer
Meltano in GitLab

[@bencodezen](#)

A little bit about you...

Survey Results

Get to know your neighbors

1. Name
2. Skills
3. Fun Fact

Before we get started...

FORMAT

1. Learn
2. Question
3. Apply

FORMAT

1. Learn

2. Question

3. Apply

Explanations
Examples

Clarifications
What-ifs

Exercises
Experiment

PARTICIPATION TIPS

Raise your hand for **questions** at **any** time!

There are no wrong questions here.

PARTICIPATION TIPS

All slides and examples are **public**.

No need to hurry and copy notes before we switch slides.

PARTICIPATION TIPS

Please **no recording.**

Out of respect for you and your participants' privacy

PARTICIPATION TIPS

These are **guidelines**. Not rules.

Feel free to question and/or disagree.

Your opinion and experience matter too.

Choose what works best for you and your team.

QUESTIONS?

LET'S GET STARTED!

VUE.JS INTRO

WHAT ARE CLIENT-SIDE FRAMEWORKS?

WHAT ARE CLIENT-SIDE FRAMEWORKS?

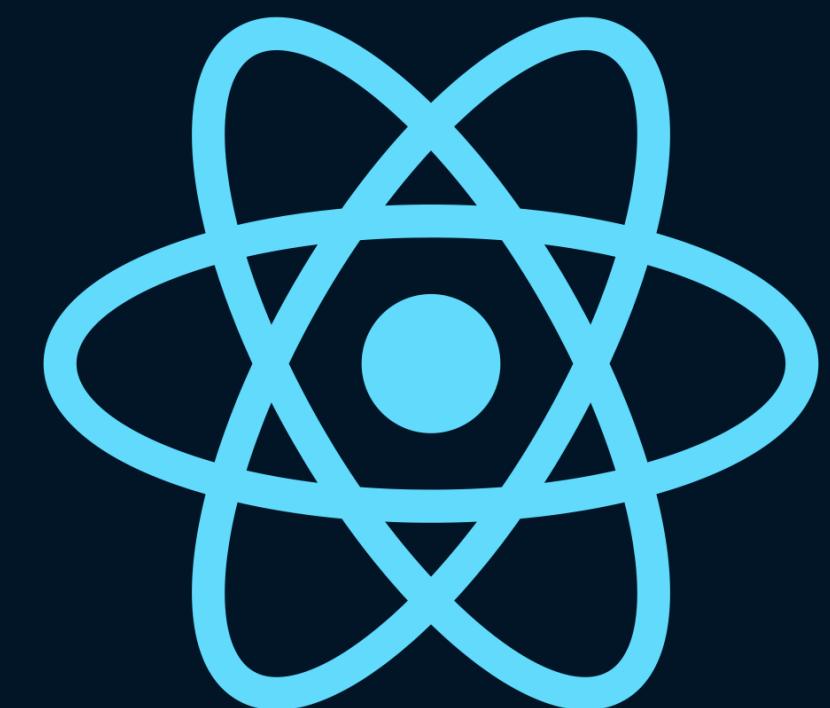
Allows frontend developers to take granular control of your site or application's user experience.

NUMBER OF DAYS SINCE

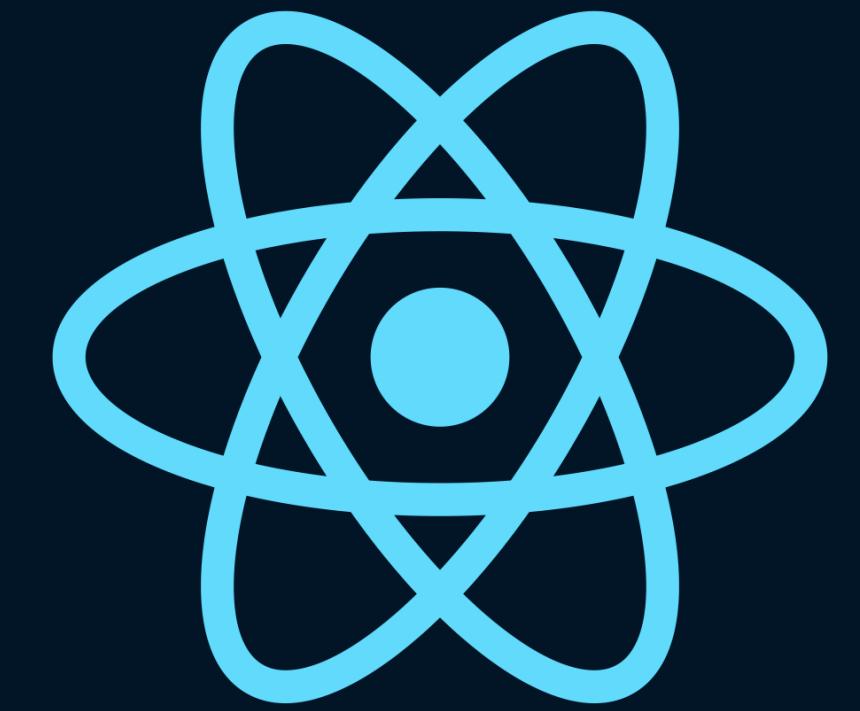
THIS DEPARTMENT
HAS WORKED
DAYS
WITHOUT
A LOST TIME
ACCIDENT
AVOID ACCIDENTS

LAST NEW JS FRAMEWORK

MOST POPULAR FRAMEWORKS



MOST POPULAR FRAMEWORKS

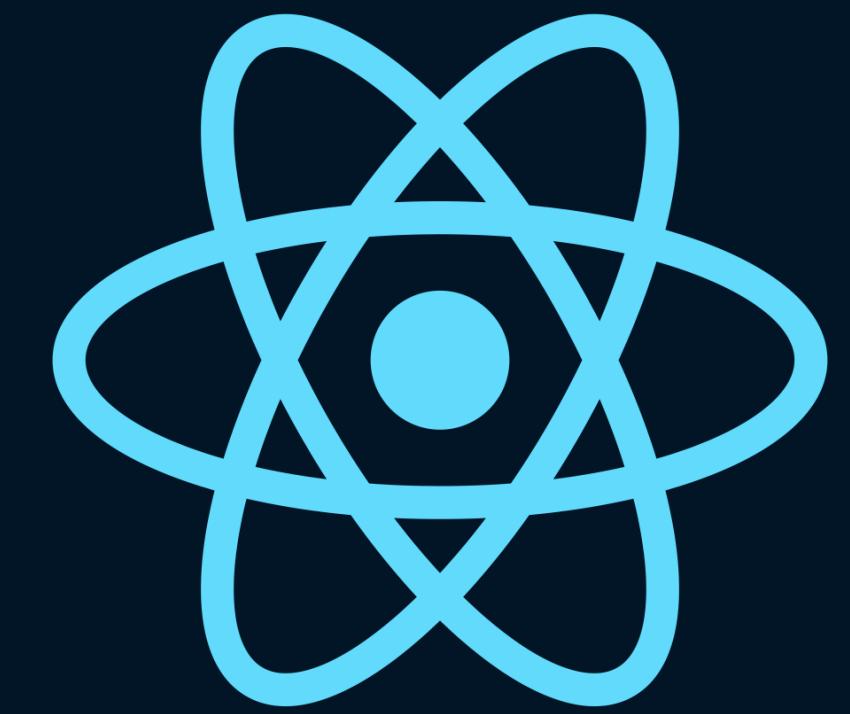


ANGULAR OVERVIEW

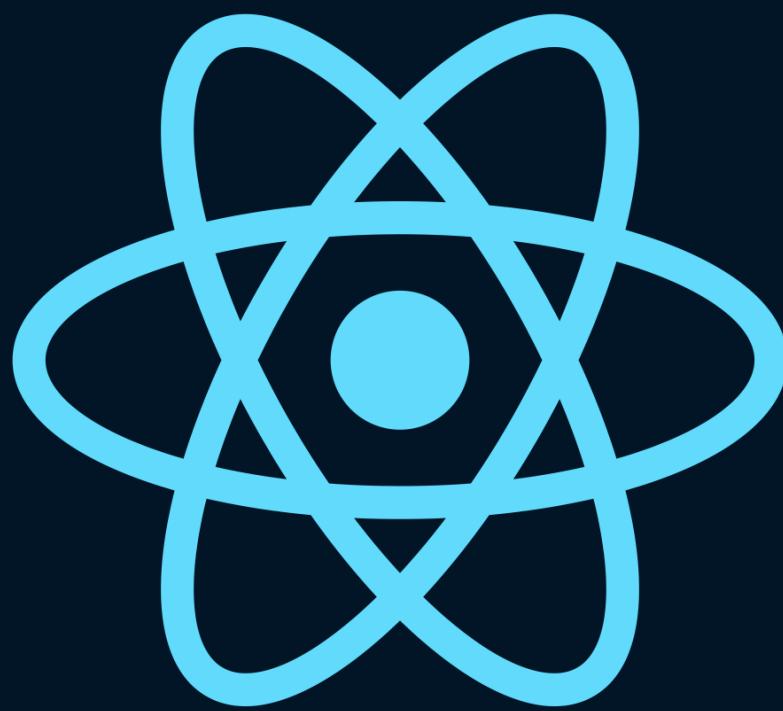


- + Developer and maintained by Google
- + Easy to get something up and running pretty quickly
- Most of what was happening seemed like magic
- Opinionated on how you should built your app

MOST POPULAR

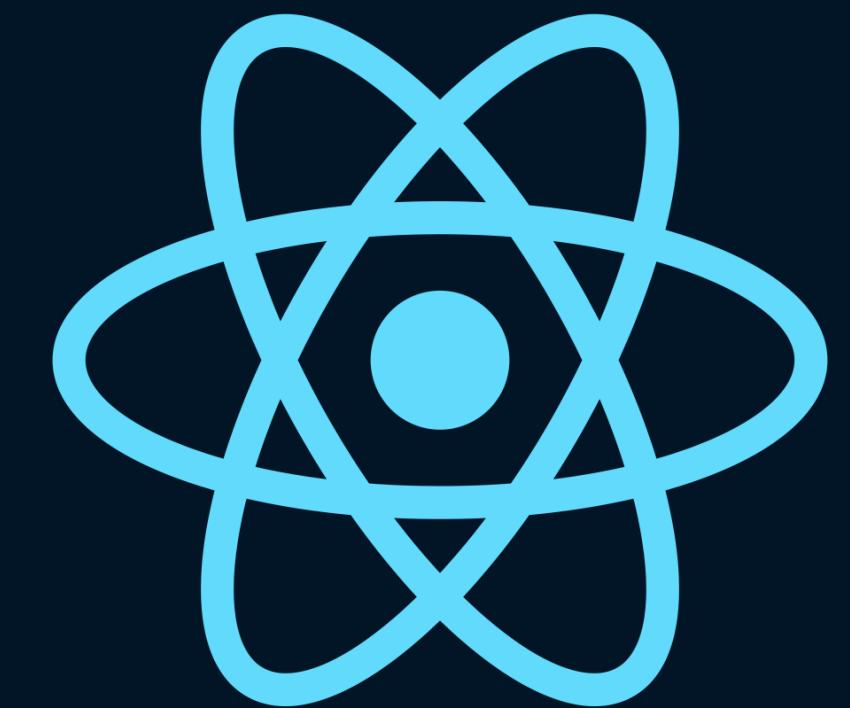


REACT OVERVIEW



- + Developed by Facebook and has an MIT license
- + Great performance due to the use of the virtual DOM
- + You got a lot better at vanilla JavaScript really quickly
- Just getting started had a high learning curve
- Unfriendly to non-JavaScript developers
- High pace of development resulting in unexpected bugs

MOST POPULAR



COMPARING FRAMEWORKS

	Angular	React
Founded By	Google	Facebook
How Opinionated?	Opinionated	Flexible
Learning Curve	Easy to start Hard to master	Hard to start Intermediate to master
Template	ngHTML	JSX

```
import React from 'react';
import { render } from 'react-dom';
import Hello from './Hello';

const styles = {
  fontFamily: 'Avenir, Helvetica, Arial, sans-serif',
  textAlign: 'center',
  color: '#2c3e50',
  marginTop: '60px'
};

const App = () => (
  <div style={styles} className="app" id="app">
    <h1>This is a React.js component file!</h1>
  </div>
);

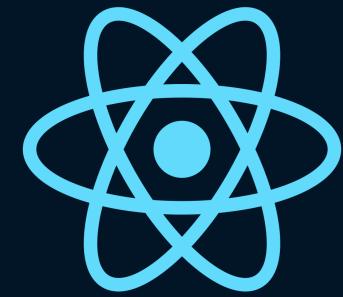
export default App
```

JSX EXAMPLE

COMPARING FRAMEWORKS



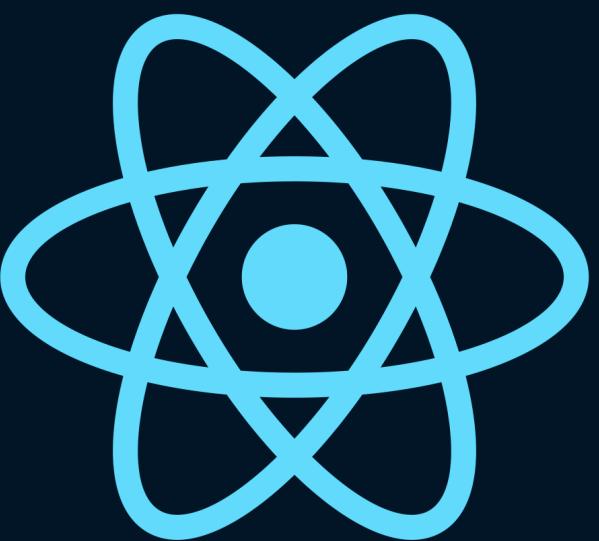
Angular



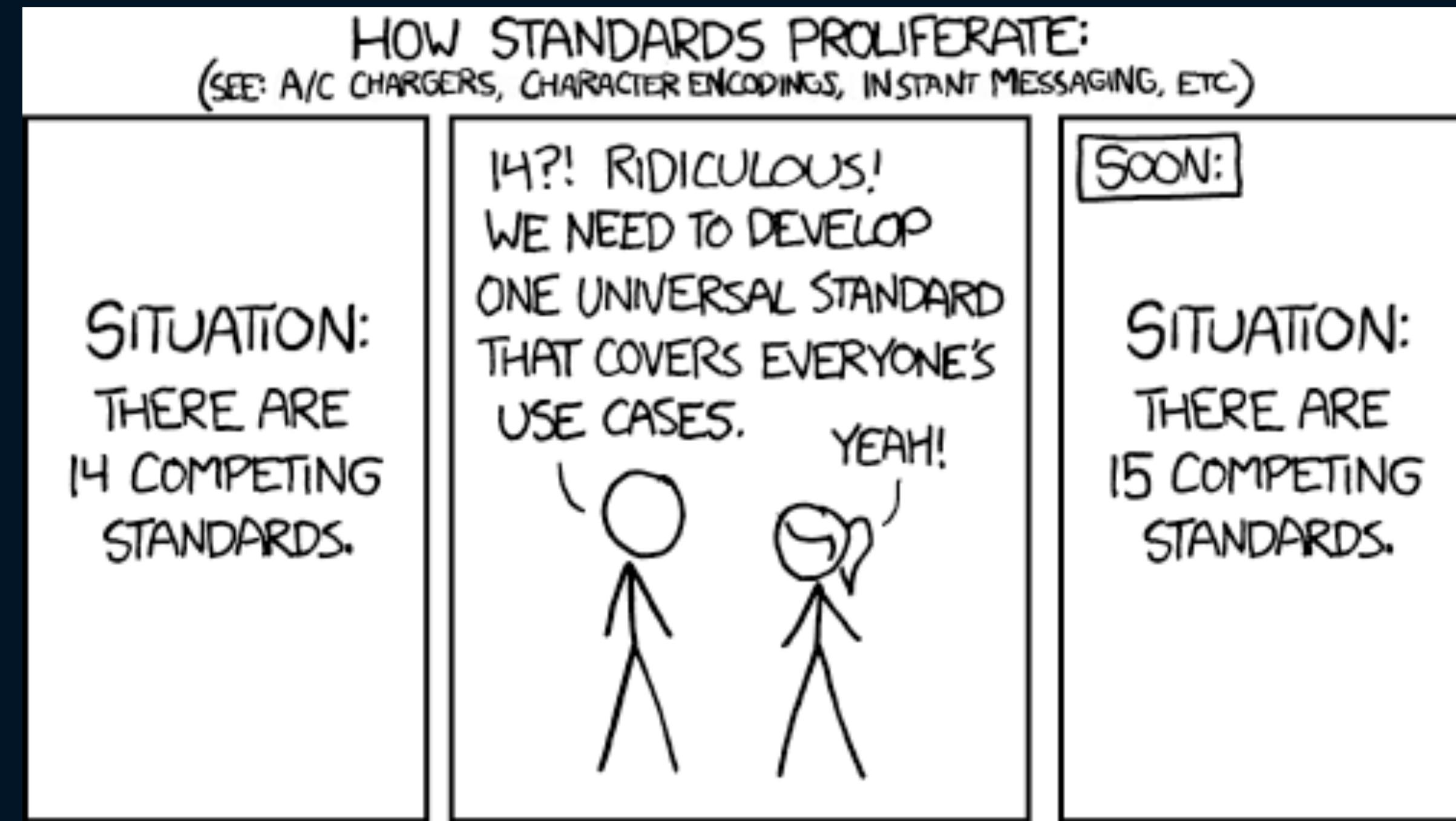
React

	Angular	React
Founded By	Google	Facebook
How Opinionated?	Opinionated	Flexible
Learning Curve	Easy to start Hard to master	Hard to start Intermediate to master
Template	ngHTML	JSX

COMPARING FRAMEWORKS

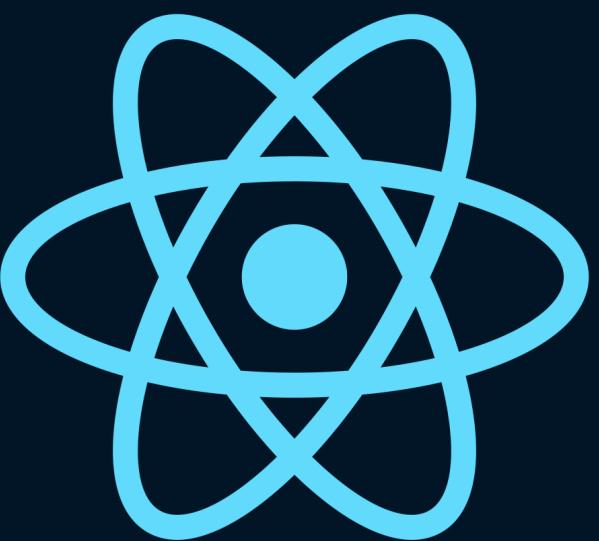


*As of April 11th, 2018



Credit - xkcd: Standards (<https://xkcd.com/927/>)

COMPARING FRAMEWORKS



*As of April 11th, 2018

COMPARING FRAMEWORKS



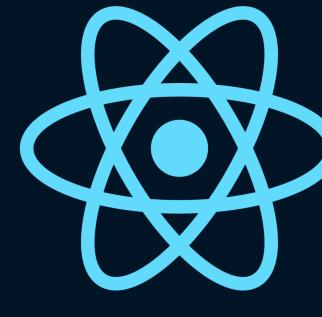
*As of April 11th, 2018

VUE OVERVIEW



- + Open-source framework with no corporate influence
- + Takes the best of both worlds and bring them together
- + It does not alienate non-JavaScript developers
- + Great performance that is on par if not better than React.js
- + Flexible and accommodating to how you prefer to build apps
- No corporate backing and has a "smaller" community compared to React / Angular

COMPARING THE TOP 3

	 Angular	 React	 Vue
Founded By	Google	Facebook	Evan You
How Opinionated?	Opinionated	Flexible	Guidance as needed
Learning Curve	Easy to start Hard to master	Hard to start Moderate to master	Easy to start Moderate to master
Template	ngHTML	JSX	vHTML (optional)
GitHub Stars	93.4k (62% v1)*	93.6k*	91k*

*As of April 18th, 2018

QUESTIONS?

**WHICH FRAMEWORK
SHOULD YOU
CHOOSE?**

IT DEPENDS



CHOOSE VUE

NO WAY!

TELL ME MORE!

AN OVERVIEW OF

VUE.JS BASICS

AN INSTANCE OF VUE.JS

```
const app = new Vue({  
  el: '#app',  
  data: {  
    firstName: 'Ben',  
    lastName: 'Hong',  
    conference: 'Peers Conference'  
  }  
})
```

AN INSTANCE OF VUE.JS

```
const app = new Vue({  
  el: '#app', ←  
  data: {  
    firstName: 'Ben',  
    lastName: 'Hong',  
    conference: 'Peers Conference'  
  }  
})
```

Uses a CSS selector to determine where this Vue instance will live.

AN INSTANCE OF VUE.JS

```
const app = new Vue({  
  el: '#app',  
  data: {  
    firstName: 'Ben',  
    lastName: 'Hong',  
    conference: 'Peers Conference'  
  }  
})
```

A data store for the Vue
to access in the app.

EXPRESSIONS IN VUE.JS

```
const app = new Vue({  
  el: '#app',  
  data: {  
    firstName: 'Ben',  
    lastName: 'Hong',  
    conference: 'Peers Conference'  
  }  
})
```

```
<!-- Using data in your template is easy! -->  
<div id="app">  
  <p>{{ firstName }} {{ lastName }} is at {{ conference }}!</p>  
</div>
```

EXPRESSIONS IN VUE.JS

```
const app = new Vue({  
  el: '#app',  
  data: {  
    firstName: 'Ben',  
    lastName: 'Hong',  
    conference: 'Peers Conference'  
  }  
})
```

```
<!-- This is how it renders! -->  
<div id="app">  
  <p>Ben Hong is at Peers Conference!</p>  
</div>
```

EXPRESSIONS IN VUE.JS

<!-- Using data in your template is easy! -->

```
<p>{{ firstName }} {{ lastName }} is at {{ conference }}!</p>
```

<!-- You can even do math! -->

```
<p>Let's perform calculations! {{ 11 * 11 / 4 }}</p>
```

<!-- Any simple expression just works -->

```
<p>{{ firstName.toUpperCase() }} IS EXCITED!</p>
```

<!-- Basic logic works too! -->

```
<p>{{ onSale ? price * 0.8 : price }}</p>
```

QUESTIONS?

EXERCISE #1: CAPTAIN MARVEL

INSTRUCTIONS

1. Create a new Vue instance that will live in the <main> element.
2. Create a data store that will have:
 - ▶ Captain Marvel's first name
 - ▶ Captain Marvel's last name
 - ▶ Captain Marvel's hero name
3. Refactor static data on site using data property

**LET'S TALK ABOUT
DIRECTIVES**



Directives are the part of
Vue.js that are a bit **magical**...

WHAT ARE DIRECTIVES?

They are **Vue specific syntax** that allow you to accomplish common goals without worrying about how it is implemented

- ▶ **v-if**
- ▶ **v-for**
- ▶ **v-else-if**
- ▶ **v-bind**
- ▶ **v-else**
- ▶ **v-on**
- ▶ **v-show**
- ▶ **v-model**

V-IF / V-ELSE-IF / V-ELSE

Allows you to determine whether an element on the page should **exist**

```
<article>
  <h1>Dashboard</h1>
  <section v-if="userPermission === 'admin'">
    ...
  </section>
  <section v-else-if="userPermission === 'user'">
    ...
  </section>
  <section v-else>
    ...
  </section>
</article>
```

V-SHOW

Allows you to determine whether an element on the page should be **visible**

VS

V-IF / V-ELSE-IF / V-ELSE

Allows you to determine whether an element on the page should **exist**

CODE DEMO

QUESTIONS?

DIRECTIVES: V-ON

V-ON

Allows us to attach methods to HTML elements
when a certain event occurs

```
<!-- The long form -->
<div v-on:click="alert('Alohomora!')">...</div>
```

```
<!-- The common shortcut -->
<div @click="alert('Alohomora!')">...</div>
```

QUESTIONS?

V-MODEL

Allows us to use two-way binding

```
<div id="app">
  <h1>{{ msg }}</h1>
  <label for="msg">Message</label>
  <input id="msg" type="text" v-model="msg" />
</div>
```

```
<script>
new Vue({
  el: '#app',
  data: {
    msg: ''
  }
})
</script>
```

CODE DEMO

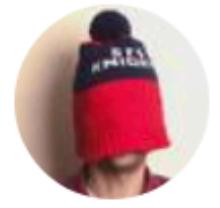
QUESTIONS?

EXERCISE #2: COUNTER

INSTRUCTIONS

1. Make the count a dynamic data property
2. Hide the text based on whether the count is even
3. Write a method to increment the count dynamically
4. Write a method to decrement the count dynamically
5. Use two way binding to change the amount that the counter can change by

QUICK BREAK



Sam Saccone

@samccone

Following



99.7% of software development in one requirement

A user should be able to view a list of items.

10:36 AM - 1 Dec 2017

610 Retweets 1,834 Likes



26



610



1.8K



DIRECTIVES: V-FOR

V-FOR

Allows us to “render a list of items based on an array (or object).”

```
<!-- Given the data model on the right -->
<ul>
  <li v-for="house in Hogwarts">
    {{ house }}
  </li>
</ul>

<!-- It will render to -->
<ul>
  <li>Ravenclaw</li>
  <li>Hufflepuff</li>
  <li>Slytherin</li>
  <li>Gryffindor</li>
</ul>
```

```
// Given this sample Vue instance
new Vue({
  ...
  data: {
    Hogwarts: [
      'Ravenclaw',
      'Hufflepuff',
      'Slytherin',
      'Gryffindor'
    ]
  }
})
```

CODE DEMO

QUESTIONS?

V-BIND

Allows us to manipulate HTML attributes
with dynamic data

```
<!-- The long form -->
<div v-bind:class="{ active: isActive }">...</div>

<!-- The common shortcut -->
<div :class="{ active: isActive }">...</div>

<!-- If isActive is true, it will render -->
<div class="active">...</div>
```

CODE DEMO

QUESTIONS?

EXERCISE #3: HP WORLD

INSTRUCTIONS

1. Convert all "Static {property}" to data properties
2. Loop through the data property `wallet` to generate the same markup as the flat HTML version
3. Dynamically bind a class name that will be identical to the key as you looping through your data
4. Write methods to wire up buttons to do function commented above each button

LET'S TALK CLI

AS WE COME TO A CLOSE...

FINAL THOUGHTS



#1. ACCESSIBLE TO
DEVELOPERS OF
PRETTY MUCH ALL
ABILITIES


```
import React from 'react';
import { render } from 'react-dom';
import Hello from './Hello';

const styles = {
  fontFamily: 'Avenir, Helvetica, Arial, sans-serif',
  textAlign: 'center',
  color: '#2c3e50',
  marginTop: '60px'
};

const App = () => (
  <div style={styles} className="app" id="app">
    <h1>This is a React.js component file!</h1>
  </div>
);

export default App
```

JSX EXAMPLE

```
<template>
  <div id="app" class="app">
    <h1>This is a Vue.js component file!</h1>
  </div>
</template>

<script>
export default {
  name: 'App'
}
</script>

<style>
.app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

SINGLE FILE COMPONENT

**#2. THEIR DOCS
ARE ONE OF THE
BEST OUT THERE**



 Vue.js

[Learn](#) ▾ [Ecosystem](#) ▾ [Team](#) [Support Vue](#) ▾ [Translations](#) ▾

Special Sponsors




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

Patreon Sponsors





[Become a Sponsor](#)

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.



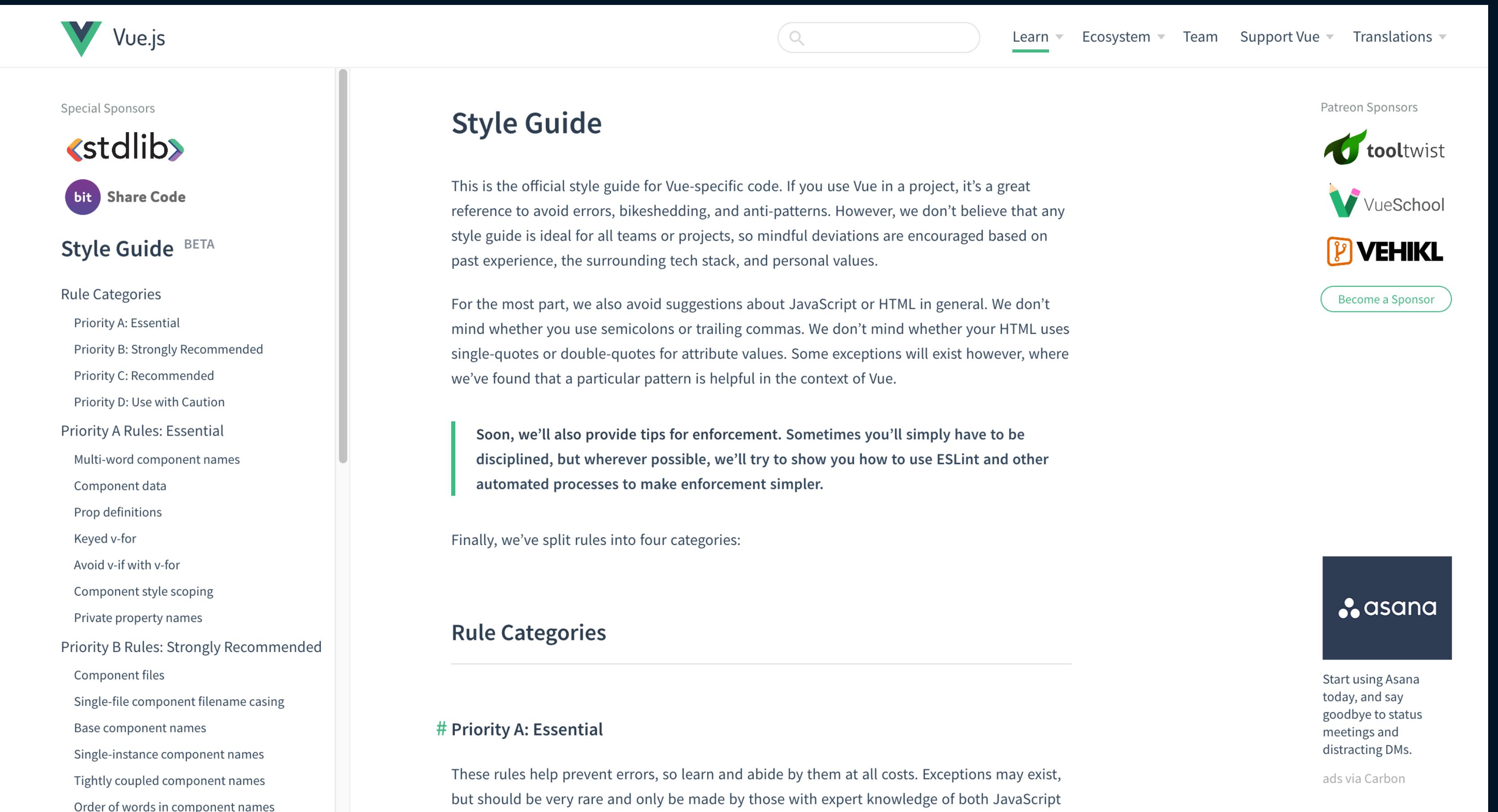
Accelerate App Adoption
ads via Carbon

VUE.JS DOCS

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



#3. IT IS THERE
WHEN YOU NEED IT
AND THEN IT GETS
OUT OF YOUR WAY.



The screenshot shows the official Vue.js Style Guide page. At the top, there's a navigation bar with the Vue.js logo, a search bar, and links for Learn, Ecosystem, Team, Support Vue, and Translations. Below the navigation, there are sections for Special Sponsors (stdlib, bit Share Code) and Patreon Sponsors (tooltwist, VueSchool, VEHIKL). A "Become a Sponsor" button is also present. The main content area features a section titled "Style Guide" with a beta status. It discusses the purpose of the style guide, emphasizing that it's for Vue-specific code and that deviations are encouraged based on context. It also notes that suggestions about JavaScript or HTML in general are avoided. A sidebar on the left lists "Rule Categories" under Priority A (Essential) and Priority B (Strongly Recommended), including topics like Multi-word component names, Component data, Prop definitions, Keyed v-for, Avoid v-if with v-for, Component style scoping, Private property names, and more. The main content also mentions future enforcement tips using ESLint and other automated processes. A "Rule Categories" section follows, with a "# Priority A: Essential" heading and a note about preventing errors. An Asana advertisement is on the right, encouraging users to start using Asana to say goodbye to status meetings and distracting DMs.

Style Guide

This is the official style guide for Vue-specific code. If you use Vue in a project, it's a great reference to avoid errors, bikeshedding, and anti-patterns. However, we don't believe that any style guide is ideal for all teams or projects, so mindful deviations are encouraged based on past experience, the surrounding tech stack, and personal values.

For the most part, we also avoid suggestions about JavaScript or HTML in general. We don't mind whether you use semicolons or trailing commas. We don't mind whether your HTML uses single-quotes or double-quotes for attribute values. Some exceptions will exist however, where we've found that a particular pattern is helpful in the context of Vue.

Soon, we'll also provide tips for enforcement. Sometimes you'll simply have to be disciplined, but wherever possible, we'll try to show you how to use ESLint and other automated processes to make enforcement simpler.

Finally, we've split rules into four categories:

Rule Categories

Priority A: Essential

These rules help prevent errors, so learn and abide by them at all costs. Exceptions may exist, but should be very rare and only be made by those with expert knowledge of both JavaScript

Special Sponsors
 stdlib
 bit Share Code

Patreon Sponsors
 tooltwist
 VueSchool
 VEHIKL

Become a Sponsor

Priority A: Essential

- Priority B: Strongly Recommended
- Priority C: Recommended
- Priority D: Use with Caution

Priority A Rules: Essential

- Multi-word component names
- Component data
- Prop definitions
- Keyed v-for
- Avoid v-if with v-for
- Component style scoping
- Private property names

Priority B Rules: Strongly Recommended

- Component files
- Single-file component filename casing
- Base component names
- Single-instance component names
- Tightly coupled component names
- Order of words in component names

Start using Asana today, and say goodbye to status meetings and distracting DMs.

ads via Carbon

VUE.JS STYLE GUIDE

<https://vuejs.org/v2/style-guide/>

**Vue.js is the most
compassionate framework
in the market right now.**

Gives me what I want
when I **need** it, and then it
gets out of my way.

Sarah Drasner

SO WHAT ELSE IS THERE TO LEARN ABOUT?

- ▶ State Management
- ▶ Animations
- ▶ Custom Events
- ▶ Computed Properties
- ▶ Managing CSS in Vue.js
- ▶ Slots
- ▶ Watch Property
- ▶ Component Props
- ▶ Lifecycle Hooks
- ▶ Mixins
- ▶ Filters
- ▶ And much more!

RECOMMENDED RESOURCES

Paid

- ▶ [Udemy: Vue.js 2 - The Complete Guide](#)
- ▶ [Frontend Masters: Introduction to Vue.js](#)
- ▶ [Vue Mastery](#)
- ▶ [Vue School](#)

Free

- ▶ [Official Vue.js Guide](#)
- ▶ [VueMastery: Intro to Vue.js](#)

POST-WORKSHOP SURVEY

<https://bencodezen.typeform.com/to/mwdQXo>

QUESTIONS?

<https://bencodezen.typeform.com/to/mwdQXo>

THANKS EVERYONE!

@BENCODEZEN

