Vue.js – Big Picture

WHY VUE.JS?

Daniel Stern
CODE WHISPERER

@danieljackstern

github.com/danielstern

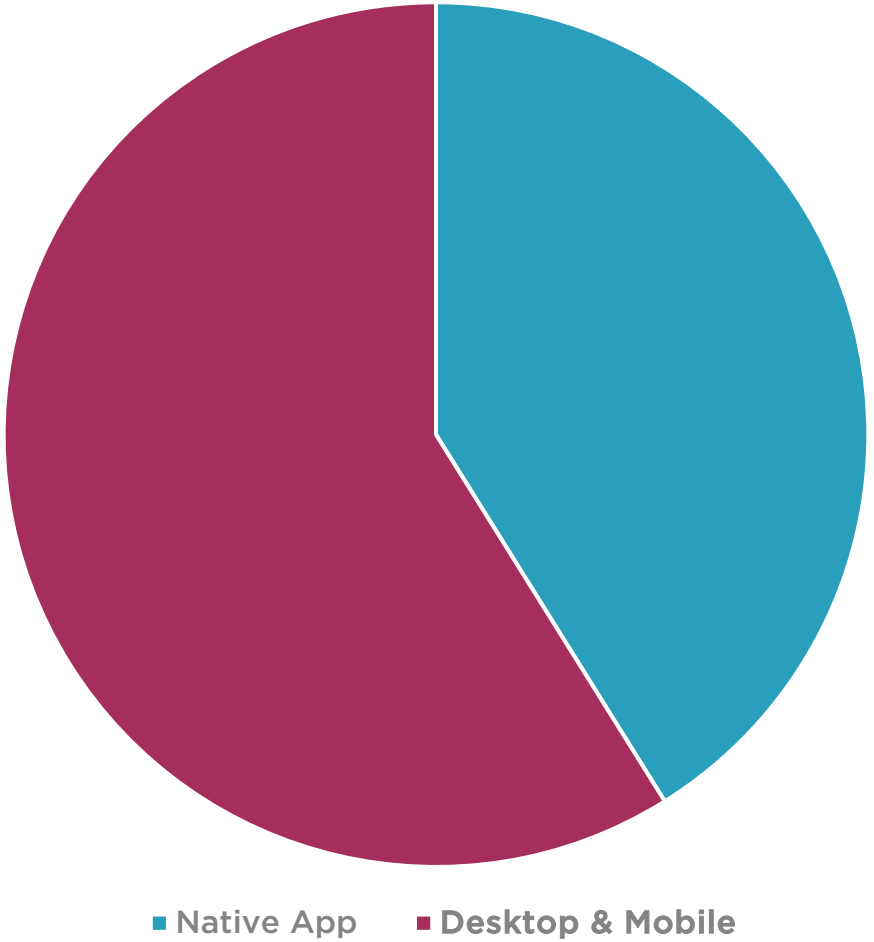
Course Overview

- Understand exactly what Vue is and if it can help achieve your business goals
- Get an overview of the core features Vue developers must master
- Know the specific challenges Vue can help you solve
- Gain context by comparing to other popular frameworks
- Continue learning with more Vue resources

Creating Business Solutions With Vue.js

E-commerce Transactions by Environment, 2017

Subset of market where Vue is useful



“In 2017, retail e-commerce sales worldwide reached $2.304 trillion, a 24.8% increase over the previous year.”

- eMarketer (estimate)

Vue.js and E-commerce

Vue apps can be built quickly by relatively inexperienced teams

Fast performance and small package (ideal for mobile)

Everything you need to build a view layer for your application

The Hidden Costs of Not Using Vue

Mobile users get bored and navigate away before your framework of choice has even loaded

Redrawing lists or just updating the DOM is so slow, your website seems like it has frozen altogether on mobile devices

Developers employed to update a messy and disorganized application take a long and expensive time to do so

Why Vue?

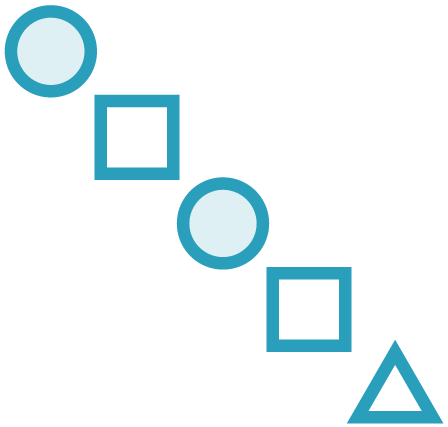
“Some are born great, some achieve greatness, and some have greatness thrust upon ’em.”

William Shakespeare, *Twelfth Night*

jQuery
Angular
React

vue

Why Vue?



Works for new or existing applications

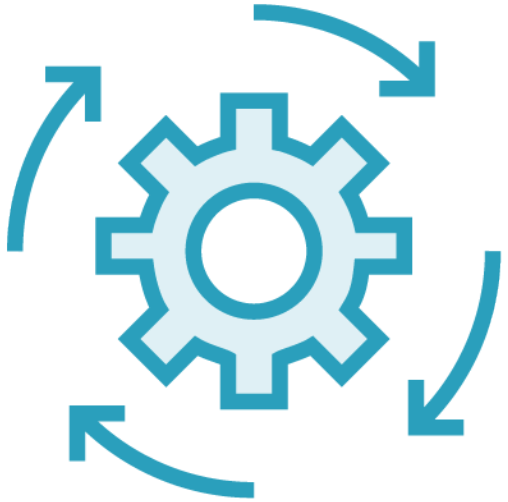
Features are familiar to React and Angular developers

Fast, easy to understand and to maintain

#1

105,000K+ Stars on Github

Who Uses Vue?

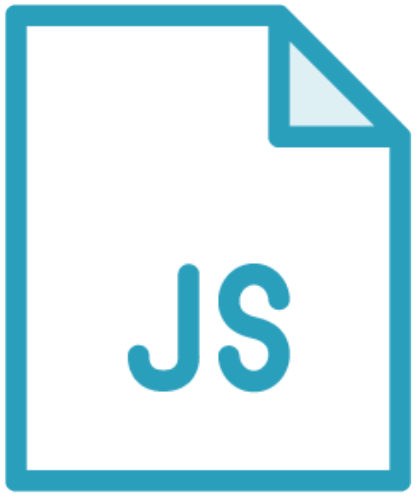


Small teams or individuals creating prototypes or very small apps

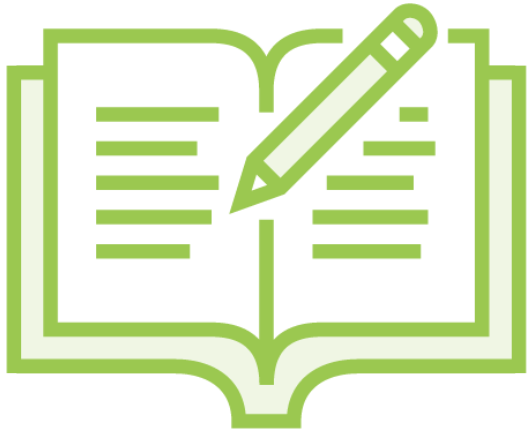
Large teams collaborating on building large web applications

Seasoned developers brought in to fix a problematic area of a legacy website

Prerequisites for Using Vue



Basic knowledge of JavaScript, CSS and HTML



Functional text editor (Atom or WebStorm are recommended)

Basic local server (Apache, http-server, Vue CLI)

What Is Vue?

What Is Vue?

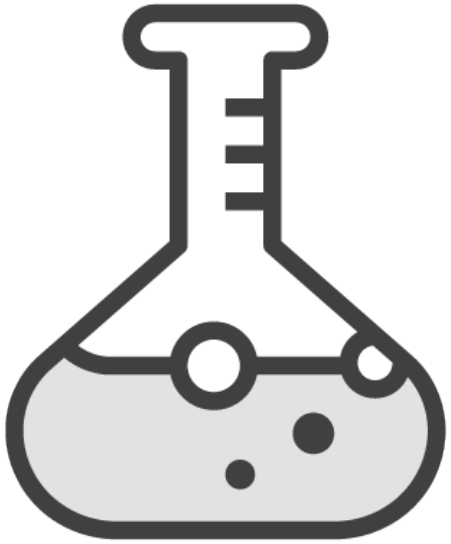
A JavaScript library (A large block of JavaScript code that is executed before your own code is executed and which gives context to code you write)

A render engine, taking a data model and components as input and outputting HTML

A collaboration tool, allowing for the segmentation of applications into smaller parcels of work (using components)

Vue's Core Features

Versatile components



Built-in reactivity

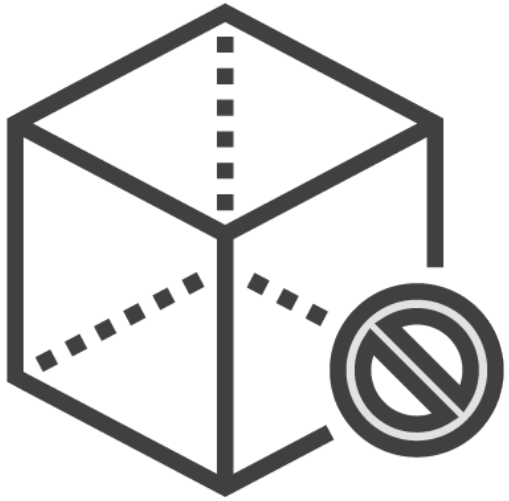
Fast initial load, fast render

v-for
v-if
v-bind

Familiar directives

Vue's Proprietary File Type - ".vue"

.vue files can contain HTML, CSS and JS Code



.vue files can't be read by web browsers (build step is required)

Allows all the code for a single component to be in one file

```
<template>
  <p>
    Meaning of life: {{message}}
  </p>
</template>

<script>
  module.exports = {
    data(){
      return {
        message: 42
      }
    }
  }
</script>

<style scoped>
  p { color : blue }
</style>
```

- Template contains HTML
- Interpolation can occur within template
- Script contains application code
- Exported value is automatically turned into Vue component
- The number 42 will appear as message in the template above
- Styles will be applied to this component only (scoped)

Finding Vue on the Net

Contribution, issues and source code on GitHub

Library downloaded from CDN or NPM

Documentation on *vuejs.org*

Vue Capabilities and Limitations

Vue Capabilities and Limitations

What Vue Does

Outputs HTML code for browsers

Maps user interactions (e.g, clicking, hovering) to JavaScript methods you write

Uses sophisticated techniques to speed the rendering of the application

Works in most browsers without any build step

What Vue Does Not Do

Does not create a server for your application (compliment with Vue-CLI)

Does not test code for errors

Does not provide significant utility methods

Does not facilitate routing (Compliment with Vue-Router)

Does not provide own solution for modules beyond native ES6

Reactivity and Vue

Reactivity

The way an application changes what you see in response to activity like the user hitting a button, or an incoming message from the server.

Reactivity Example

```
[`apples`,`pears`]
```

Your Shopping List

- Apples
- Pears

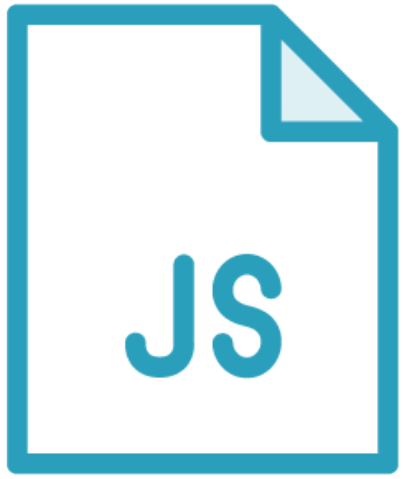
Reactivity Example

```
[`apples`,`pears`,`grapes`]
```

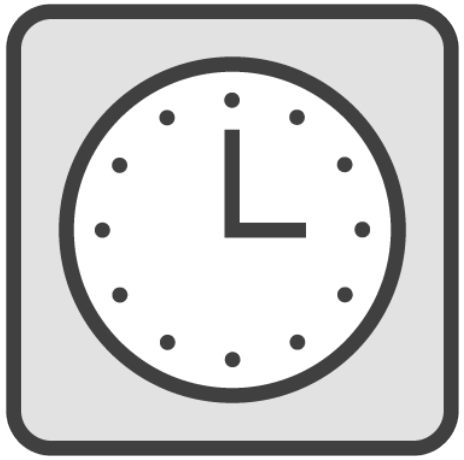
Your Shopping List

- Apples
- Pears
- Grapes

Why is Reactivity Important?



JavaScript does not have built-in reactivity

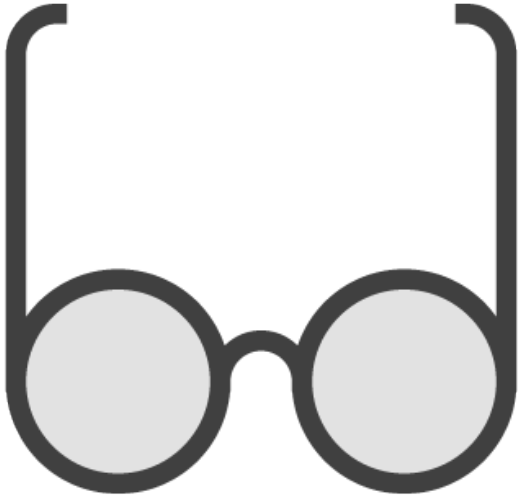


Reactivity that is not implemented correctly can be extremely slow

Vue and similar libraries solve this problem differently

How Vue Handles Reactivity

Changes to data model are automatically detected



What the user sees is updated automatically and instantly

Elements are reused and changes are batched to improve performance

Vue's Reactivity: Developer vs. User

Code is easier to write and maintain

Vue uses virtual DOM to make large updates fast and declarative

Less code, fixing bugs is easier

The application is fast and enjoyable

Application doesn't seem to get slower despite more activity

User experiences fewer bugs

Reactivity and Vue – Challenges

Vue components update automatically when model changes

Reactivity is simple for numbers and strings (immutable)

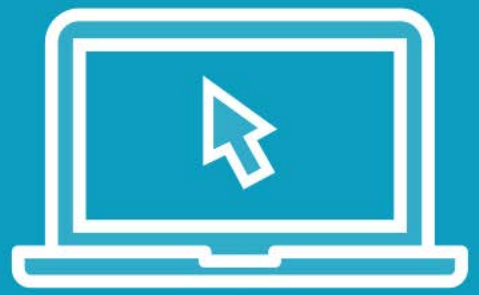
Reactivity is not so simple for arrays or objects (mutable)

Vue can’t always tell when an array has been updated

With mutable data, view may not update as expected

A Simple Vue Application

Demo



A basic to-do list application

Things to note...

- No build step
- Compact file size
- Vue is being downloaded from CDN

Take some time to explore...

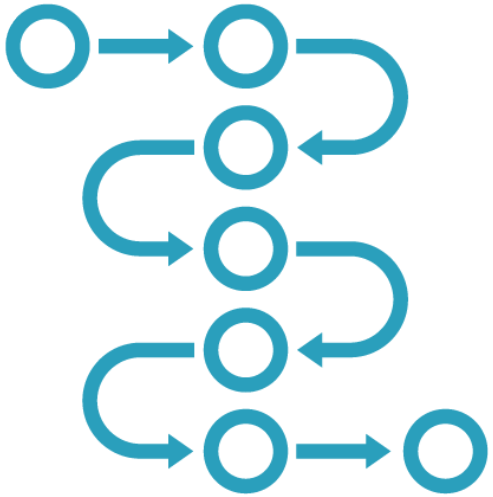
- Interact with the app:
  http://danielstern.ca/vue
- Clone the app, modify it, and read my detailed comments
  https://github.com/danielstern/vue-todomvc

Where Should You Use Vue?

Where Should You Use Vue?

Modernizing Legacy Applications

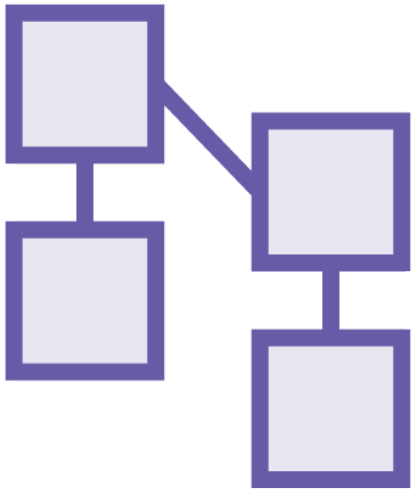
Building Applications as a Team



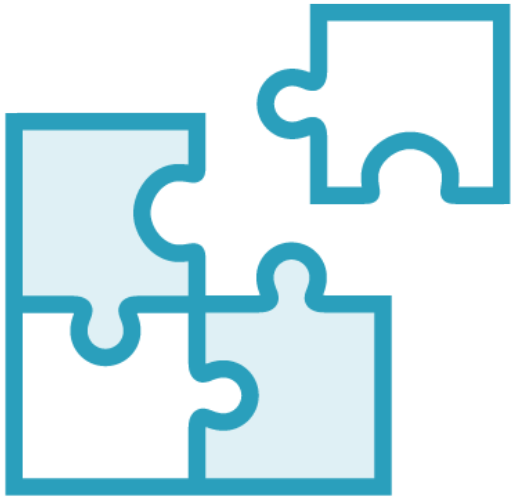
Prototyping

Modernizing Legacy Applications with Vue

Vue can be incrementally included



Problem areas can be replaced without changing application structure



Vue co-exists well with React, Angular and jQuery

Benefits to affected areas are felt immediately

“Cowards die many times before their deaths; The valiant never taste of death but once.”

William Shakespeare, *Julius Caesar*

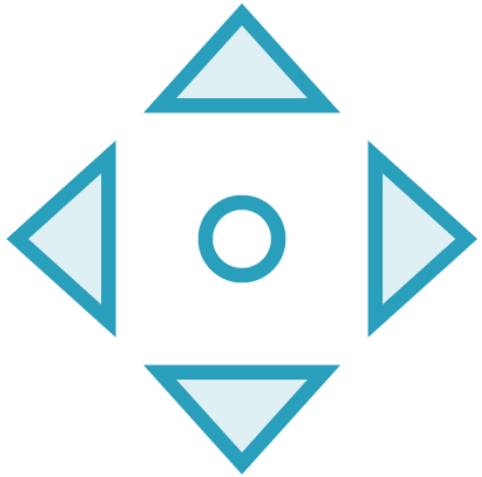
When Should You Modernize Your Legacy Application with Vue?

Simple changes (especially to the view layer) take a long time and are so complicated, only senior devs can do them

Requests for new features in the view layer are denied because "implementing them would be too difficult" or because making the change "might break something else"

Application code is all ad-hoc (no consistent use of frameworks or best practices), new developers have to learn how everything works from the ground up

Building Applications as a Team



Components allow separation of concerns

HTML, CSS and JavaScript developers can work together

Growing proportion of new team members are already familiar with Vue

Ecosystem libraries provide necessary tooling for production applications

Prototyping

Vue can be set up very quickly (at minimum, almost no setup at all)

Prototype apps can include many elements with no performance issues

Conversion to a full-scale Vue app is straightforward

The Vue Ecosystem

What Is The Vue Ecosystem?

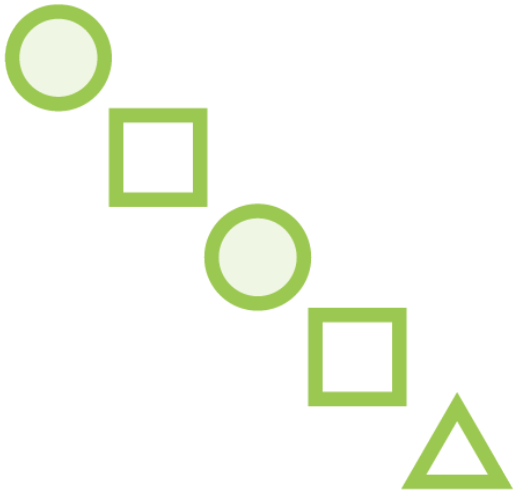
Vue is small library and its core features are limited to quickly rendering your application's view

To achieve the same functionality as React or Angular (which too have their own ecosystems) other libraries need to be used together with Vue

A Few of Vue's Complimentary Libraries

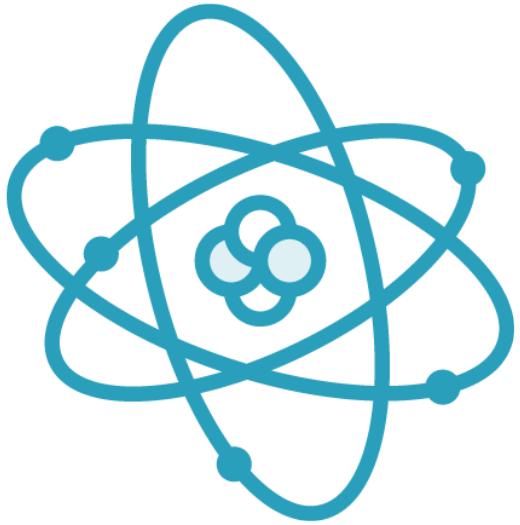
nuxt

Universal Rendering



vuex

State Management



vuetify

Component Framework

vue-cli

Scaffolding & Build

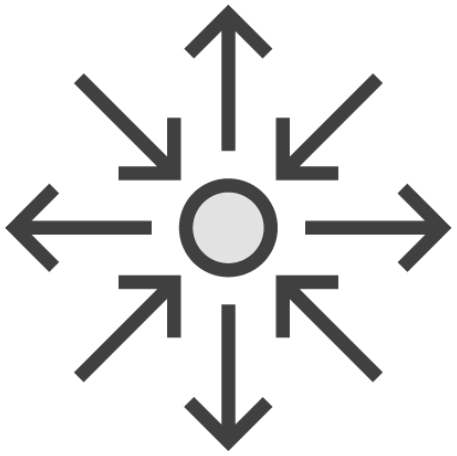
Nuxt

Used to create universal (server-rendered) Vue applications

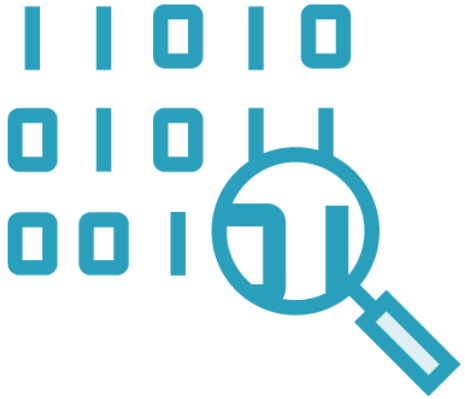
Many universal challenges are simplified by nuxt

Faster loading times on all devices, especially mobile

Vuex



Provides a way of updating your state

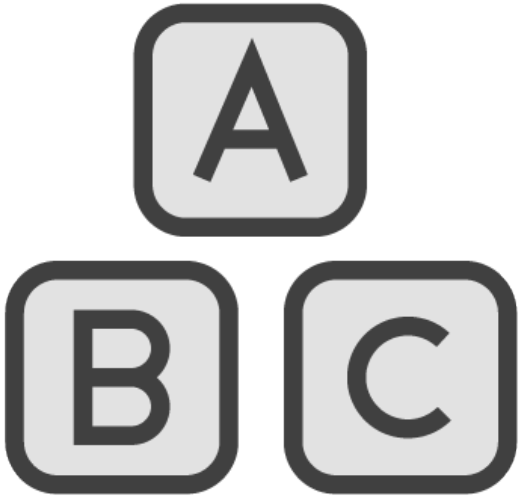


Similar to Flux or Redux

Used to keep applications with massive state manageable

Vuetify

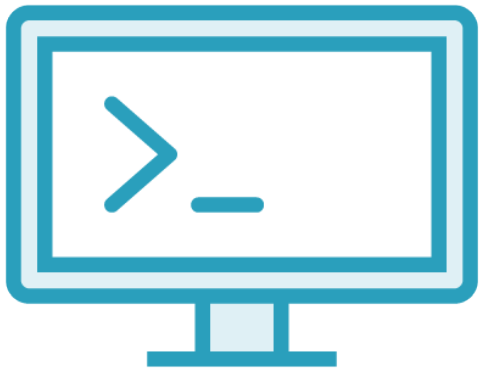
Contains sleek and performant Vue components



Has a material design aesthetic

Used as an alternative to creating own solutions for UI

Vue-CLI



Tool for creating, building and updating Vue applications

Provides standard way to update app across whole team

Allows junior developers to add to the application's structure

Facilitates productivity among large teams and big projects

```
#vue cli commands
> vue serve

> vue build

> vue create
> vue ui

> vue add
```

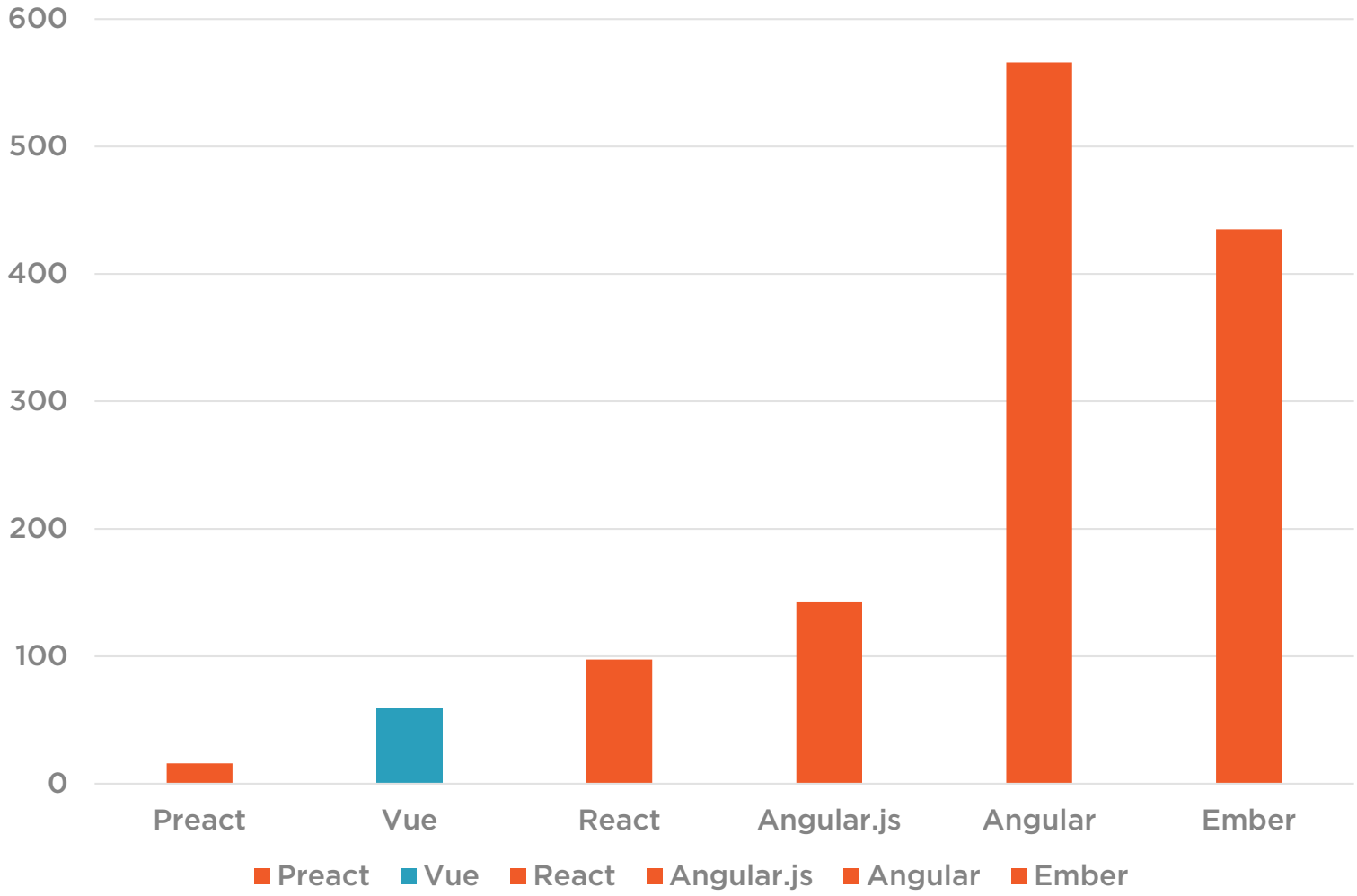
- Serves the current directory as an application with app.vue as its root
- Assembles a .vue and ES6 project into an HTML and JS one usable by browsers
- Creates a new Vue project (using CLI or UI)
- Installs plugins for Vue-CLI

Comparison to Existing Frameworks

“There is nothing either good or bad, but thinking makes it so.”

William Shakespeare, *Hamlet*

File Size of Major Frameworks (Minified)

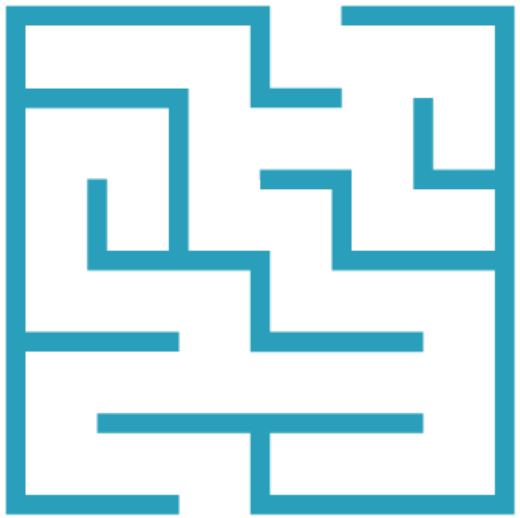


Vue vs. React

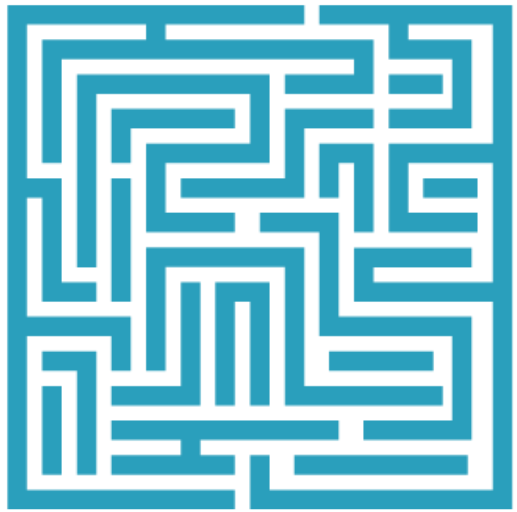
Vue	React
Builds fast and scalable applications	Builds fast and scalable applications
Speeds up render with Virtual DOM	Speeds up render with Virtual DOM
Proprietary file type, .vue, used to logically group HTML and JS	Proprietary coding language, .jsx, used to logically group HTML and JS
Add functionality with directives such as v-for and v-if	Add functionality with plain JavaScript such as maps and ternary statements
Vue-CLI makes structure easy	Create-React-App makes structure easy
Vue without build step provides excellent path to prototyping	React needs JSX – even prototypes should have build step

Vue vs. Angular

Vue



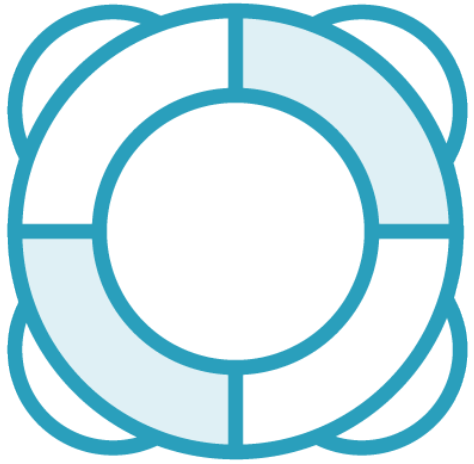
Angular



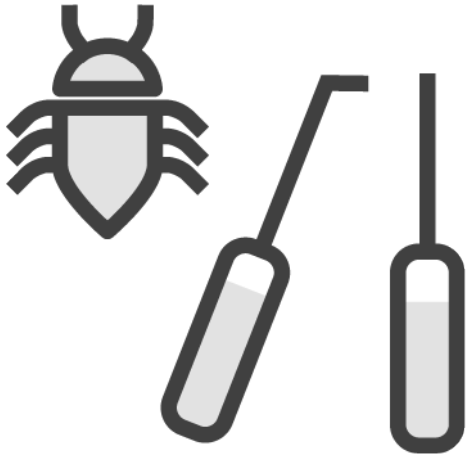
Vue vs. Angular

Vue	Angular
Simple and easy to learn due to emphasis on plain HTML and JavaScript	Learning Angular is no simple task due to emphasis on modules and TypeScript
Uses only JavaScript modules (optional)	Uses JavaScript modules *and* Angular Modules
Easy to integrate as just one part of the application	Does not tend to integrate well with other frameworks
Productivity peaks at medium-sized applications	Productivity peaks for large or very large applications
Add functionality with directives such as v-for, v-if	Add functionality with directives such as ng-for, ng-if

Vue Pitfalls and Drawbacks



Needs support from ecosystem

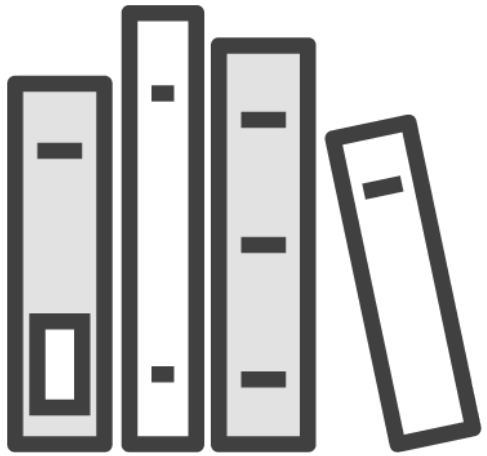


Tricky reactivity

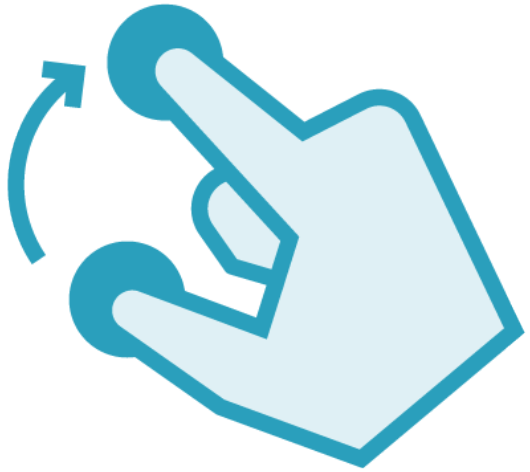
.vue file type may confuse new developers

Relatively new and untested

Why You Might Not Use Vue



Less established, fewer existing resources



Many production scenarios are yet unproven

From the beginning, It is known that the finished project will be huge

The Vue Build Step

The Vue Build Step

Uses Webpack to transform *.vue* files into JavaScript, via Vue-CLI

Vue applications using *.vue* files require a build step to work

Build step is not needed for Vue applications that do not use *.vue* files (and other advanced features like JSX and ES6 imports)

“More dependencies,
more problems.”

Code Whisperer

Benefits of Not Having a Build Step

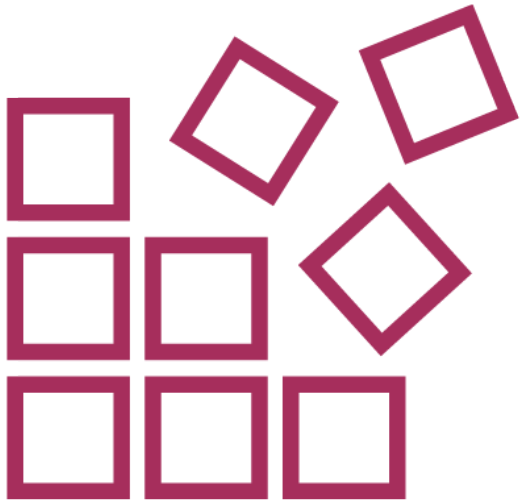
Very fast to get started

Much smaller number of dependencies

Application *can* be deployed with no preparation at all

Benefits of Having a Build Step

Use of *.vue* files, JSX, TypeScript permitted



Code can be optimized for production

ES6 Modules and experimental features can be used

Summary

Daniel Stern
CODE WHISPERER

@danieljackstern

github.com/danielstern

Executive Summary

Vue is fast, lightweight and easy to use

Vue has much in common with both Angular and React, though it is simpler

Vue applications are built from components

Vue works without a build step for prototypes, and with a build step for serious production applications

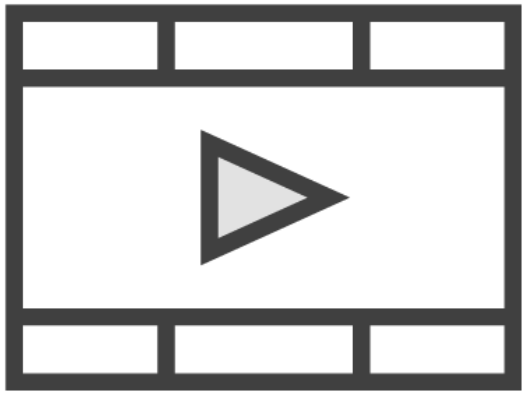
Vue has a vibrant ecosystem – without it, Vue only renders the view layer

Vue Advantages Summary

- Small package size
- Omit build step when necessary
- Familiar API
- Strong command-line tools
- Great for prototyping
- Is easy to learn and use
- Strong community support

Continue Your Education

Vue.js Fundamentals
By Jim Cooper



Vue.js Getting Started
By Chad Cambell

Check back frequently for more great courses!

Thank You!