CS 4220

- Current Trends in Web Design & Development -

Cydney Auman

AGENDA

01 What is Vue.js?

02 Why Vue.js?

03 Working with Vue.js

04 Code Demo

05 HW - Vue.js App

What is Vue.js

Vue.js (pronounced like view) is a Progressive JavaScript Frontend Framework for building user interfaces.

- Approachable

If you are already familiar with HTML, CSS and JavaScript? Start building a Vue.js application quickly and easily. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects.

- Versatile

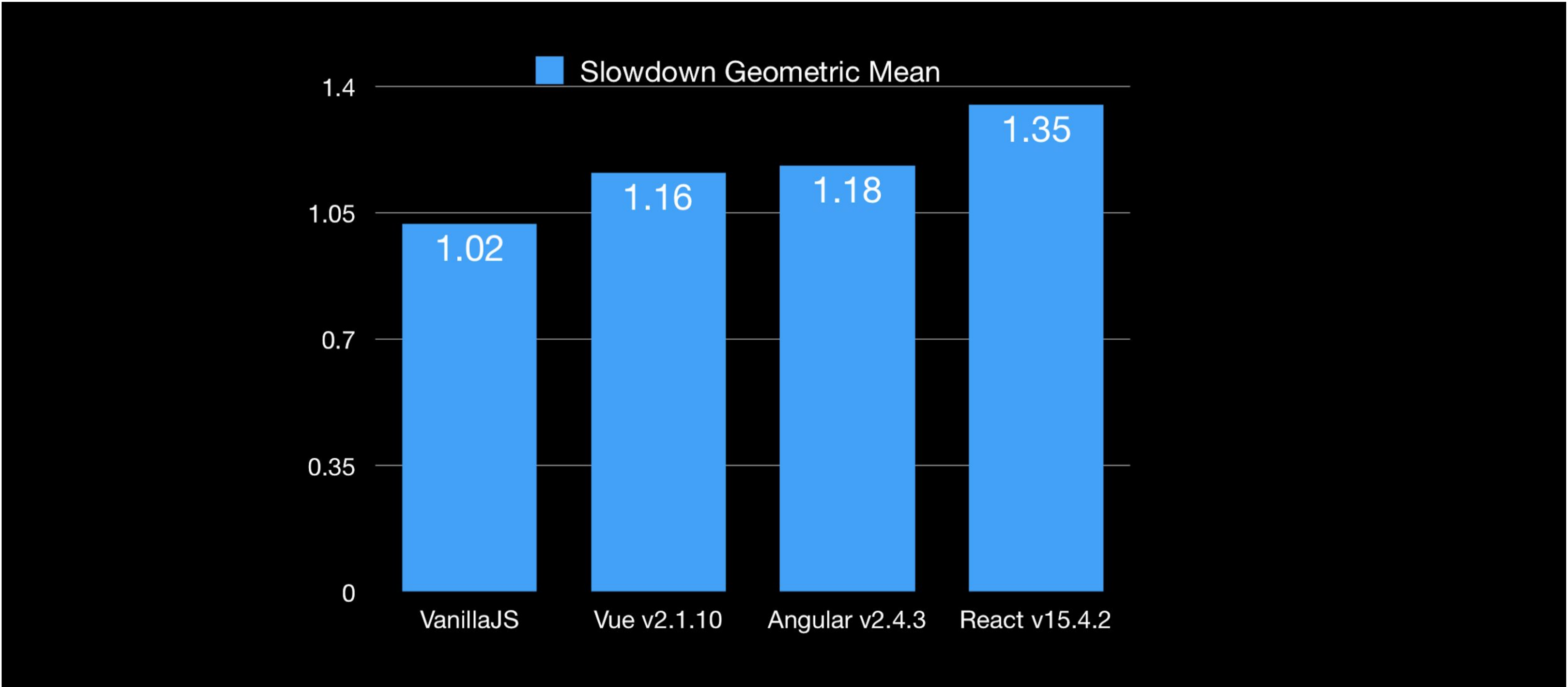
An incrementally adoptable ecosystem that scales between a library and a full-featured framework. Vue capable of powering simple Single-Page Applications to sophisticated Web Applications when used in combination with modern tooling and supporting libraries.

- Performant

Around 30KB min+gzip Runtime Blazing Fast Virtual DOM Minimal Optimization Efforts

Why Vue.js

Vue.js is Fast!



Data Source: http://stefankrause.net/js-frameworks-benchmark5/webdriver-ts/table.html

Why Vue.js - DOM Virtualization

DOM updates are expensive. DOM Recalculations are expensive and blocks the User.

Virtual DOM is essentially a way to generate a version of the DOM in-memory each time state is changed. This is then compared to the actual DOM. Now, only the parts that need to be updated are updated.. Without this concept of Virtual DOM instead everything would need to be re-render.

Vue.js uses Virtual DOM

- Lightweight copy of the actual DOM
- Vue efficiently determine what needs to be updated
- Batch updates the DOM

Working with Vue.js

The Core Library of Vue.js is available via a Content Delivery Network (CDN).

To start using Vue.js simply include the script tag in your HTML:

```
<script src="https://unpkg.com/vue@2.6.11/dist/vue.js"></script>
```

This script tag is asking the CDN to grab version 2.6.11 of Vue.js which is currently the stable build.

Working with Vue.js

Every Vue application starts by creating a new Vue instance with the Vue function:

The Vue instance will be mounted to the CSS element specified by the key el.

Additionally, when a Vue instance is created, it adds all the properties found in its data object to Vue’s reactivity system. When the values of those properties change, the view will update to match the new values.

```
const app = new Vue({
    // use CSS selector to identify the element
    el: '#grocery-app',
    data: {}
});
```

Working with Vue.js - Template Syntax

Vue.js uses an HTML-based template syntax that allows you to declaratively bind the rendered DOM to the underlying Vue instance's data. All Vue.js templates are valid HTML.

Under the hood, Vue compiles the templates into Virtual DOM render functions. Vue is able to intelligently figure out the minimal number of items to re-render and apply the minimal amount of DOM manipulations when the app state changes.

The most basic form of data binding is text interpolation using "Mustache" (double curlys): `{{ }}`.

Working with Vue.js - Template Syntax

Vue.js uses an HTML-based template syntax that allows you to declaratively bind the rendered DOM to the underlying Vue instance's data. All Vue.js templates are valid HTML.

Under the hood, Vue compiles the templates into Virtual DOM render functions. Vue is able to intelligently figure out the minimal number of items to re-render and apply the minimal amount of DOM manipulations when the app state changes.

The most basic form of data binding is text interpolation using "Mustache" (double curlys): `{{  }}`.

Template Syntax Vue.js

Vue will replace the data associated with title and course in the curly braces in the template.

```
const app = new Vue({
     // you can use CSS selector to identify the element
     el: '#app',
     data: {
         title: 'Web Design and Application Development',
         course: 'CS4220'
     }
 });
```

```
<h1>{{ title }}</h1>
<h2>{{ course }}</h2>
```

Working with Vue.js - Directives

Directives are special attributes with the **v-** prefix. Directive attribute values are expected to be a single expression. A directive’s job is to reactively apply side effects to the DOM when the value of its expression changes.

- **v-model**

Directive to create two-way data bindings on form input, textarea, and select elements.   Vue automatically picks the correct way to update the element based on the input type.

- **v-for**

Iterate and render the element or template block multiple times based on the source data provided.

- **v-show**

Toggles the element’s display property based on the truth of the expression. If true it will show else it will not.

```
console.log(‘Week 13’);
console.log(‘Code Examples’);
```

Lab, Homework and Prep

Lab Time

- Run the Code Example
- Extra Credit Review
- Work on the Lab Assignment

Preparation for Next Week

- Readings:

  Eloquent JavaScript - Chapter 13
  https://vuejs.org/v2/guide/