

CS 4220

- Current Trends in Web Design & Development -

Cydney Auman

AGENDA

- 01** Vue.js Review
- 02** Vue.js Components
- 03** Code Demo
- 04** HW Review & Final Discussion
- 05** Extra Credit

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

Vue.js Instance

Every Vue application starts by creating a new Vue instance with the Vue function. When you create a Vue instance, you pass in an options object.

The Vue instance will be mounted to the CSS element which is an option represented by the key **el**.

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

Vue.js Instance Options

Vue.js offers many options when creating a Vue instance.

data: 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.

methods: These static methods which are bound to key presses, mouse clicks and/other events. All methods will have their this context automatically bound to the Vue instance. Methods defined can be called from other methods.

computed: Values are cached and only re-computed when a dependency changes. If a dependency is out of the scope, the computed property will not be updated. Computed methods cannot accept arguments.

Vue.js Instance Options

Vue.js offers many options when creating a Vue instance.

components: Components are reusable Vue instances with a specific name. Since components are reusable Vue instances, they accept many of the same options as new Vue instance. A notable exception is they do not accept the `el` option. Because components have a specified name.

props: A list and/or hash of properties that are exposed to accept data from the parent component or the Vue instance.

template: HTML represented as a string template to be used for the Vue instance. The template can be associated to a component and will be used when the component is.

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.

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-if/v-else-if/v-else**

Directive to conditionally render the html element based on the expressions evaluation of true or false.

- **v-bind**

Directive to dynamically bind one or more attributes. Attribute include images, href and CSS styling.

- **v-cloak**

Directive that is used for hiding un-compiled mustache bindings until the Vue instance is ready.


```
console.log( 'Week 15' );  
console.log( 'Code Examples' );
```

Lab, Homework and Prep



Lab Time

- Lab Week 13 Review
- Discuss the Final and associated Extra Credit
- Work on the Final