Template Syntax

With Vue being an MVVM tool, there needs be a way to bind the models to their respective views. This way, data being handled by an underlying model is printed in the browser for the user's consumption.

Views are represented by HTML templates but with extra utility features like interpolation ({{ }}) and directives (v-on:click) to help display the models' data.

A Comparison

techniques will make us appreciate what Vue's templates have to offer. A simple and perfect example is trying to insert text into a given div:

A quick look at the conventional JavaScript DOM manipulation (without Vue)



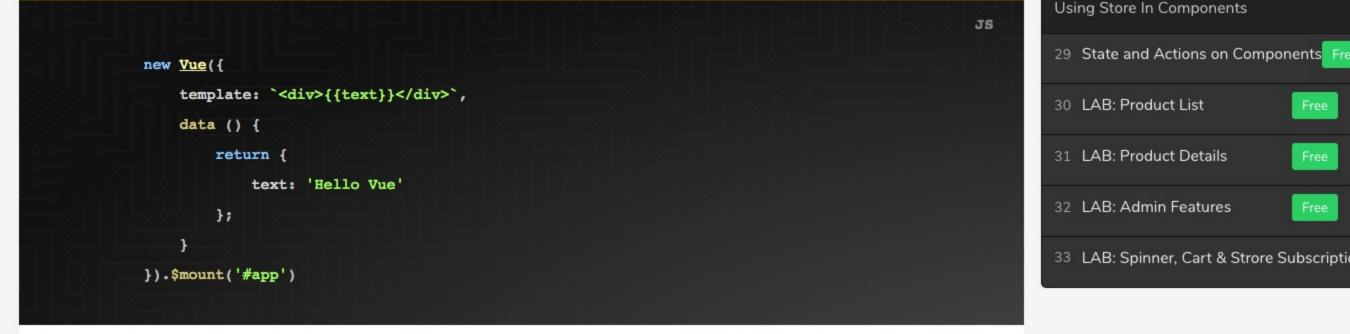
// jQuery
const divElement = \$('#text');
divElement.text('Hello Vue')

It's even more tedious with JavaScript:

// JavaScript
const divElement = document.getElementById('text');
const textNode = document.createTextNode('Hello Vue');

divElement.appendChild(textNode);

With Vue, it's dead simple and declarative:



Interpolation

The template (view) and data (model) are split into different entities but bound

together using the object properties. Let's explain some binding concept:

The concept of using double curly braces as a placeholder for data properties in a Vue template is known as an interpolation:

<div>{{text}}</div>

The double curly braces are popularly known as Mustache syntax/tag. Vue is not the

The mustache tag will be replaced with what ever the data holds during render. The amazing thing is that when the bound data changes, the template automatically

gets updated. This is one painful limitation our JavaScript example has. We would

only tool that uses it so don't get thrown off when you see it somewhere else.

have to check for changes and do the update manually.

Directives

Directives are custom HTML attributes that allow you to bind data to both view and

attributes. It also helps with binding DOM events.

<div v-text="text"></div>

```
<!-- same as -->
<div>{{text}}</div>

Directive Attributes
```

<!-- Another example with href -->

v-on which is Vue's event binding directive:

<a :href="url">{{text}}

<!-- WRONG!!! -->

<div id="{{id}}">{{text}}</div>

You might be tempted to use interpolation (mustache) in HTML attributes:

Vue will scream errors once it encounters that.

The v-bind directive is responsible for such tasks. So you could safely do this:

HTML

HTML

JS

HTML

<!-- VERY CORRECT!!! -->

<div v-bind:id="id">{{text}}</div>

```
If that seems like a lot of key strokes, shorthand is allowed:

HTML

| Compared to the strokes of the strokes
```

Same pattern works for event binding as well. Rather, than v-bind, we will use

Vue allows you to modify the (default) behavior of some attributes right inside the

template. A good guess is that you have seen something like this:

// Send to server

function handleFormSubmit(event) {
 event.preventDefault();

```
The snippet will make your browser not reload in an attempt to submit a form. You don't have to take this dirty job to the logic again; you can easily use a modifier to handle such case in your template:
```

<form v-on:submit.prevent="handleFormSubmit">
</form>

Template Expressions

<form @submit.prevent="handleFormSubmit">

<!-- Shorthand -->

</form>

Getting Started 1 Introduction 2 Tools 3 Our Task 4 Hello World Vue Basics 5 Vue CLI 6 Components and the Vue Instance Fre 7 Template Syntax 8 Conditional & List Rendering 9 Styles & Classes Routing 10 Single Page Apps 11 Creating Routes 12 Route Outlets & Links 13 Nested & Dynamic Routes Forms 14 Two Way Binding (Reactivity) 15 Form Validation 16 Handling Form Submissions API Backend 17 Prelude 18 Setup 19 Provisioning a MongoDB Database Fr 20 Enabling CORS 21 Schemas & Models 22 Routes and Controllers 23 Testing with POSTman Vuex 24 The Current Problem 25 State Management 26 Getters 27 Mutations 28 Actions