Vue.js

Quickstart - you can include Vue with:
<script src="https://unpkg.com/vue"></script>

Vue Architecture

Basic Vue Application

```
<div id="my-app">
  {{ message.toUpperCase() }}
</div>
<script>
  const app = new Vue({
    el: '#my-app',
    data: {
      message: 'Hello C4Q!'
    }
  })
</script>
```

This vue app renders as:

```
HELLO C4Q!
```

The "data" object contains all of the information the app knows about.

"el" is the id of the HTML element for this Vue app. You can have more than one vue app on a page with different ids.

Properties for Vue Config

`el : string`
Declares which HTML element this Vue instance should be bound to.

`data : obj`
Declares the starting value of all the data that this Vue instance can use.

`methods : obj`
Declares all of the functions that this Vue instance has access to.

`created : ftn`
A function which is run upon creation of the Vue instance. Useful for AJAX.

... and many more (see the API documentation)

Templating

```
<div id="my-app">
  <p>{{ message }}</p>
  <p>{{ value + 3 }}</p>
  <p>{{ getResult() }}</p>
</div>
```

```
const app = new Vue({
  el: '#my-app',
  data: {
    message: 'Hello C4Q!',
    value: 7
  },
  methods: {
    getResult() {
      return 'Good morning'
    }
  }
})
```

```
Hello C4Q!
10
Good morning
```

The basic templating syntax allows you to use values in data, interpreted values, and also the results of methods to get data for the page.

Data Binding

Binding data into the DOM

v-bind:attribute

```
<img v-bind:src="imageURL">
<div v-bind:style="{color: getColor()}">
```

v-for

```
<span v-for="tag of tags">{{tag}}</span>
```

v-if

```
<div v-if="str.length">{{str}}</div>
```

v-html & v-text

```
<p v-html="getTemplate()"></p>
```

Getting data out of the DOM

v-on:event

```
<button v-on:click="doLogin">Login</button>
<input v-on:keyup="keyCount += 1">
<input v-on:keyup.enter="showMessage('hi')">
```

Two-way binding (for form elements)

v-model

```
<input v-model="user.username">
<textarea v-model="comment"></textarea>
```

v-bind

You can use v-bind to give Vue control of any html attribute. v-bind can use a variable, expression, or function result as the value to bind.

v-text & v-html

v-text escapes any HTML characters (e.g. < and >) before display. v-html injects HTML elements. Watch out for XSS attacks using v-html!

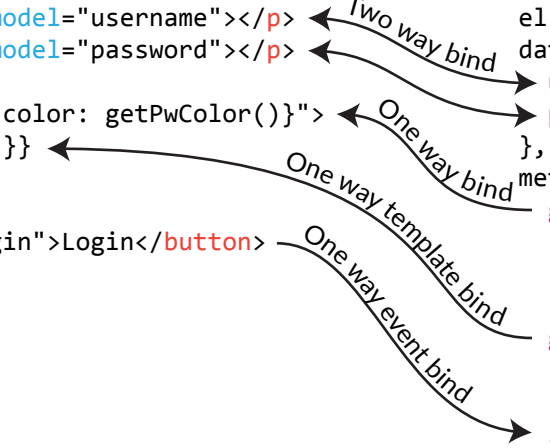
v-on

v-on allows Vue to handle any DOM event. You can use a function handle, or expression (including function calls) as the handler for v-on events. These can get pretty fancy & powerful!

Using Binding

```
<div id="my-app">
  <p>Username: <input v-model="username"></p>
  <p>Password: <input v-model="password"></p>
  <p>Password Strength:
    <span v-bind:style="{color: getPwColor()}">
      {{ getPwStrength() }}
    </span>
  </p>
  <button v-on:click="login">Login</button>
</div>
```

```
const app = new Vue({
  el: '#my-app',
  data: {
    username: '', //starts empty
    password: '' // starts empty
  },
  methods: {
    getPwColor() {
      return this.getPwStrength() > 9 ?
        'green' : 'red'
    },
    getPwStrength() {
      return this.password.length //todo
    },
    login(evt) {
      if (this.getPwStrength() > 9) {
        // todo login via AJAX
        this.username = ''
        this.password = ''
    }}})
```



Username:
Password:
Password Strength: 0
Login

Gotchas

!! Modifying objects and arrays sometimes doesn't trigger a re-render, use Vue.$set or arr.$set if you're adding/removing elements.

!! The "data" object can be an object in the main Vue app, but must be declared as a function that returns an object in a component.

!! Everything must go in the Vue config object. If it's not in data, methods, etc, then Vue doesn't know about it.

Created by Erty Seidohl.
Apr 4, 2017. Vue 2.0.