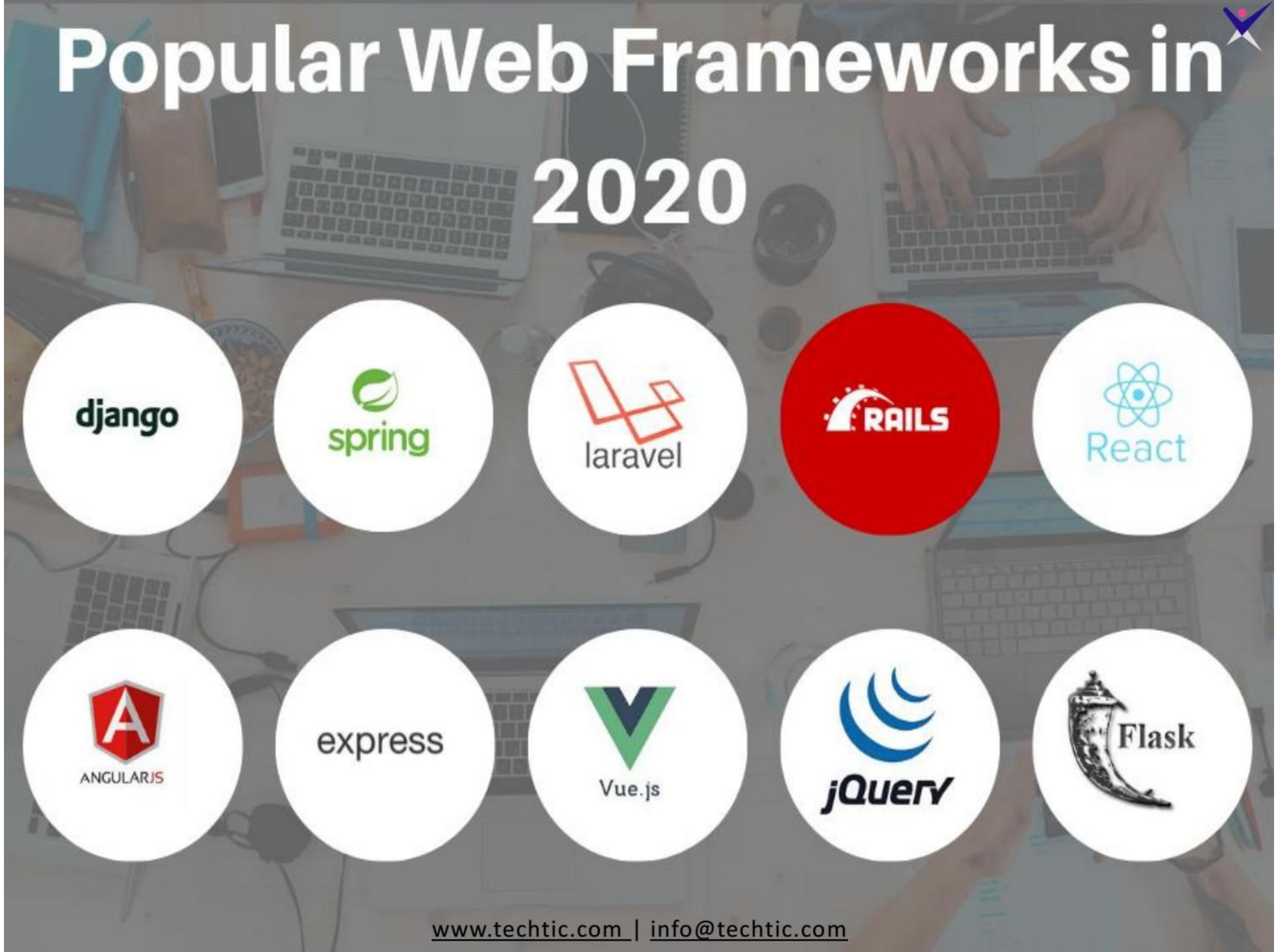
Industry 4.0 Academy

Introduction to VueJS

Popular Web Frameworks in 2020



django

spring

laravel

RAILS

React

ANGULARJS

express

Vue.js

jQuery

Flask

Framework & Libraries

Libraries & Framework	Language
Django	Framework using Python
Spring	Framework using Java
Laravel	Framework using PHP
Rails	Framework using Ruby
React JS	Frontend Library using JS
Angular	Framework using JS
Express	Backend Framework using JS
Vue JS	Framework using JS
Jquery	Frontend Library using JS

Why Vue

- Low learning curve
- Widely use or **rising technology**
- Nearly as powerful as other framework (ReactJS, Angular)
- Very modular / flexible
- New, but stable technology

Getting started with Vue JS

Install Nodejs and npm

https://nodejs.org/en/

Install Vue

```
npm install --global @vue/cli (PC)
```

sudo npm install --global @vue/cli (MAC)

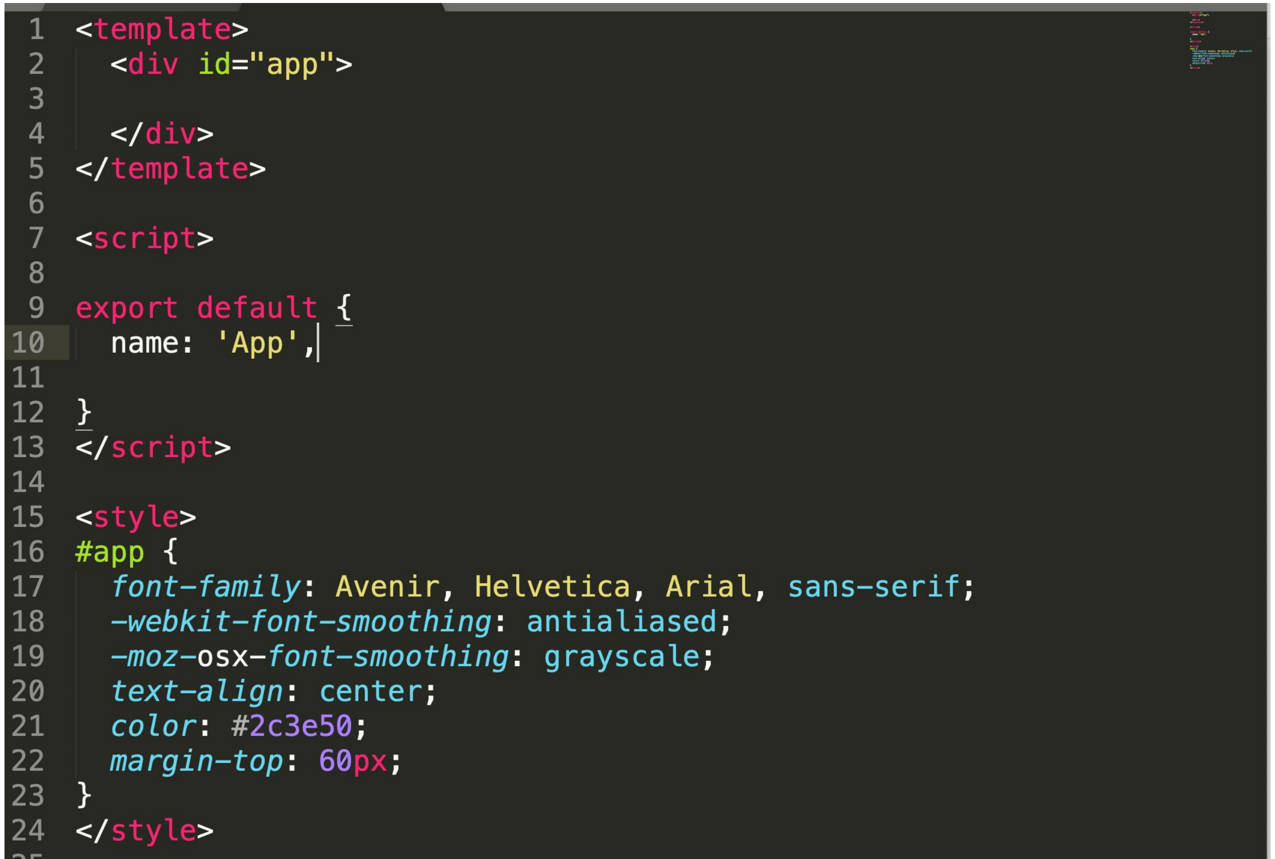
To create and Start the project

```
Vue create hello-vue

Select default (Vue2)

cd hello-vue
```

Install vue.js plugin on Sublime

1) Ctrl + shift + p / cmd + shift + p -> Install Package Control
2) Ctrl + shift + p / cmd + shift + p -> Install package
3) Wait for 5-10s another window will pop up, then you look for vue complete package
4) Go down right and look for Vue Component (from plain text)

Starter code



```
<template>
  <div id="app">

  </div>
</template>

<script>

export default {
  name: 'App',

}
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

Hello Vue! (template)

```
   <template>

  <div id="app">

    {{msg}}

  </div>

</template>
```

Hello Vue! (script)

```
<script>

  export default {

    name: 'App',

    data() {

      return {

        msg: "Hello World",

      }

    }

  }

</script>
```

Creating a vue instance

1) Start creating a vue instance as follows:

```
var app = new Vue({

   // options

 })
```

1- Create variables inside script

```
<script>
     export default {
          name: 'App',
          data() {
               return {
                    firstname : "Wan Muzaffar",
                    lastname  : "Wab Hashim",
                    htmlcontent : "<div><h1>Hello vue!</h1></div>"
      }
}
}
</script>
```

2 - Rendering with {{}} in html part

```
  <div id = "vue_det">

         <h1>Firstname : {{firstname}}</h1>

         <h1>Lastname : {{lastname}}</h1>

         <div>{{htmlcontent}}</div>

      </div>
```

Event Handling in Vuejs

```
<div id="example-1">

  <button v-on:click="counter += 1">Add 1</button>

  <p>The button above has been clicked {{ counter }} times.</p>

</div>

var example1 = new Vue({

  el: '#example-1',

  data: {

    counter: 0

  }

})
```

Event Handling with methods (1)

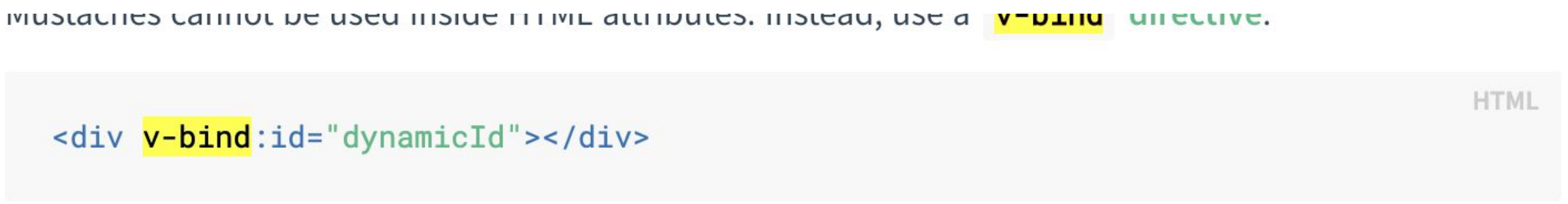
```
<div id="example-2">

  <!-- `greet` is the name of a method defined below -->

  <button v-on:click="greet">Greet</button>

</div>
```

Event handling with methods (2)

```
var example2 = new Vue({
      el: '#example-2',
      data: {
            name: 'Vue.js'
      },
      // define methods under the `methods` object
      methods: {
            greet: function (event) {
      // `this` inside methods points to the Vue instance
            alert('Hello ' + this.name + '!')
      // `event` is the native DOM event
            if (event) {
            alert(event.target.tagName)
            }
      }
      }
})
```

Binding variables to attributes using v-bind

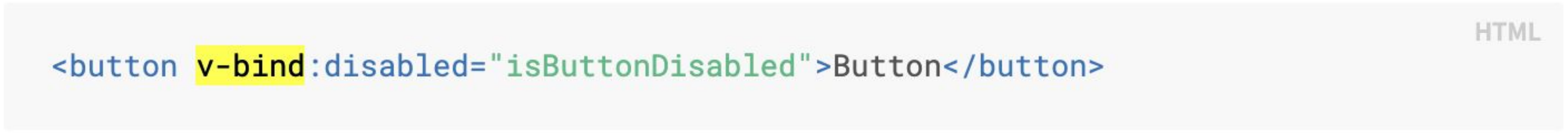
We use v-bind when we want to bind the attribute to the element



Mustaches cannot be used inside HTML attributes. Instead, use a v-bind directive.

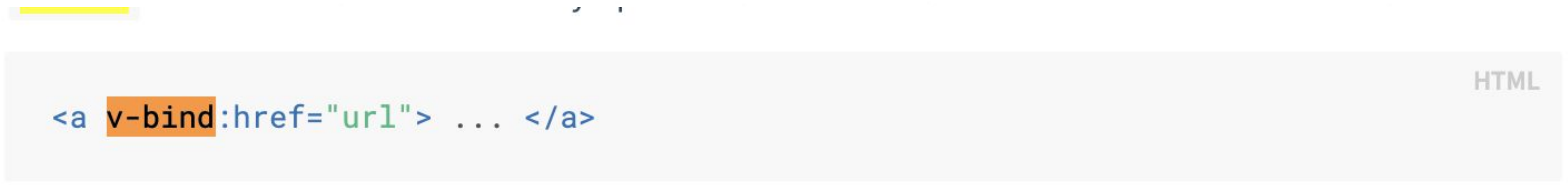
HTML

```html
<div v-bind:id="dynamicId"></div>
```



HTML

```html
<button v-bind:disabled="isButtonDisabled">Button</button>
```



HTML

```html
<a v-bind:href="url"> ... </a>
```

Conditional rendering - v-if

The directive `v-if` is used to conditionally render a block. The block will only be rendered if the directive's expression returns a truthy value.

```
<h1 v-if="awesome">Vue is awesome!</h1>
```

You may also add an else statement for the block:

```
<h1 v-if="awesome">Vue is awesome!</h1>

<h1 v-else>Oh no 😢</h1>
```

List rendering - v-for (1)

We can use the `v-for` directive to render a list of items based on an array.

```
<ul id="example-1">

  <li v-for="item in items" :key="item.message">

    {{ item.message }}

  </li>

</ul>
```

List rendering - v-for (2)

```
var example1 = new Vue({
  el: '#example-1',
  data: {
    items: [
      { message: 'Foo' },
      { message: 'Bar' }
    ]
  }
})
```

Data binding with v-model

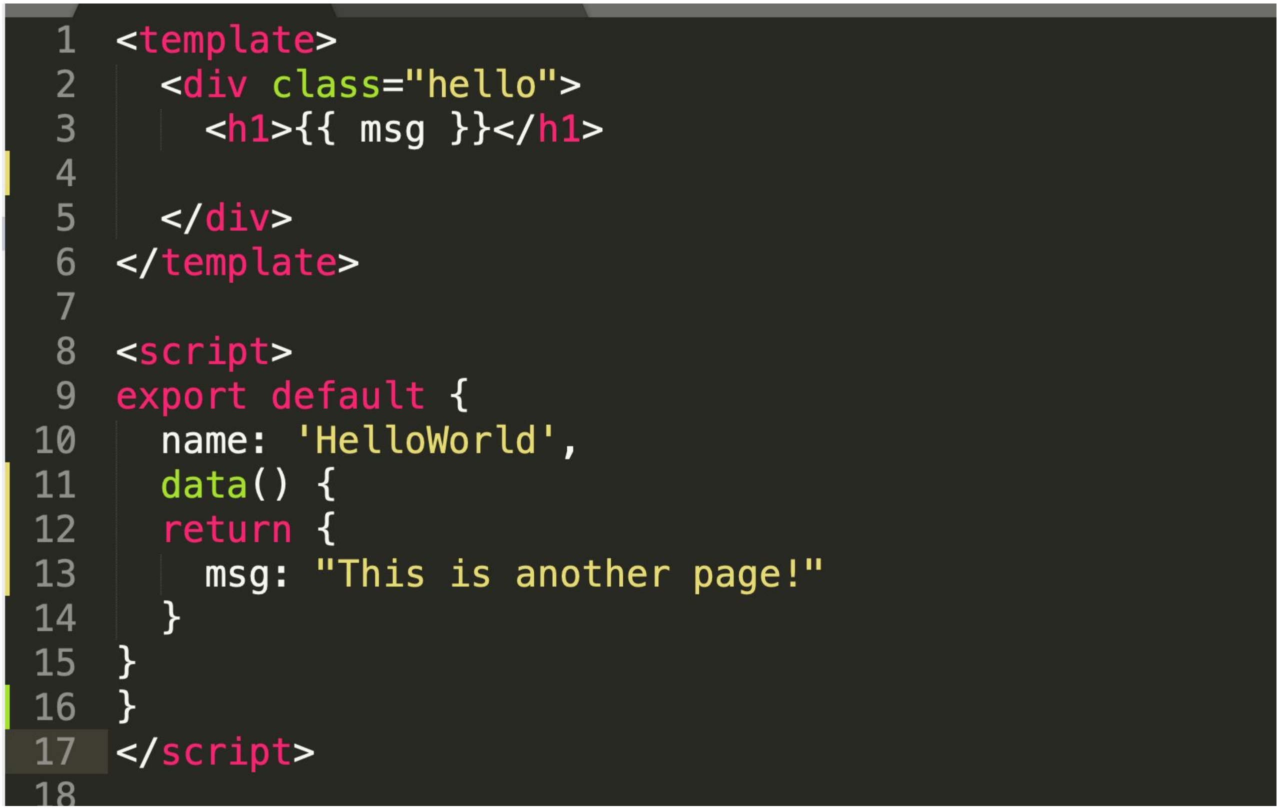
You can use the `v-model` directive to create two-way data bindings on form input, textarea, and select elements. It automatically picks the correct way to update the element based on the input type.

Example:

```
<input v-model="message" placeholder="edit me">
<p>Message is: {{ message }}</p>

<span>Multiline message is:</span>

<p style="white-space: pre-line;">{{ message }}</p>

<br>

<textarea v-model="message" placeholder="add multiple lines"></textarea>
```

`More example :` https://vuejs.org/v2/guide/forms.html

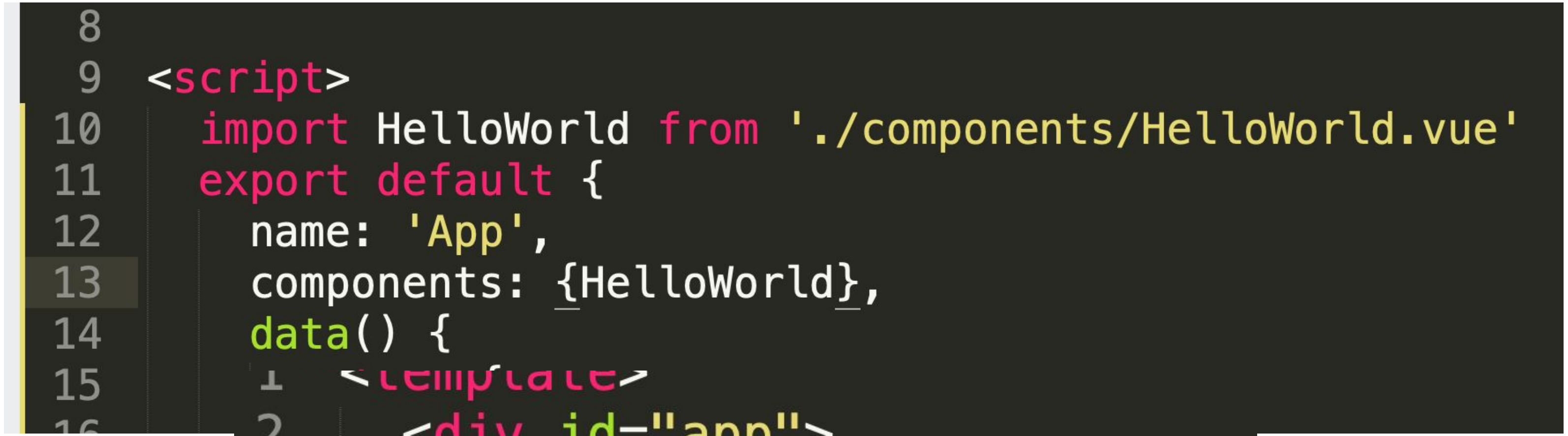
Segrating pages into multiple components (1)



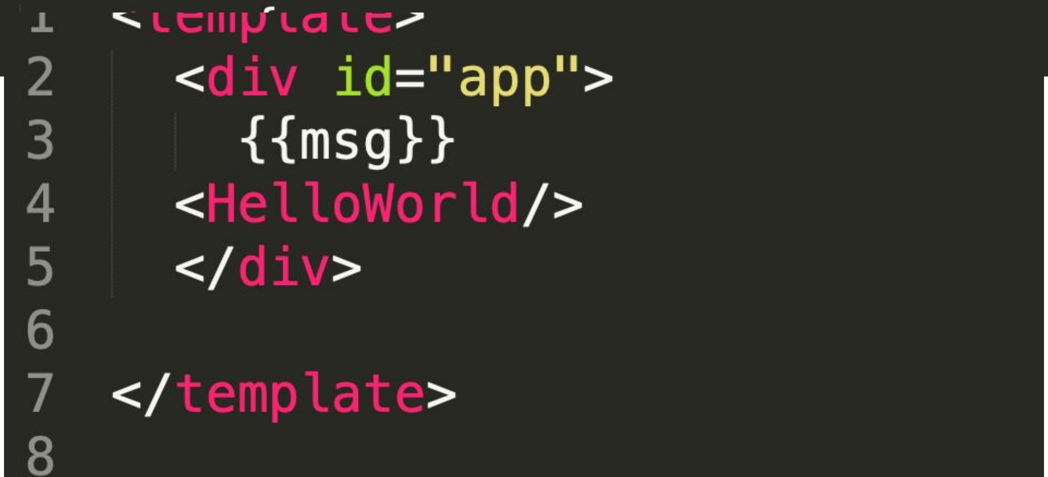
```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>

  </div>
</template>

<script>
export default {
  name: 'HelloWorld',
  data() {
  return {
    msg: "This is another page!"
  }
}
}
</script>
```

Inside new components, export the items (take attention on the exported name)

Import and call the components

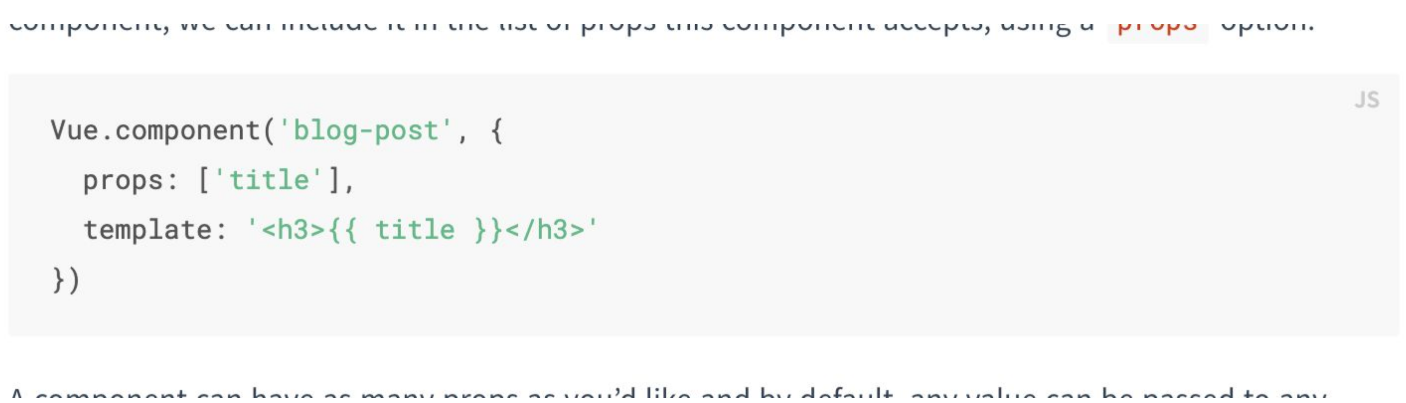


```
<script>
  import HelloWorld from './components/HelloWorld.vue'
  export default {
    name: 'App',
    components: {HelloWorld},
    data() {
```



```
<template>
  <div id="app">
    {{msg}}
  <HelloWorld/>
  </div>

</template>
```

Passing data in components (through props)

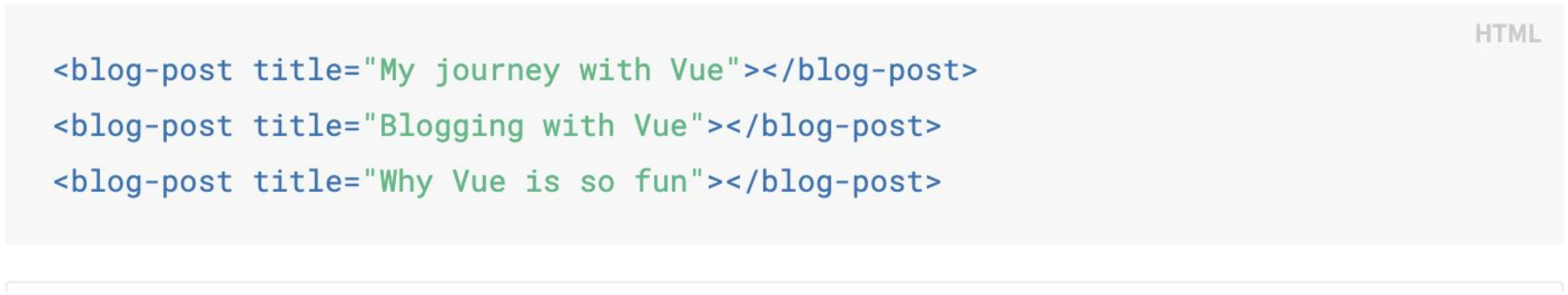


component, we can include it in the list of props this component accepts, using a `props` option.

JS

```js
Vue.component('blog-post', {
  props: ['title'],
  template: '<h3>{{ title }}</h3>'
})
```

A component can have as many props as you'd like and by default, any value can be passed to any



HTML

```html
<blog-post title="My journey with Vue"></blog-post>
<blog-post title="Blogging with Vue"></blog-post>
<blog-post title="Why Vue is so fun"></blog-post>
```

Vue router

Add vue router into your project using the following command line:

```
vue add router
```

Setting up routing

```
import Vue from 'vue'
import VueRouter from 'vue-router'


Vue.use(VueRouter)



const router = new VueRouter({
 mode: 'history',
 base: process.env.BASE_URL,
 routes
})


export default router
```

Create route inside

```
const routes = [
 {
   path: '/',
   name: 'Main',
   component: Main
 },
 {
   path: '/detail/:sendDate/:returnDate/:city',
   name: 'Detail',
   component: Detail
 },
 {
   path: '/about',
   name: 'About',
   component: () => import(/* webpackChunkName: "about" */ '../views/About.vue')
 },
 {
   path: '/car/:id',
   name:'Car',
   component: Car
```

Inside main.js modify as follow to include route

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'


Vue.config.productionTip = false


new Vue({
 router,
 render: h => h(App)
}).$mount('#app')
```

Retrieving data from params

```
sendData:this.$route.params.sendDate,
            returnDate:this.$route.params.returnDate,
            city:this.$route.params.city
```

Link to video

https://youtu.be/ovRR0dTppfE - 1st Day

https://youtu.be/FLJzYpXTwvE - 2nd Day

https://youtu.be/jRSNqWZZL3o - 3rd Day

https://youtu.be/ZRFi9umwBCc - 4th Day