Creating Vue.js Components and Using Template Syntax

Jim Cooper
SOFTWARE ENGINEER

@jimthecoop jcoop.io

Agenda

Creating Components

Using Bindings to Display Data and Handle Events

Conditionally Displaying Elements with v-if and v-show

Repeating Elements with v-for

Styling Components

Working with Component Lifecycle Hooks

Reducing Duplication with Mixins

Global Components

vs

Single-file Components

```
new Vue({
  template: '<div>Hello World</div>',
  el: '#app',
});
```

Components Are Vue Instances

```
new Vue({
  template: '<div>Hello World</div>',
  el: '#app',
});
```

Global Components

```
const World = Vue.component(
  'World',
  { template: '<span>World</span>' },
);

new Vue({
  template: '<div>Hello <World></div>',
  el: '#app',
});
```

Global Components

Global Components Limitations

Global Variables

String Templates

CSS is Not Encapsulated

No Build-time Compilation Support

HelloWorld.vue

```
<template>
</template>

<script>
</script>

<style>
</style>
```

Single-file Components

HelloWorld.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>

<script>
</script>

<style>
</style>
```

Single-file Components

HelloWorld.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>

<script>
  export default { name: 'HelloWorld'};
</script>


<style>
</style>
```

Single-file Components

HelloWorld.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>

<script>
  export default { name: 'HelloWorld'};
</script>

<style>
  .hello {
    color: red;
  }
</style>
```

Single-file Components

Summary

Creating Components

Using Bindings to Display Data and Handle Events

Conditionally Displaying Elements with v-if and v-show

Repeating Elements with v-for

Styling Components

Working with Component Lifecycle Hooks

Reducing Duplication with Mixins