

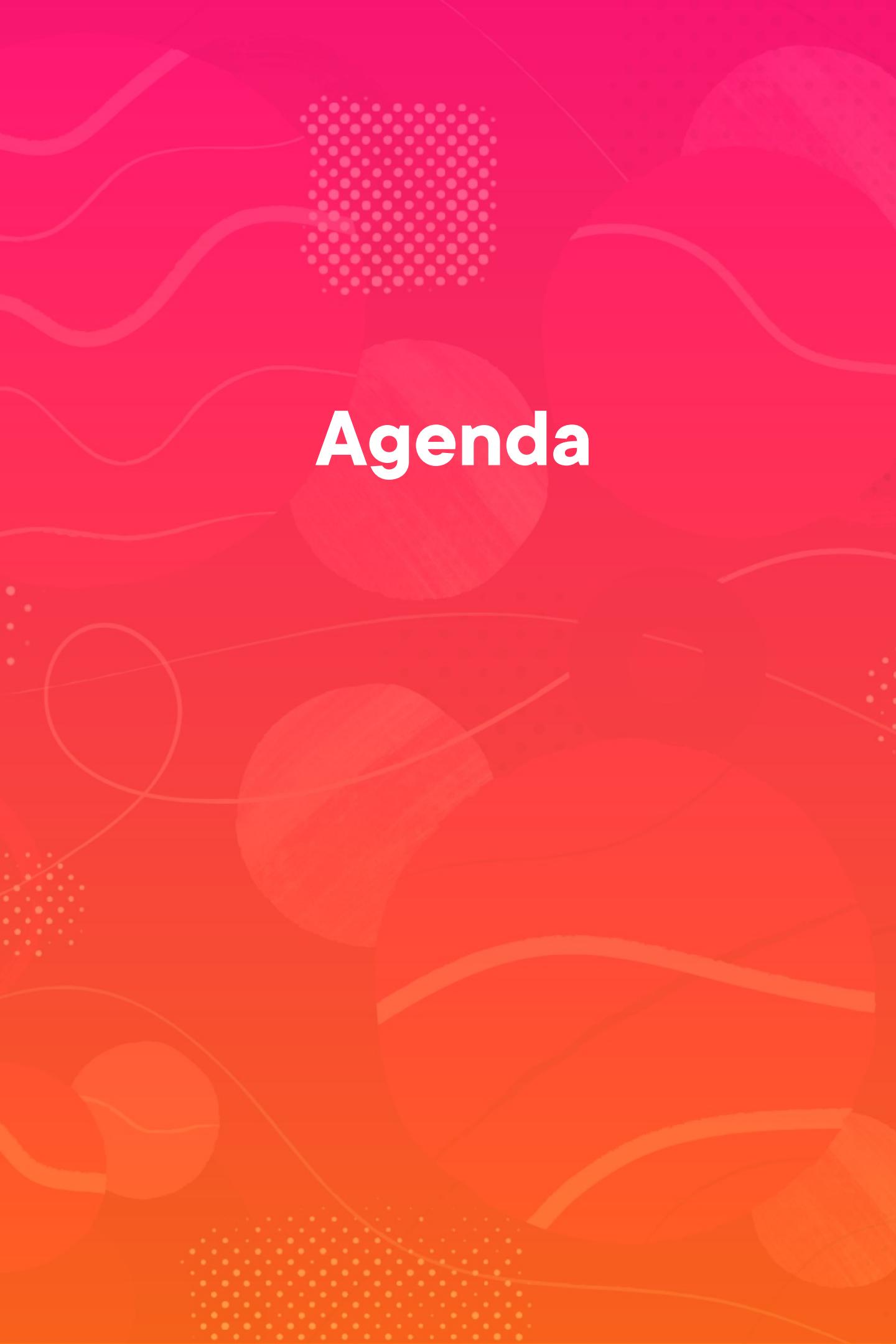
Using Vue Template Syntax



Jim Cooper

Software Engineer

@jimthecoop | jcoop.io



Agenda

- Binding data to element attributes with `v-bind`
- Binding to user events with `v-on`
- Using computed properties
- Displaying data with interpolation
- Conditionally displaying elements with `v-if` and `v-show`
- Repeating elements with `v-for`
- Working with component lifecycle hooks
- Reducing duplication with mixins



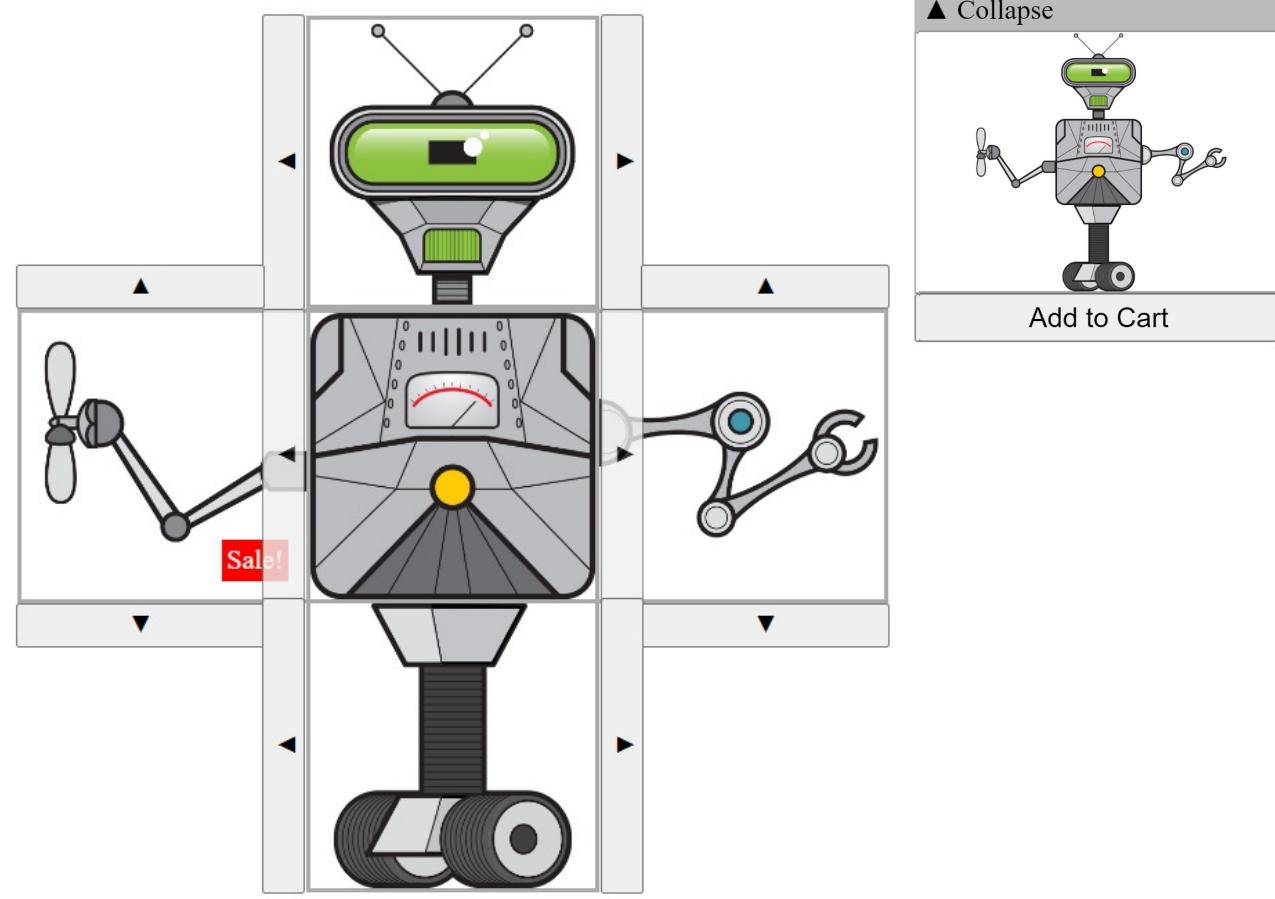


Build-a-Bot

Build

Cart 2

Build
Sidebar



Component Basics



HomePage.vue

```
<template>
</template>

<script>
</script>

<style>
</style>
```

Single-file Components



HomePage.vue

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

Single-file Components



HomePage.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>
<script>
  export default { name: 'HelloWorld' };
</script>
<style>
</style>
```

Single-file Components



HomePage.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>
<script>
  export default { name: 'HelloWorld' };
</script>
<style>
  .hello { color: red; }
</style>
```

Single-file Components



HomePage.vue

```
<template>
  <span class="hello">Hello World!</span>
</template>
<script>
  export default { name: 'HelloWorld' };
</script>
<style>
  .hello { color: red; }
</style>
```

Single-file Components



Options API

vs

Composition API



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

Options API vs Composition API



```
<script>
export default {
  name: RobotBuilder'
  created() {
    this.getParts();
  },
  data() {
    return {
      addedToCart: false,
      cart: [],
    }
  }
};
</script>
```

```
<script setup>
import { onCreated } from vue;

let addedToCart = false;
let cart = [];

onCreated(() => {
  this.getParts();
});

</script>
```

Options API vs Composition API



Local Components

vs

Global Components



```
</template>
...
<main>
  <HomePage />
</main>
</template>
```

```
<script>
import HomePage from './components/HomePage.vue';

export default {
  name: 'App',
  components: {
    HomePage,
  },
}
</script>
```

Local Components



```
import { createApp } from 'vue';
import App from './App.vue';

createApp(App).mount('#app');
```

Global Components



```
import { createApp } from 'vue';
import App from './App.vue';

import HomePage from './components/HomePage.vue';

const app = createApp(App).mount('#app');
app.component('HomePage', HomePage);
```

Global Components



```
</template>
...
<main>
  <HomePage />
</main>
</template>

<script>
import HomePage from './components/HomePage.vue';

export default {
  name: 'App',
  components: {
    HomePage,
  },
};
</script>
```

Global Components



```
</template>
...
<main>
  <HomePage />
</main>
</template>
```

```
<script>
  export default {
    name: 'App',
  };
</script>
```

Global Components



Global Components Drawbacks

Global Variables
**Increased Bundle
Sizes**



Summary

- Binding data to element attributes with v-bind
- Binding to user events with v-on
- Using computed properties
- Displaying data with interpolation
- Conditionally displaying elements with v-if and v-show
- Repeating elements with v-for
- Working with component lifecycle hooks
- Reducing duplication with mixins



Up Next:

Using the Composition and Reactivity APIs

