

Vue.js

Marek Jakimiuk



Introduction.

Statistics.

(March 13,
2019)

React.js

 Watch	6,646
 Star	124,269
 Fork	22,577

Vue.js

 Watch	5,780
 Star	130,533
 Fork	18,635

Angular.js

 Watch	4,244
 Star	59,432
 Fork	28,922

Statistics.

(February 13, 2019 – March 13, 2019)

React.js



Vue.js



Angular.js



Which big companies use Vue.js?

- GitLab
- Behance
- Livestorm
- Fox News
- Alibaba

Sources:

<https://www.monterail.com/state-of-vuejs-report>

<https://www.quora.com/Which-big-companies-use-Vue-js>

Why Vue.js?

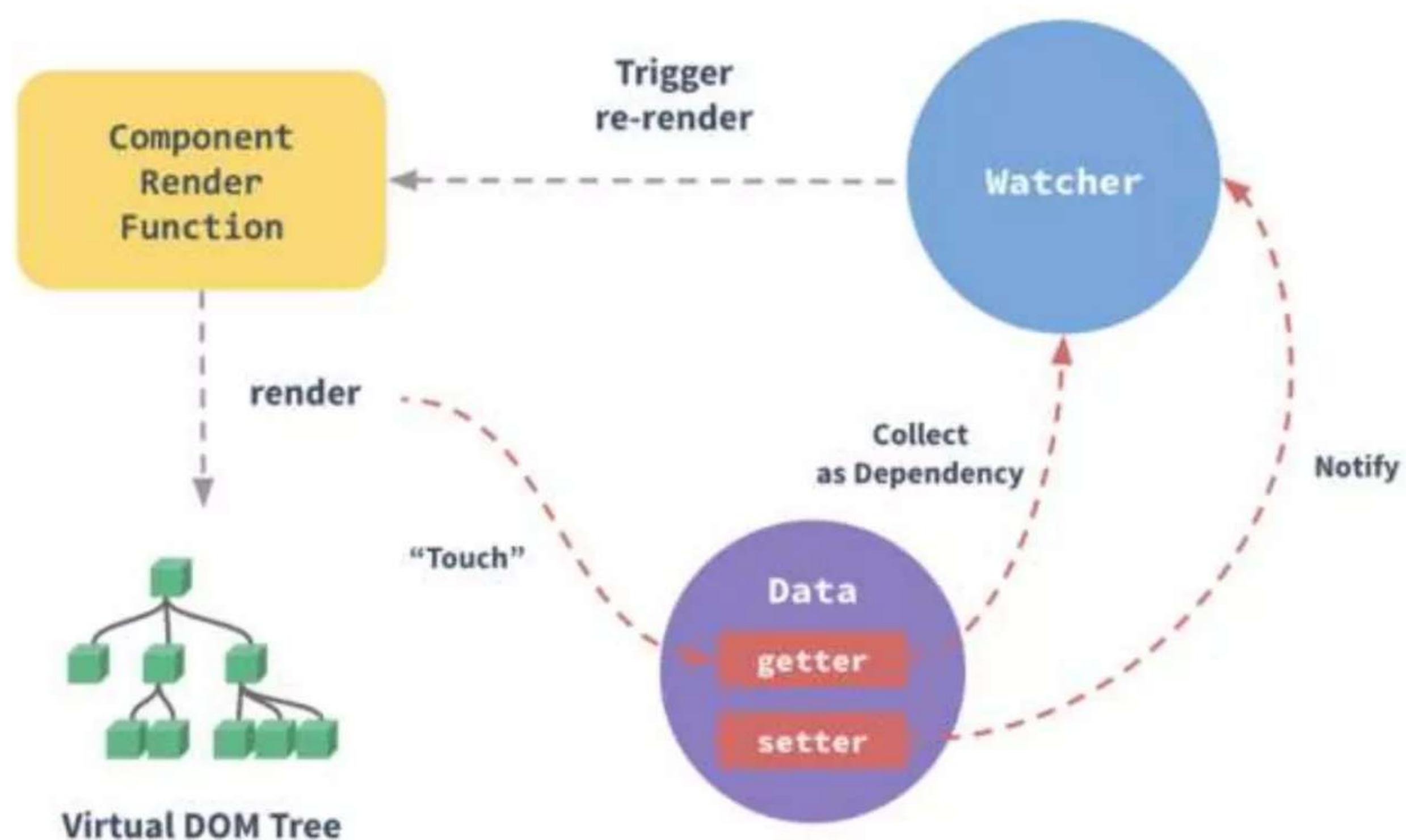
- Easy to start
- Vue template file component
- Virtual DOM
- Scoped css
- Built-in Transitions & Animation support
- Documentation
- Community
- Ecosystem

Ecosystem.

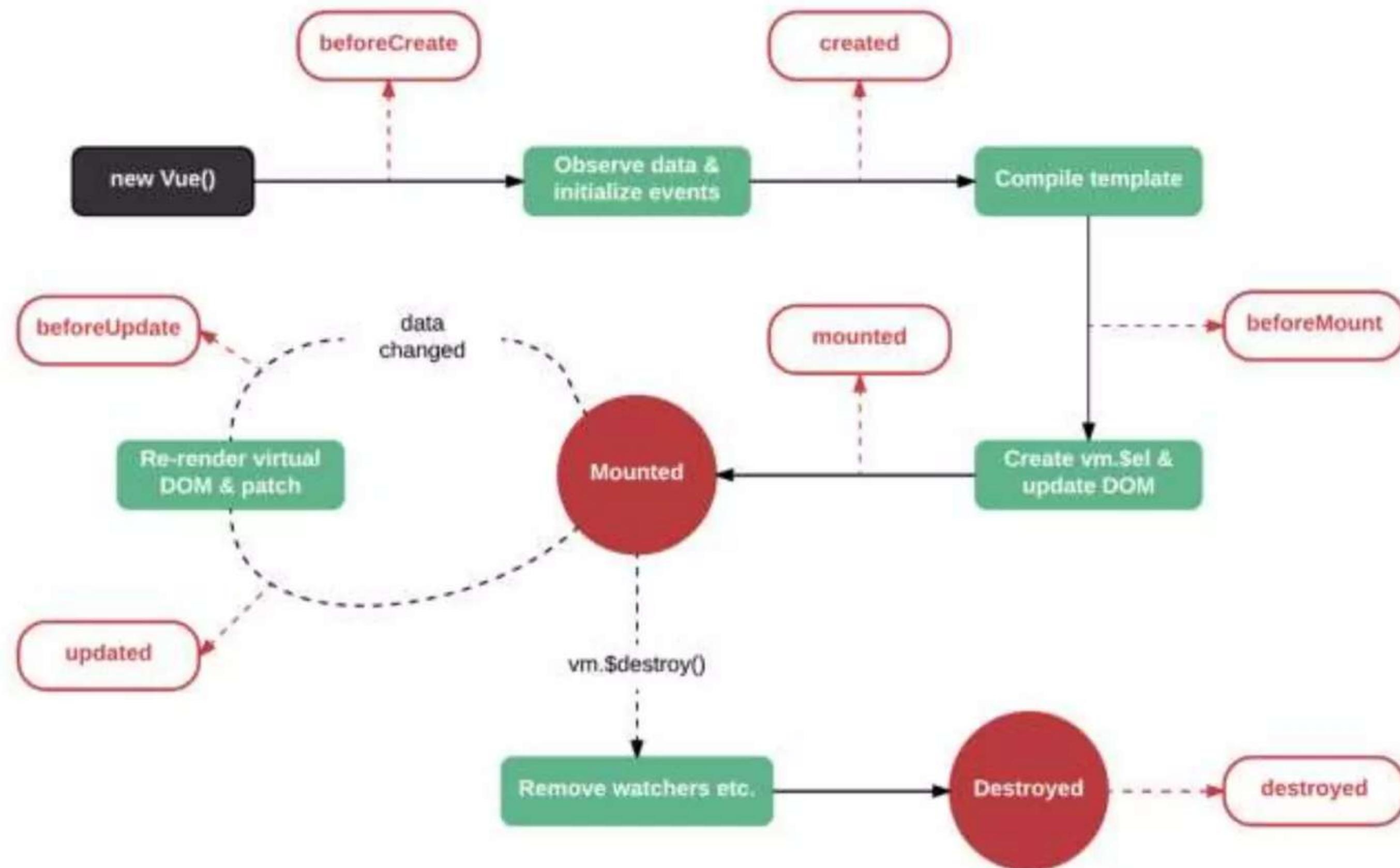
- vue-router - Single-page application routing
- vuex - Large-scale state management
- vue-cli - Project scaffolding
- vue-loader - Single File Component (*.vue file) loader for webpack
- vue-rx - RxJS integration
- vue-devtools - Browser DevTools extension
- vue-server-renderer - Server-side rendering support
- vue-class-component - TypeScript decorator for a class-based API

How it works?

- Inspired by Model-view-viewmodel (MVVM) architectural pattern.
- Dependency tracking system with getters and setters



Lifecycle & hooks



Enough Introduction !



Structure.

Vue component.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component VS Basic template.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js
App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://npmcdn.com/vue/dist/vue.js"></script>
  </head>
  <body>

    <div id="app">
      {{ msg }}
    </div>

    <script>
      new Vue({
        el: '#app',
        data: {
          msg: 'Hello World'
        }
      })
    </script>

  </body>
</html>
```

Vue component VS Basic template.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js
App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

~~<!DOCTYPE html>~~

~~<html>~~

~~<head>~~

~~<script src="https://npmcdn.com/vue/dist/vue.js"></script>~~

~~</head>~~

~~<body>~~

~~<div id="app">~~

~~{{ msg }}~~

~~</div>~~

~~<script>~~

~~new Vue({~~

~~el: '#app',~~

~~data: {~~

~~msg: 'Hello World'~~

~~}~~

~~})~~

~~</script>~~

~~</body>~~

~~</html>~~

Vue component.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component - view template.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js
App"/>
  </div>
</template>
<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component - script.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>
```

```
<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>
```

```
<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component - style.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>
```

```
<style scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component - style preprocessor.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style lang="scss" scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

```
npm install -D sass-loader node-sass
```

```
module.exports = {
  module: {
    rules: [
      // ... other rules omitted

      // this will apply to both plain '.scss' files
      // AND '<style lang="scss">' blocks in '.vue' files
      { test: /\.scss$/,
        use: [
          'vue-style-loader',
          'css-loader',
          'sass-loader'
        ]
      }
    ],
    // plugin omitted
  }
}
```

Vue component - style scoped.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style lang="scss" scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

Vue component - style scoped.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>
```

```
<style lang="scss" scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

```
<!-- Src -->
<template>
  <div class="example">hi</div>
</template>

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

Vue component - style scoped.

```
<!-- Basic Vue template - ./app.vue -->

<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style lang="scss" scoped>
  div {
    font-family: 'Avenir', Helvetica, Arial, sans-serif;
    color: #2c3e50;
  }
</style>
```

```
<!-- Src -->
<template>
  <div class="example">hi</div>
</template>

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

```
<!-- Out -->
<template>
  <div class="example" data-v-f3f3eg9>hi</div>
</template>

<style>
  .example[data-v-f3f3eg9] {
    color: red;
  }
</style>
```

Structure of Vue component.

```
export default {  
    name: 'RangeSlider',  
    components: {}, // when component uses other components  
    data() {}, // component variables  
    props: {}, // component properties  
    computed: {}, // component properties  
    watch: {}, // watch variables  
    filters: {}, // filters  
    methods: {} // methods  
  
    // Component Lifecycle hooks  
    beforeCreate() {},  
    created() {},  
    beforeMount() {},  
    mounted() {},  
    beforeUpdate() {},  
    updated() {},  
    activated() {},  
    deactivated() {},  
    beforeDestroy() {},  
    destroyed() {},  
    errorCaptured() {}  
};
```

Structure of Vue component.

```
export default {  
  name: 'RangeSlider',  
  components: {}, // when component uses other components  
  data() {}, // component variables  
  props: {}, // component properties  
  computed: {}, // component properties  
  watch: {}, // watch variables  
  filters: {}, // filters  
  methods: {} // methods
```

```
// Component Lifecycle hooks  
  beforeCreate() {},  
  created() {},  
  beforeMount() {},  
  mounted() {},  
  beforeUpdate() {},  
  updated() {},  
  activated() {},  
  deactivated() {},  
  beforeDestroy() {},  
  destroyed() {},  
  errorCaptured() {}  
};
```

Structure of Vue component.

```
export default {  
  name: 'RangeSlider',  
  components: {}, // when component uses other components  
  data() {}, // component variables  
  props: {}, // component properties  
  computed: {}, // component properties  
  watch: {}, // watch variables  
  filters: {}, // filters  
  methods: {} // methods
```

```
// Component Lifecycle hooks
```

```
  beforeCreate() {},  
  created() {},  
  beforeMount() {},  
  mounted() {},  
  beforeUpdate() {},  
  updated() {},  
  activated() {},  
  deactivated() {},  
  beforeDestroy() {},  
  destroyed() {},  
  errorCaptured() {}  
};
```

Interpolations.

Interpolations - Text.

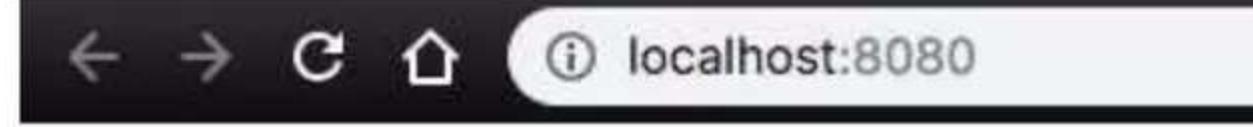
```
<template>
<div id="app">
  <p>Message: {{ msg }}</p>
  <p v-once>This will never change: {{ msg }}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      msg: 'Hello World!'
    }
  }
}
</script>
```

Interpolations - Text.

```
<template>
<div id="app">
  <p>Message: {{ msg }}</p>
  <p v-once>This will never change: {{ msg }}</p>
</div>
</template>
```

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      msg: 'Hello World!'
    }
  }
}
</script>
```



Message: Hello World!

This message will never change: Hello World!

Interpolations - About data.

data Must Be a Function

When we defined the `<button-counter>` component, you may have noticed that `data` wasn't directly provided an object, like this:

```
data: {  
  count: 0  
}
```

Instead, a component's `data` option must be a function, so that each instance can maintain an independent copy of the returned data object:

```
data: function () {  
  return {  
    count: 0  
  }  
}
```

If Vue didn't have this rule, clicking on one button would affect the data of *all other instances*, like below:

You clicked me 0 times. You clicked me 0 times. You clicked me 0 times.

Interpolations - Raw HTML.

```
<template>
<div id="app">
  <p v-html="msg"></p>
  <p v-once>This message will never change: {{ msg
}}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      msg: '<h1>Hello <strong>World</strong></h1>'
    }
  }
}
</script>
```

Interpolations - Raw HTML.

```
<template>
<div id="app">
  <p v-html="msg"></p>
  <p v-once>This message will never change: {{ msg }}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      msg: '<h1>Hello <strong>World</strong></h1>'
    }
  }
}
</script>
```



Hello World

This message will never change: <h1>Hello World</h1>

Interpolations - Attributes.

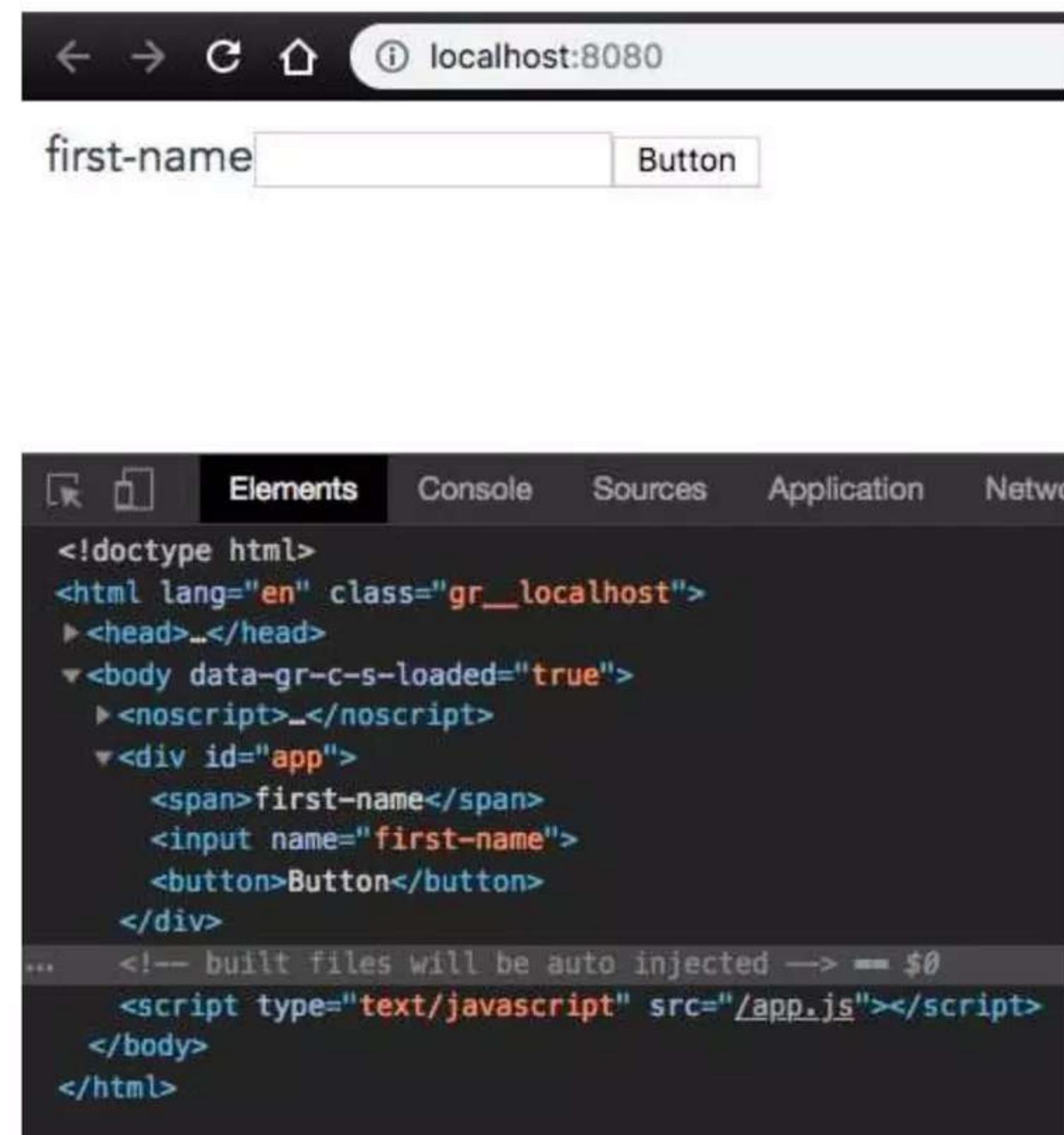
```
<template>
<div id="app">
  <span>{{ dynamicName }}</span>
  <input v-bind:name="dynamicName"/>
  <button v-bind:disabled="isButtonDisabled">Button</button>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      isButtonDisabled: false,
      dynamicName: 'first-name'
    }
  }
}
</script>
```

Interpolations - Attributes.

```
<template>
<div id="app">
  <span>{{ dynamicName }}</span>
  <input v-bind:name="dynamicName"/>
  <button v-bind:disabled="isButtonDisabled">Button</button>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      isButtonDisabled: false,
      dynamicName: 'first-name'
    }
  }
}
</script>
```



The screenshot shows a browser window with the URL `localhost:8080`. In the search bar, the text "first-name" is entered. Below the search bar is a button labeled "Button". The browser's developer tools are open, specifically the "Elements" tab, which displays the rendered HTML of the application. The rendered HTML includes a `<div id="app">` element containing a `first-name`, an `<input name="first-name">`, and a `<button>Button</button>`. The developer tools also show the original script code at the bottom of the tree view.

```
<!doctype html>
<html lang="en" class="gr__localhost">
  <head>...
  <body data-gr-c-s-loaded="true">
    <noscript>...
    <div id="app">
      <span>first-name</span>
      <input name="first-name">
      <button>Button</button>
    </div>
    ...
    <!-- built files will be auto injected --> == $0
    <script type="text/javascript" src="/app.js"></script>
  </body>
</html>
```

Interpolations - JavaScript expression.

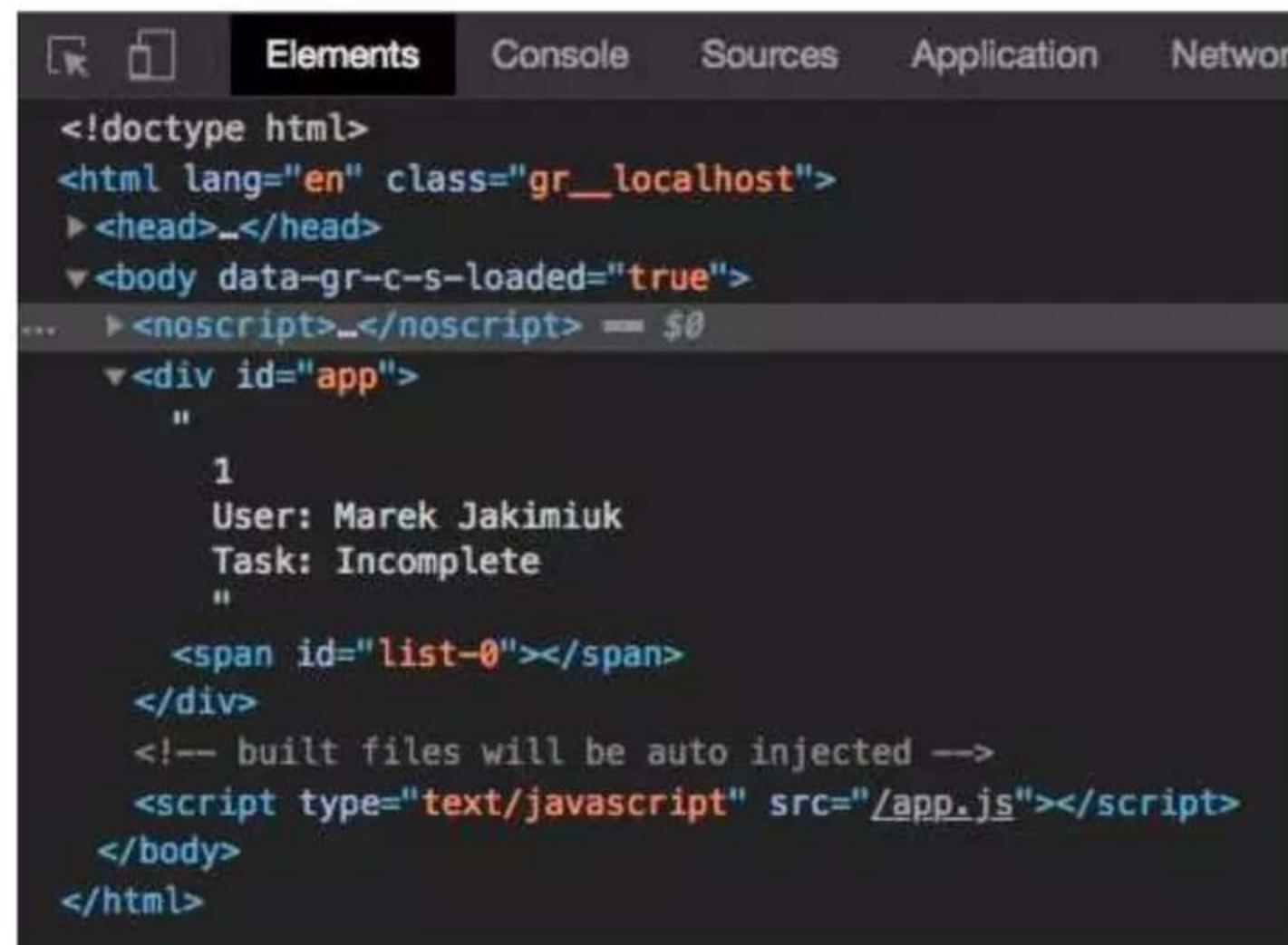
```
<template>
<div id="app">
  {{ index + 1 }}
  User: {{ user.toLowerCase()
    .split(' ')
    .map((s) => s.charAt(0).toUpperCase() + s.substring(1))
    .join(' '))
  }}
  Task: {{ done ? 'Completed' : 'Incomplete' }}
  <span v-bind:id="list-' + index"></span>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      index: 0,
      done: false,
      user: 'marek jakimiuk'
    }
  }
}
</script>
```

Interpolations - JavaScript expression.

```
<template>
<div id="app">
  {{ index + 1 }}
  User: {{ user.toLowerCase()
    .split(' ')
    .map((s) => s.charAt(0).toUpperCase() + s.substring(1))
    .join(' '))
  }}
  Task: {{ done ? 'Completed' : 'Incomplete' }}
  <span v-bind:id="list-' + index"></span>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      index: 0,
      done: false,
      user: 'marek jakimiuk'
    }
  }
}
</script>
```



Directives.

Directives - V-BIND.

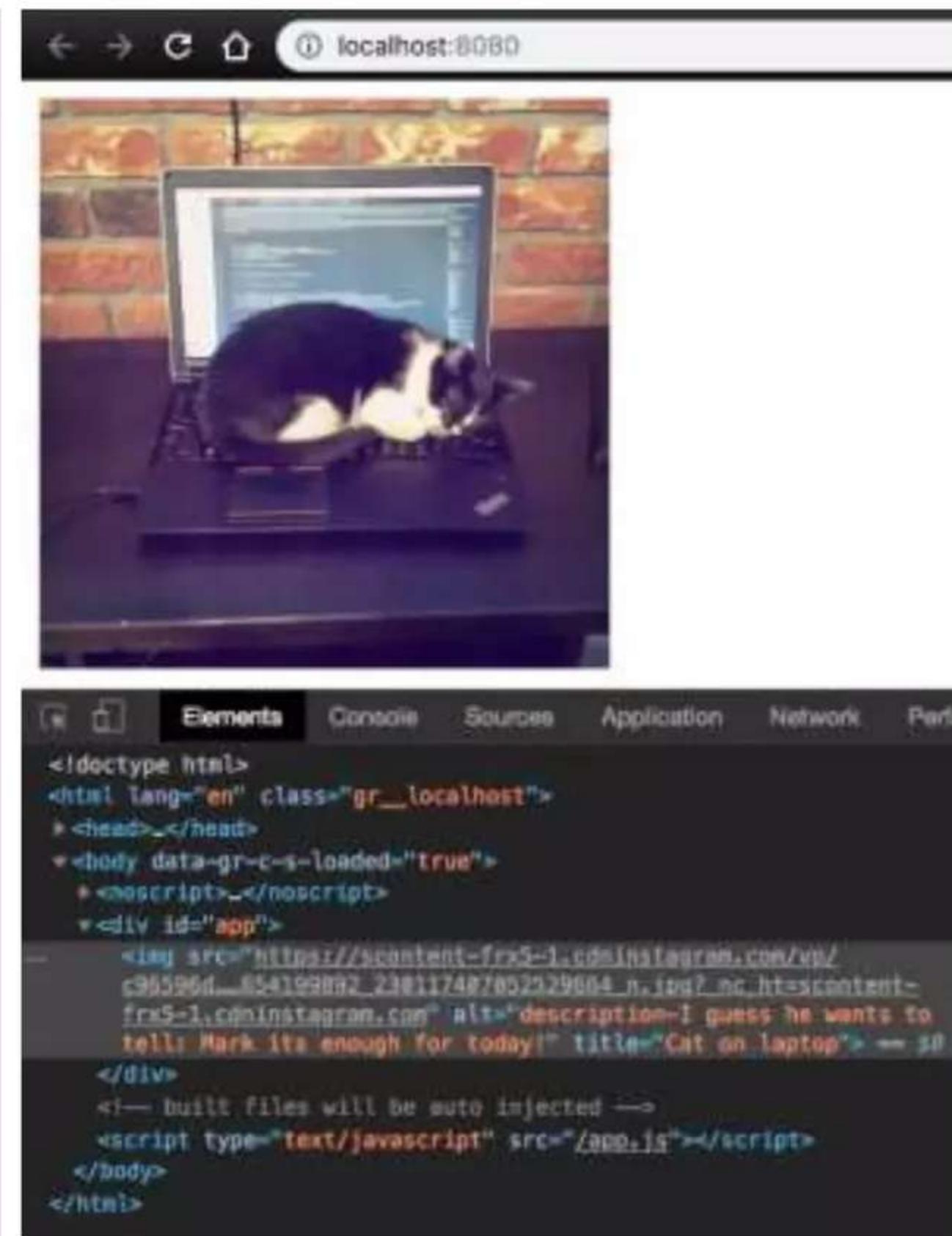
```
<template>
<div id="app">
  
  <!-- shorthand - 
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      title: 'Cat on laptop',
      alt: 'I guess he wants to tell: Mark its enough for today!',
      src: 'https://scontent-frx5-
1.cdninstagram.com/vp/c96596da302ef2bfc8351a70180a32bd/5D
29AE53/t51.2885-
15/sh0.08/e35/s640x640/15035741_1827043654199892_23011
7407052529664_n.jpg?_nc_ht=scontent-frx5-1.cdninstagram.com'
    }
  }
}
</script>
```

Directives - V-BIND.

```
<template>
<div id="app">
  
  <!-- shorthand - 
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      title: 'Cat on laptop',
      alt: 'I guess he wants to tell: Mark its enough for today!',
      src: 'https://scontent-frx5-
1.cdninstagram.com/vp/c96596da302ef2bfc8351a70180a32bd/5D
29AE53/t51.2885-
15/sh0.08/e35/s640x640/15035741_1827043654199892_23011
7407052529664_n.jpg?_nc_ht=scontent-frx5-1.cdninstagram.com'
    }
  }
}
</script>
```



Directives - V-ON - Component data.

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

  <p>Counter : {{ counter }}</p>

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

  <!-- shorthand -->
  <button @click="counter -= 1">Subtract</button>

</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      counter: 0
    }
  }
}
</script>
```

Counter : 0

Add Subtract

Counter : 5

Add Subtract

Counter : 2

Add Subtract

Directives - V-ON - Methods.

```
<template>
<div id="app">
  <p>Counter : {{ counter }}</p>
  <!-- full syntax -->
  <button v-on:click="counter += 1">Add</button>
  <!-- shorthand -->
  <button @click="counter -= 1">Subtract</button>

  <button @click="reset">Reset</button>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      counter: 0
    }
  },
  methods: {
    reset: function () {
      this.counter = 0
    }
  }
}
</script>
```

Directives - V-ON - Click and key events.

Event	Modifiers
v-on:click .native	
v-on:click .stop	
v-on:click .prevent	
v-on:click .passive	2.3
v-on:click .capture	
v-on:click .self	
v-on:click .once	2.1.4

Mouse	Modifiers
v-on:click .left	2.2
v-on:click .right	2.2
v-on:click .middle	2.2

Keyboard	Modifiers
v-on:keyup .ctrl	2.1
v-on:keyup .alt	2.1
v-on:keyup .shift	2.1
v-on:keyup .meta	2.1
v-on:keyup .enter	
v-on:keyup .tab	
v-on:keyup .delete	
v-on:keyup .esc	
v-on:keyup .up	
v-on:keyup .down	
v-on:keyup .left	
v-on:keyup .right	

Directives - V-ON - Event modifiers.

```
<template>
<div id="app">
  <form v-on:submit.prevent="onSubmit">
    <input type="text" name="first-name"/>
    <input type="text" name="last-name"/>
    <button type="submit">Send</button>
  </form>
</div>
</template>
```

```
<script>
export default {
  name: 'app',
  methods: {
    onSubmit: function () {
      // for example - some validate instructions
    }
  }
}
</script>
```

Watchers, Filters & Computed Properties.

Watchers.

Watchers.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      fullName: 'Marek Jakimiuk'
    }
  },
  watch: {
    firstName: function (val) {
      this.fullName = val + ' ' + this.lastName
    },
    lastName: function (val) {
      this.fullName = this.firstName + ' ' + val
    }
  }
}
</script>
```

Watchers.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      fullName: 'Marek Jakimiuk'
    }
  },
  watch: {
    firstName: function (val) {
      this.fullName = val + ' ' + this.lastName
    },
    lastName: function (val) {
      this.fullName = this.firstName + ' ' + val
    }
  }
}
</script>
```

Watchers.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      fullName: 'Marek Jakimiuk'
    }
  },
  watch: {
    firstName: function (val) {
      this.fullName = val + ' ' + this.lastName
    },
    lastName: function (val) {
      this.fullName = this.firstName + ' ' + val
    }
  }
}
</script>
```

```
<template>
  <div id="app">
    {{ firstName }}
    {{ lastName }}
    {{ fullName }}
  </div>
</template>
```



Computed Properties.

Computed properties.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      score: [2, 3, 4, 5]
    }
  },
  computed: {
    fullName: function () {
      return this.firstName + ' ' + this.lastName
    },
    userScore: function () {
      return this.score.reduce(function (prevValue, currentValue) {
        return prevValue + currentValue
      }, 0)
    }
  }
}
</script>
```

Computed properties.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      score: [2, 3, 4, 5]
    }
  },
  computed: {
    fullName: function () {
      return this.firstName + ' ' + this.lastName
    },
    userScore: function () {
      return this.score.reduce(function (prevValue, currentValue) {
        return prevValue + currentValue
      }, 0)
    }
  }
}
</script>
```

Computed properties.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      score: [2, 3, 4, 5]
    }
  },
  computed: {
    fullName: function () {
      return this.firstName + ' ' + this.lastName
    },
    userScore: function () {
      return this.score.reduce(function (prevValue, currentValue) {
        return prevValue + currentValue
      }, 0)
    }
  }
}</script>
```

Computed properties.

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      firstName: 'Marek',
      lastName: 'Jakimiuk',
      score: [2, 3, 4, 5]
    },
    computed: {
      fullName: function () {
        return this.firstName + ' ' + this.lastName
      },
      userScore: function () {
        return this.score.reduce(function (prevValue, currentValue) {
          return prevValue + currentValue
        }, 0)
      }
    }
  }
}</script>
```

```
<template>
<div id="app">
  {{ fullName }}
  {{ userScore }}
</div>
</template>
```

Filters.

Filters.

```
<template>
<div id="app">
  <p>{{ message }}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      message: 'Example how vue filter works.'
    }
  },
  filters: {
    capitalize: function (value) {
      if (!value) return ""
      value = value.toString()
      return value.toUpperCase()
    }
  }
}
</script>
```

Filters.

```
<template>
<div id="app">
  <p>{{ message }}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      message: 'Example how vue filter works.'
    }
  },
  filters: {
    capitalize: function (value) {
      if (!value) return ''
      value = value.toString()
      return value.toUpperCase()
    }
  }
}
</script>
```

Filters.

```
<template>
<div id="app">
  <p>{{ message | capitalize }}</p>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      message: 'Example how vue filter works.'
    }
  },
  filters: {
    capitalize: function (value) {
      if (!value) return ''
      value = value.toString()
      return value.toUpperCase()
    }
  }
}
</script>
```

Methods.

Methods.

```
<template>
  <div id="app">
    <p> Bulb state : {{ bulb ? 'ON' : 'OFF' }} </p>
    <button v-on:click="toggle"> Toggle </button>
  </div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      bulb: true
    }
  },
  methods: {
    toggle: function () {
      console.log('E.g. Event : ', event.currentTarget)
      this.bulb = !this.bulb
    }
  }
}
</script>
```

Methods.

```
<template>
  <div id="app">
    <p> Bulb state : {{ bulb ? 'ON' : 'OFF' }} </p>
    <button v-on:click="toggle"> Toggle </button>
  </div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      bulb: true
    }
  },
  methods: {
    toggle: function () {
      console.log('E.g. Event : ', event.currentTarget)
      this.bulb = !this.bulb
    }
  }
}
</script>
```

Methods.

```
<template>
  <div id="app">
    <p> Bulb state : {{ bulb ? 'ON' : 'OFF' }} </p>
    <button v-on:click="toggle"> Toggle </button>
  </div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      bulb: true
    }
  },
  methods: {
    toggle: function () {
      console.log('E.g. Event : ', event.currentTarget)
      this.bulb = !this.bulb
    }
  }
}
</script>
```

Form Input Bindings.

Two-way Data Bindings.

Binding - Input text.

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

<script>
export default {
  name: 'app',
  data: function () {
    return {
      message: ""
    }
  }
}
</script>
```

Binding - Input text.

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

```
<script>
export default {
  name: 'app',
  data: function () {
    return {
      message: ''
    }
  }
}
</script>
```

edit me

Message is:

Binding - Checkbox.

```
<template>
<div id="app">
  <input type="checkbox" id="checkbox" v-model="checked">
  <label for="checkbox">{{ checked }}</label>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      checked: false
    }
  }
}
</script>
```

Binding - Checkbox.

```
<template>
  <input type="checkbox" id="checkbox" v-model="checked">
  <label for="checkbox">{{ checked }}</label>
  <label for="checkbox">{{ checked }}</label>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      checked: false
    }
  }
}
</script>
```

false

Binding - Multi Checkbox.

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

    <input type="checkbox" id="jack" value="Jack" v-model="checkedNames">
    <label for="jack">Jack</label>

    <input type="checkbox" id="john" value="John" v-model="checkedNames">
    <label for="john">John</label>

    <input type="checkbox" id="mike" value="Mike" v-model="checkedNames">
    <label for="mike">Mike</label>

    <br>
    <span>Checked names: {{ checkedNames }}</span>
  </div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      checkedNames: []
    }
  }
}
</script>
```

Jack John Mike
Checked names: []

Binding - Multi Checkbox.

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

    <input type="checkbox" id="jack" value="Jack" v-model="checkedNames">
    <label for="jack">Jack</label>

    <input type="checkbox" id="john" value="John" v-model="checkedNames">
    <label for="john">John</label>

    <input type="checkbox" id="mike" value="Mike" v-model="checkedNames">
    <label for="mike">Mike</label>

  <br>
  <span>Checked names: {{ checkedNames }}</span>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      checkedNames: []
    }
  }
}
</script>
```

Jack John Mike
Checked names: []

Binding - Select.

```
<template>
<div id="app">
<select v-model="selected">
  <option disabled value="">Please select one</option>
  <option>A</option>
  <option>B</option>
  <option>C</option>
</select>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      selected: ""
    }
  }
}
</script>
```

Please select one

Class Binding.

Class Binding (Passing Object).

Class Binding - Passing Object.

```
<template>
  <div id="app">
    <div v-bind:class="{ active: isActive }"></div>
  </div>
</template>

<script>
export default {
  data() {
    return {
      isActive: true,
      hasError: false
    }
  }
}
</script>
```

```
▼<div id="app"> == $0
  <div class="active"></div>
  </div>
```

Class Binding - Passing Object.

```
<template>
<div id="app">
<div class="static"
  v-bind:class="{ active: isActive, 'text-danger': hasError }">
  <!-- Content -->
</div>
</div>
</template>

<script>
export default {
  data() {
    return {
      isActive: true,
      hasError: false
    }
  }
}
</script>
```

```
▼<div id="app">
  <div class="static active"></div> == $0
  | </div>
```

Class Binding (Passing Array).

Class Binding - Passing Array.

```
<template>
<div id="app">
  <div v-bind:class="[activeClass, errorClass]">
    <!-- Content -->
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeClass: 'active',
      errorClass: 'text-danger'
    }
  }
}
</script>
```

```
▼<div id="app">
  <div class="active text-danger"></div> == $0
  </div>
```

Class Binding - Passing Array.

```
<template>
<div id="app">
  <div v-bind:class="[activeClass, errorClass]">
    <!-- Content -->
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeClass: 'active',
      errorClass: 'text-danger'
    }
  }
}
</script>
```

```
▼<div id="app">
  <div class="active text-danger"></div> == $0
</div>
```

Style Binding.

Style Binding.

```
<template>
<div id="app">
  <div v-bind:style="{ color: activeColor, fontSize: fontSize + 'px' }">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeColor: 'red',
      fontSize: 30
    }
  }
}
</script>
```

Style Binding.

```
<template>
<div id="app">
  <div v-bind:style="{ color: activeColor, fontSize: fontSize + 'px' }">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeColor: 'red',
      fontSize: 30
    }
  }
}
</script>
```

Lorem ipsum dolor sit amet
consectetur adipisicing elit.

Style Binding.

```
<template>
<div id="app">
  <div v-bind:style="{ color: activeColor, fontSize: fontSize + 'px' }">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeColor: 'red',
      fontSize: 30
    }
  }
}
</script>
```

Lore*um ipsum dolor sit amet
consectetur adipisicing elit.*

▼<div id="app">
 <div style="color: red; font-size: 30px;">
 Lore*um ipsum dolor sit amet consectetur
adipisicing elit.*
 </div> == \$0
</div>

Style Binding.

```
<template>
<div id="app">
  <button @click="fontSize += 1">+</button>
  <button @click="fontSize -= 1">-</button>
  <div v-bind:style="{ color: activeColor, fontSize: fontSize + 'px' }">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
</template>

<script>
export default {
  data() {
    return {
      activeColor: 'red',
      fontSize: 30
    }
  }
}
</script>
```

+ -

Lorem ipsum dolor sit amet
consectetur adipisicing elit.

Conditional Rendering.

Conditional Rendering - if.

```
<template>
<div id="app">
  <h1 v-if="awesome">
    😊 Vue is awesome!
  </h1>
</div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      awesome: true
    }
  }
}
</script>
```

← → ⌂ ⌄ 127.0.0.1:8080

😍 **Vue is awesome!**

Conditional Rendering - else

```
<template>
  <div id="app">
    <h1 v-if="awesome">
      😊 Vue is awesome!
    </h1>
    <h1 v-else>
      😢 It's impossible!
    </h1>
  </div>
</template>

<script>
export default {
  name: 'app',
  data: function () {
    return {
      awesome: false
    }
  }
}
</script>
```

← → ⌂ ⌄ 127.0.0.1:8080

😍 **Vue is awesome!**

← → ⌂ ⌄ 127.0.0.1:8080

😢 **It's impossible!**

Conditional Rendering - else-if.

```
<template>
  <div id="app">
    <div v-if="type === 'A'">
      A
    </div>
    <div v-else-if="type === 'B'">
      B
    </div>
    <div v-else-if="type === 'C'">
      C
    </div>
    <div v-else>
      Not A/B/C
    </div>
  </div>
</template>
```

Conditional Rendering - else-if.

```
<template v-if="loginType === 'username'>  
  <label>Username</label>  
  <input placeholder="Enter your username" key="username-input">  
</template>  
<template v-else>  
  <label>Email</label>  
  <input placeholder="Enter your email address" key="email-input">  
</template>
```

Username

[Toggle login type](#)

Transitions & Animations.

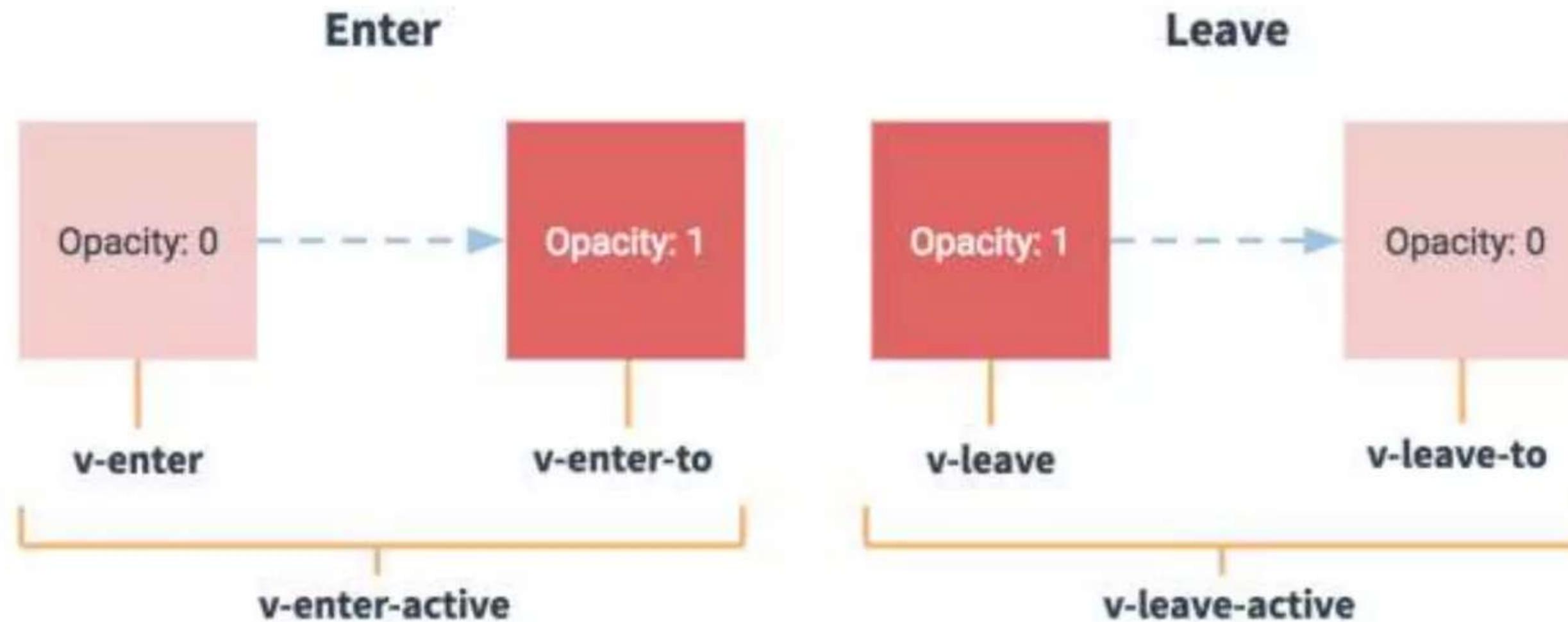
Transitions & Animations.

```
<template>
<div id="app">
  <button v-on:click="show = !show">
    Toggle
  </button>
  <p v-if="show">hello</p>
</div>
</template>

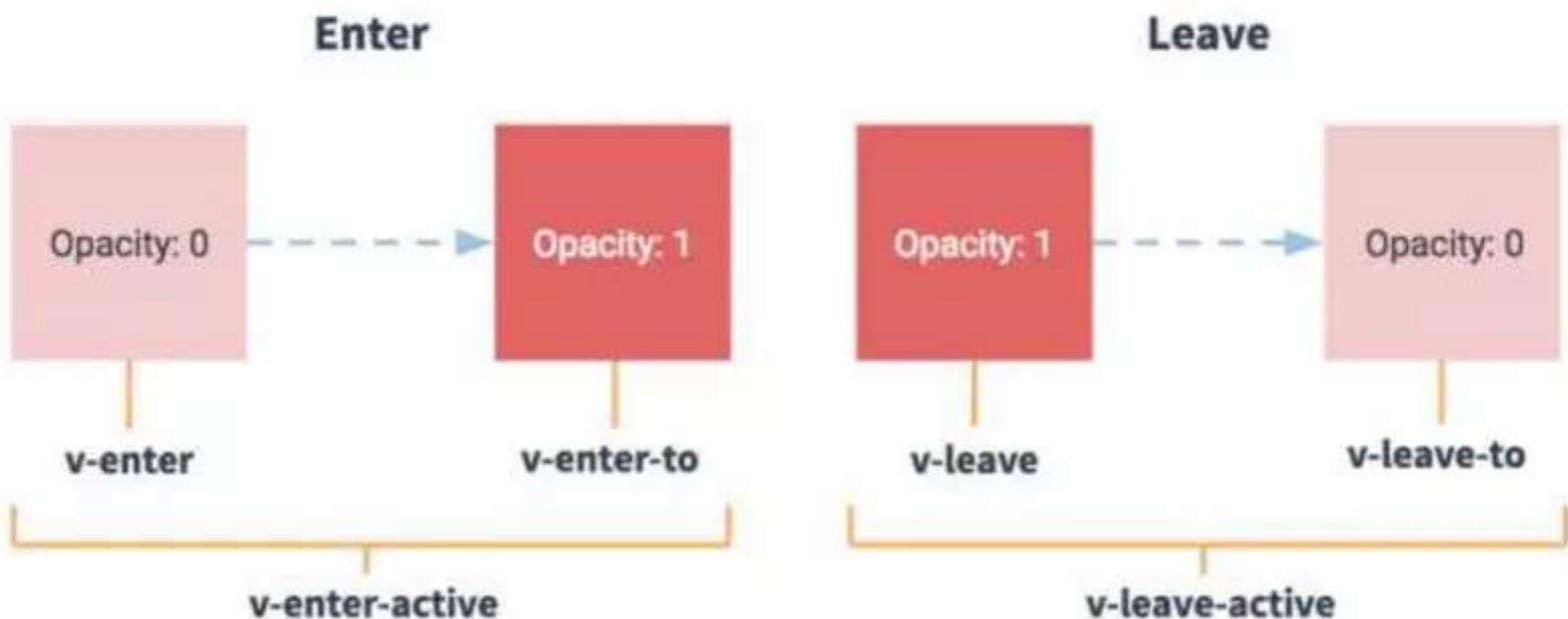
<script>
export default {
  data() {
    return {
      show: true
    }
  }
}
</script>
```



Enter/Leave & List Transitions.



Enter/Leave & List Transitions.



Toggle

```
▼<div id="app">
  <button>
    Toggle
  </button>
  <h1>Hello!</h1>
  ...
  </div> == $0
</div>
```

Transitions & Animations.

```
<template>
<div id="app">
  <button v-on:click="show = !show">
    Toggle
  </button>
  <transition name="fade">
    <p v-if="show">hello</p>
  </transition>
</div>
</template>

/* ...script */
```

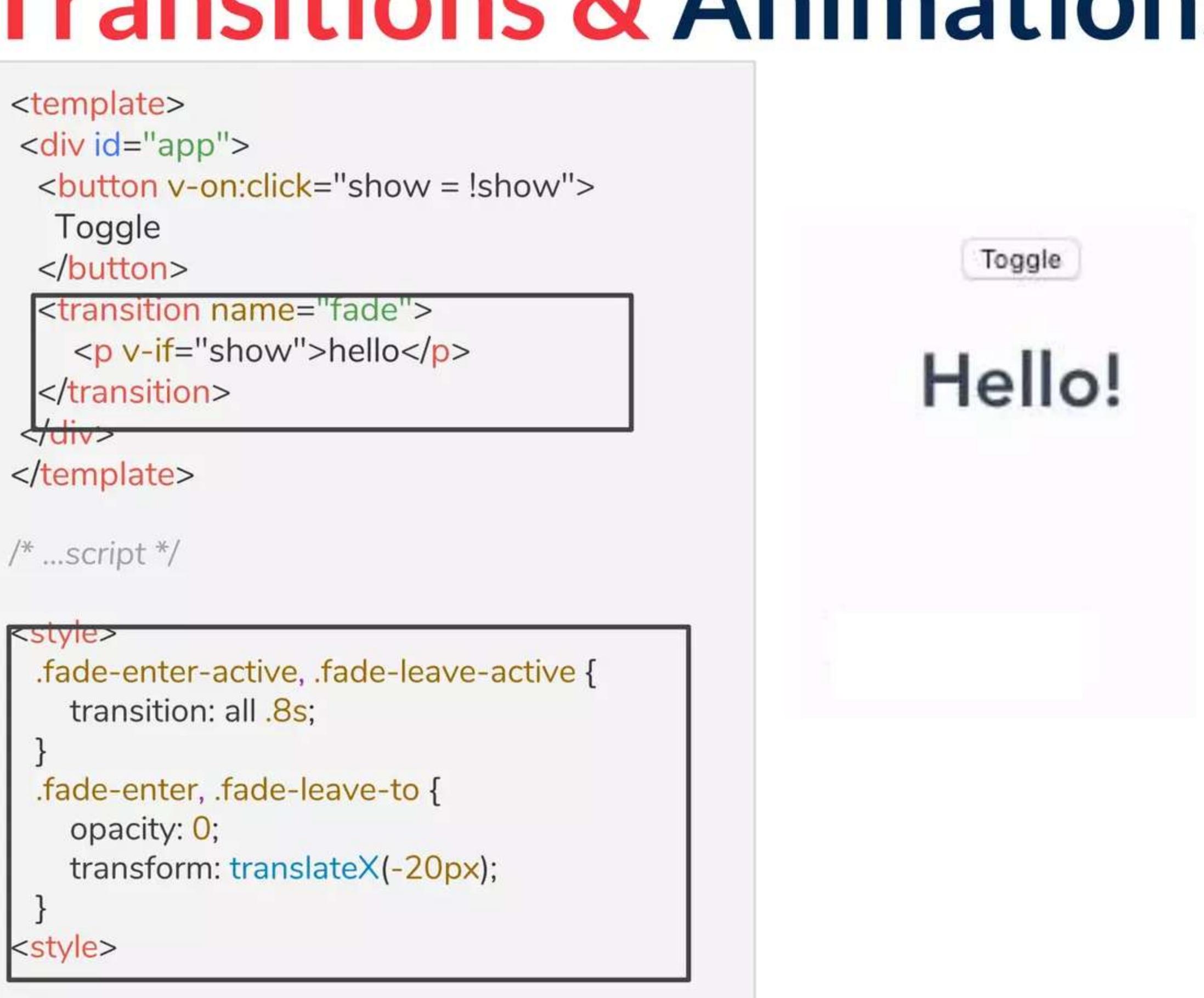
```
<style>
.fade-enter-active, .fade-leave-active {
  transition: all .8s;
}
.fade-enter, .fade-leave-to {
  opacity: 0;
  transform: translateX(-20px);
}
<style>
```

Transitions & Animations.

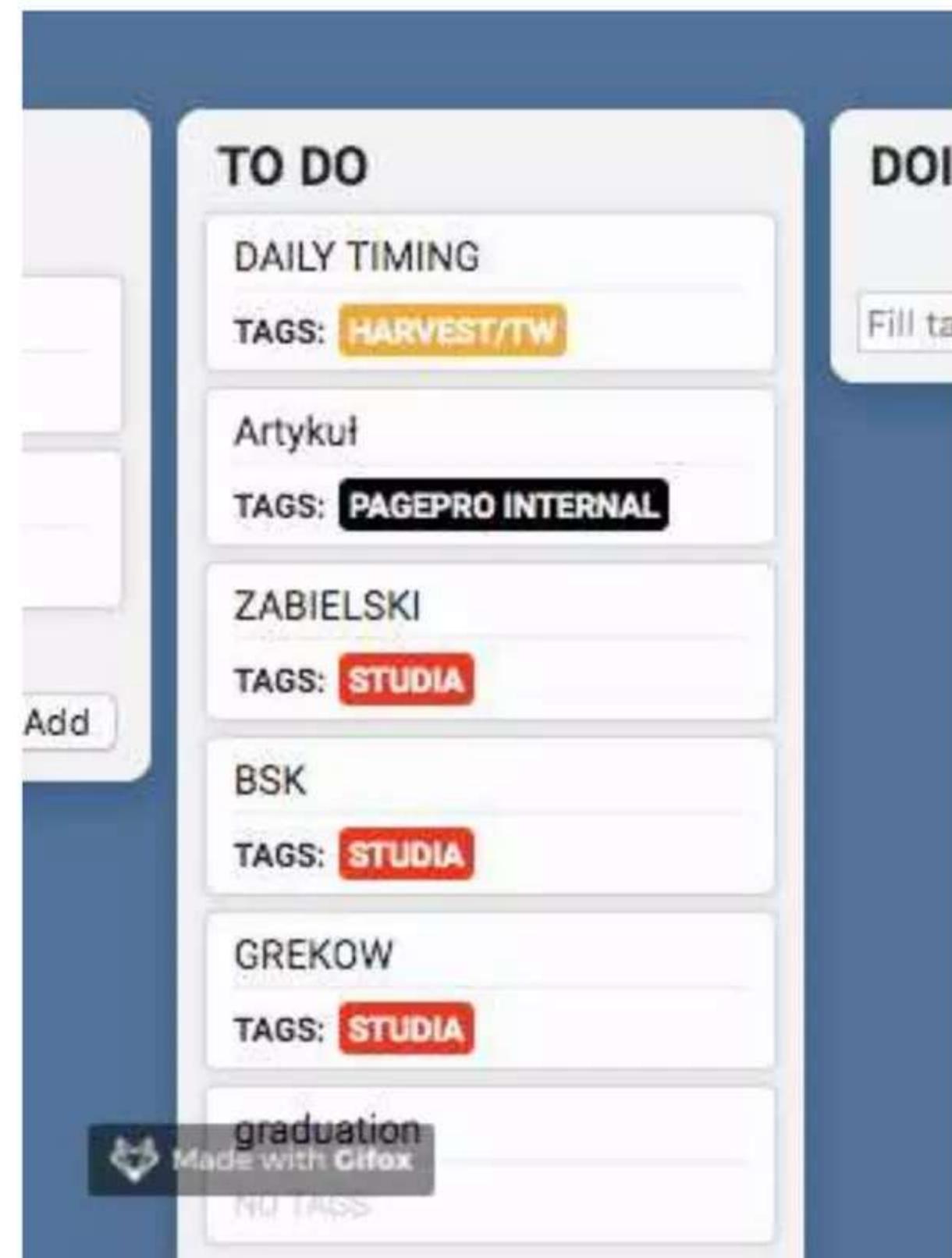
```
<template>
<div id="app">
  <button v-on:click="show = !show">
    Toggle
  </button>
  <transition name="fade">
    <p v-if="show">hello</p>
  </transition>
</div>
</template>

/* ...script */
```

```
<style>
.fade-enter-active, .fade-leave-active {
  transition: all .8s;
}
.fade-enter, .fade-leave-to {
  opacity: 0;
  transform: translateX(-20px);
}
<style>
```



Transitions & Animations - Groups.



List Rendering.

List Rendering Array.

List Rendering - Array.

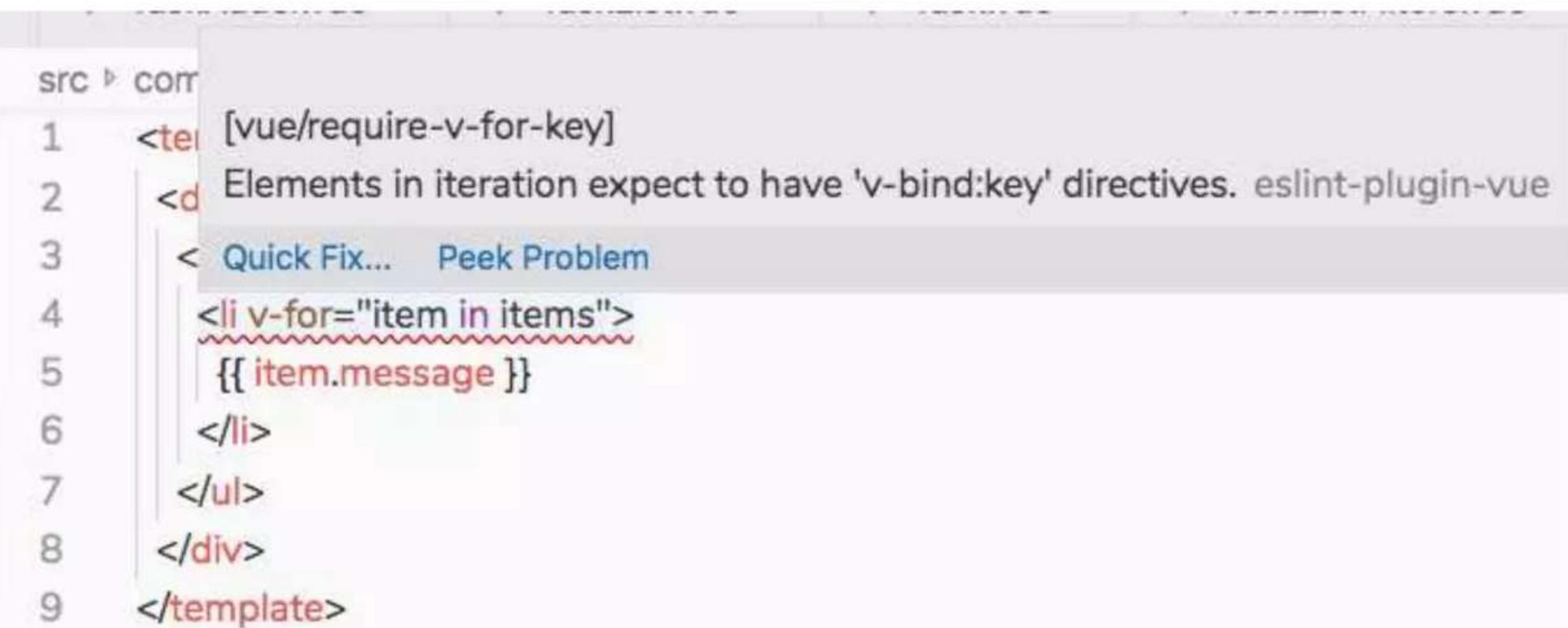
```
<template>
<div id="app">
<ul>
  <li v-for="item in items">
    {{ item.message }}
  </li>
</ul>
</div>
</template>

<script>
export default {
  name: 'MessageBox',
  data() {
    return {
      items: [
        { message: 'Foo' },
        { message: 'Bar' }
      ]
    }
  }
}
</script>
```

List Rendering - Array.

```
<template>
<div id="app">
<ul>
  <li v-for="item in items">
    {{ item.message }}
  </li>
</ul>
</div>
</template>

<script>
export default {
  name: 'MessageBox',
  data() {
    return {
      items: [
        { message: 'Foo' },
        { message: 'Bar' }
      ]
    }
  }
}
</script>
```



List Rendering - Array.

```
<template>
<div id="app">
<ul>
  <li v-for="item in items" v-bind:key="item.message">
    {{ item.message }}
  </li>
</ul>
</div>
</template>

<script>
export default {
  name: 'MessageBox',
  data() {
    return {
      items: [
        { message: 'Foo' },
        { message: 'Bar' }
      ]
    }
  }
}
</script>
```

The screenshot shows a code editor interface with a tooltip overlay. The tooltip is titled '[vue/require-v-for-key]' and contains the message 'Elements in iteration expect to have 'v-bind:key' directives. eslint-plugin-vue'. It includes 'Quick Fix...' and 'Peek Problem' buttons. Below the tooltip, the Vue.js template code is displayed, specifically the part where the `v-for` directive is used without a corresponding `v-bind:key`.

```
src ▶ corr [vue/require-v-for-key]
1   <tei> Elements in iteration expect to have 'v-bind:key' directives. eslint-plugin-vue
2   <d>
3   < Quick Fix... Peek Problem
4   <li v-for="item in items">
5     {{ item.message }}
6   </li>
7   </ul>
8   </div>
9   </template>
```

List Rendering Object.

List Rendering - Object.

```
<template>
<div id="app">
<ul>
  <li v-for="value in user">
    {{ value }}
  </li>
</ul>
</div>
</template>

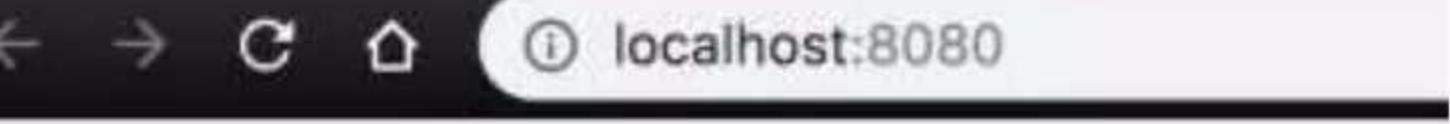
<script>
export default {
  name: 'Profile',
  data() {
    return {
      user: {
        firstName: 'John',
        lastName: 'Doe',
        age: 30
      }
    }
  }
}
</script>
```



List Rendering - Object.

```
<template>
<div id="app">
<ul>
  <li v-for="(value, index) in user" v-bind:key="index">
    {{ index }}: {{ value }}
  </li>
</ul>
</div>
</template>

<script>
export default {
  name: 'Profile',
  data() {
    return {
      user: {
        firstName: 'John',
        lastName: 'Doe',
        age: 30
      }
    }
  }
}
</script>
```



firstName: John
lastName: Doe
age: 30

List Rendering - Object.

```
{ "10-02-2019": [ { "index": 0, "amount": "-31,00", "name": "Jedzenie poza domem", "title": "PAYU", "date": "10-02-2019", "timestamp": "2019-02-10T15:45:09.3254180Z" }, { "index": 1, "amount": "-10,00", "name": "Przejazdy", "title": "SunUp *Bartom*", "date": "08-02-2019", "timestamp": "2019-02-08T20:19:03.6296000Z" }, { "index": 2, "amount": "-31,20", "name": "Żywność i chemia domowa", "title": "PSS Spolem Sklep nr3", "date": "09-02-2019", "timestamp": "2019-02-09T18:00:43.6621660Z" } ] }
```



List Rendering - Object.

```
<h2>Transactions</h2>
<ol>
<li
  class="daily-list-item"
  v-for="(group, index) in transactionsGrouped"
  :key="index">
  <div class="daily-list-cat">
    <span class="daily-list-cat__hour">
      {{ index | momentDate('DD-MM-YYYY', 'DD') }}
    </span>
    <span class="daily-list-cat__day">
      {{ index | momentDate('DD-MM-YYYY', 'MMM') }}
    </span>
  </div>
  <ul>
    <li v-for="transaction in group"
        v-bind:key="transaction.index"
        v-bind:class="[
          parseInt(transaction.amount, 10) < -20 ? 'negative' : '',
          transaction.isIncluded ? 'included' : ''
        ]">
      <!-- list transactions per day -->
    </li>
  </ul>
</ol>
```

```
{
  "10-02-2019": [
    {
      "index": 0,
      "amount": "-31,80",
      "name": "Jedzenie poza domem",
      "title": "PAYU",
      "date": "10-02-2019",
      "timestamp": "2019-02-10T15:45:09.3254180Z"
    },
    ...
  ],
  "08-02-2019": [
    {
      "index": 2,
      "amount": "-10,80",
      "name": "Przejazdy",
      "title": "SunUp *Bartom",
      "date": "08-02-2019",
      "timestamp": "2019-02-08T20:19:03.6296000Z"
    },
    ...
  ],
  "09-02-2019": [
    {
      "index": 3,
      "amount": "-31,20",
      "name": "Żywność i chemia domowa",
      "title": "PSS Spolem Sklep nr3",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    },
    ...
  ]
}
```

List Rendering - Object.

```
<h2>Transactions</h2>
<ol>
<li
  class="daily-list-item"
  v-for="(group, index) in transactionsGrouped"
  :key="index">
  <div class="daily-list-cat">
    <span class="daily-list-cat__hour">
      {{ index | momentDate('DD-MM-YYYY', 'DD') }}
    </span>
    <span class="daily-list-cat__day">
      {{ index | momentDate('DD-MM-YYYY', 'MMM') }}
    </span>
  </div>
  <ul>
    <li v-for="(transaction) in group"
        v-bind:key="transaction.index"
        v-bind:class="[
          parseInt(transaction.amount, 10) < -20 ? 'negative' : '',
          transaction.isIncluded ? 'included' : ''
        ]">
      <!-- list transactions per day -->
    </li>
  </ul>
</ol>
```

```
{ [
  "10-02-2019": [
    {
      "index": 0,
      "amount": "-31,00",
      "name": "Jedzenie poza domem",
      "title": "PAYU",
      "date": "10-02-2019",
      "timestamp": "2019-02-10T15:45:09.3254180Z"
    },
    ...
  ],
  "08-02-2019": [
    {
      "index": 2,
      "amount": "-10,80",
      "name": "Przejazdy",
      "title": "SumUp *Bartom",
      "date": "08-02-2019",
      "timestamp": "2019-02-08T20:19:03.6296000Z"
    },
    ...
  ],
  "09-02-2019": [
    {
      "index": 3,
      "amount": "-31,20",
      "name": "Żywność i chemia domowa",
      "title": "PSS Spolem Sklep nr3",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    },
    ...
  ]
}]
```

Transactions			
10	Feb	PAYU Jedzenie poza domem 16:02:32	-31,00 X
		[NETFLIX] - STYCZEŃ,... TV, internet, telefon 16:02:28	-26,00 X
08	Feb	SumUp *Bartom Przejazdy 21:02:62	-10,80 X
		ZABKA Z5838 K.1 Żywność i chemia domowa 18:02:23	-11,77 X
09	Feb	PSS Spolem Sklep nr3 Żywność i chemia domowa 19:02:66	-31,20 X
		"Jubileuszowy" Żywność i chemia domowa 19:02:66	-7,43 X
		"Jubileuszowy" Żywność i chemia domowa 19:02:65	-20,00 X

List Rendering - Object.

```
<h2>Transactions</h2>
<ol>
<li
  class="daily-list-item"
  v-for="(group, index) in transactionsGrouped"
  :key="index">
  <div class="daily-list-cat">
    <span class="daily-list-cat__hour">
      {{ index | momentDate('DD-MM-YYYY', 'DD') }}
    </span>
    <span class="daily-list-cat__day">
      {{ index | momentDate('DD-MM-YYYY', 'MMM') }}
    </span>
  </div>
  <ul>
    <li v-for="(transaction) in group"
        v-bind:key="transaction.index"
        v-bind:class="[
          parseInt(transaction.amount, 10) < -20 ? 'negative' : '',
          transaction.isIncluded ? 'included' : ''
        ]">
        <!-- list transactions per day -->
    </li>
  </ul>
</ol>
```

```
{ [
  "10-02-2019": [
    {
      "index": 0,
      "amount": "-31,00",
      "name": "Jedzenie poza domem",
      "title": "PAYU",
      "date": "10-02-2019",
      "timestamp": "2019-02-10T15:45:09.3254180Z"
    },
    {
      "index": 1,
      "amount": "-26,00",
      "name": "[NETFLIX] - STYCZEŃ...",
      "title": null,
      "date": "10-02-2019",
      "timestamp": "2019-02-10T15:45:09.3254180Z"
    }
  ],
  "08-02-2019": [
    {
      "index": 2,
      "amount": "-10,80",
      "name": "Przejazdy",
      "title": "SumUp *Bartom",
      "date": "08-02-2019",
      "timestamp": "2019-02-08T20:19:03.6296000Z"
    },
    {
      "index": 3,
      "amount": "-31,20",
      "name": "Zabka Z5838 K.1",
      "title": "PSS Spolem Sklep nr3",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    }
  ],
  "09-02-2019": [
    {
      "index": 4,
      "amount": "-7,43",
      "name": "Jubileuszowy",
      "title": "Jubileuszowy",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    },
    {
      "index": 5,
      "amount": "-20,00",
      "name": "Jubileuszowy",
      "title": "Jubileuszowy",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    }
  ]
]
```

Transactions			
10 Feb	PAYU Jedzenie poza domem 16:02:32	-31,00	X
08 Feb	[NETFLIX] - STYCZEŃ... TV, internet, telefon 16:02:28	-26,00	X
09 Feb	SumUp *Bartom Przejazdy 21:02:62	-10,80	X
09 Feb	ZABKA Z5838 K.1 Zywnosc i chemia domowa 18:02:23	-11,77	X
	PSS Spolem Sklep nr3 Zywnosc i chemia domowa 19:02:66	-31,20	X
	"Jubileuszowy" Zywnosc i chemia domowa 19:02:66	-7,43	X
	"Jubileuszowy" Zywnosc i chemia domowa 19:02:65	-20,00	X

List Rendering - Array.

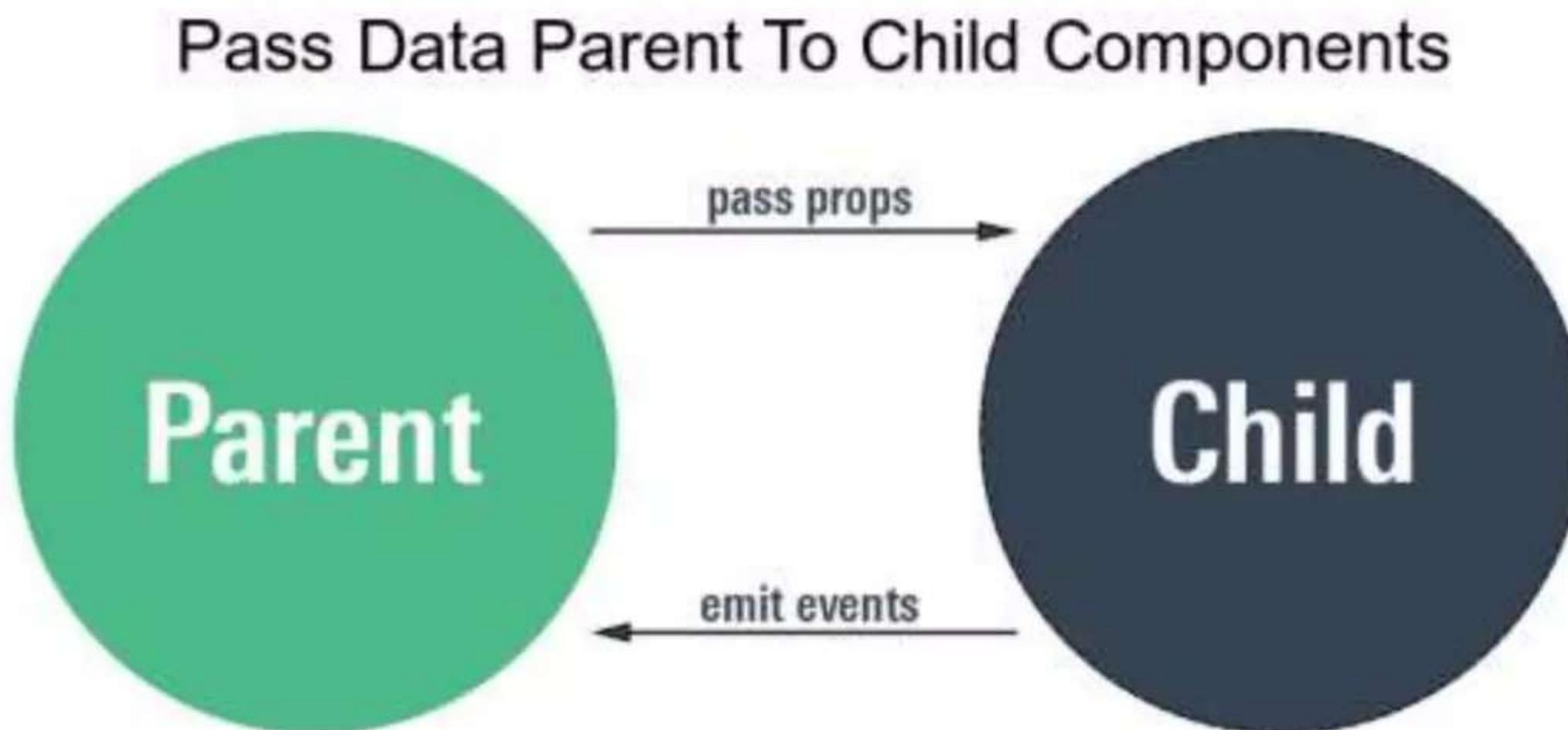
```
<h2>Transactions</h2>
<ol>
<li
  class="daily-list-item"
  v-for="(group, index) in transactionsGrouped"
  :key="index">
  <div class="daily-list-cat">
    <span class="daily-list-cat__hour">
      {{ index | momentDate('DD-MM-YYYY', 'DD') }}
    </span>
    <span class="daily-list-cat__day">
      {{ index | momentDate('DD-MM-YYYY', 'MMM') }}
    </span>
  </div>
  <ul>
    <li v-for="(transaction) in group"
        v-bind:key="transaction.index"
        v-bind:class="[
          parseInt(transaction.amount, 10) < -20 ? 'negative' : '',
          transaction.isIncluded ? 'included' : ''
        ]">
      <!-- list transactions per day -->
    </li>
  </ul>
</ol>
```

```
{ [
  "10-02-2019": [
    {
      "index": 0,
      "amount": "-31,00",
      "name": "Jedzenie poza domem",
      "title": "PAYU",
      "date": "10-02-2019",
      "timestamp": "2019-02-10T15:45:09.3254180Z"
    },
    ...
  ],
  "08-02-2019": [
    {
      "index": 2,
      "amount": "-10,80",
      "name": "Przejazdy",
      "title": "SumUp *Bartom",
      "date": "08-02-2019",
      "timestamp": "2019-02-08T20:19:03.6296000Z"
    },
    ...
  ],
  "09-02-2019": [
    {
      "index": 3,
      "amount": "-31,20",
      "name": "Żywność i chemia domowa",
      "title": "PSS Spolem Sklep nr3",
      "date": "09-02-2019",
      "timestamp": "2019-02-09T18:00:43.6621660Z"
    },
    ...
  ]
}]
```

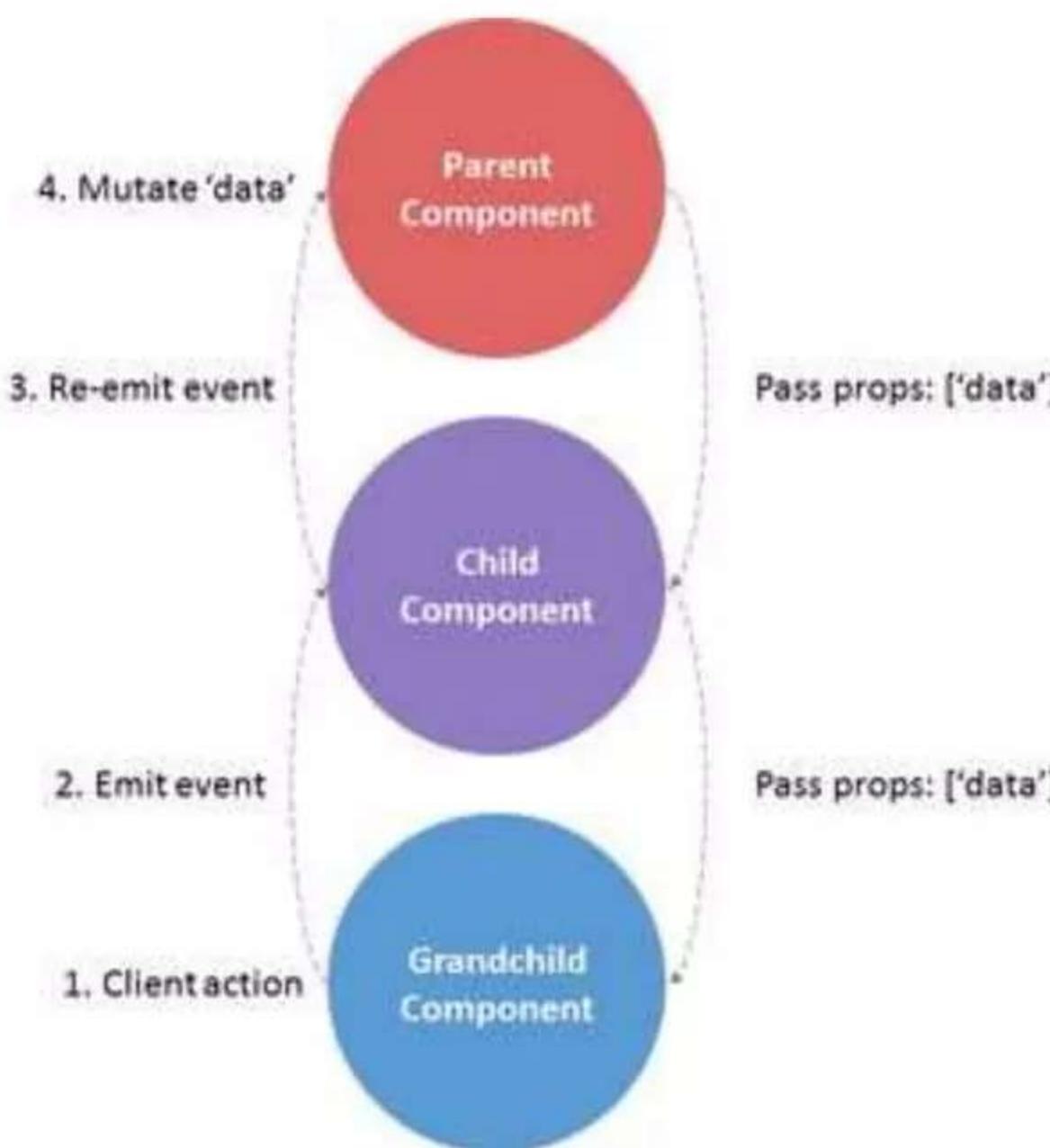
Transactions			
10	Feb	PAYU Jedzenie poza domem 16:02:32	-31,00 X
		[NETFLIX] - STYCZEŃ,... TV, internet, telefon 16:02:28	-26,00 X
08	Feb	SumUp *Bartom Przejazdy 21:02:62	-10,80 X
		ZABKA Z5838 K.1 Żywność i chemia domowa 18:02:23	-11,77 X
09	Feb	PSS Spolem Sklep nr3 Żywność i chemia domowa 19:02:66	-31,20 X
		"Jubileuszowy" Żywność i chemia domowa 19:02:66	-7,43 X
		"Jubileuszowy" Żywność i chemia domowa 19:02:65	-20,00 X

Component Communication.

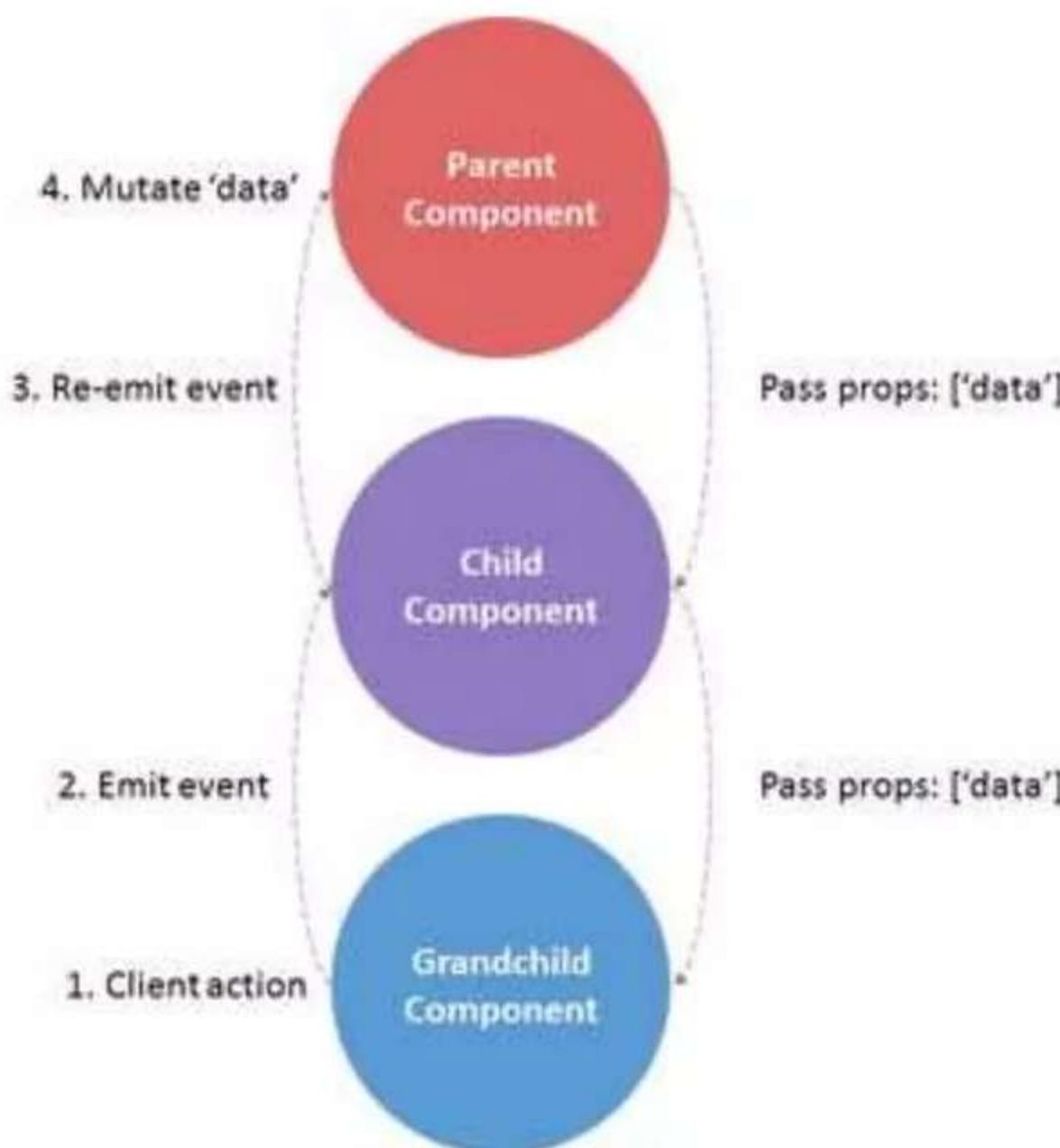
Component Communication - Basics.



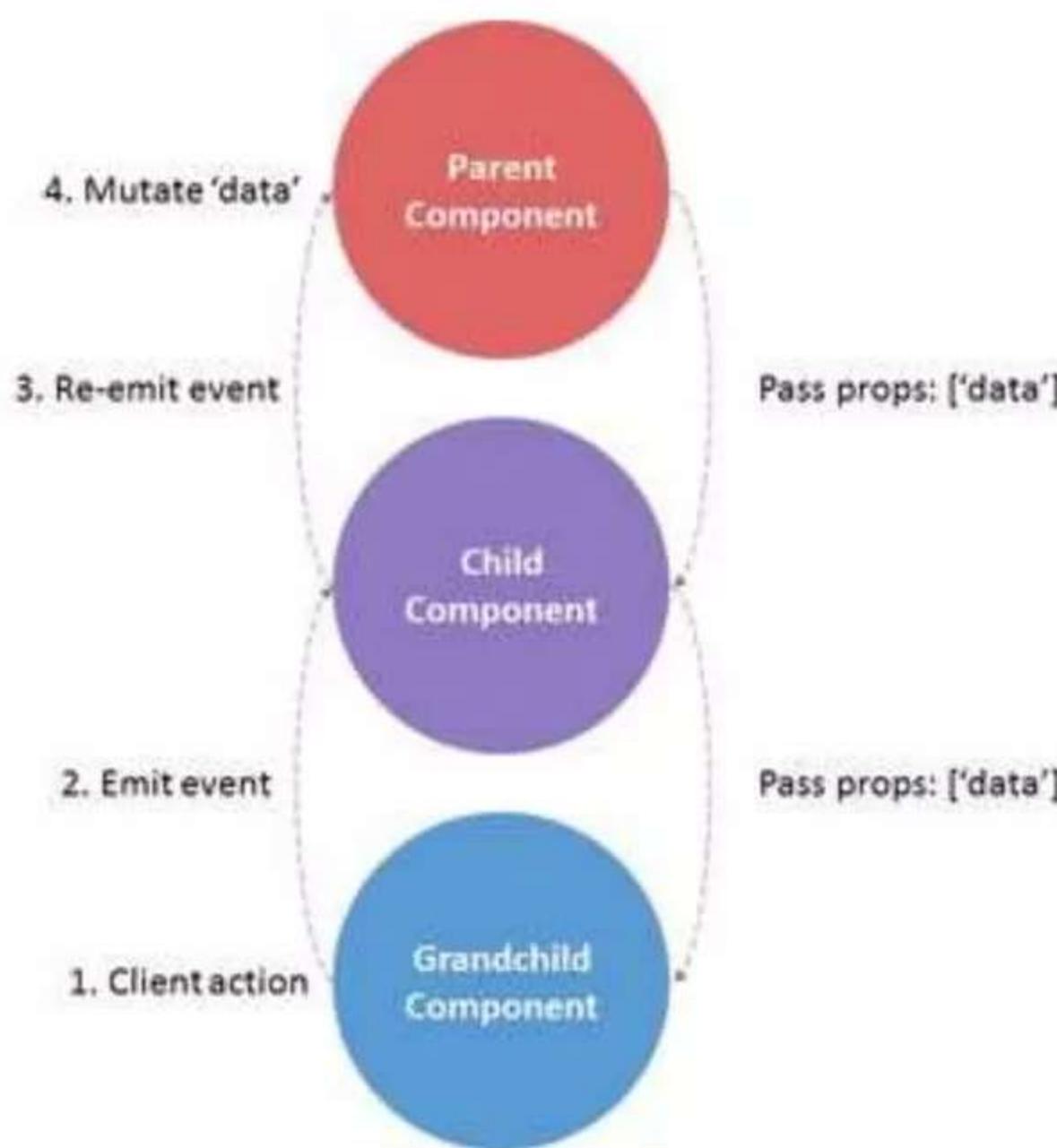
Component Communication - Basics.



Component Communication - Basics.

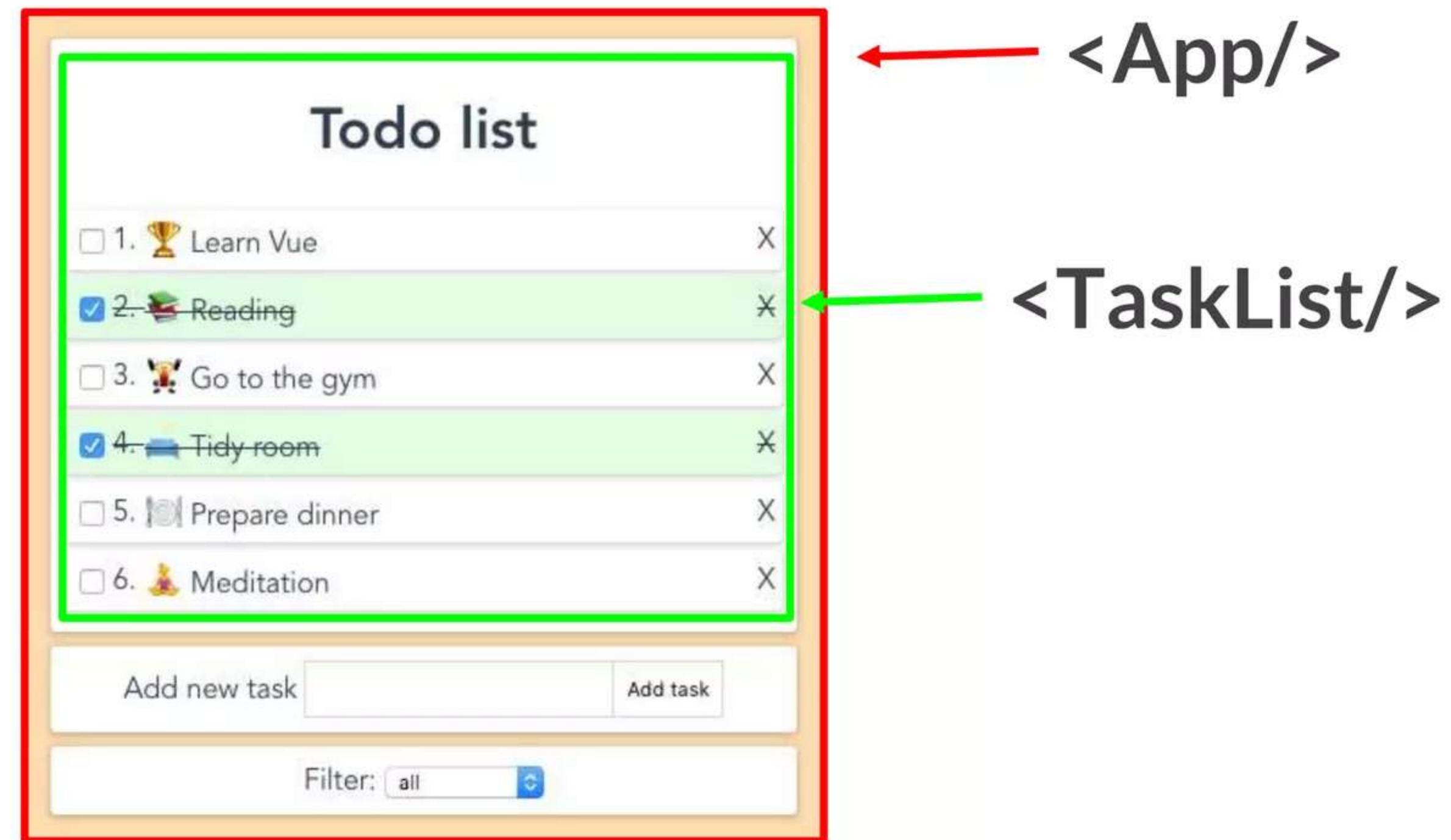
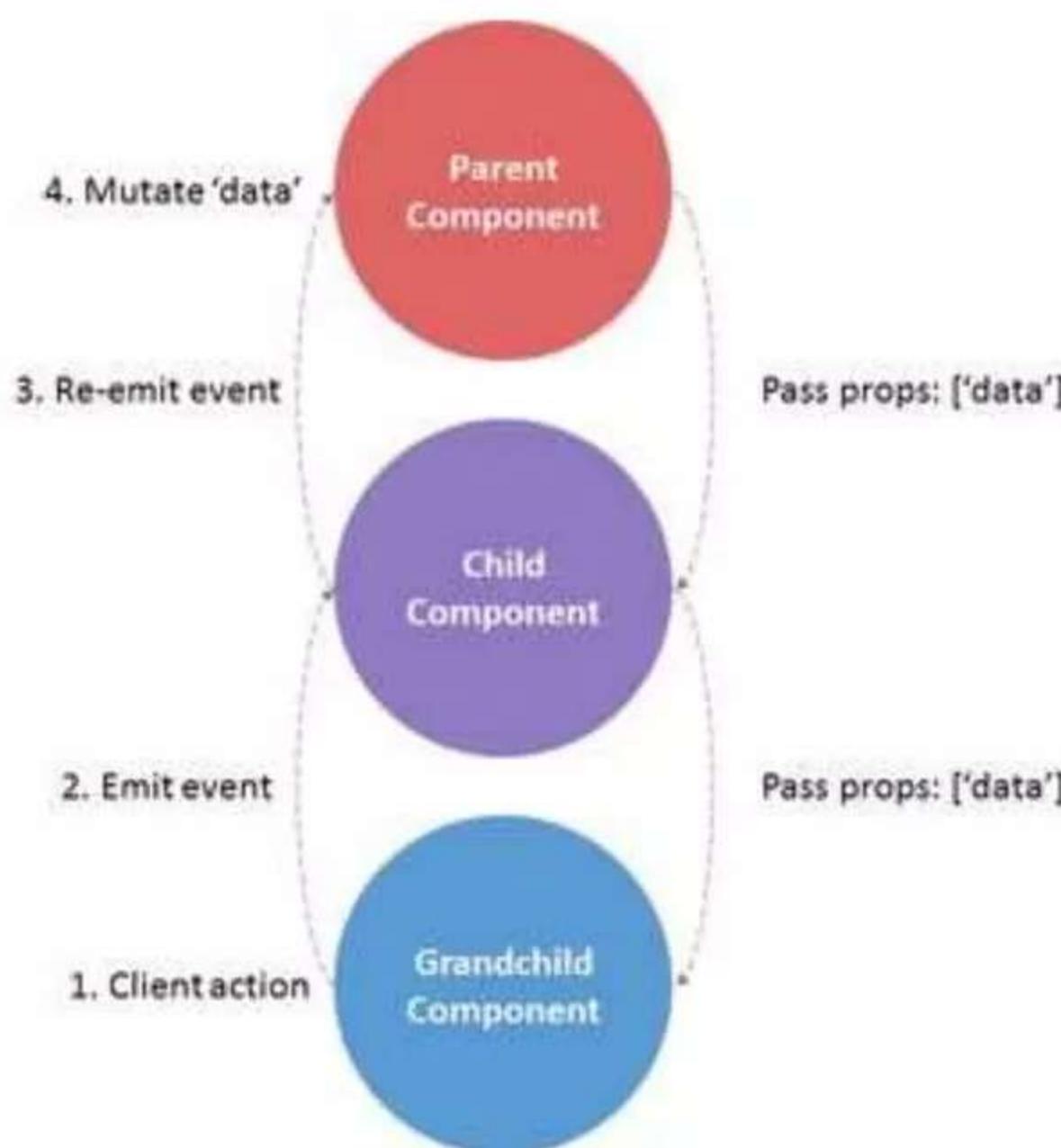


Component Communication - Basics.

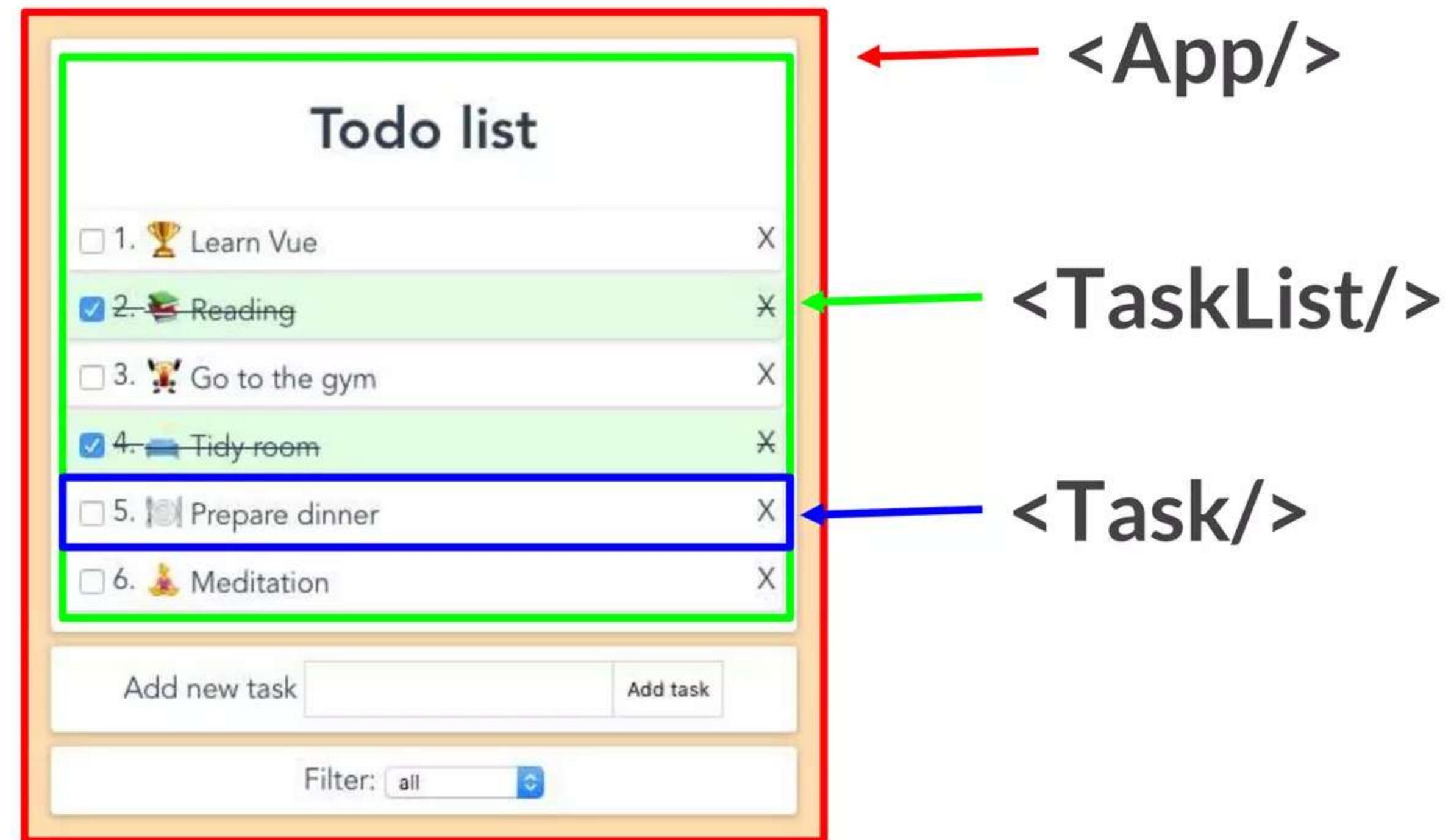
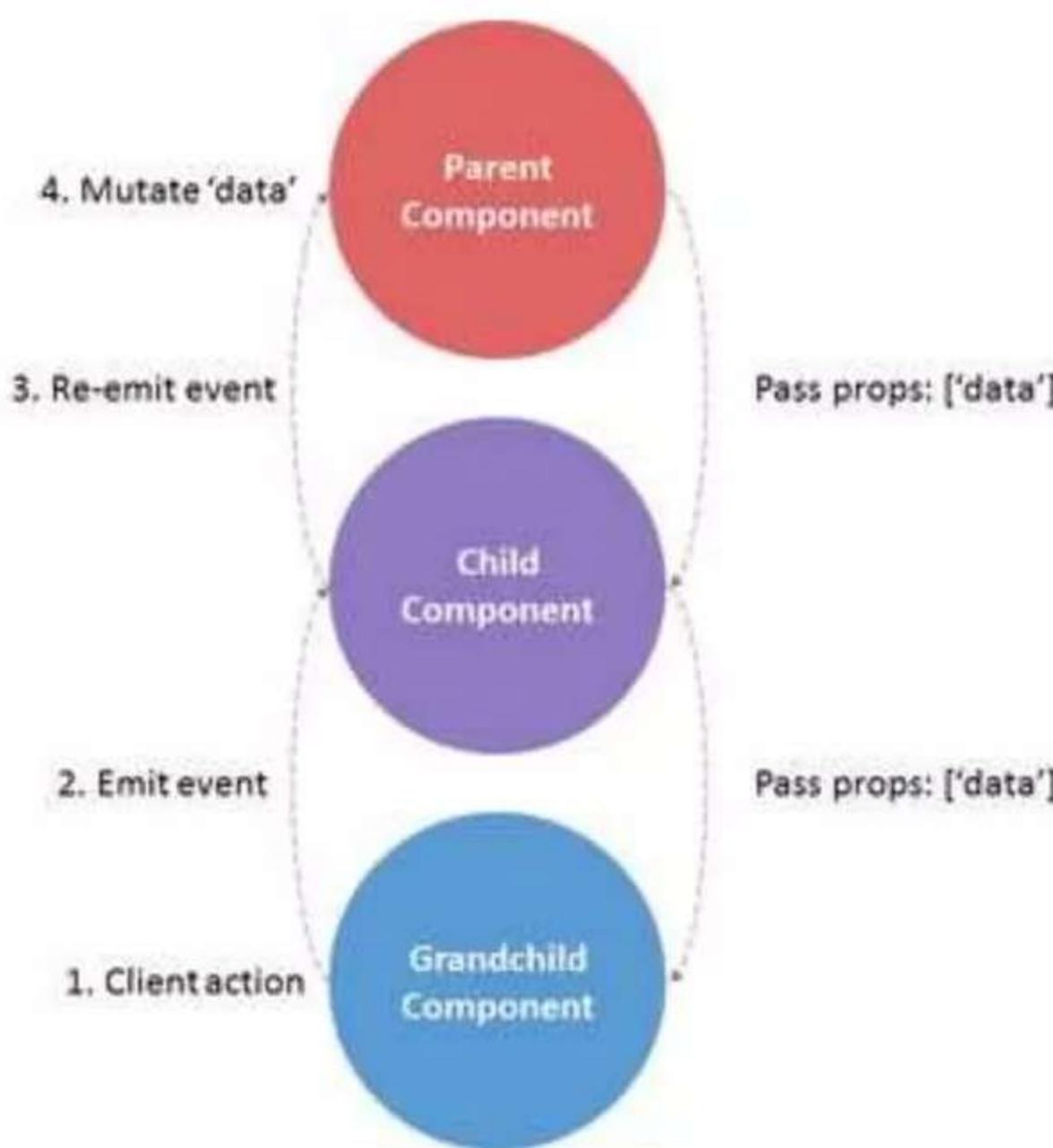


<App/>

Component Communication - Basics.

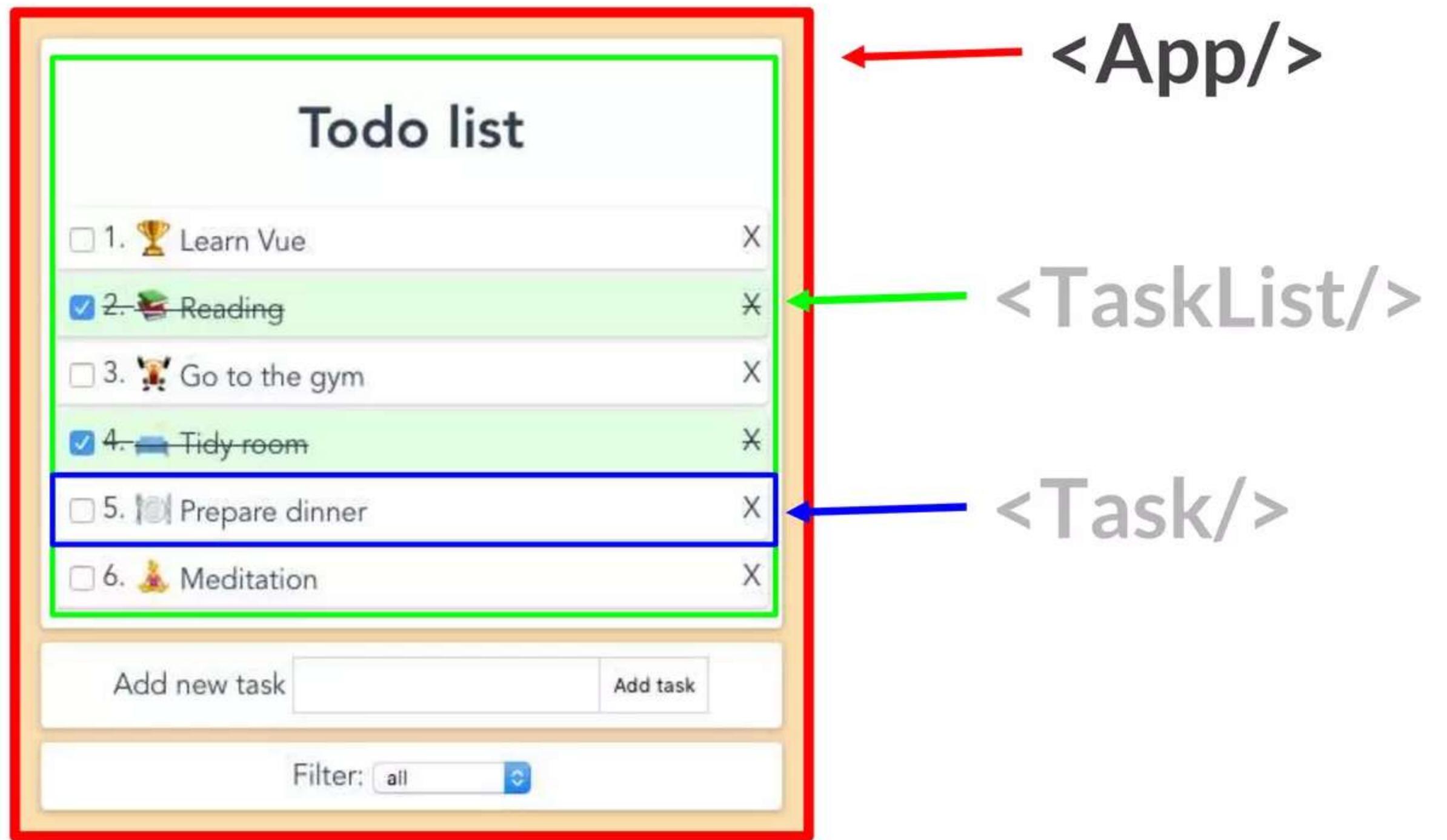


Component Communication - Basics.



Component Communication (Props).

Component Communication - Pass Props.

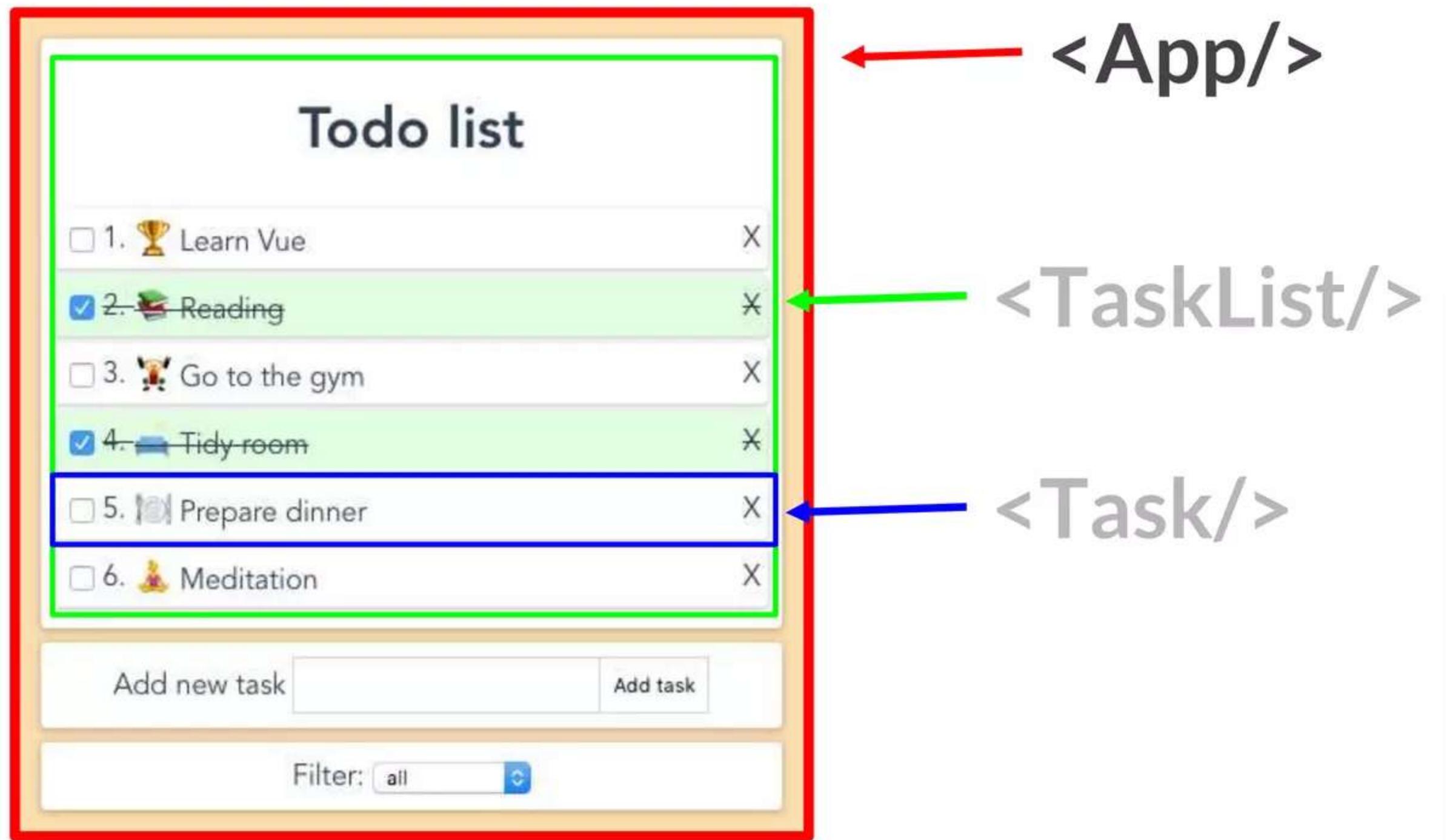


```
<!-- <App> -->
<template>
<div id="app">
  <TaskList
    :title="listTitle"
    :tasks="filteredTasks"/>
</div>
</template>

<script>
import TaskList from '@/components/TaskList.vue'

export default {
  name: 'app',
  components: {
    TaskList
  },
  data: function () {
    return {
      listTitle: 'Todo list',
      tasks: [...]
    }
  }
}
</script>
```

Component Communication - Pass Props.

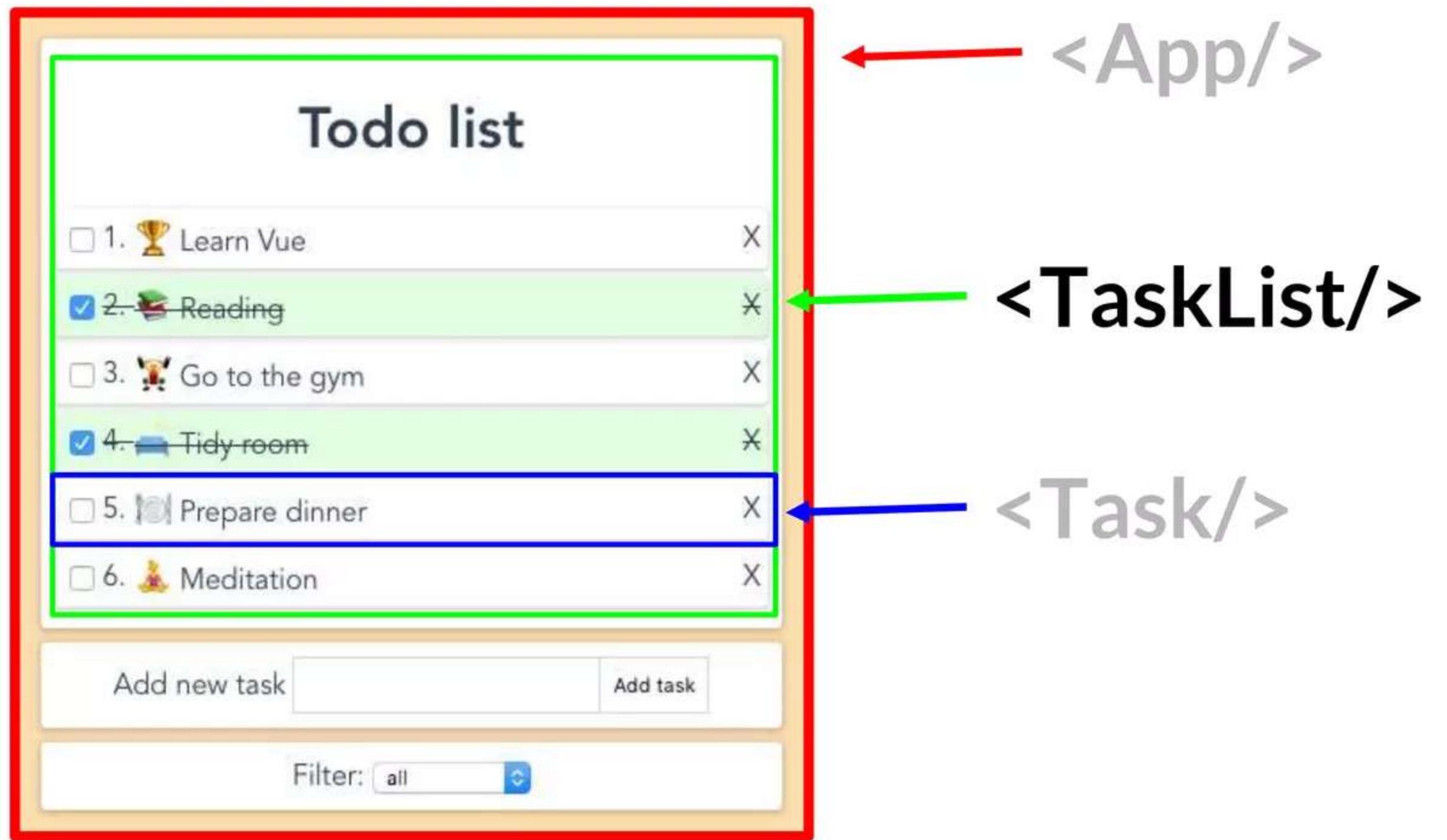


```
<!-- <App> -->
<template>
<div id="app">
  <TaskList
    :title="listTitle"
    :tasks="filteredTasks"/>
</div>
</template>

<script>
import TaskList from '@/components/TaskList.vue'

export default {
  name: 'app',
  components: {
    TaskList
  },
  data: function () {
    return {
      listTitle: 'Todo list',
      tasks: [...]
    }
  }
}
</script>
```

Component Communication - Pass Props.



```
<!-- <TaskList/> -->
<template>
<div class="container">
  <h1>{{ title }}</h1>
  <ul>
    <Task
      v-bind:key="index"
      :task="task"
      :index="index"
      v-for="(task, index) in tasks"
      @deleteTask="deleteTask"/>
  </ul>
</div>
</template>

<script>
import Task from '@/components/Task.vue'

export default {
  name: 'TaskList',
  components: {
    Task
  },
  props: {
    title: String,
    tasks: Array
  },
  </script>
```

Component Communication - Pass Props.



<App/>

<TaskList/>

<Task/>

```
<!-- <TaskList> -->
<template>
<li class="c-task" v-bind:class="task.done ? 'is-done' : "">
  <input v-model="task.done" type="checkbox"/>
  <span>{{ index + 1 }}. </span>
  <span>
    {{ task.name }}
  </span>
  <span class="c-btn-delete" v-on:click="deleteTask(index)">X</span>
</li>
</template>

<script>
export default {
  name: 'Task',
  props: {
    task: Object,
    index: Number
  },
  methods: {
    deleteTask: function (taskId) {
      this.$emit('deleteTask', taskId)
    }
  }
}
</script>
```

Component Communication (Emit).

Component Communication - Emit.



<App/>

<TaskList/>

<Task/>

```
<!-- <TaskList> -->
<template>
<li class="c-task" v-bind:class="task.done ? 'is-done' : "">
  <input v-model="task.done" type="checkbox"/>
  <span>{{ index + 1 }}. </span>
  <span>
    {{ task.name }}
  </span>
  <span class="c-btn-delete" v-on:click="deleteTask(index)">X</span>
</li>
</template>

<script>
export default {
  name: 'Task',
  props: {
    task: Object,
    index: Number
  },
  methods: {
    deleteTask: function (taskId) {
      this.$emit('deleteTask', taskId)
    }
  }
}
</script>
```

Component Communication - Emit.

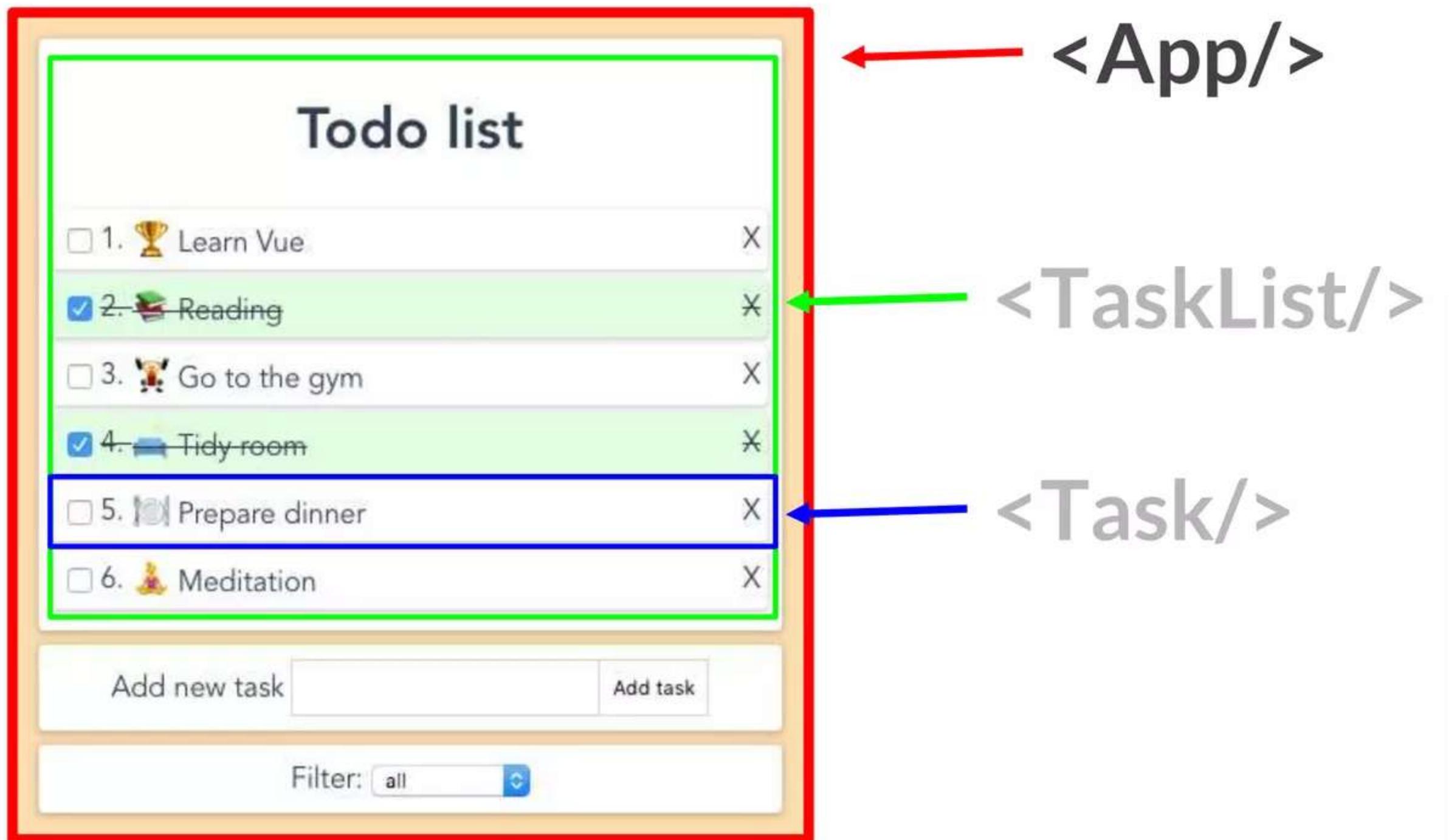


```
<!-- <TaskList/> -->
<template>
<div class="container">
  <h1>{{ title }}</h1>
  <ul>
    <Task
      v-bind:key="index"
      :task="task"
      :index="index"
      v-for="(task, index) in tasks"
      @deleteTask="deleteTask"/>
  </ul>
</div>
</template>

<script>
import Task from '@/components/Task.vue'

export default {
  name: 'TaskList',
  components: {
    Task
  },
  props: {
    title: String,
    tasks: Array
  },
  </script>
```

Component Communication - Receive Emit.



```
<!-- <App> -->
<template>
<div id="app">
  <TaskList
    :title="listTitle"
    :tasks="filteredTasks"
    @deleteTask="deleteTask"/>
</div>
</template>

<script>
import TaskList from '@/components/TaskList.vue'

export default {
  name: 'app',
  components: {
    TaskList
  },
  data: function () {
    return {
      listTitle: 'Todo list',
      tasks: [...]
    }
  },
  methods: {
    deleteTask: function (taskId) {
      this.tasks.splice(taskId, 1)
    }
  }
}
</script>
```

Component Communication (Props).

Practice.

Todo list

- 1. 🏆 Learn Vue X
- 2. 📖 Reading X
- 3. 🏋️ Go to the gym X
- 4. 🪡 Tidy room X
- 5. 🍲 Prepare dinner X
- 6. 🧘 Meditation X

Add new task

Add task

Filter:

all

Thank You :)