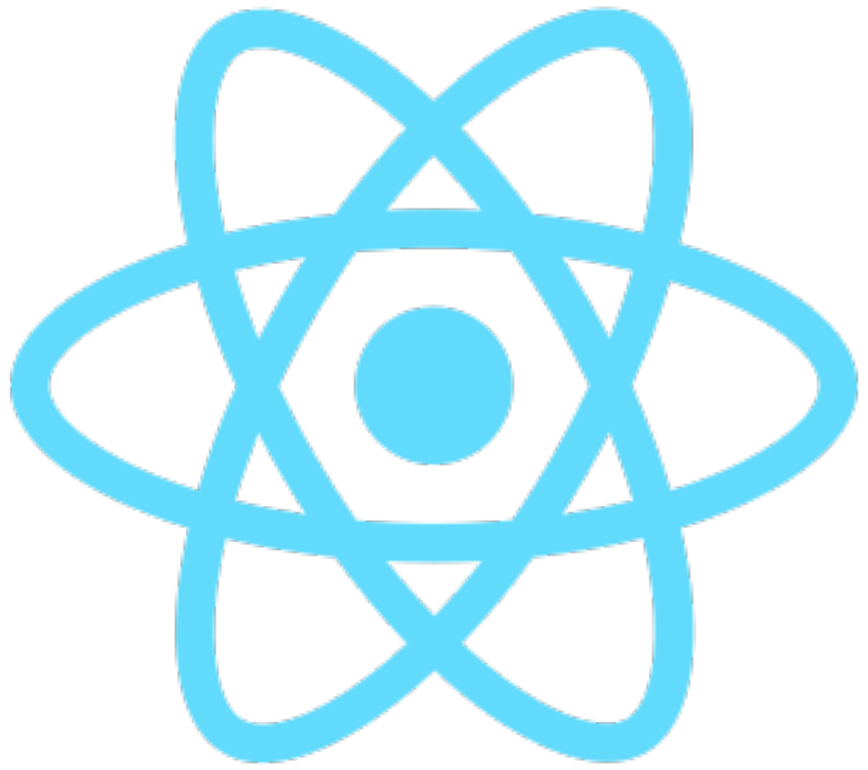


Vue vs React

By example

Viet Tran - CTO Skylab

Introduction & History

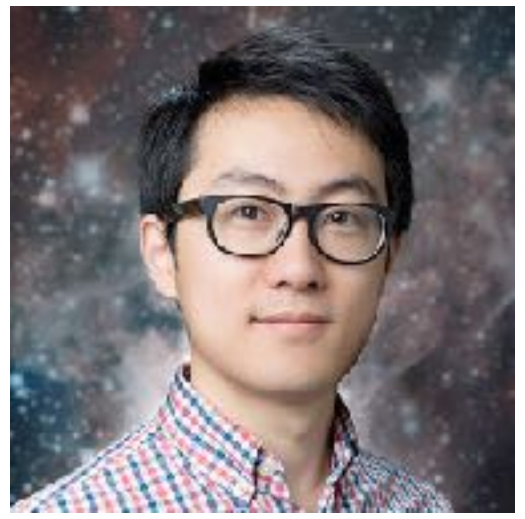
First released in Mar 2013

by

Facebook

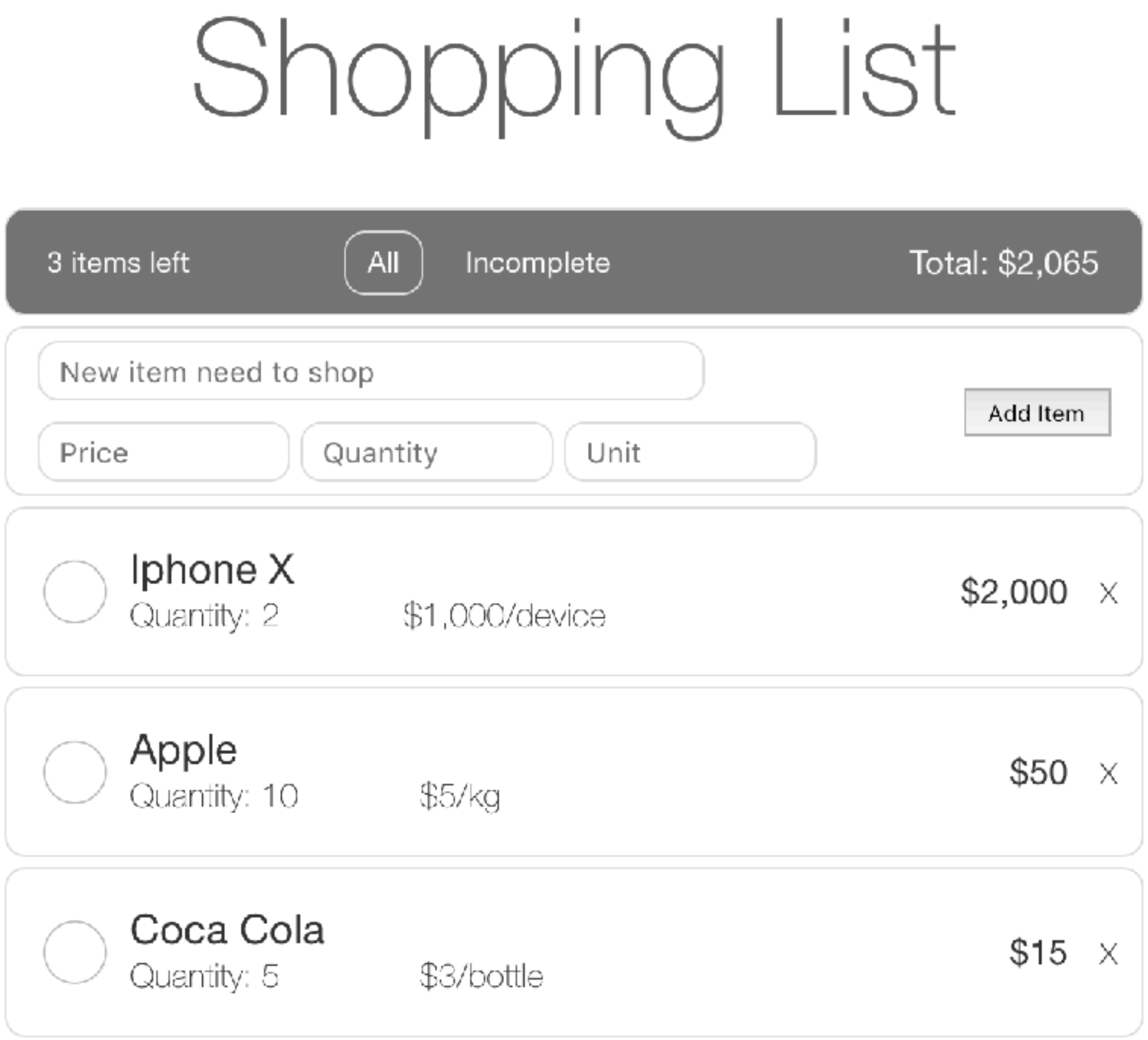
First released in Feb 2014

by



Evan You

Vue vs React by example



Shopping List

3 items left

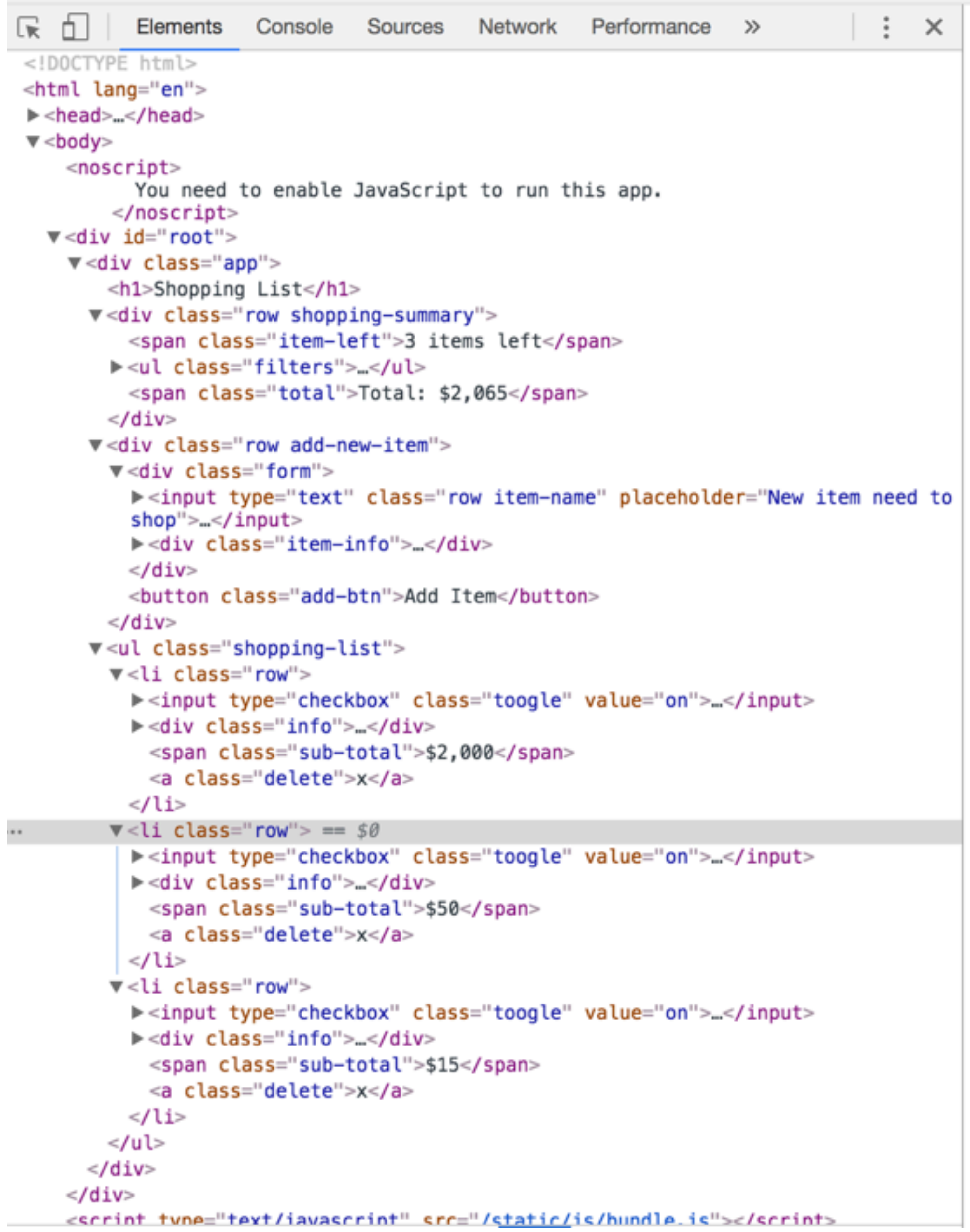
Total: $2,065

○ Iphone X
Quantity: 2 $1,000/device $2,000 X

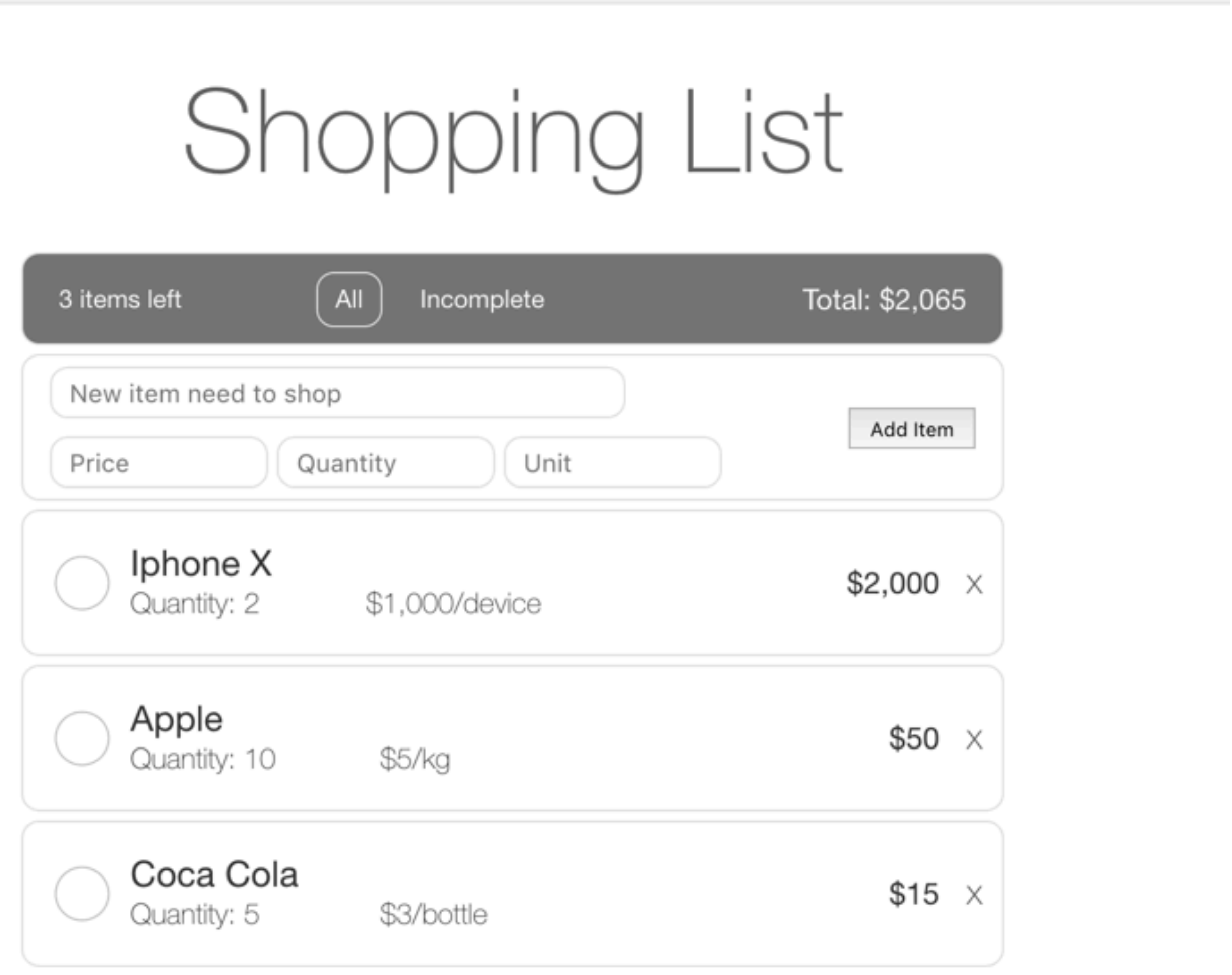
○ Apple
Quantity: 10 $5/kg $50 X

○ Coca Cola
Quantity: 5 $3/bottle $15 X

Shopping List without React/Vue



```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body>
    <noscript>
          You need to enable JavaScript to run this app.
        </noscript>
  ▼<div id="root">
    ▼<div class="app">
        <h1>Shopping List</h1>
      ▼<div class="row shopping-summary">
          <span class="item-left">3 items left</span>
        ▶<ul class="filters">…</ul>
          <span class="total">Total: $2,065</span>
        </div>
      ▼<div class="row add-new-item">
        ▼<div class="form">
          ▶<input type="text" class="row item-name" placeholder="New item need to
          shop">…</input>
          ▶<div class="item-info">…</div>
          </div>
          <button class="add-btn">Add Item</button>
        </div>
      ▼<ul class="shopping-list">
        ▼<li class="row">
          ▶<input type="checkbox" class="toogle" value="on">…</input>
          ▶<div class="info">…</div>
            <span class="sub-total">$2,000</span>
            <a class="delete">x</a>
          </li>
…       ▼<li class="row"> == $0
          ▶<input type="checkbox" class="toogle" value="on">…</input>
          ▶<div class="info">…</div>
            <span class="sub-total">$50</span>
            <a class="delete">x</a>
          </li>
        ▼<li class="row">
          ▶<input type="checkbox" class="toogle" value="on">…</input>
          ▶<div class="info">…</div>
            <span class="sub-total">$15</span>
            <a class="delete">x</a>
          </li>
        </ul>
      </div>
    </div>
    <script type="text/javascript" src="/static/js/bundle.js"></script>
```



Shopping List

3 items left | All | Incomplete | Total: $2,065

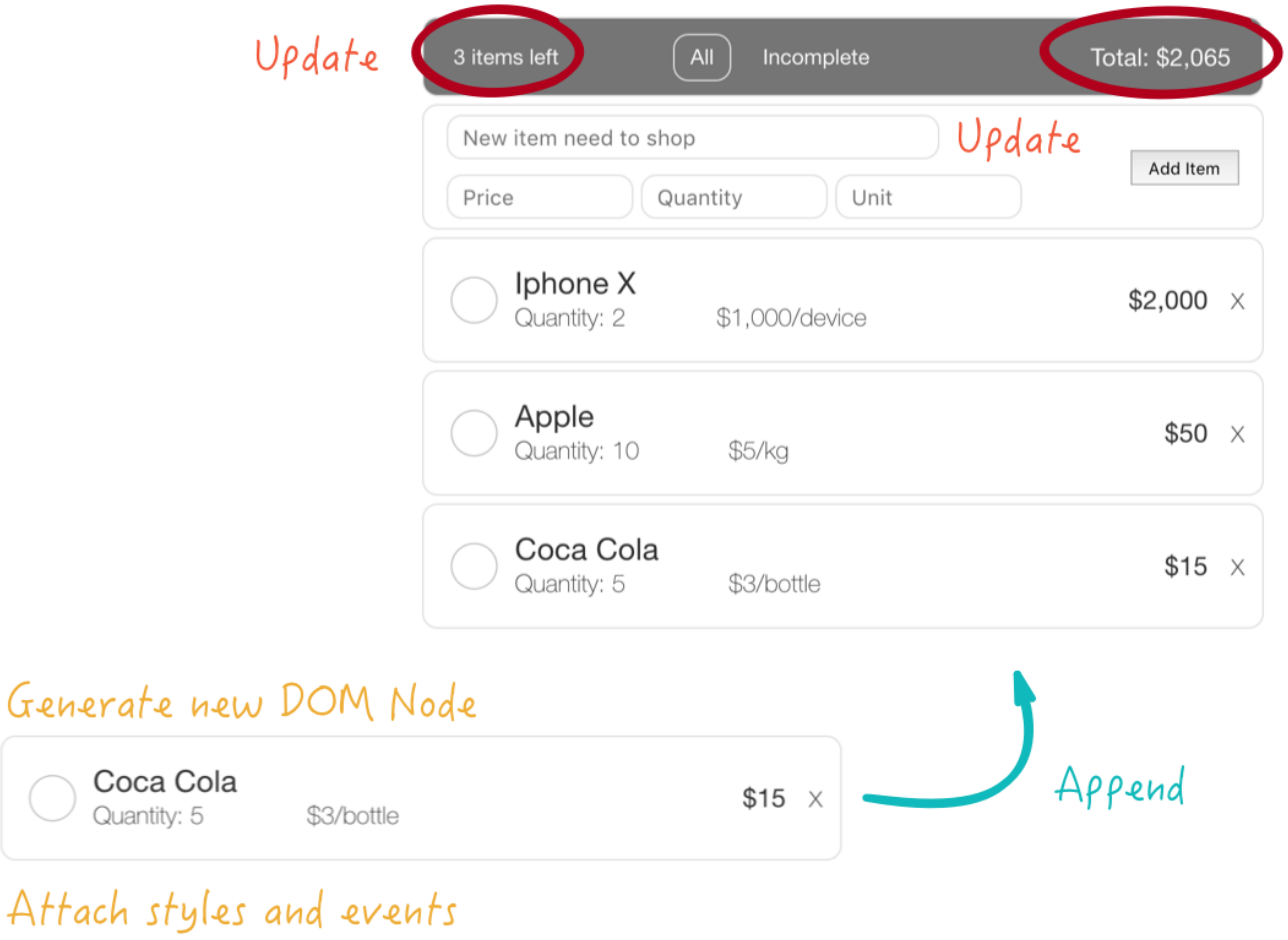
○ Iphone X — Quantity: 2 — $1,000/device — $2,000 X

○ Apple — Quantity: 10 — $5/kg — $50 X

○ Coca Cola — Quantity: 5 — $3/bottle — $15 X

DOM is everywhere

Shopping List without React/Vue



Iphone X
Quantity: 2 $1,000/device $2,000 X

Apple
Quantity: 10 $5/kg $50 X

Coca Cola
Quantity: 5 $3/bottle $15 X

Update

Update

Generate new DOM Node

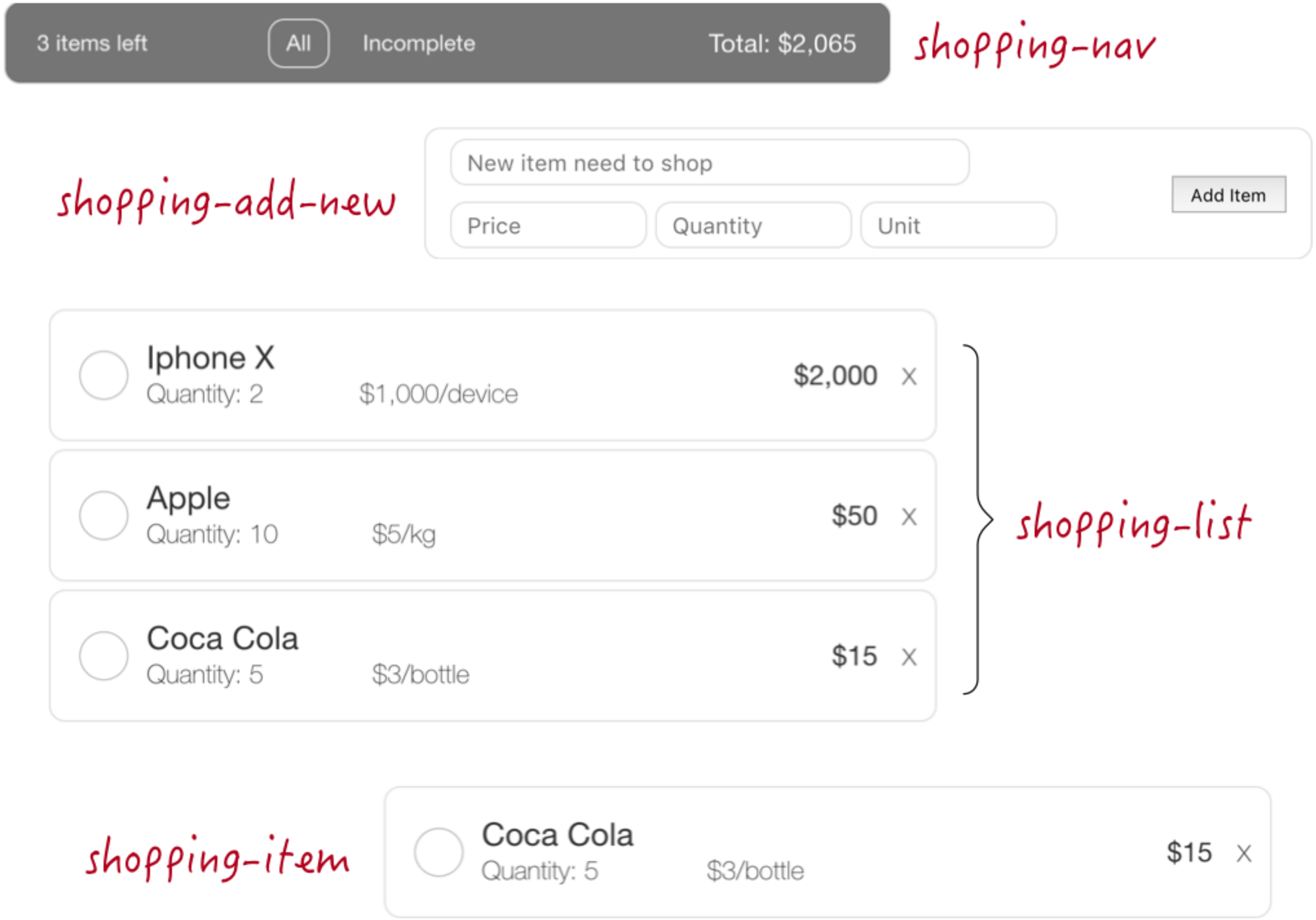
Coca Cola
Quantity: 5 $3/bottle $15 X

Append

Attach styles and events

When we add a new shopping item

Shopping List with React/Vue



○ Iphone X Quantity: 2 $1,000/device $2,000 X

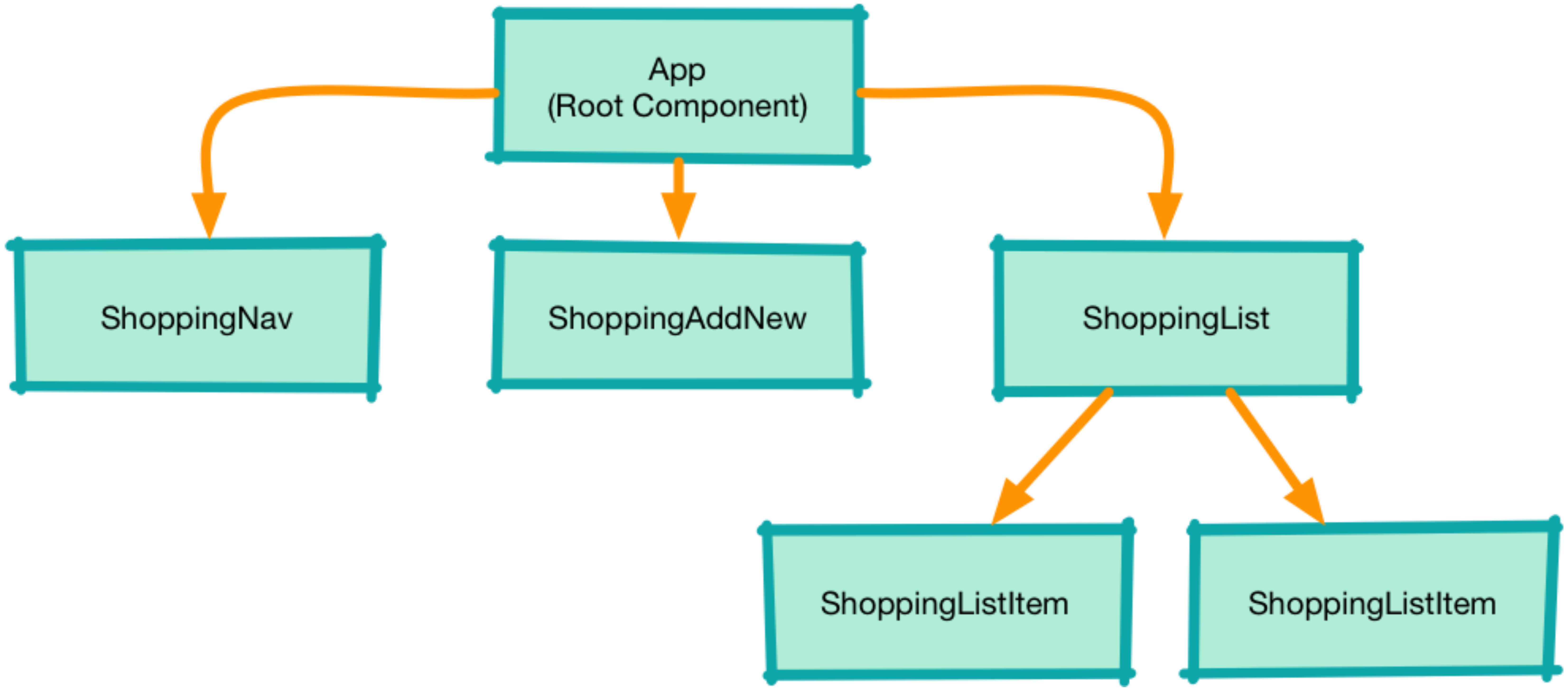
○ Apple Quantity: 10 $5/kg $50 X

○ Coca Cola Quantity: 5 $3/bottle $15 X

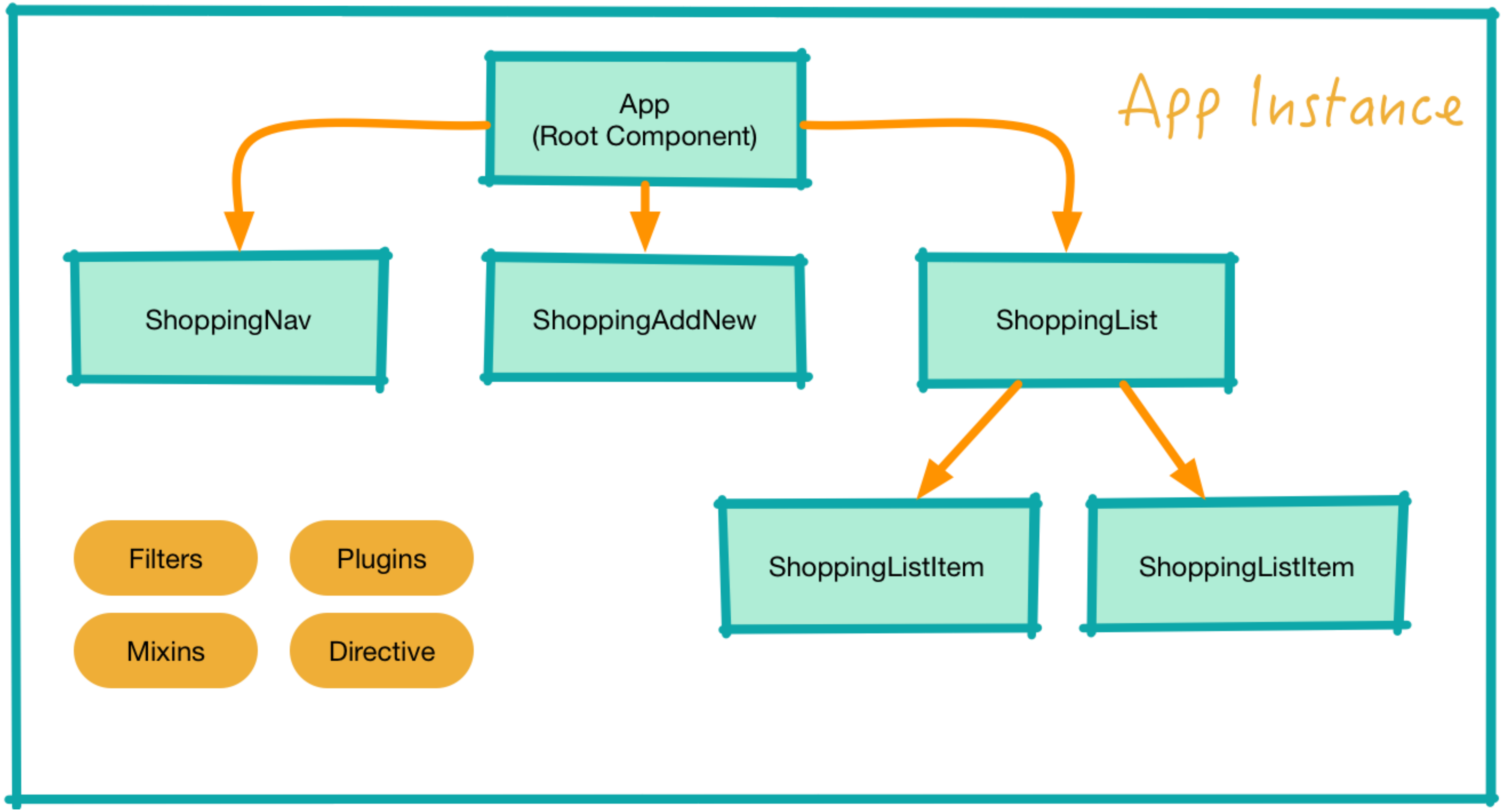
○ Coca Cola Quantity: 5 $3/bottle $15 X

Everything is Virtual DOM/Component

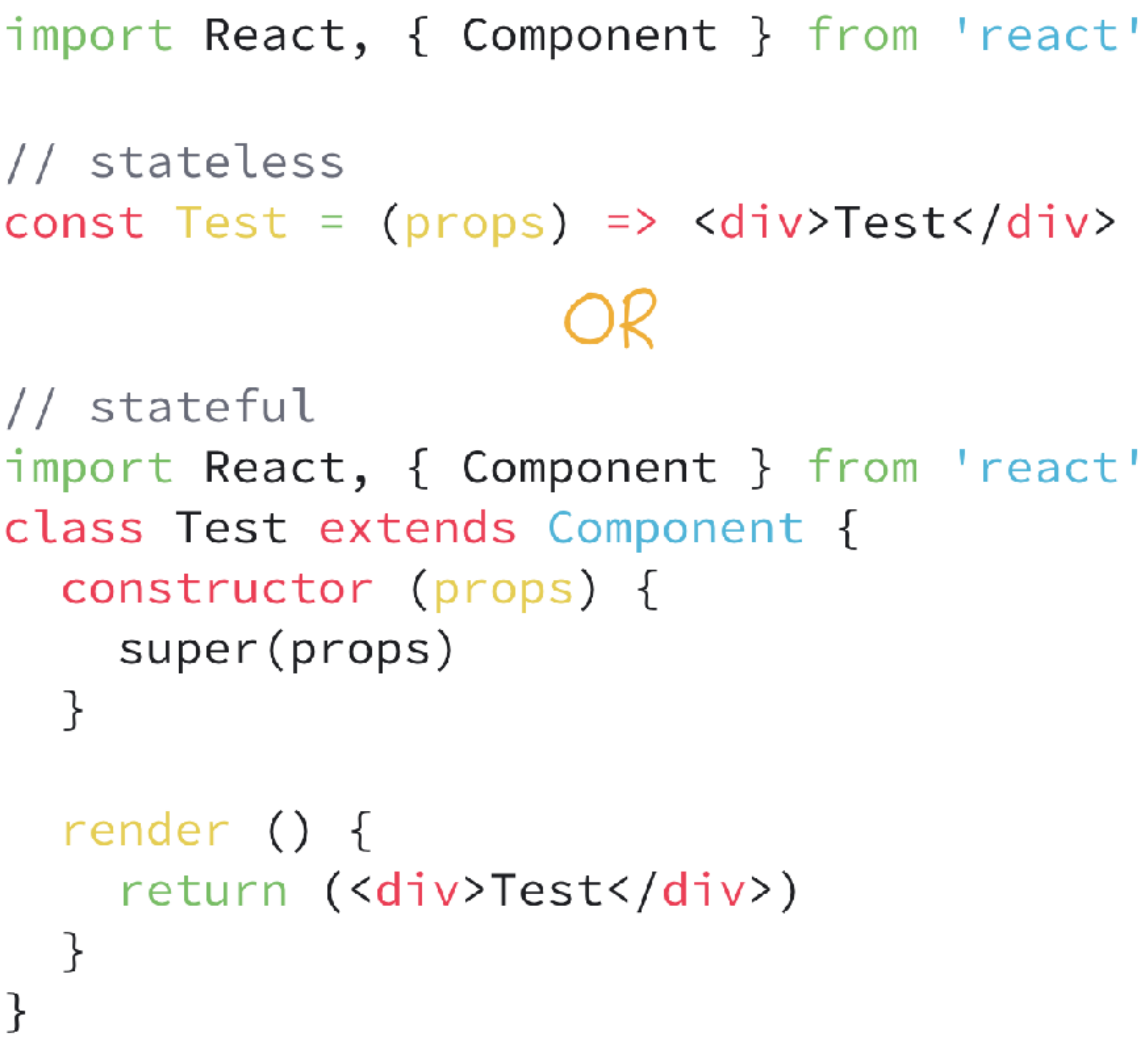
Components in React



Components in Vue



Create a component (React)



```
import React, { Component } from 'react'

// stateless
const Test = (props) => <div>Test</div>
```

OR

```
// stateful
import React, { Component } from 'react'
class Test extends Component {
  constructor (props) {
    super(props)
  }

  render () {
    return (<div>Test</div>)
  }
}
```

Create a component (Vue)

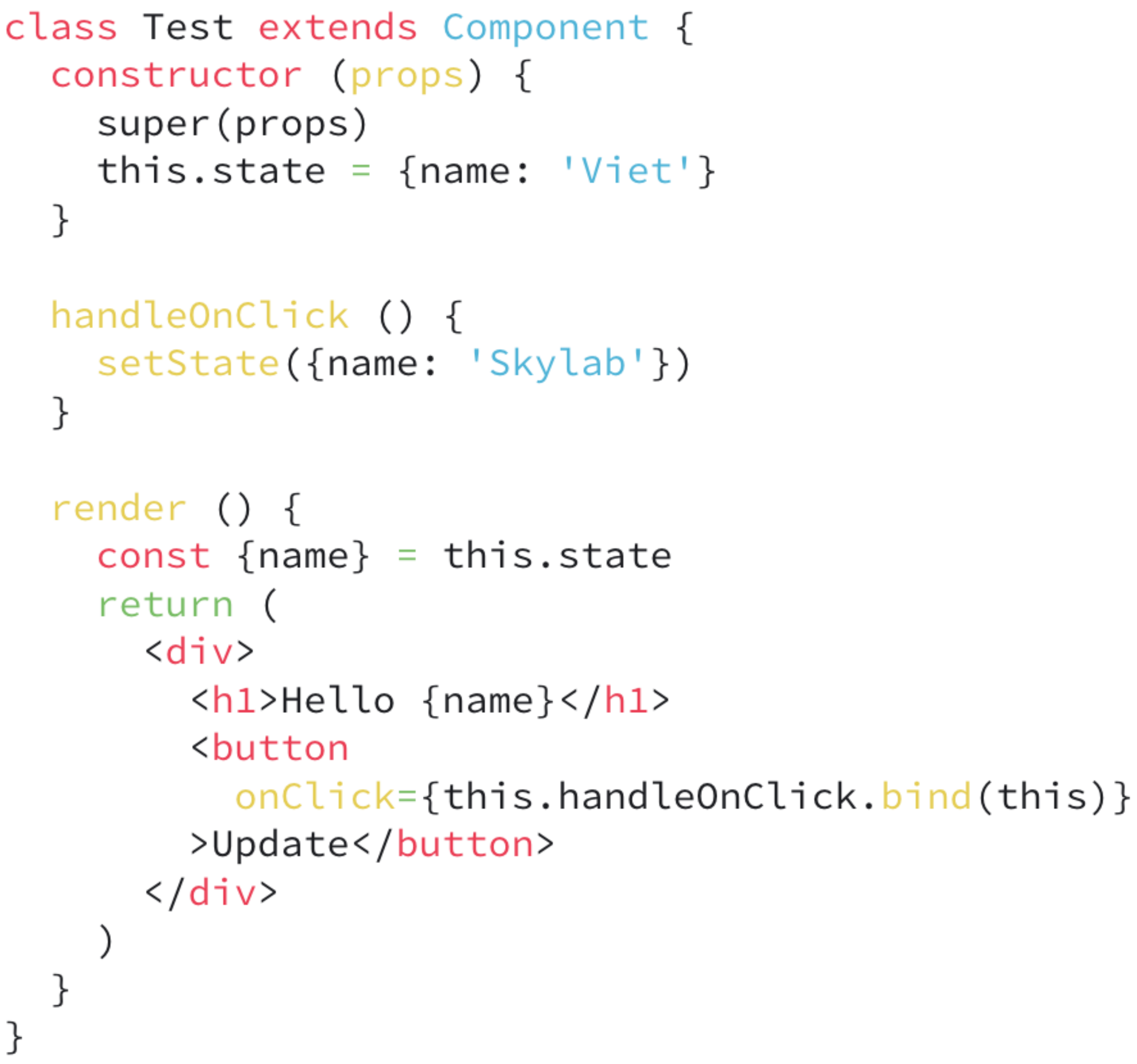
```
<template>
  <div>Test</div>
</template>

<script>
export default {
  name: 'Test'
}
</script>

<style scoped>
</style>
```

Single file component

```
Vue.component('test', {
  template: '<div>Test</div>'
})
```

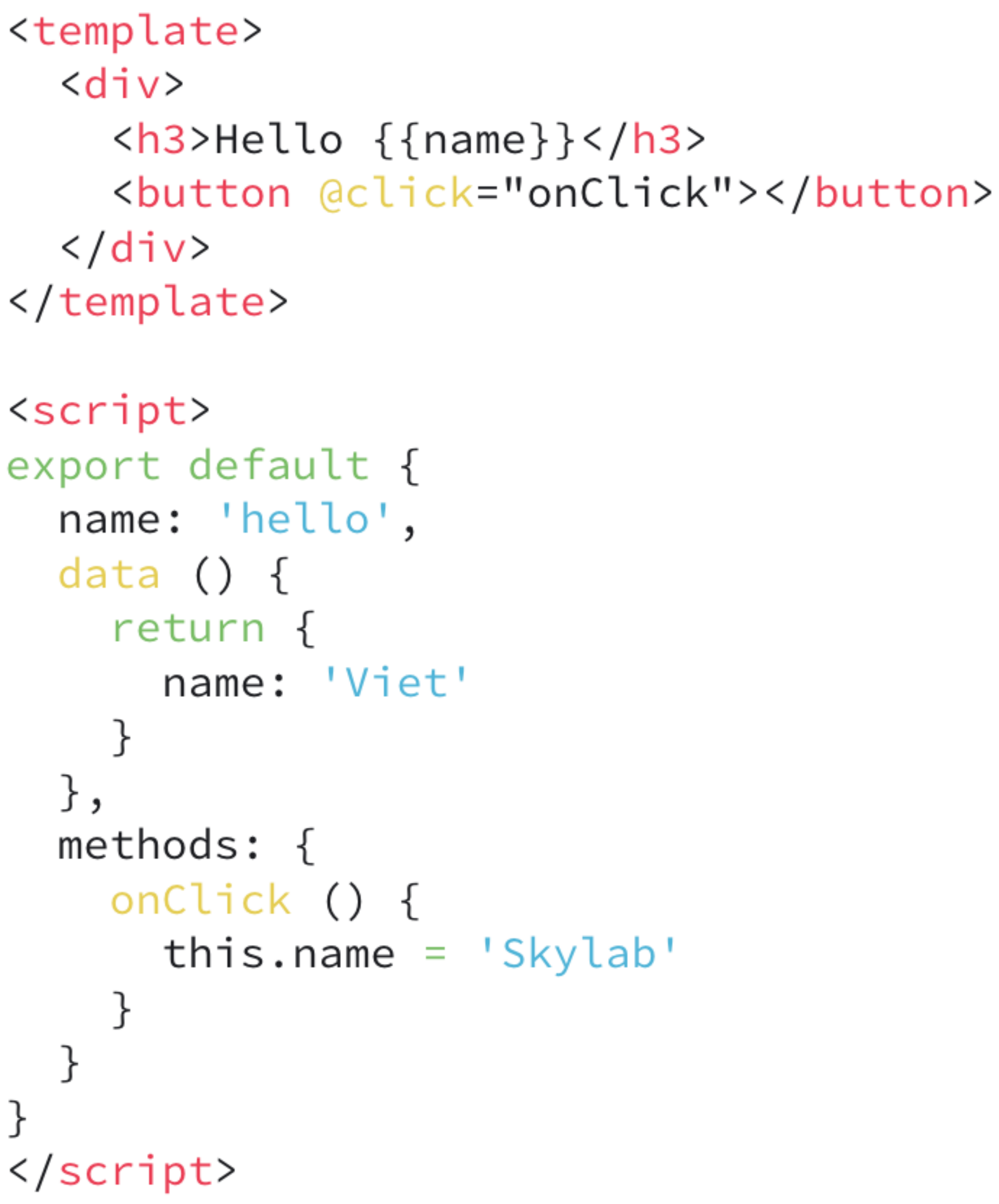
Dynamic component/Register component

Updating and events (React)



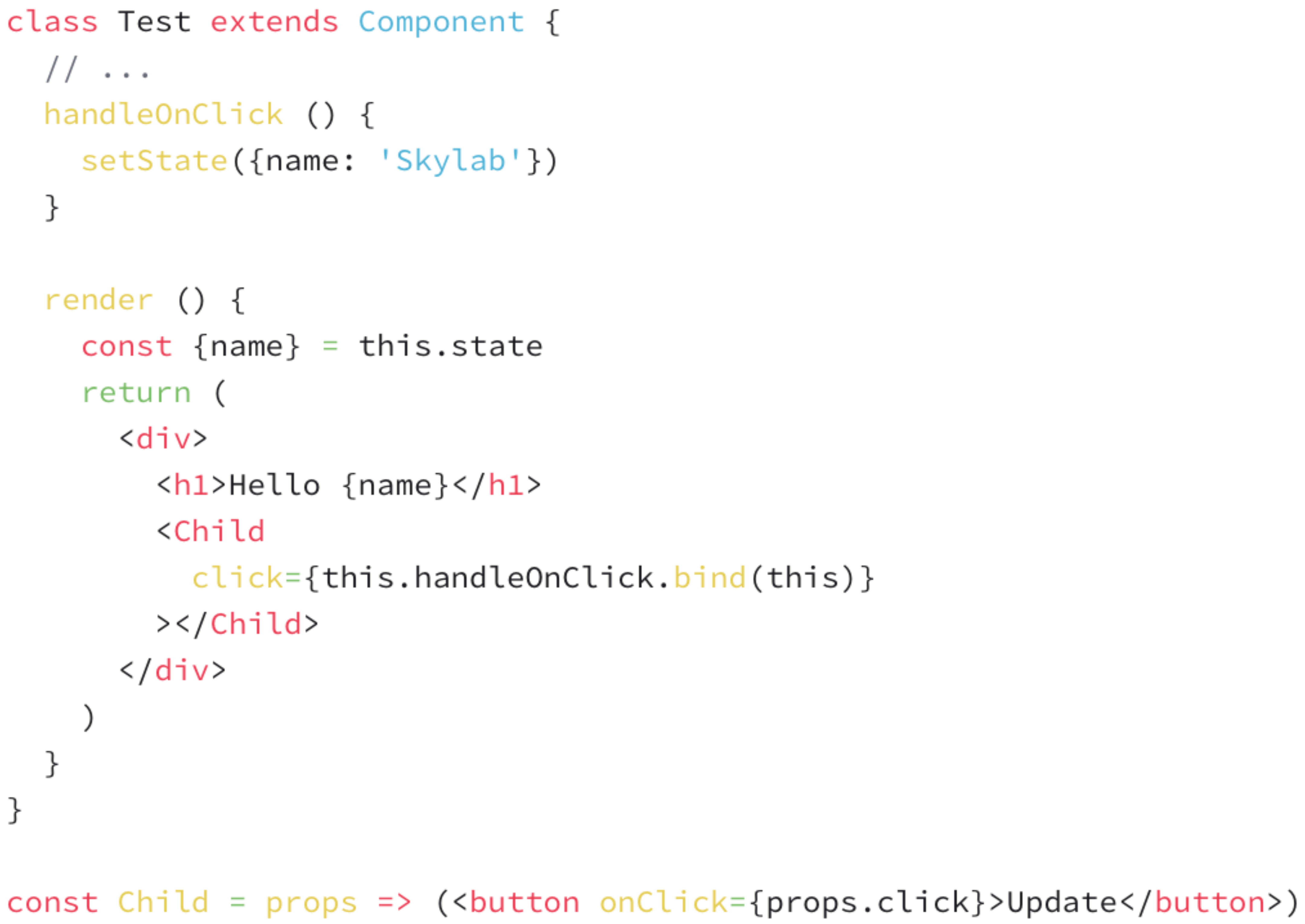
```
class Test extends Component {
  constructor (props) {
    super(props)
    this.state = {name: 'Viet'}
  }

  handleOnClick () {
    setState({name: 'Skylab'})
  }

  render () {
    const {name} = this.state
    return (
      <div>
        <h1>Hello {name}</h1>
        <button
          onClick={this.handleOnClick.bind(this)}
        >Update</button>
      </div>
    )
  }
}
```

Updating and events (Vue)



```
<template>
  <div>
    <h3>Hello {{name}}</h3>
    <button @click="onClick"></button>
  </div>
</template>

<script>
export default {
  name: 'hello',
  data () {
    return {
      name: 'Viet'
    }
  },
  methods: {
    onClick () {
      this.name = 'Skylab'
    }
  }
}
</script>
```

Passing events to children (React)



```
class Test extends Component {
  // ...
  handleOnClick () {
    setState({name: 'Skylab'})
  }

  render () {
    const {name} = this.state
    return (
      <div>
        <h1>Hello {name}</h1>
        <Child
          click={this.handleOnClick.bind(this)}
        ></Child>
      </div>
    )
  }
}

const Child = props => (<button onClick={props.click}>Update</button>)
```

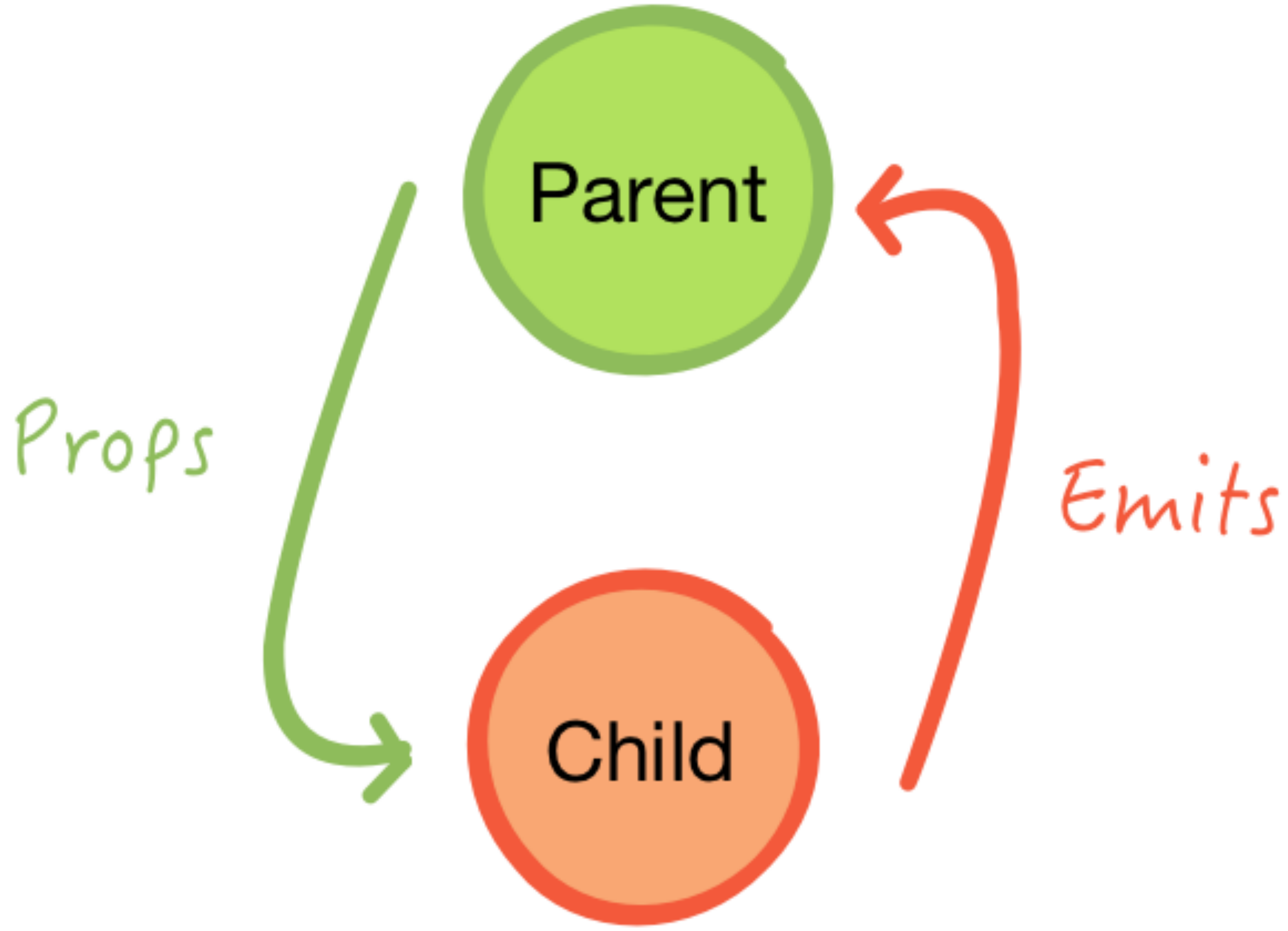
Passing events to children (Vue)

Test.vue

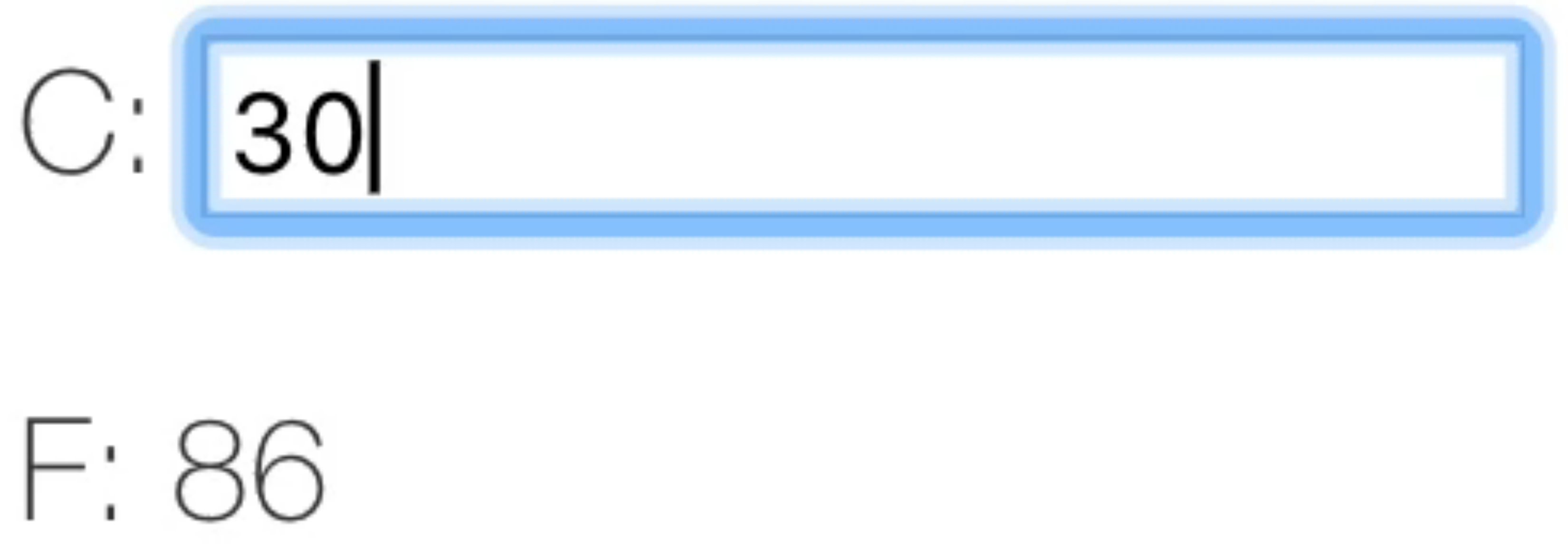
```
<template>
  <div>
    <h3>Hello {{name}}</h3>
    <child @change="onChange" />
  </div>
</template>

<script>
export default {
  name: 'hello',
  data () {
    return {
      name: 'Viet'
    }
  },
  methods: {
    onChange (val) {
      this.name = val
    }
  }
}
</script>
```

Child.vue

```
<template>
  <button @click="$emit('change',
'Skylab')">Update</button>
</template>

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



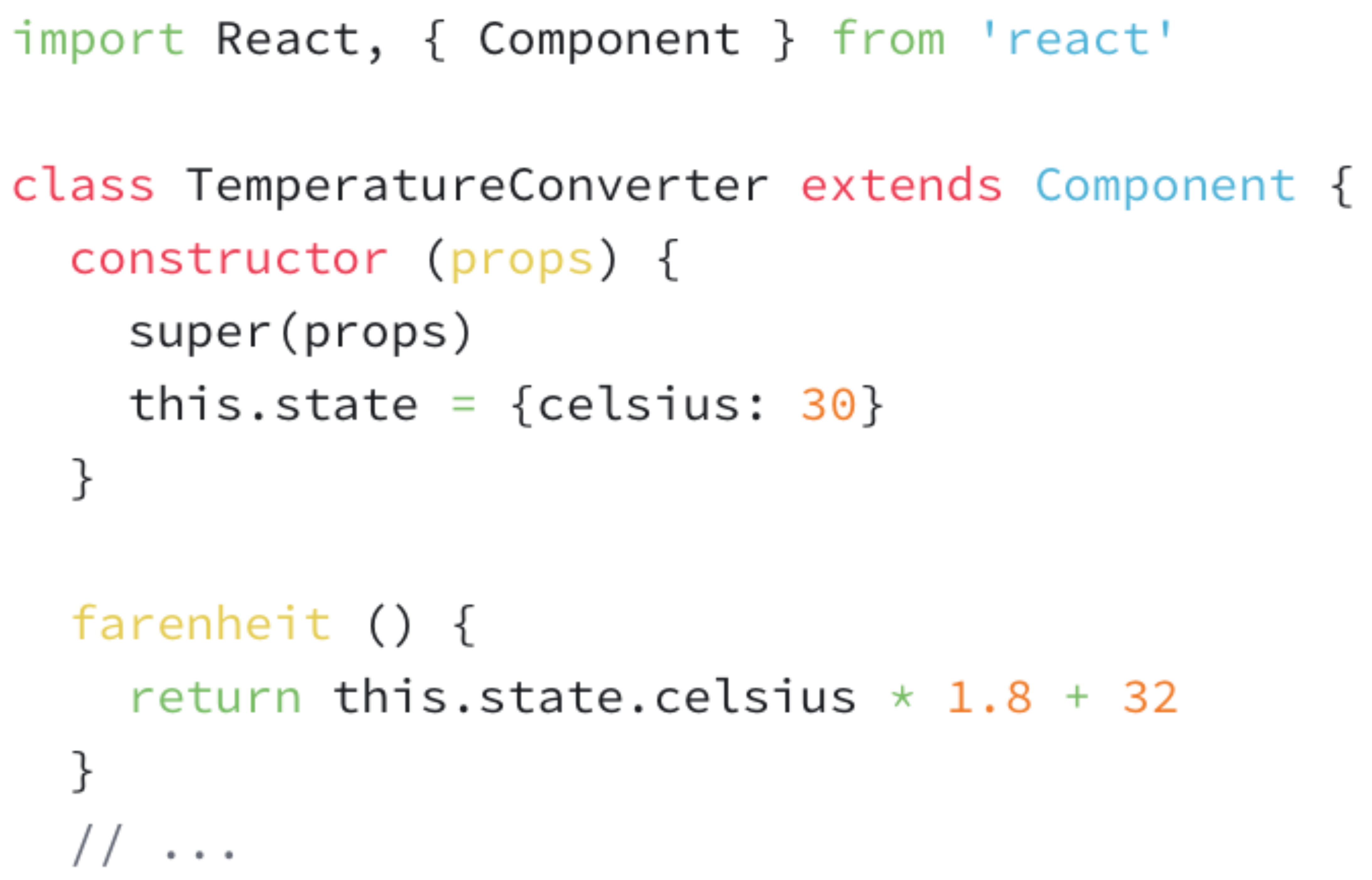
Two-way data binding



C: 30

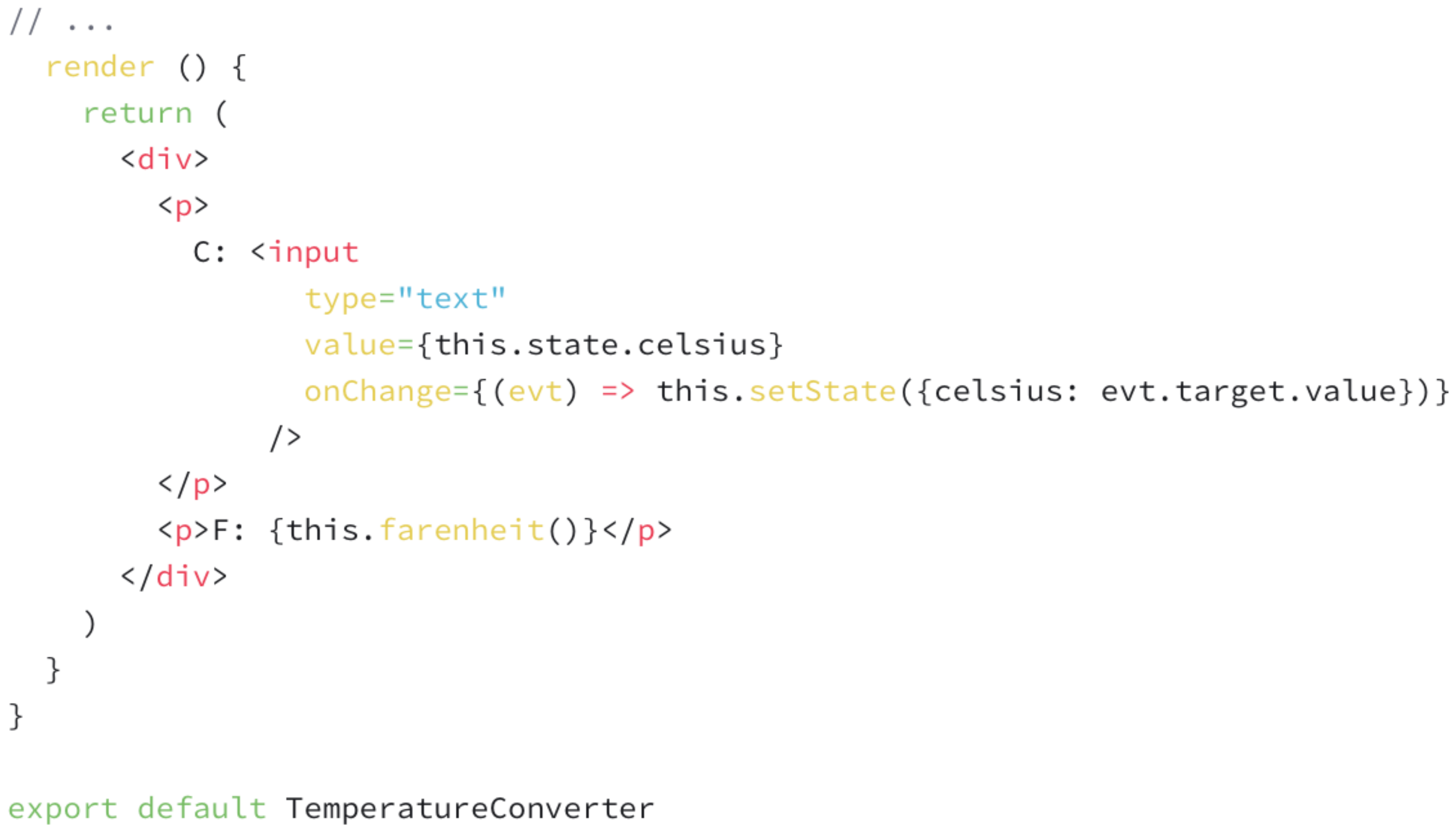
F: 86

Two-way data binding (React)



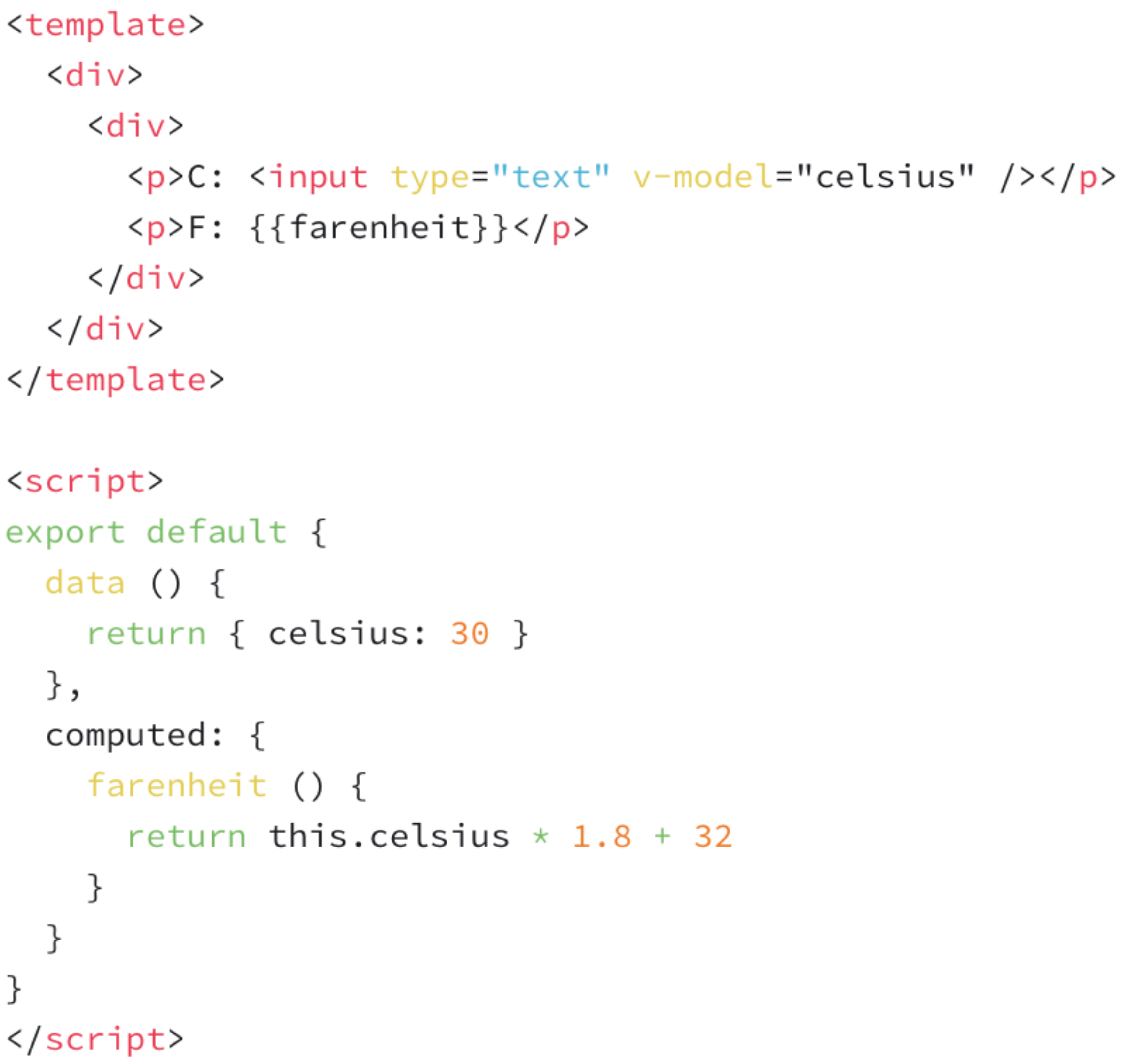
```
import React, { Component } from 'react'

class TemperatureConverter extends Component {
  constructor (props) {
    super(props)
    this.state = {celsius: 30}
  }

  farenheit () {
    return this.state.celsius * 1.8 + 32
  }
  // ...
```

Two-way data binding (React)



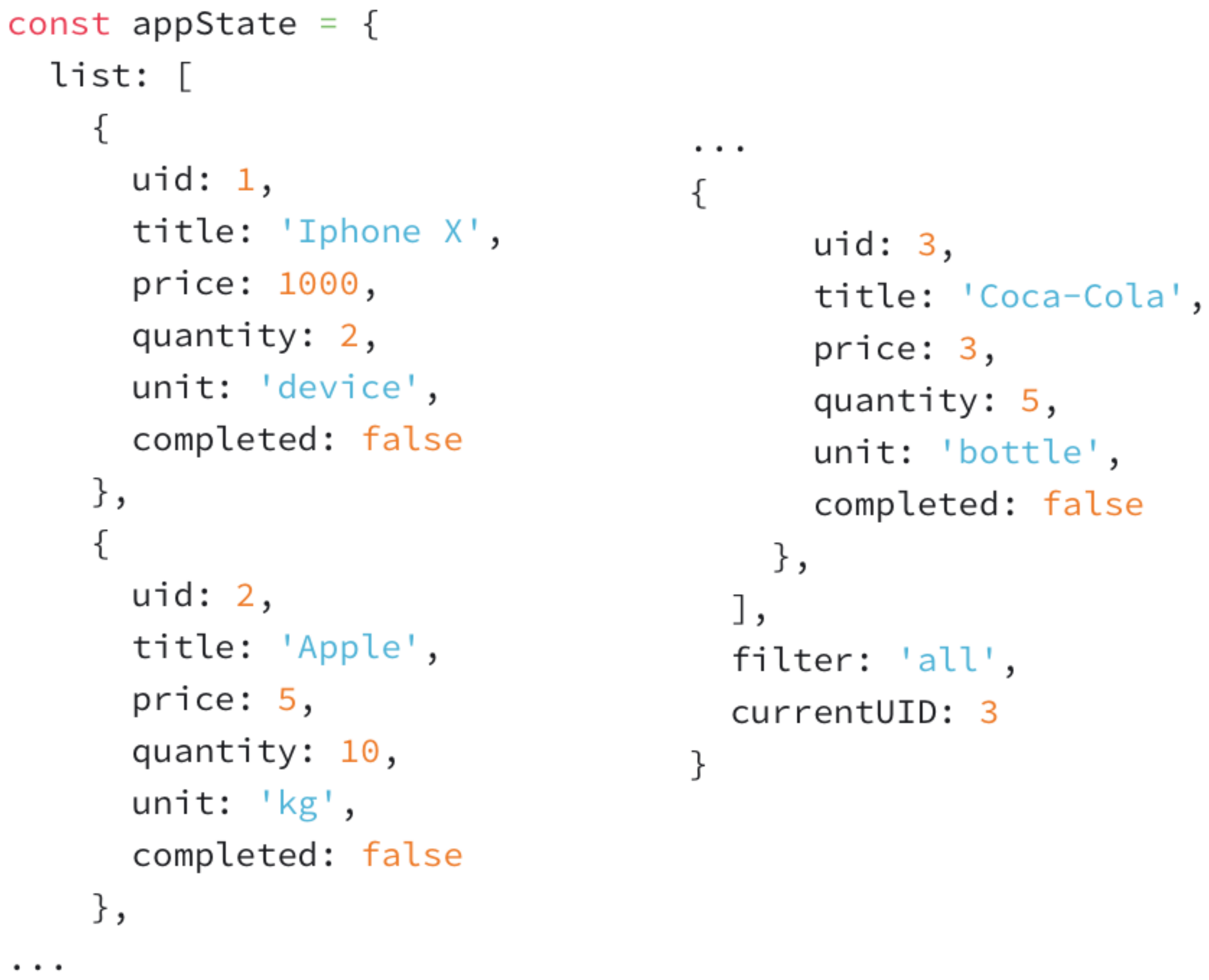
```
// ...
  render () {
    return (
      <div>
        <p>
          C: <input
                type="text"
                value={this.state.celsius}
                onChange={(evt) => this.setState({celsius: evt.target.value})}
              />
        </p>
        <p>F: {this.farenheit()}</p>
      </div>
    )
  }
}

export default TemperatureConverter
```

Two-way data binding (Vue)



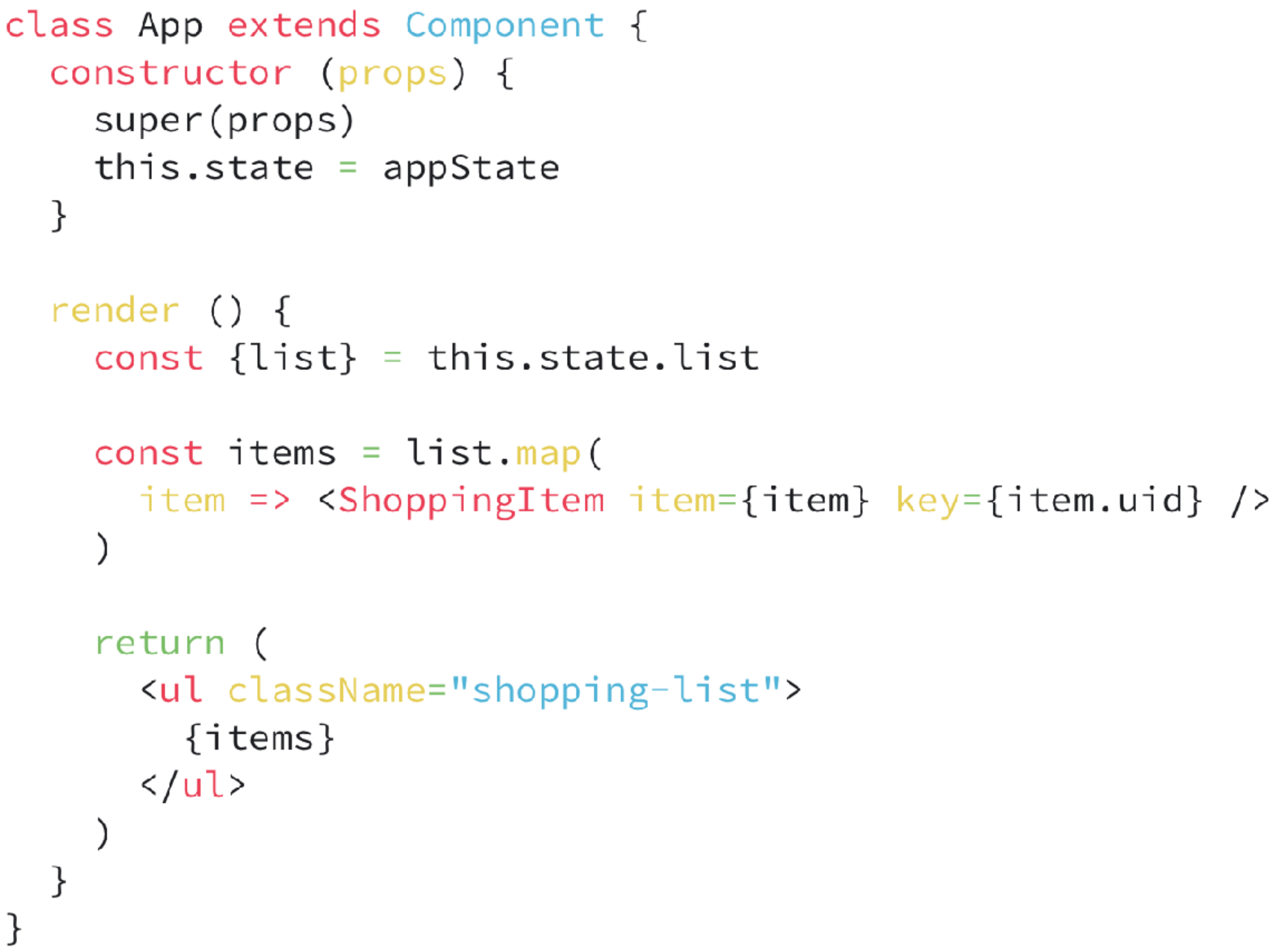
```
<template>
  <div>
    <div>
      <p>C: <input type="text" v-model="celsius" /></p>
      <p>F: {{farenheit}}</p>
    </div>
  </div>
</template>

<script>
export default {
  data () {
    return { celsius: 30 }
  },
  computed: {
    farenheit () {
      return this.celsius * 1.8 + 32
    }
  }
}
</script>
```

App State for Shopping List



```
const appState = {
  list: [
    {
      uid: 1,
      title: 'Iphone X',
      price: 1000,
      quantity: 2,
      unit: 'device',
      completed: false
    },
    {
      uid: 2,
      title: 'Apple',
      price: 5,
      quantity: 10,
      unit: 'kg',
      completed: false
    },
...
...
{
      uid: 3,
      title: 'Coca-Cola',
      price: 3,
      quantity: 5,
      unit: 'bottle',
      completed: false
    },
  ],
  filter: 'all',
  currentUID: 3
}
```

Render with iterating in React

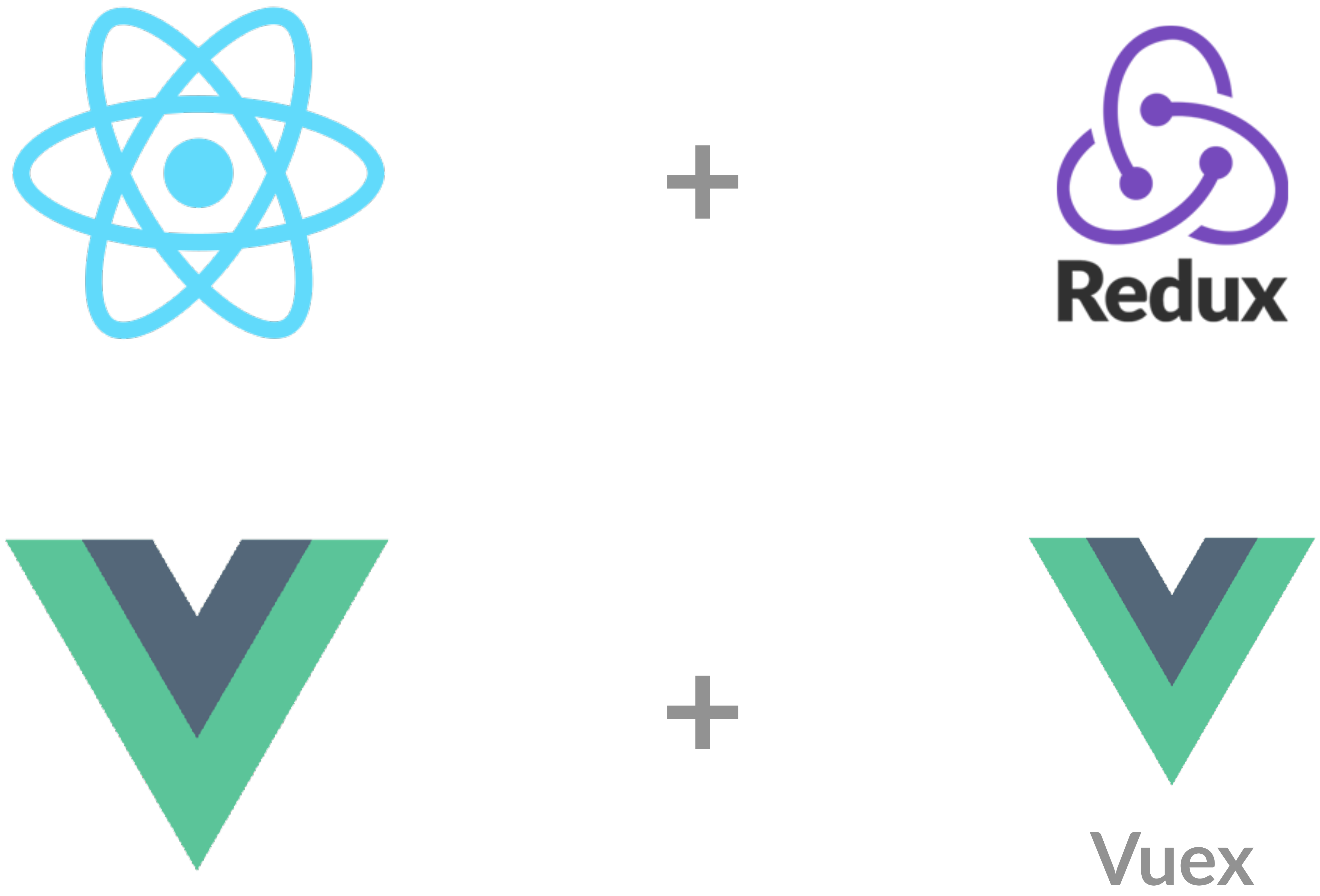


```
class App extends Component {
  constructor (props) {
    super(props)
    this.state = appState
  }

  render () {
    const {list} = this.state.list

    const items = list.map(
      item => <ShoppingItem item={item} key={item.uid} />
    )

    return (
      <ul className="shopping-list">
        {items}
      </ul>
    )
  }
}
```

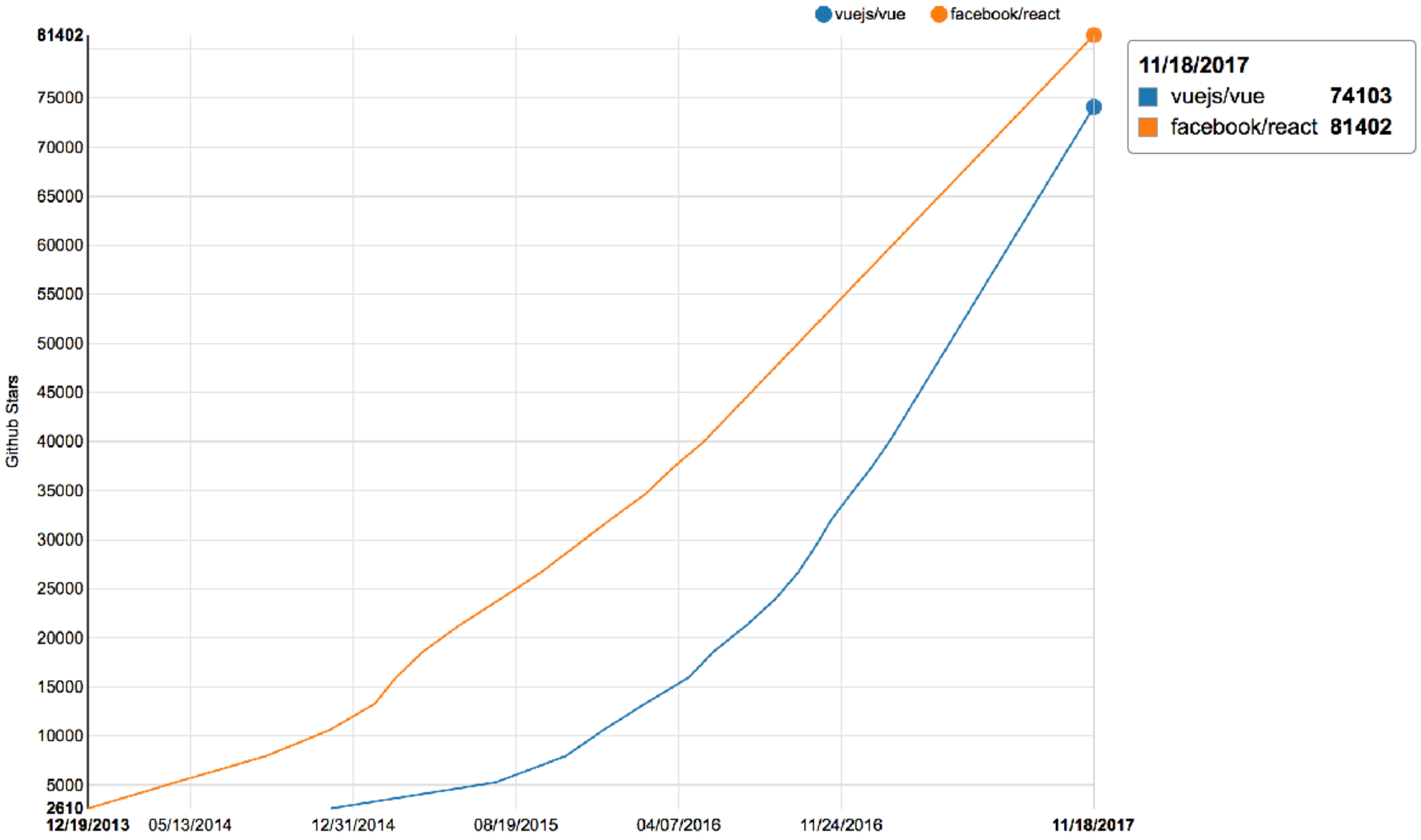
Render with iterating in Vue

```
<template>
  <ul>
    <shopping-item v-for="item in list" :key="item.uid" :item="item"/>
  </ul>
</template>

<script>
import ShoppingItem from '@/components/ShoppingItem'
export default {
  data () {
    return appState
  },
  computed () {
    list () {
      return appState.list
    }
  }
  components: {
    ShoppingItem
  }
}
</script>
```

State Management

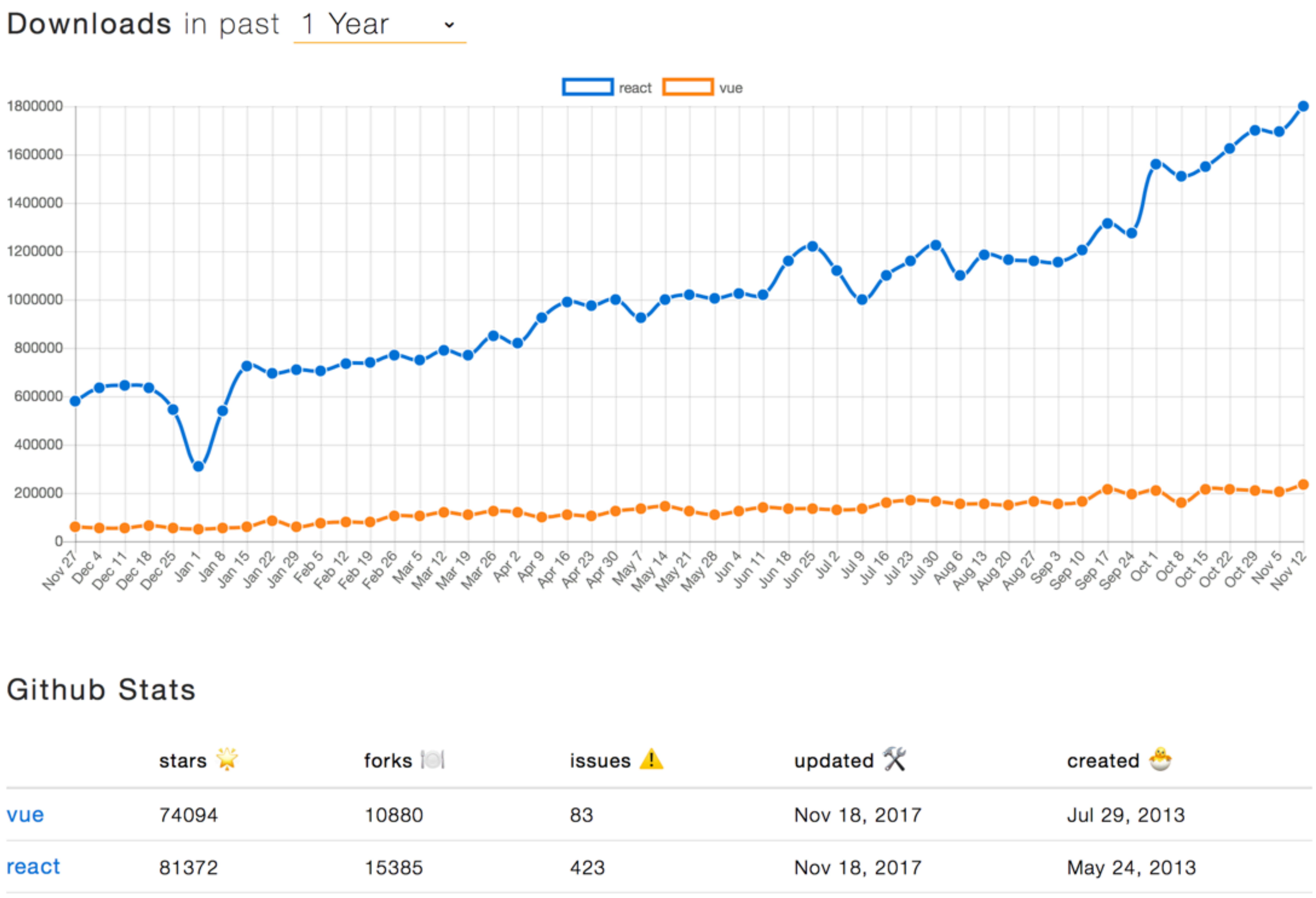


Statistics - Star History



http://www.timqian.com/star-history/#vuejs/vue&facebook/react

Statistics - NPM Trends



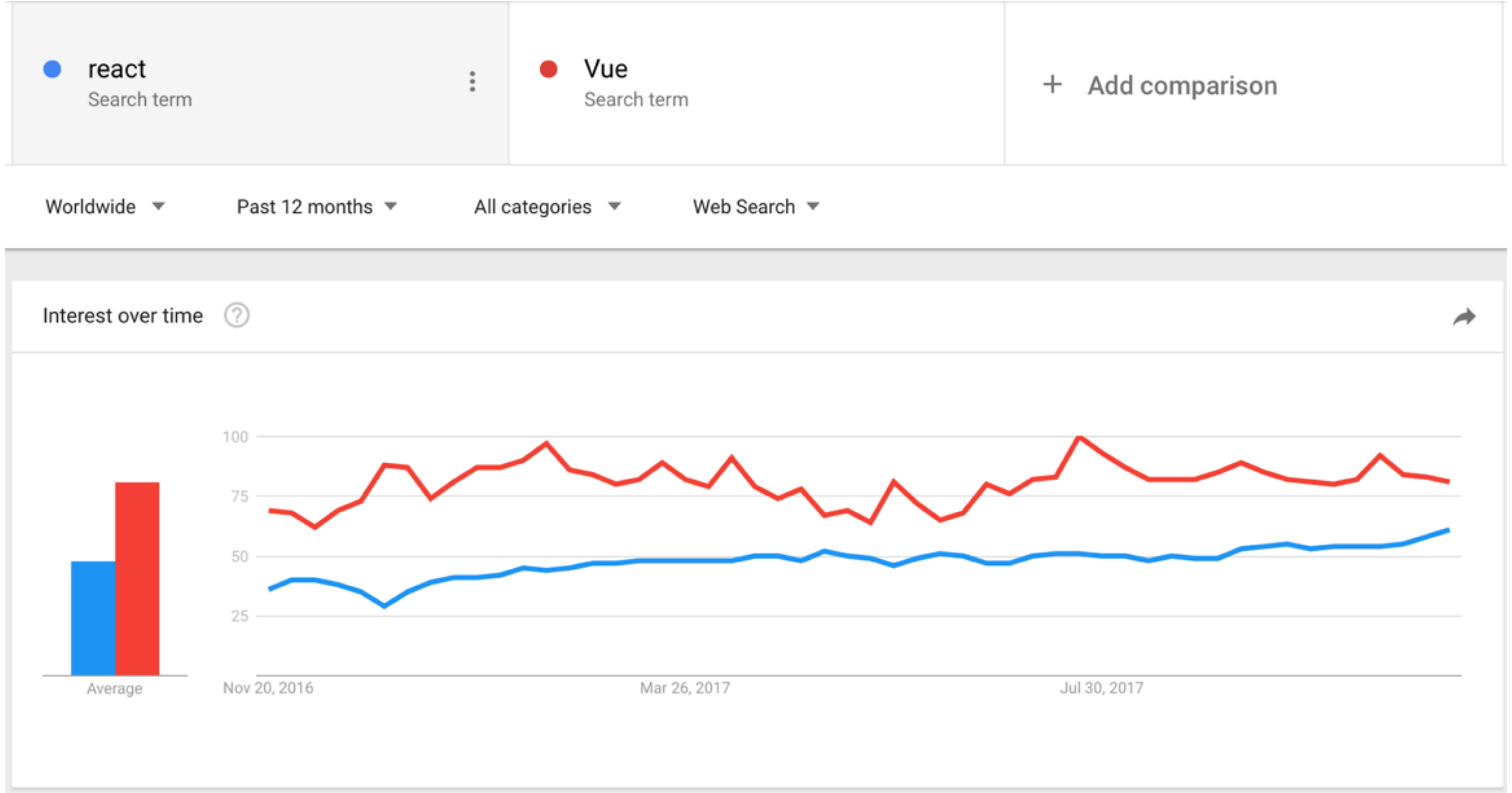
Downloads in past 1 Year ⌄

Github Stats

	stars 🌟	forks 🍽️	issues ⚠️	updated 🛠️	created 🐣
vue	74094	10880	83	Nov 18, 2017	Jul 29, 2013
react	81372	15385	423	Nov 18, 2017	May 24, 2013

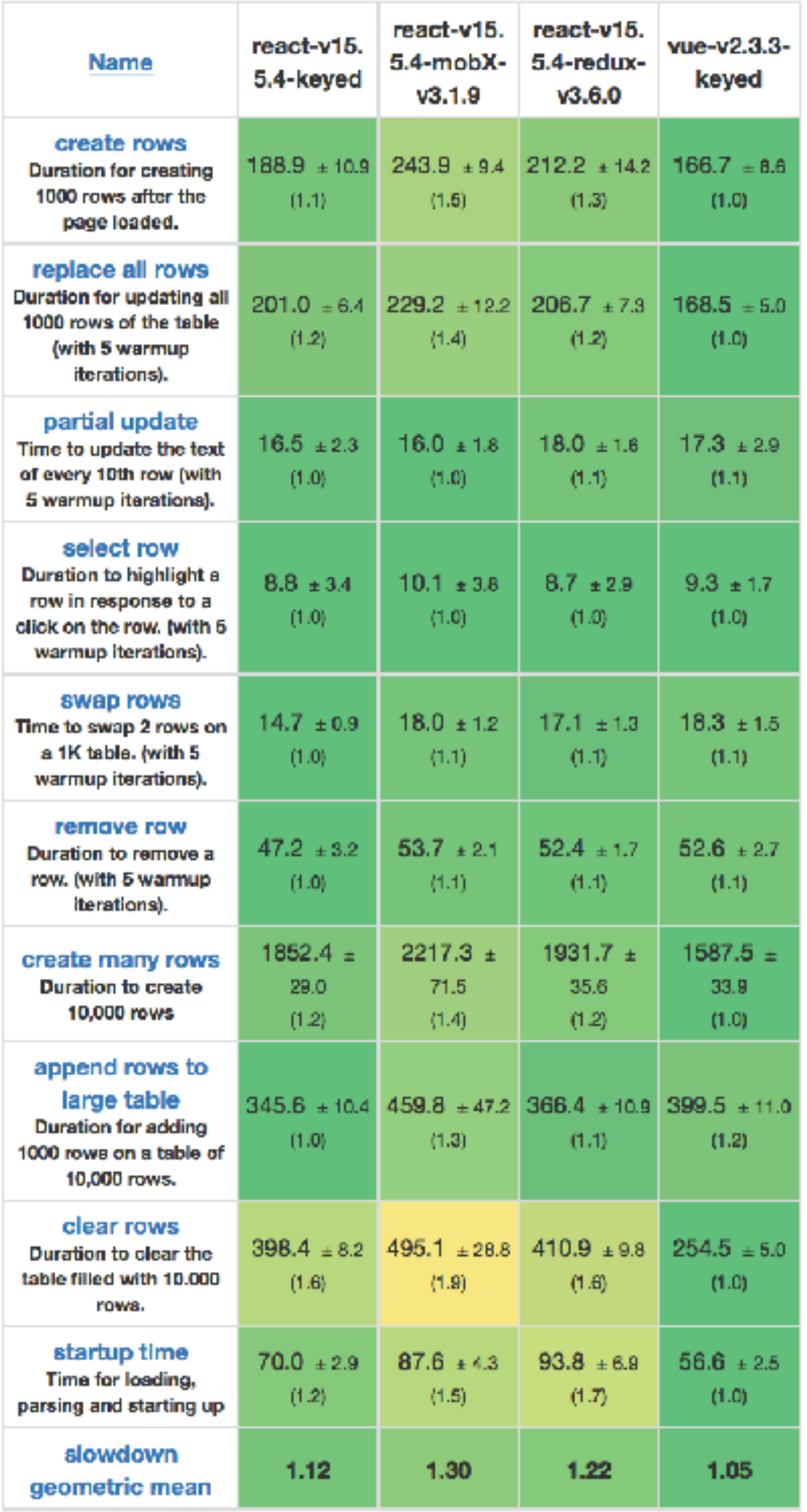
http://www.npmtrends.com/react-vs-vue

Statistics - Google Trends



https://trends.google.com/trends/explore?q=react,Vue

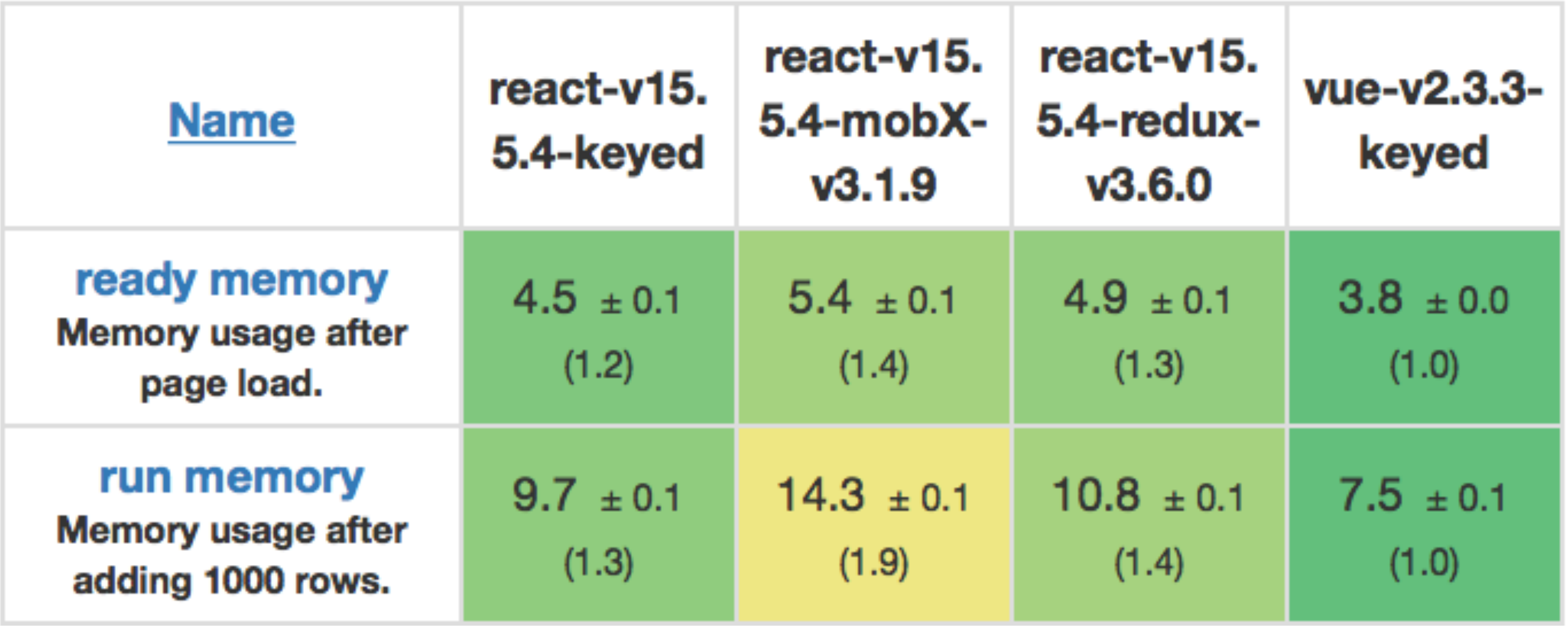
Performance - Duration in milliseconds



Name	react-v15.5.4-keyed	react-v15.5.4-mobX-v3.1.9	react-v15.5.4-redux-v3.6.0	vue-v2.3.3-keyed
create rows Duration for creating 1000 rows after the page loaded.	188.9 ± 10.9 (1.1)	243.9 ± 9.4 (1.5)	212.2 ± 14.2 (1.3)	166.7 ± 8.6 (1.0)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	201.0 ± 6.4 (1.2)	229.2 ± 12.2 (1.4)	206.7 ± 7.3 (1.2)	168.5 ± 5.0 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	16.5 ± 2.3 (1.0)	16.0 ± 1.8 (1.0)	18.0 ± 1.6 (1.1)	17.3 ± 2.9 (1.1)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	8.8 ± 3.4 (1.0)	10.1 ± 3.8 (1.0)	8.7 ± 2.9 (1.0)	9.3 ± 1.7 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	14.7 ± 0.9 (1.0)	18.0 ± 1.2 (1.1)	17.1 ± 1.3 (1.1)	18.3 ± 1.5 (1.1)
remove row Duration to remove a row. (with 5 warmup iterations).	47.2 ± 3.2 (1.0)	53.7 ± 2.1 (1.1)	52.4 ± 1.7 (1.1)	52.6 ± 2.7 (1.1)
create many rows Duration to create 10,000 rows	1852.4 ± 29.0 (1.2)	2217.3 ± 71.5 (1.4)	1931.7 ± 35.6 (1.2)	1587.5 ± 33.9 (1.0)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	345.6 ± 10.4 (1.0)	459.8 ± 47.2 (1.3)	366.4 ± 10.9 (1.1)	399.5 ± 11.0 (1.2)
clear rows Duration to clear the table filled with 10.000 rows.	398.4 ± 8.2 (1.6)	495.1 ± 28.8 (1.9)	410.9 ± 9.8 (1.6)	254.5 ± 5.0 (1.0)
startup time Time for loading, parsing and starting up	70.0 ± 2.9 (1.2)	87.6 ± 4.3 (1.5)	93.8 ± 6.9 (1.7)	56.6 ± 2.5 (1.0)
slowdown geometric mean	**1.12**	**1.30**	**1.22**	**1.05**

http://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html

Performance - Memory Size



Name	react-v15.5.4-keyed	react-v15.5.4-mobX-v3.1.9	react-v15.5.4-redux-v3.6.0	vue-v2.3.3-keyed
ready memory Memory usage after page load.	4.5 ± 0.1 (1.2)	5.4 ± 0.1 (1.4)	4.9 ± 0.1 (1.3)	3.8 ± 0.0 (1.0)
run memory Memory usage after adding 1000 rows.	9.7 ± 0.1 (1.3)	14.3 ± 0.1 (1.9)	10.8 ± 0.1 (1.4)	7.5 ± 0.1 (1.0)

http://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html

No framework/library is better

Just we are better

Q & A

Thank you

Viet Tran - CTO Skylab
viettranx@gmail.com