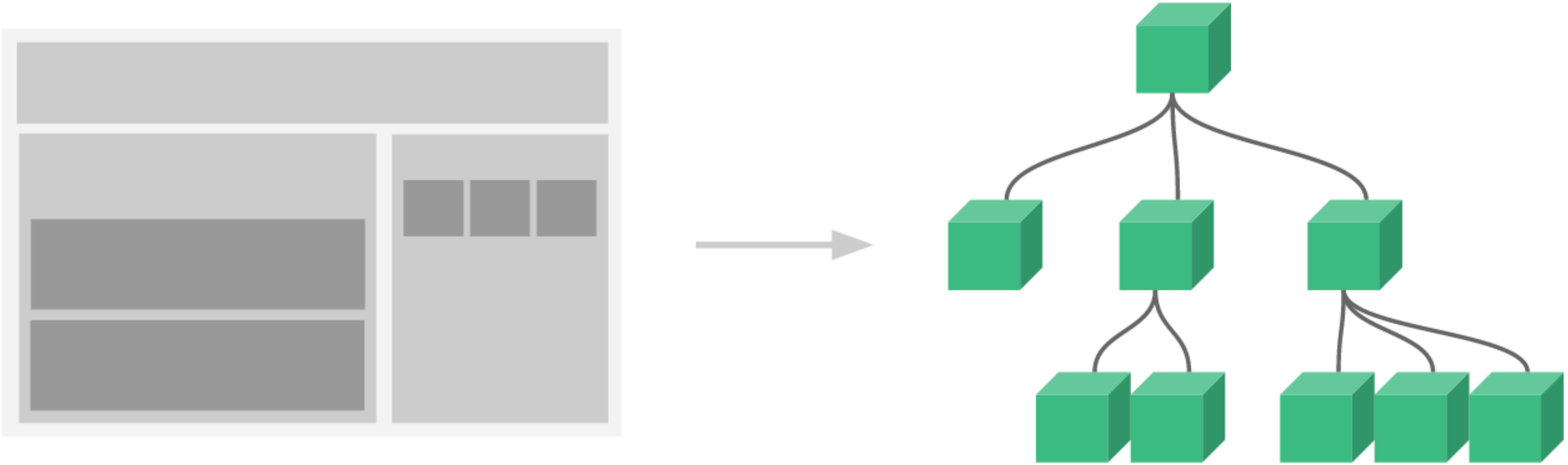
Component

VueJs

Thavorac

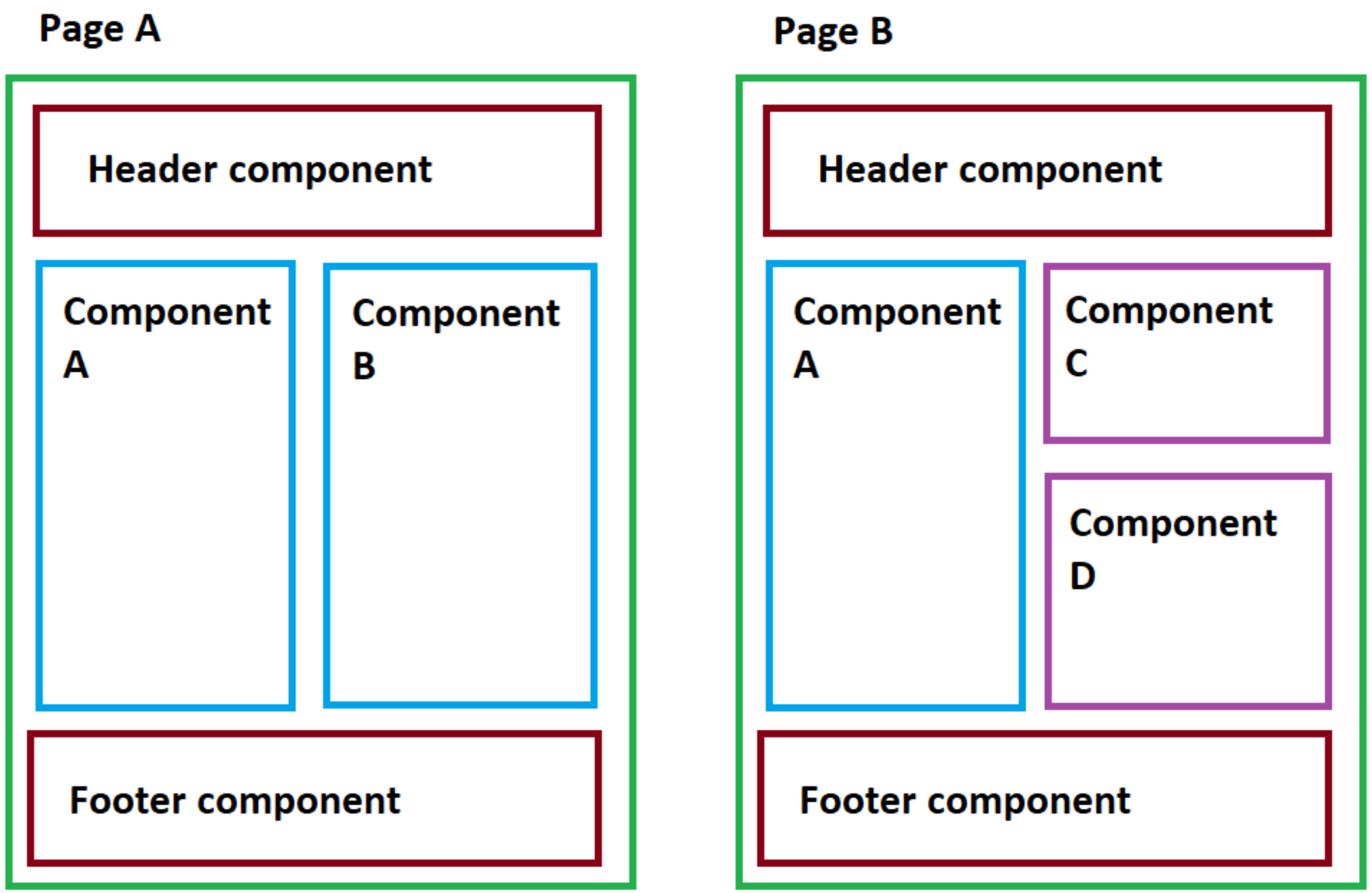


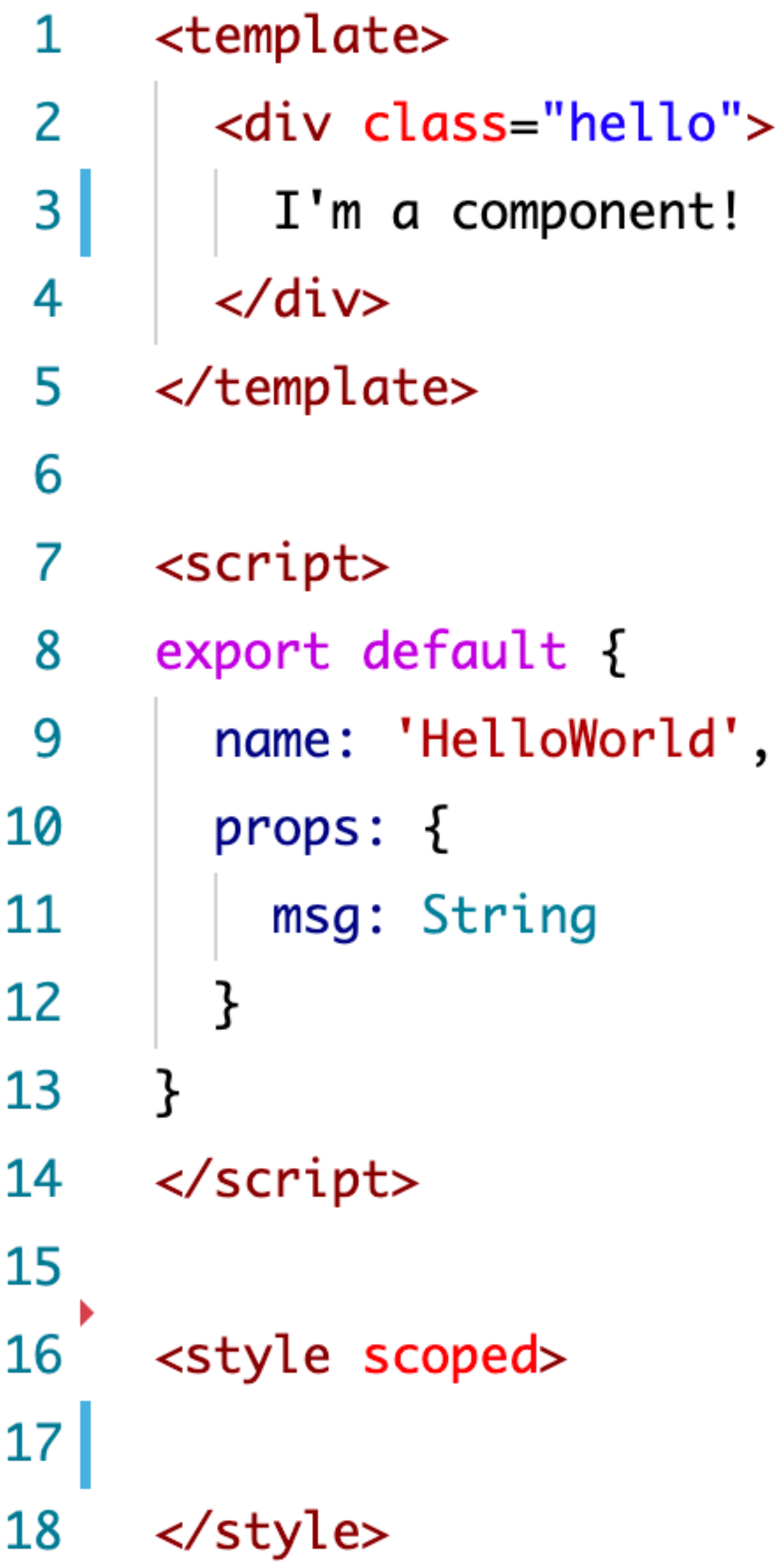
VueJs app is basically a component tree

What is VueJs Component?

The **component** is just like a Vue instance which has its own lifecycle, and has the same property as Vue instance.

The Vue component is attached itself to other component or Vue instance and can be re-use multiple time.





```
<template>
  <div class="hello">
    I'm a component!
  </div>
</template>

<script>
export default {
  name: 'HelloWorld',
  props: {
    msg: String
  }
}
</script>

<style scoped>

</style>
```

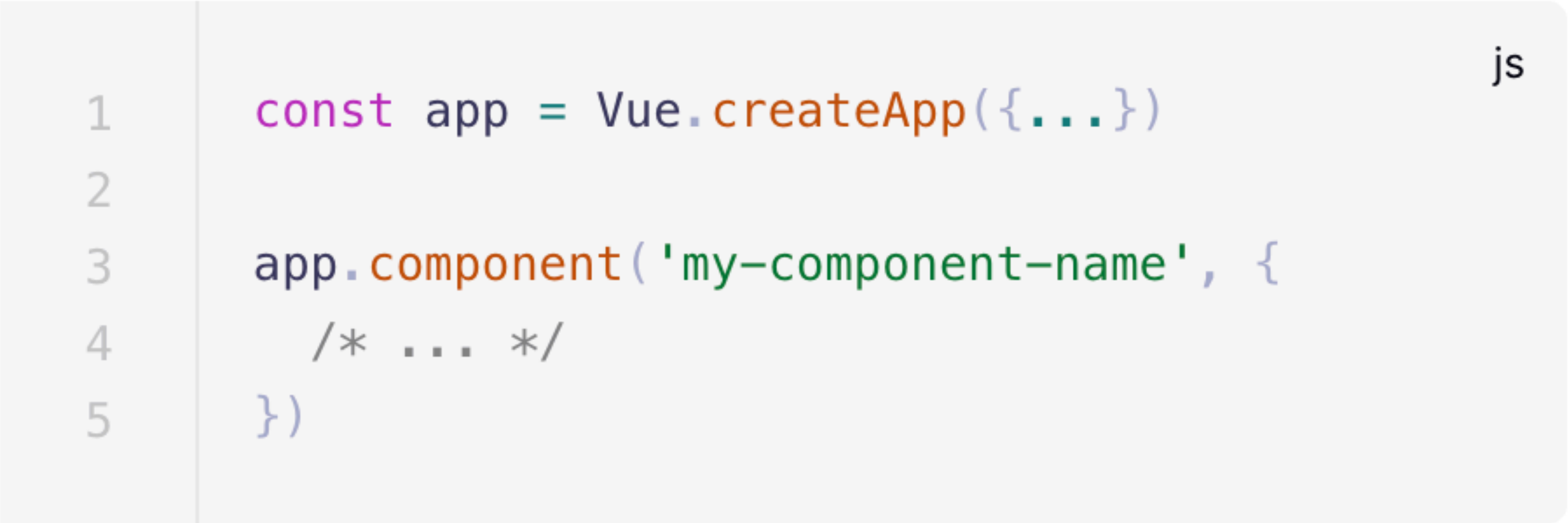
Here is how a component looks like.

There 3 sections, just like a Vue instance:

- HTML Template
- Script
- Style

Component Registration

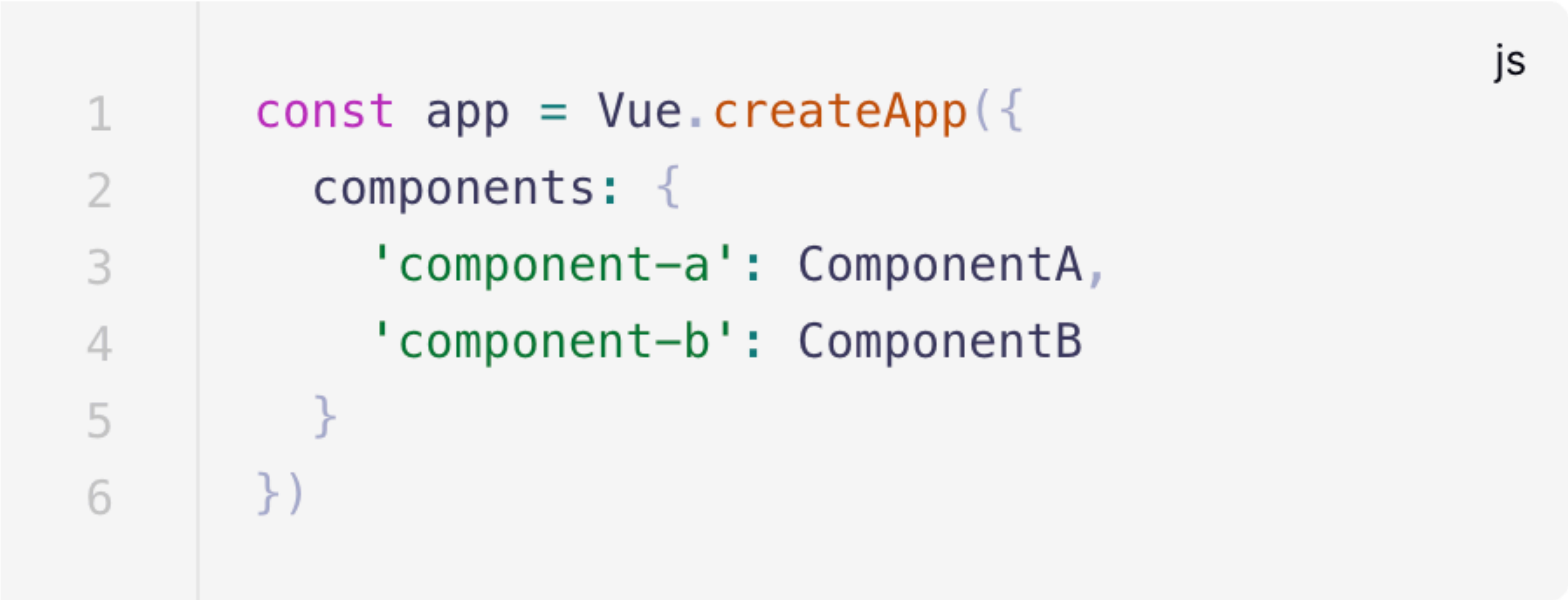
Global



```js
const app = Vue.createApp({...})

app.component('my-component-name', {
  /* ... */
})
```

It can be used by any components of this Vue instance.

Local



```js
const app = Vue.createApp({
  components: {
    'component-a': ComponentA,
    'component-b': ComponentB
  }
})
```

It can not be used by any sub component.

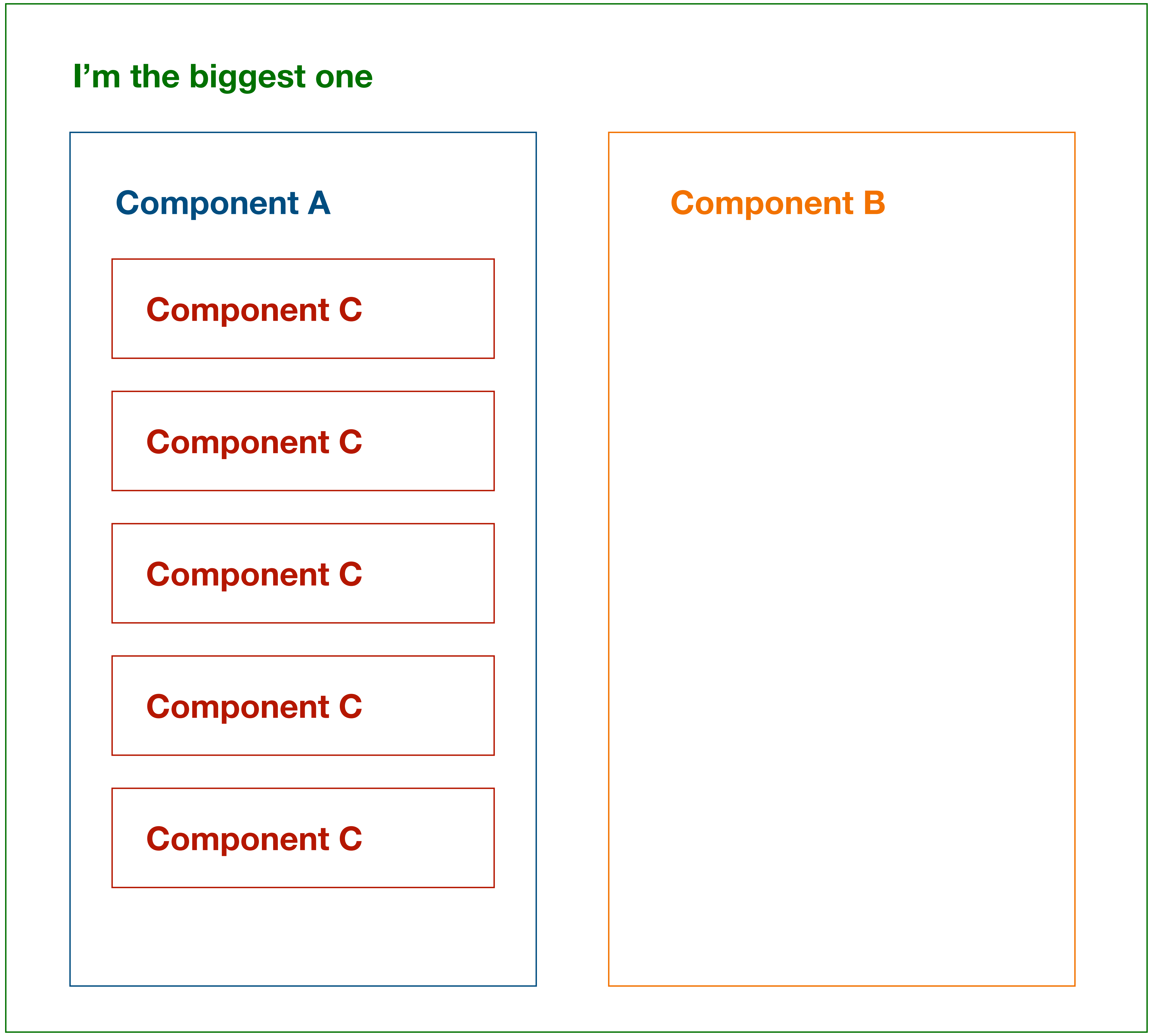
Exercise 1.

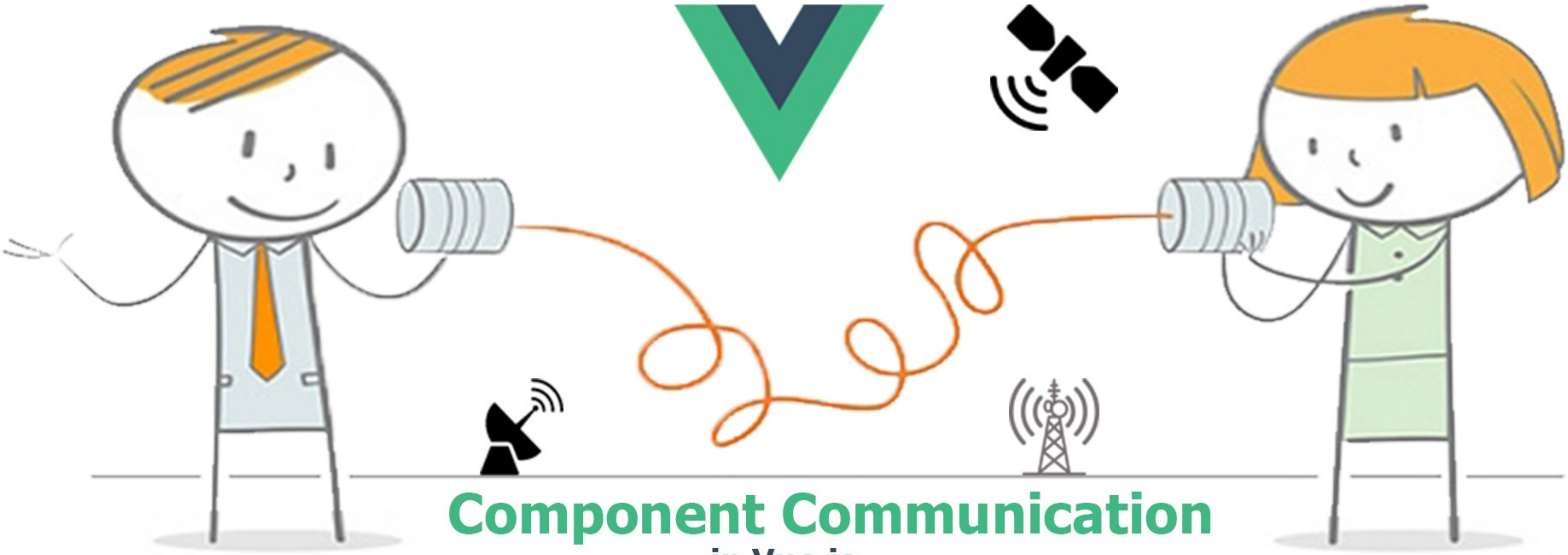
Create a page like this:

There are 3 components:

- Component A
- Component B
- Component C

You can either register this component locally or globally.



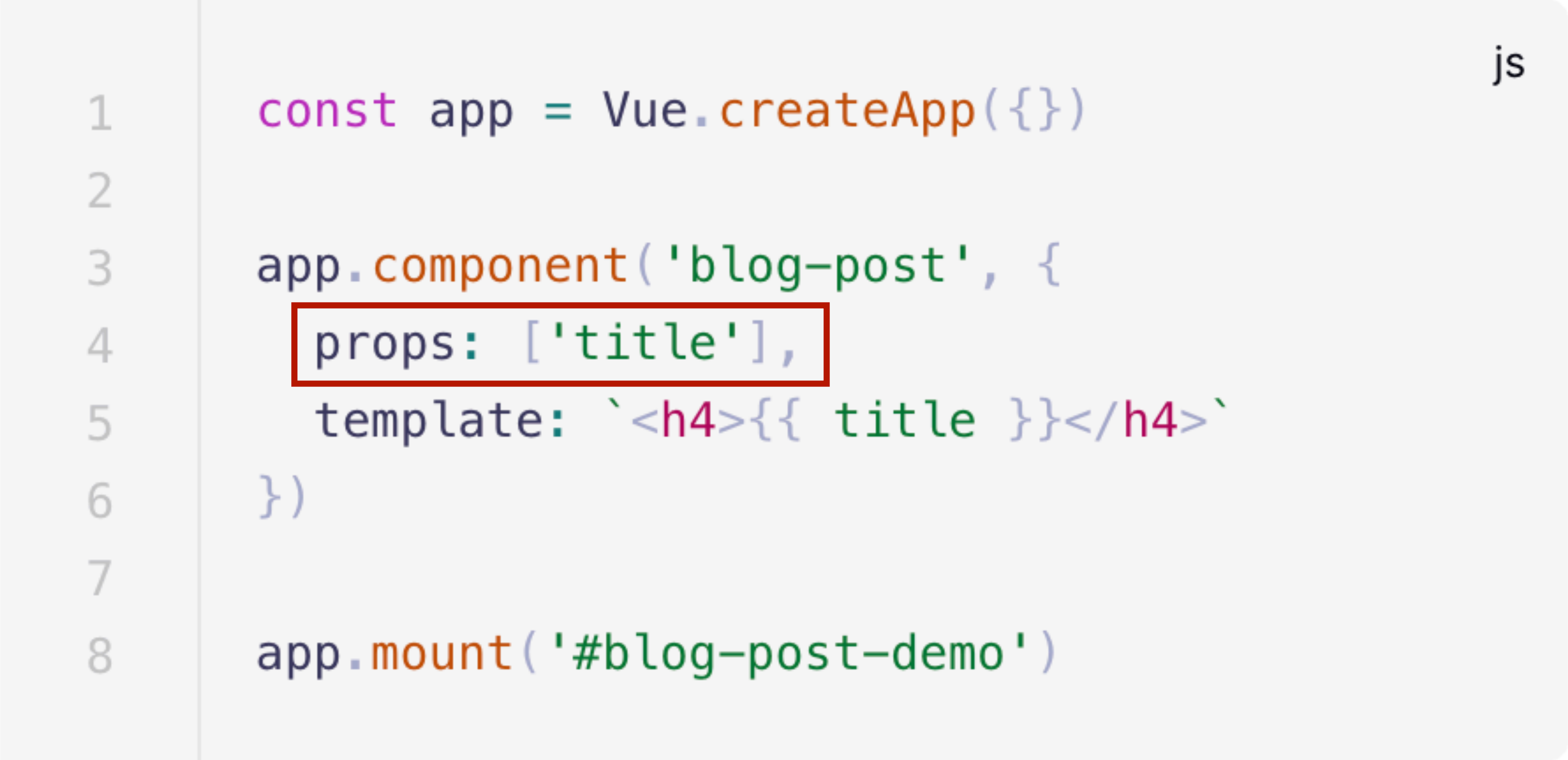


Component Communication

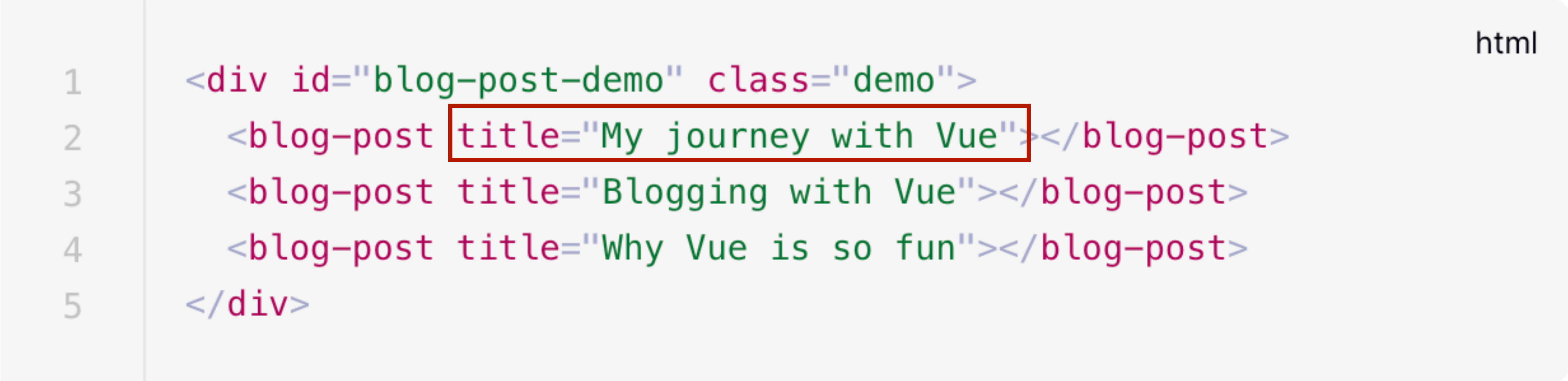
in Vue.js

Passing data from parent to child

Use “props”



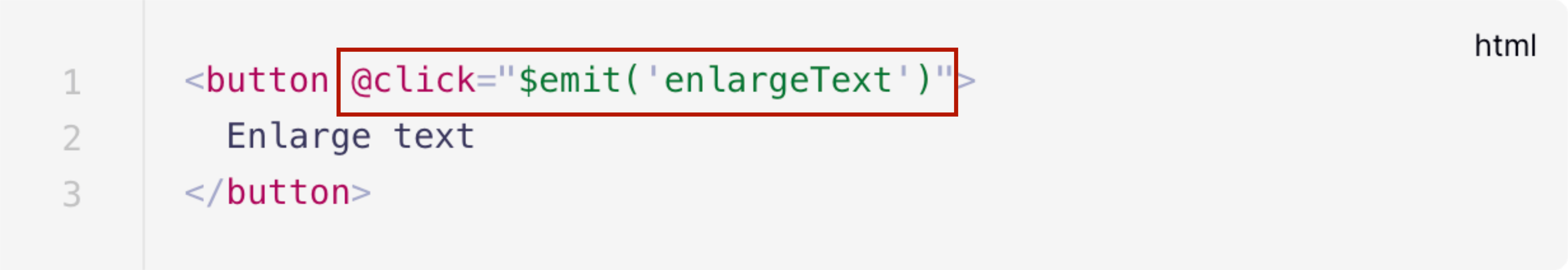
```js
const app = Vue.createApp({})

app.component('blog-post', {
  props: ['title'],
  template: `<h4>{{ title }}</h4>`
})

app.mount('#blog-post-demo')
```



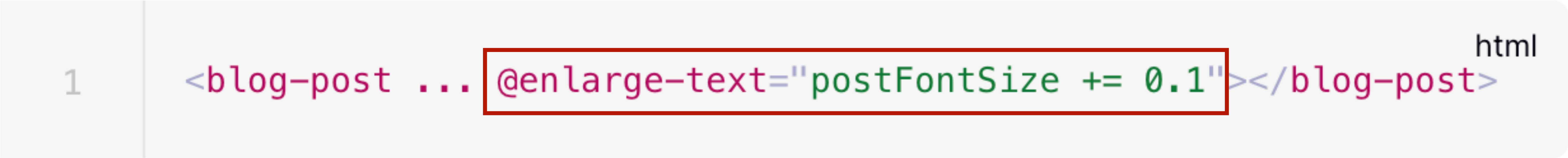
```html
<div id="blog-post-demo" class="demo">
  <blog-post title="My journey with Vue"></blog-post>
  <blog-post title="Blogging with Vue"></blog-post>
  <blog-post title="Why Vue is so fun"></blog-post>
</div>
```

Passing data back from child to parent

Use “$emit”



```html
<button @click="$emit('enlargeText')">
  Enlarge text
</button>
```



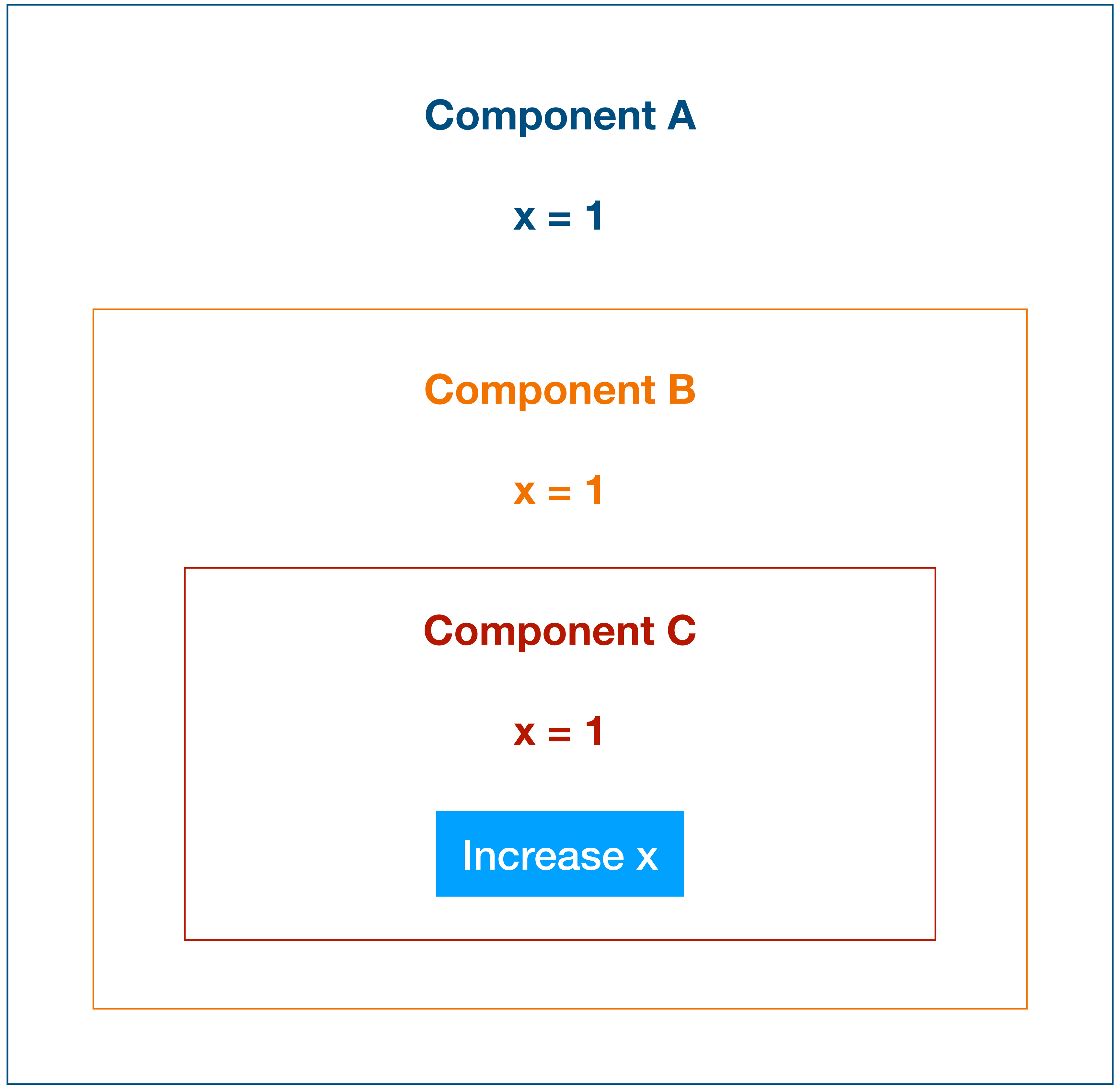
```html
<blog-post ... @enlarge-text="postFontSize += 0.1"></blog-post>
```

Exercise 2.

Create 3 components like the figure:

Value of x is passed from component A until component C.

When you click on button “Increase x” of component C, this value will be updated back to component A



Component A

x = 1

Component B

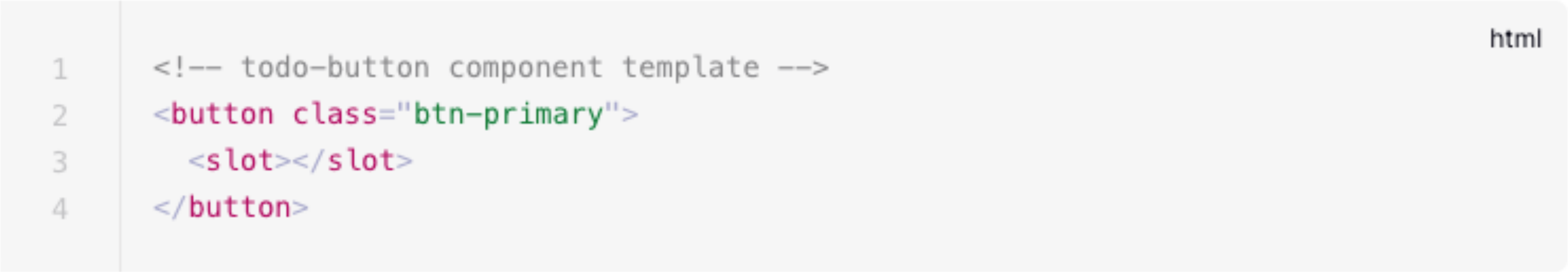
x = 1

Component C

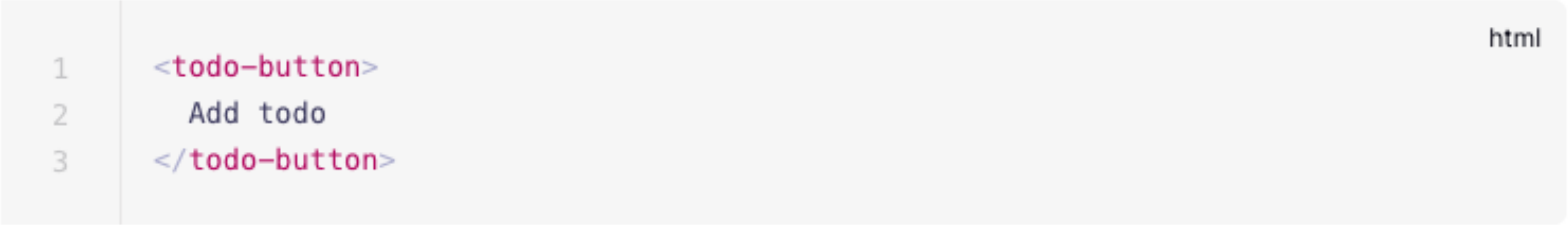
x = 1

Passing content to child component

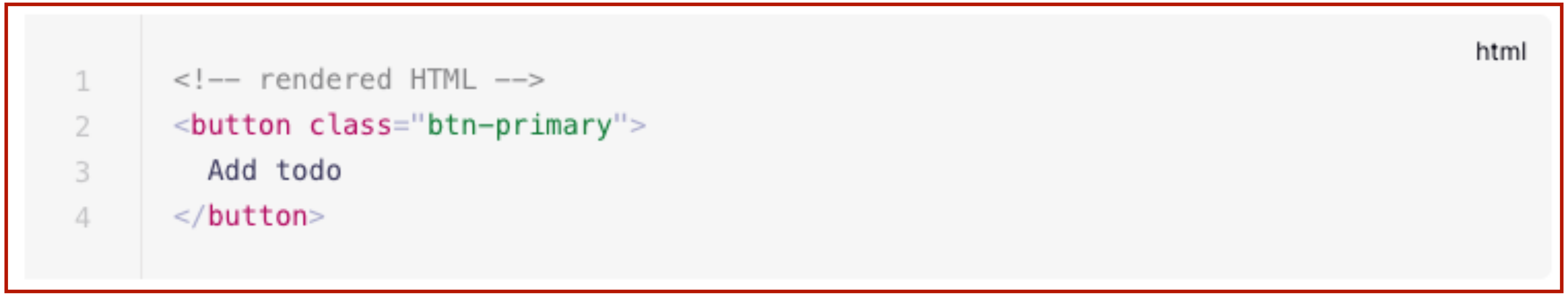
Vue implements a content distribution API, using the `<slot>` element to serve as distribution outlets for content. It turns your component to be flexible for re-use purpose.



```html
<!-- todo-button component template -->
<button class="btn-primary">
  <slot></slot>
</button>
```



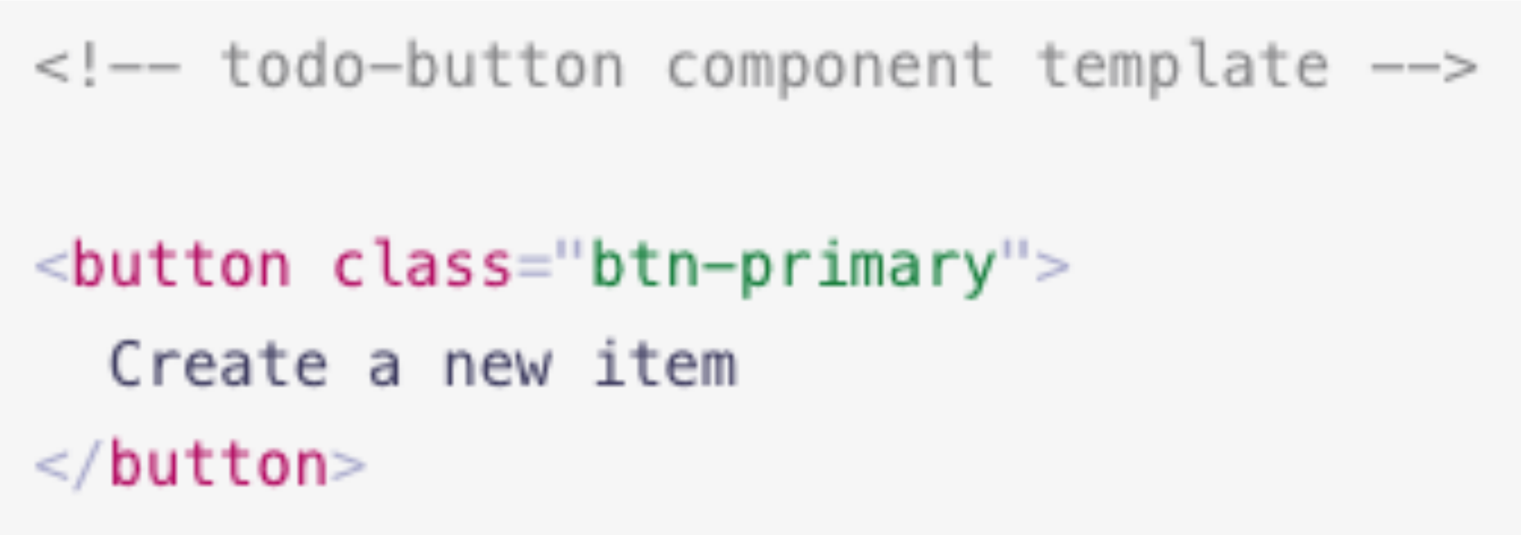
```html
<todo-button>
  Add todo
</todo-button>
```



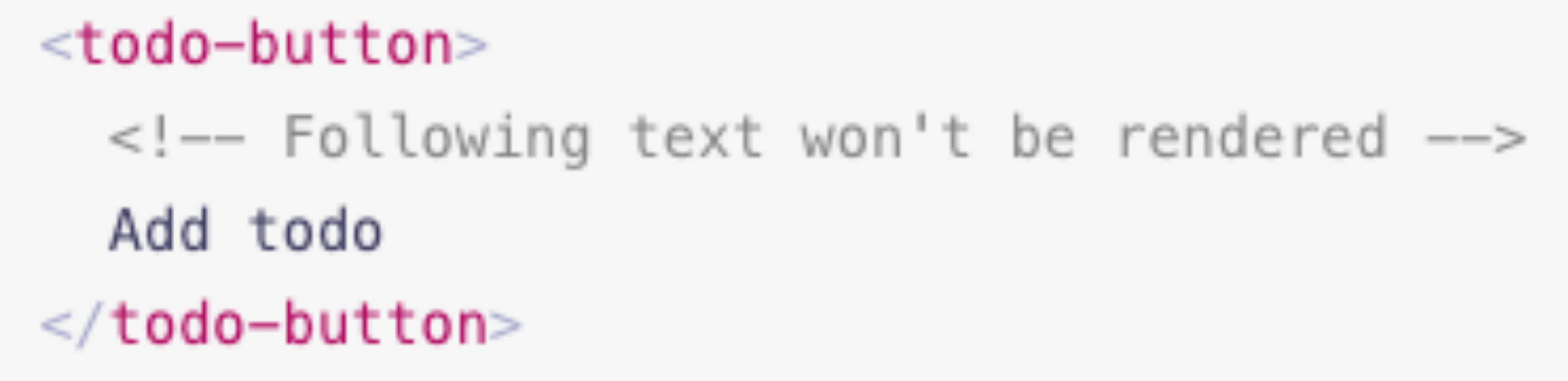
```html
<!-- rendered HTML -->
<button class="btn-primary">
  Add todo
</button>
```

Some cases with <slot></slot>

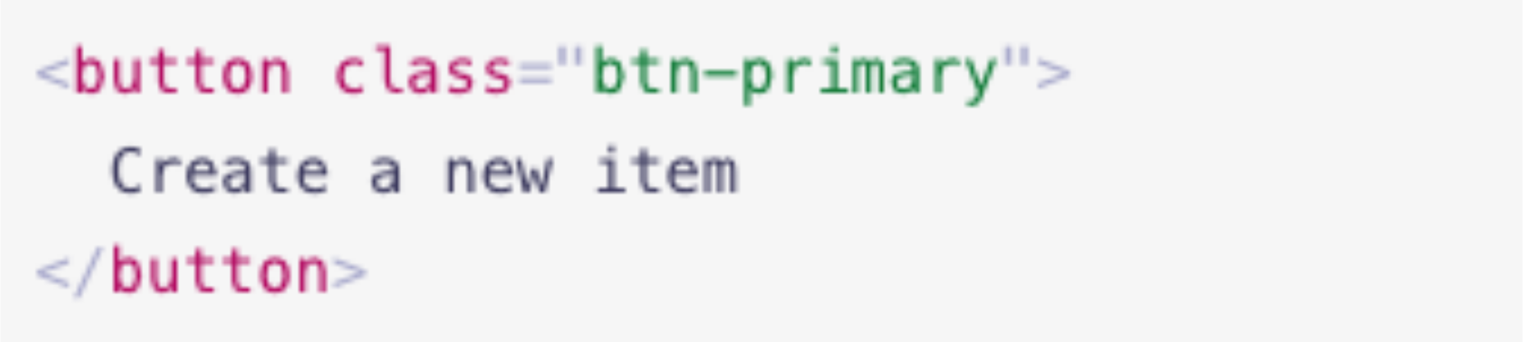
What if you don’t declare <slot> inside component but you passed content to it:



```
<!-- todo-button component template -->

<button class="btn-primary">
  Create a new item
</button>
```

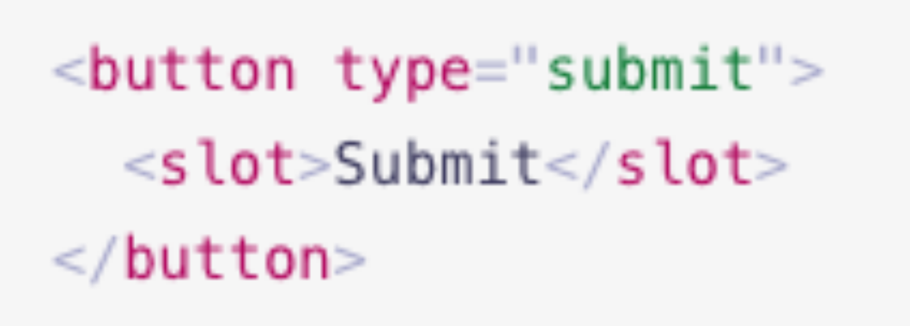


```
<todo-button>
  <!-- Following text won't be rendered -->
  Add todo
</todo-button>
```

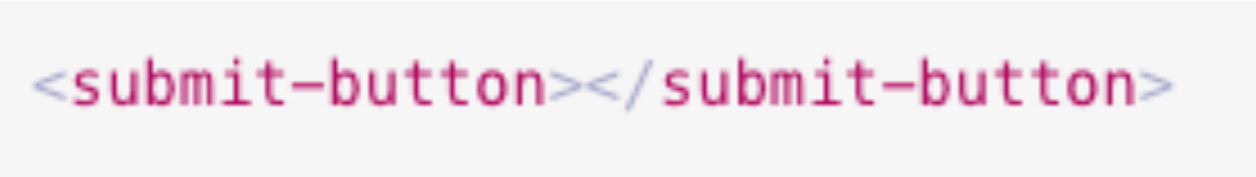


```
<button class="btn-primary">
  Create a new item
</button>
```

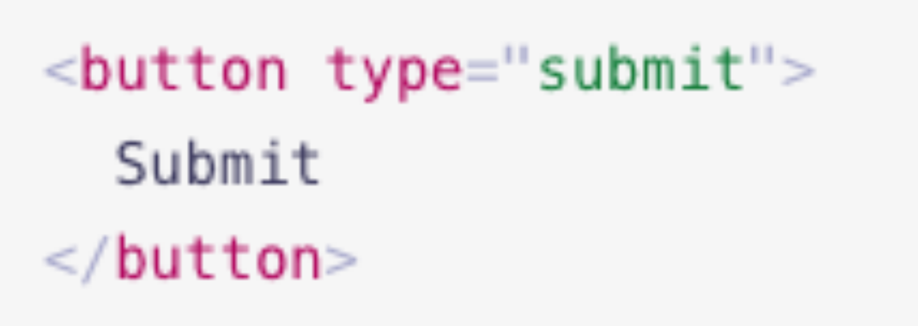
How can I have a default content, if no content is given to the component



```
<button type="submit">
  <slot>Submit</slot>
</button>
```



```
<submit-button></submit-button>
```



```
<button type="submit">
  Submit
</button>
```