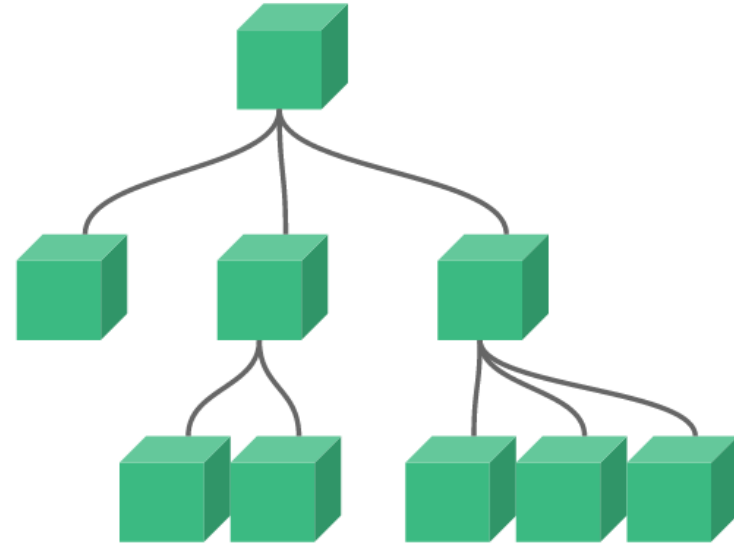
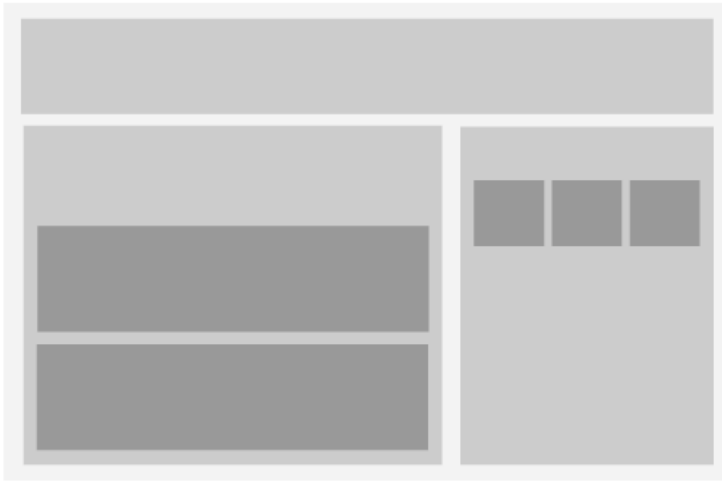


Vue Component

Bok, Jong Soon
javaexpert@nate.com
<https://github.com/swacademy/Vue.js>

Vue Component

- 화면의 영역을 구분하여 개발할 수 있는 Vue의 기능
- Component 기반으로 화면을 개발하게 되면 Code의 재사용성이 올라가고 빠르게 제작할 수 있다.



Vue Component (Cont.)

■ Component 생성 코드 형식

- 하나의 Component의 구성은 **template, script, style**로 구성될 수 있다.
- 즉, 각각 HTML, JavaScript, CSS로 대응되는 개념이다.
- 최소한 한 개의 **<template>** 은 있어야 한다.
- Component를 생성하는 Code 형식은 다음과 같다.

```
Vue.component('Component 이름', {  
  template : 'HTML Code들'  
});
```

Vue Component (Cont.)

■ Component 생성 후 표시하기

- 간단한 App Header Component를 생성해 보자.

```
Vue.component('app-header', {  
  template : '<h1>Header Component</h1>'  
});
```

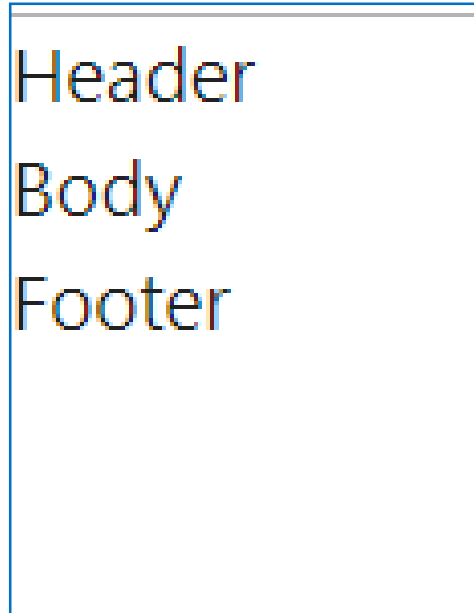
- 이제 등록한 Component를 화면에서 표시하려면 아래와 같이 Component Tag(Component 이름)을 추가한다.

```
<div id="app">  
  <app-header></app-header>  
</div>
```

- 그러면 아래와 같이 표시된다.

```
<div id="app">  
  <h1>Header Component</h1>  
</div>
```

Vue Component (Cont.)



```
<div id="app">
  <my-header></my-header>
  <my-body></my-body>
  <my-footer></my-footer>
</div>
<script>
  Vue.component('my-header', {
    template : '<div>Header</div>'
  });
  Vue.component('my-body', {
    template : '<div>Body</div>'
  });
  Vue.component('my-footer', {
    template : '<div>Footer</div>'
  });
  let vm = new Vue({
    el : '#app'
  })
</script>
```

Lab : @vue/cli를 이용하여 Vue Component 구성하기



Vue Component (Cont.)

- Component 등록 방법 2가지
 - 전역 Component 등록하는 방법
 - 앞 Slide 방법
 - 지역 Component 등록하는 방법

```
var appHeader = {  
  template: '<h1>Header Component</h1>'  
}
```

```
new Vue({  
  components: {  
    'app-header': appHeader  
  }  
})
```





Lab : Global Component 만들기

Lab - 전역 Component 만들기

```
8     <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9 </head>
10 <body>
11     <ul id="example">
12         <list-item></list-item>
13     </ul>
14
15     <script>
16         Vue.component('list-item', {
17             template : '<li>foo</li>'
18         })
19
20         new Vue({
21             el : '#example'
22         })
23     </script>
24 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

- foo

Lab - 전역 Component 만들기 (Cont.)

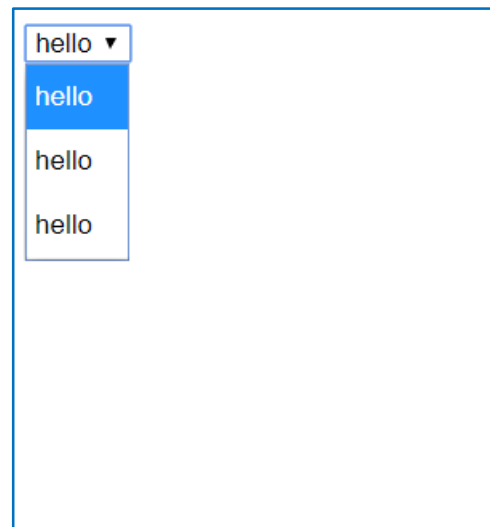
```
8     <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9 </head>
10 <body>
11     <ul id="example">
12         <list-item></list-item>
13     </ul>
14
15     <script>
16         Vue.component('list-item', {
17             template : '<li>foo {{ contents }} </li>',
18             data : function(){
19                 return {contents : 'bar'}
20             }
21         })
22
23         new Vue({
24             el : '#example'
25         })
26     </script>
27 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

- foo bar

Lab - 전역 Component 만들기 (Cont.)

```
8      <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9  </head>
10 <body>
11   <div id="app">
12     <select-component></select-component>
13   </div>
14
15   <script id="selectTemplate">
16     <select>
17       <option-component></option-component>
18       <option-component></option-component>
19       <option-component></option-component>
20     </select>
21   </script>
22   <script>
23     Vue.component('select-component', {
24       template: '#selectTemplate'
25     });
26     Vue.component('option-component', {
27       template : '<option>hello</option>'
28     })
29     new Vue({
30       el : '#app'
31     })
32   </script>
33 </body>
```



Lab - 전역 Component 만들기 (Cont.)

```
8     <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9 </head>
10 <body>
11     <div id="fruits-list">
12         <fruits-list-title></fruits-list-title>
13         <fruits-list-description></fruits-list-description>
14         <fruits-list-table></fruits-list-table>
15     </div>
16
17     <script>
18         Vue.component('fruits-list-title', {
19             template : '<h1>과일 목록</h1>'
20         })
21         Vue.component('fruits-list-description', {
22             template: '<p>각 계절 대표적 과일의 목록</p>'
23         })
```

Lab - 전역 Component 만들기 (Cont.)

```
24 Vue.component('fruits-list-table', {
25   template: `
26     <table>
27       <tr>
28         <th>계절</th>
29         <th>과일</th>
30       </tr>
31       <tr>
32         <td>봄</td>
33         <td>딸기</td>
34       </tr>
35       <tr>
36         <td>여름</td>
37         <td>수박</td>
38       </tr>
39       <tr>
40         <td>가을</td>
41         <td>포도</td>
42       </tr>
43       <tr>
44         <td>겨울</td>
45         <td>귤</td>
46       </tr>
47     </table>`
48 })
```

```
50 new Vue({
51   el : '#fruits-list'
52 })
53 </script>
54 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

과일 목록

각 계절 대표적 과일의 목록

계절 과일

봄 딸기
여름 수박
가을 포도
겨울 귤

Lab - 전역 Component 만들기

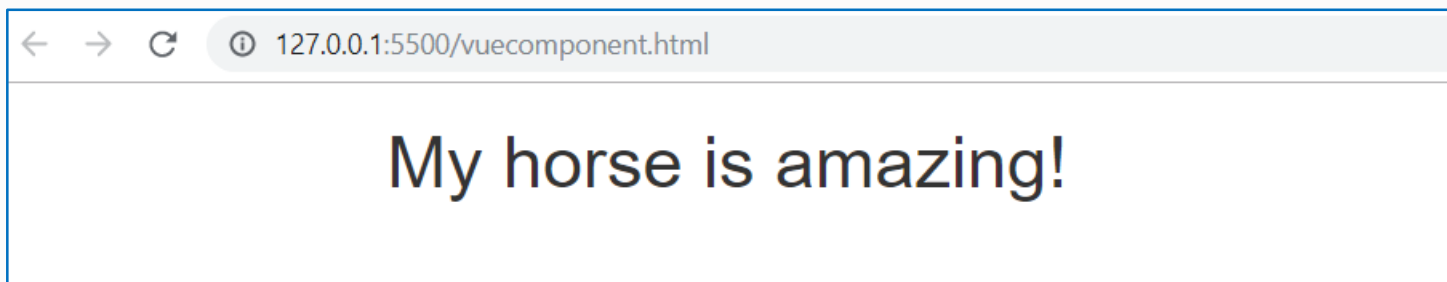
```
8   <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
10 </head>
11 <body>
12   <div class="container">
13     <story></story>
14   </div>
15
16   <script>
17     Vue.component('story', {
18       template: '<h1>My horse is amazing!</h1>'
19     });
20
21     new Vue({
22       el : '.container'
23     })
24   </script>
25 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

My horse is amazing!

Lab - 전역 Component 만들기

```
8      <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet">
9      <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
10     </head>
11     <body>
12       <div class="container">
13         <story></story>
14       </div>
15       <template id="story-template">
16         <h1>My horse is amazing!</h1>
17       </template>
18       <script>
19         Vue.component('story', {
20           template: '#story-template'
21         });
22
23         new Vue({
24           el: '.container'
25         })
26       </script>
27     </body>
```





Lab : Local Component 만들기



Lab – 지역 Component 만들기

```
8   <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
10  </head>
11  <body>
12    <div class="container">
13      <story v-bind:story="{plot: 'My horse is amazing.', writer: 'Mr. Weebl'}">
14      </story>
15      <story v-bind:story="{plot: 'Narwhals invented Shish Kebab.', writer: 'Mr. Weebl'}">
16      </story>
17      <story v-bind:story="{plot: 'The dark side of the Force is stronger.', writer: 'Darth Vader'}">
18      </story>
19    </div>
20    <template id="story-template">
21      <h1>{{ story.writer }} said "{{ story.plot }}"</h1>
22    </template>
23    <script>
24      Vue.component('story', {
25        props: ['story'],
26        template: "#story-template"
27      });
28
29      new Vue({
30        el : '.container'
31      })
32    </script>
33  </body>
```

127.0.0.1:5500/vuecomponent.html

Mr. Weebl said "My horse is amazing."

Mr. Weebl said "Narwhals invented Shish Kebab."

Darth Vader said "The dark side of the Force is stronger."

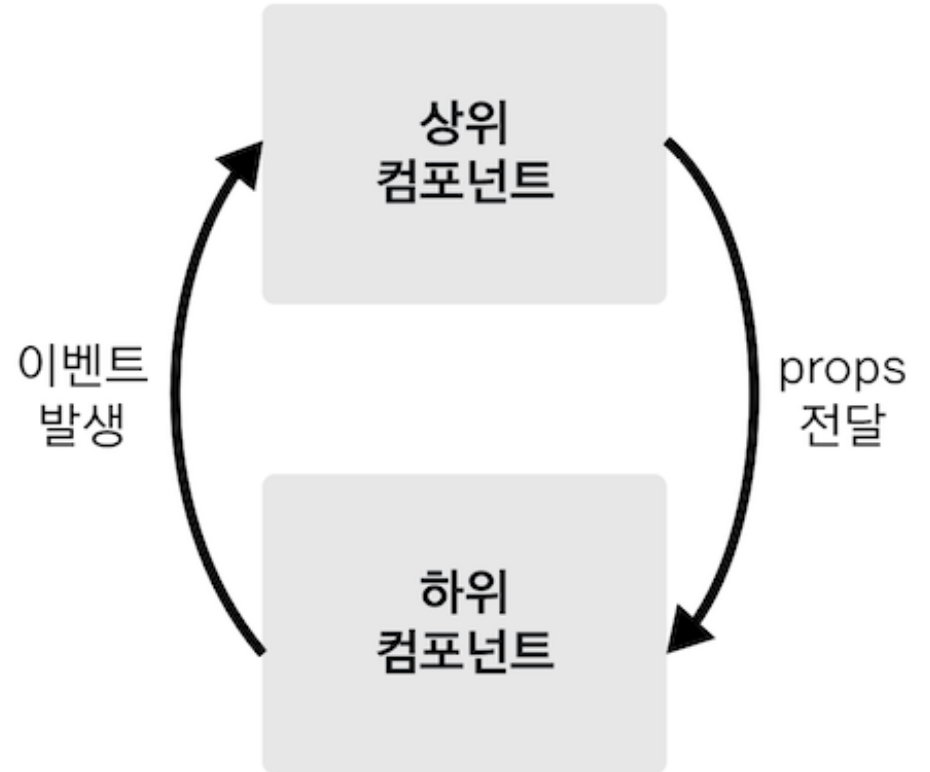


Lab : @vue/cli를 이용한 Global, Local Component 생성하기



Component Communication

- Vue Component는 각각 고유한 Data 유효 범위를 갖는다.
- 따라서, Component간에 Data를 주고 받기 위해서는 다음과 같은 규칙을 따라야 한다.
 - 상위에서 하위로 Data를 내려줌 → **props** 속성
 - 하위에서 상위로 Event를 올려줌 → **Event** 발생
→ **\$emit, \$on**
 - 형제간 → **eventbus**



props 속성

- Component 사이에 Data를 전달할 수 있는 Component 통신 방법
- 상위 Component에서 하위 Component로 내려보내는 Data 속성
- Code 형식
 - props 속성을 사용하기 위해서는 하위 Component의 Component 내용과 상위 Component의 Template에 각각 Code를 추가해줘야 한다.

//하위 Component 내용

```
var childComponent = {  
  props : ['props 속성 명']  
}
```

<!--상위 Component의 Template -->

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

```
  <child-component v-bind:props 속성 명="상위 Component의 data 속성">
```

```
  </child-component>
```

```
</div>
```

props 속성 (Cont.)

```
js
// 하위 컴포넌트 : childComponent
var childComponent = {
  props: ['propsdata'],
  template: '<p>{{ propsdata }}</p>'
}

// 상위 컴포넌트 : root 컴포넌트
new Vue({
  el: '#app',
  components: {
    'child-component': childComponent
  },
  data: {
    message: 'hello vue.js'
  }
})
```

```
html
<div id="app">
  <child-component v-bind:propsdata="message"></child-component>
  <!-- 위의 출력 결과는 hello vue.js -->
</div>
```

Event Emit

- Event 발생은 Component의 통신 방법 중 하위 Component에서 상위 Component로 통신하는 방법이다.
- Code 형식
 - 하위 Component의 Method나 Lifecycle Hook과 같은 곳에 다음과 같은 Code를 추가한다.

//하위 Component의 내용

this.\$emit('Event 이름')

- 그리고 나서 해당 Event를 수신하기 위해 상위 Component의 Template에 다음과 같이 구현한다.

<!-- 상위 Component의 Template -->

<div id="app">

<child-component v-on:Event 이름="상위 Component의 실행할 Method 이름 또는 연산">

</child-component>

</div>

Event Emit (Cont.)

```
js
// 하위 컴포넌트 : childComponent
var childComponent = {
  methods: {
    sendEvent: function() {
      this.$emit('update');
    }
  }
}

// 상위 컴포넌트 : root 컴포넌트
new Vue({
  el: '#app',
  components: {
    'child-component': childComponent
  },
  methods: {
    showAlert: function() {
      alert('event received');
    }
  }
})
```

```
html
<div id="app">
  <child-component v-on:update="showAlert"></child-component>
</div>
```

Lab : Component간 통신1

```
8   <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9 </head>
10 <body>
11   <div id="app">
12     <item-desc v-bind:item-name="myItem"></item-desc>
13   </div>
14
15   <script>
16     Vue.component('item-desc', {
17       props : {
18         itemName:{
19           type:String
20         }
21       },
22       template : '<p>{{ itemName }}은 편리해.</p>'
23     })
24     new Vue({
25       el : '#app',
26       data : {
27         myItem : 'pen'
28       }
29     })
30   </script>
31 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

pen은 편리해.

Lab : Component간 통신2

```
8      <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9  </head>
10 <body>
11     <div id="fruits-component">
12         <ol>
13             <!-- v-for를 사용해 각 fruit를 props(fruit-item)에 전달 -->
14             <fruits-item-name v-for="fruit in fruitsItems"
15                             :key="fruit.name" :fruits-item="fruit"></fruits-item-name>
16         </ol>
17     </div>
18
```

Lab : Component간 통신2 (Cont.)

```
19 <script>
20   Vue.component('fruits-item-name', {
21     props: {
22       fruitsItem: { // 템플릿 안에서는 케밥 케이스 사용
23         type: Object, // 객체 여부
24         required: true // 이 컴포넌트에서 반드시 필요하므로 true
25       }
26     },
27     template: '<li>{{fruitsItem.name}}</li>'
28   })
29
30   new Vue({
31     el: '#fruits-component',
32     data: { // 부모 컴포넌트에서는 배열이지만 v-for를 사용해 객체로 만들어 전달
33       fruitsItems: [
34         { name: '배' },
35         { name: '딸기' }
36       ]
37     }
38   })
39 </script>
40 </body>
```

← → ↻ ⓘ 127.0.0.1:5500/vuecomponent.html

1. 배
2. 딸기

Lab : Component간 통신3

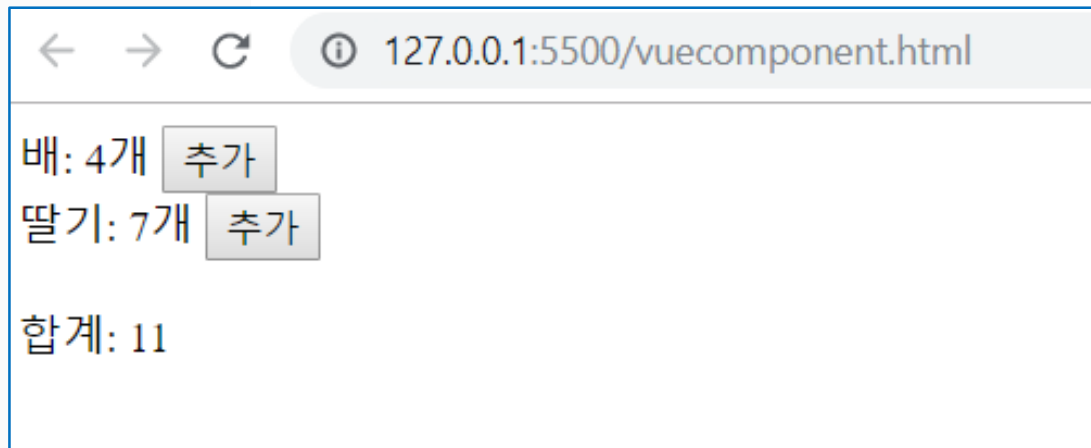
```
8      <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9  </head>
10 <body>
11     <div id="fruits-counter">
12         <div v-for="fruit in fruits">
13             {{fruit.name}}: <counter-button v-on:increment="incrementCartStatus()"></counter-button>
14         </div>
15         <p>합계: {{total}}</p>
16     </div>
```

Lab : Component간 통신3 (Cont.)

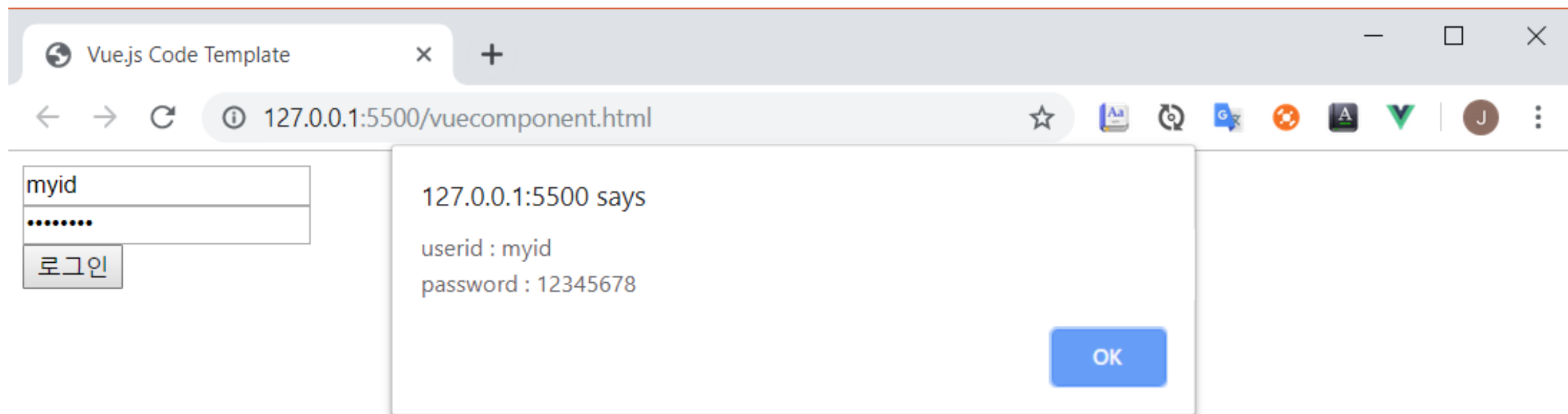
```
18     <script>
19         var counterButton = Vue.extend({
20             template: '<span>{{counter}}개 <button v-on:click="addToCart">추가</button></span>',
21             data: function () {
22                 return {
23                     counter: 0
24                 }
25             },
26             methods: {
27                 addToCart: function () {
28                     this.counter += 1
29                     this.$emit('increment') // increment 커스텀 이벤트 발생
30                 }
31             },
32         })
```

Lab : Component간 통신3 (Cont.)

```
34     new Vue({
35       el: '#fruits-counter',
36       components: {
37         'counter-button': counterButton
38       },
39       data: {
40         total: 0,
41         fruits: [
42           { name: '배' },
43           { name: '딸기' }
44         ]
45       },
46       methods: {
47         incrementCartStatus: function () {
48           this.total += 1
49         }
50       }
51     })
52 </script>
```



Lab



Lab (Cont.)

```
8     <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9 </head>
10 <body>
11     <div id="login-example">
12         <user-login></user-login>
13     </div>
14
15     <!-- template -->
16     <script type="text/x-template" id="login-template">
17         <div id="login-template">
18             <div>
19                 <input type="text" placeholder="로그인 ID" v-model="userid">
20             </div>
21             <div>
22                 <input type="password" placeholder="패스워드" v-model="password">
23             </div>
24             <button @click="login">로그인</button>
25         </div>
26     </script>
```

Lab (Cont.)

```
28 <script>
29   Vue.component('user-login', {
30     template : '#login-template',
31     data : function(){
32       return {
33         userid:'',
34         password : ''
35       }
36     },
37     methods : {
38       login:function(){
39         auth.login(this.userid, this.password)
40       }
41     }
42   })
43
44   let auth = {
45     login:function(id, pass){
46       alert('userid : ' + id + '\n' + 'password : ' + pass);
47     }
48   }
49
50   new Vue({
51     el : '#login-example'
52   });
53 </script>
```






Lab : Component간 Data 전달하기