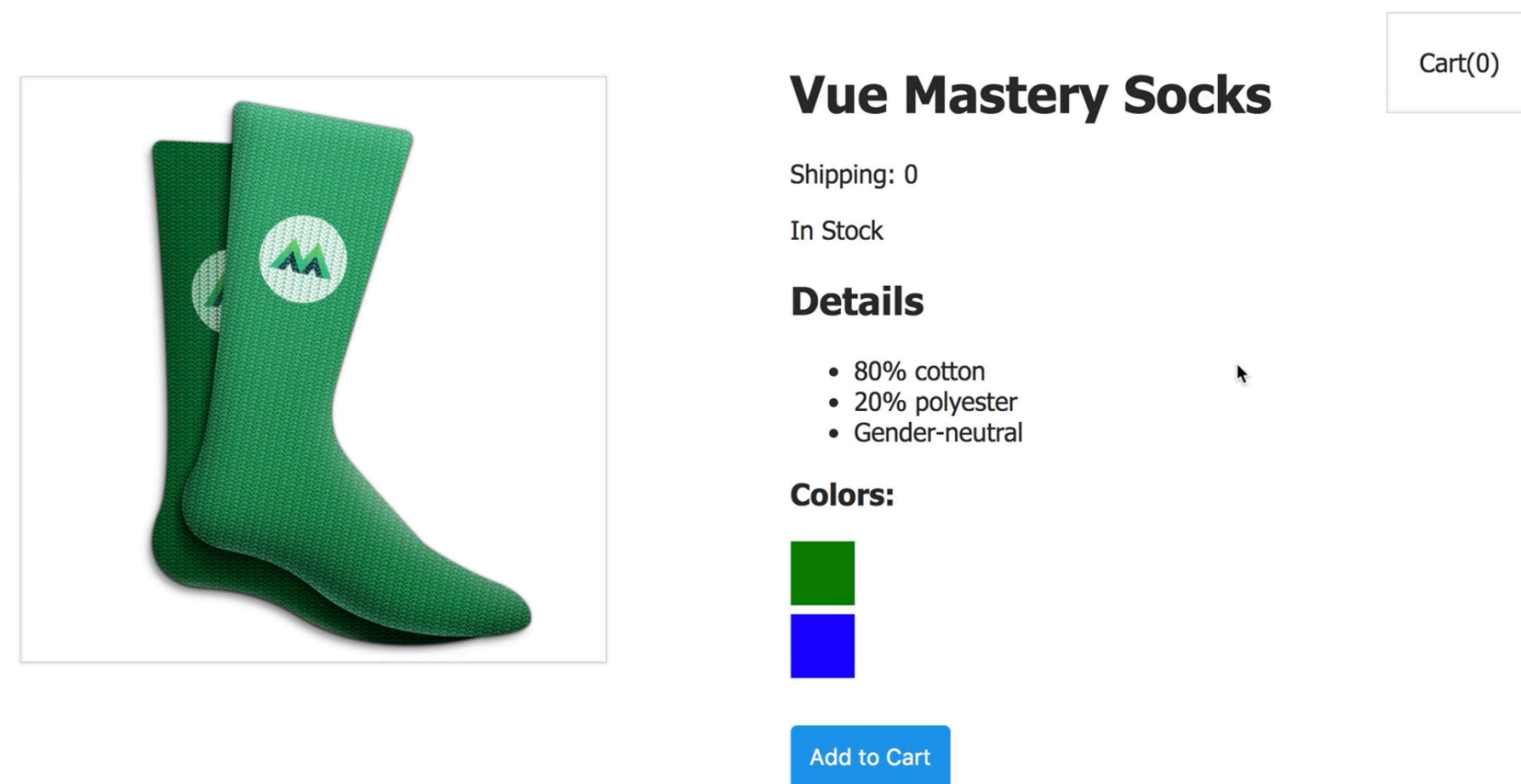


필수 요소 실습

<https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance/>

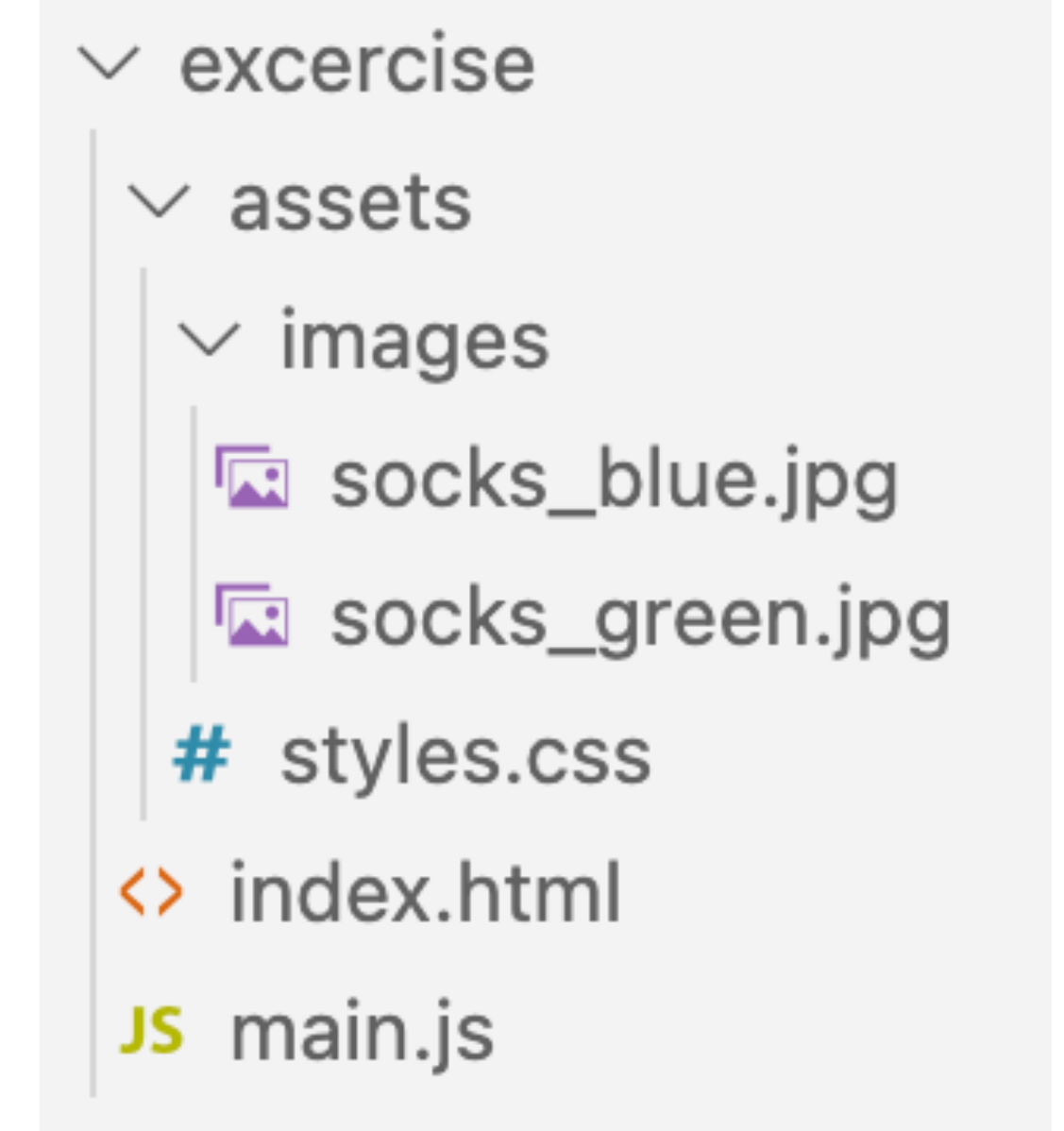
실습 내용

- Vue.js 공식 사이트에서 안내하는 학습 사이트(vuemastery)의 입문 예제를 한글로 번역함
- <https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance/>



준비 사항

- 예제 실습용 폴더에 다음 파일들을 준비
 - index.html
 - main.js
 - assets/styles.css
 - assets/images/사이즈가 같은 두 장의 jpg



- <https://drive.google.com/drive/folders/1Br-qQuoNRPa4zLxCIHjPi4nLYIBMwTKo?usp=sharing>

Vue instance 만들기

- main.js에 Vue instance를 만든 다음 데이터로 product: 'Socks'를 선언
- index.html에서 product 변수를 출력하도록 한다.

Vue instance 만들기

main.js

```
1  ✓ let app = new Vue({  
2    el: '#app',  
3  ✓  data: {  
4    |    product: 'Socks'  
5    |  }  
6  });
```

index.html

```
7  <body>  
8    <div id="app">  
9      <h1>{{ product }}</h1>  
10   </div>  
11   <script src="https://cdn.jsdelivr.net/npm/vue"></script>  
12   <script src="main.js"></script>  
13 </body>
```

Stylesheet 적용

- stylesheet를 index 파일에 포함시키고
- div (id=app) 상단에 div (class=nav-bar) 를 추가
- div (id=app) 의 자식으로 div (class=product)로 지정한다.
- div (class=product) 의 자식으로 기존 h1 태그를 이동시킨다.

Stylesheet 적용

index.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Excercise</title>
6      <link rel="stylesheet" type="text/css" href="styles.css">
7    </head>
8    <body>
9      <div class="nav-bar"></div>
10     <div id="app">
11       <div class="product">
12         <h1>{{ product }}</h1>
13       </div>
14     </div>
15     <script src="https://cdn.jsdelivr.net/npm/vue"></script>
16     <script src="main.js"></script>
17   </body>
18 </html>
```

이미지와 정보 영역 분리

- `div(class=product)` 를 다음과 같은 자식 요소로 나눈다.

```
10      <div id="app">
11          <div class="product">
12              <div class="product-image">
13                  <img src="">
14              </div>
15
16              <div class="product-info">
17                  <h1>{{ product }}</h1>
18              </div>
19          </div>
20      </div>
```

- 데이터에 `image` 변수를 추가하고 둘 중 하나의 이미지의 경로를 세팅한 다음 `img` 태그를 완성

이미지와 정보 영역 분리

main.js

```
1 let app = new Vue({
2   el: '#app',
3   data: {
4     product: 'Socks',
5     image: './assets/socks_green.jpg'
6   }
7 });
```

index.html

```
10 <div id="app">
11   <div class="product">
12     <div class="product-image">
13       
14     </div>
15     <div class="product-info">
16       <h1>{{ product }}</h1>
17     </div>
18   </div>
19 </div>
```

조건문

- 데이터에 inStock 이라는 boolean 변수를 선언
- div<class=product-info) 의 제목 하단에 p 태그를 이용해
 - true 면 <p>In Stock</p> false 면 <p>Out of Stock</p> 출력

조건문

main.js

```
1  let app = new Vue({
2    el: '#app',
3    data: {
4      product: 'Socks',
5      image: './assets/socks_green.jpg',
6      inStock: true
7    }
8  });
```

index.html

```
10  <div id="app">
11    <div class="product">
12      <div class="product-image">
13        
14      </div>
15      <div class="product-info">
16        <h1>{{ product }}</h1>
17        <p v-if="inStock">In Stock</p>
18        <p v-else>Out of Stock</p>
19      </div>
20    </div>
21  </div>
```

조건문

- 데이터에 inStock 대신 inventory 라는 정수형 데이터 선언
- inventory가 10 초과면 In Stock
- inventory가 0 초과 10이하면 Almost sold out!
- inventory가 0 이면 Out of Stock 출력하도록 수정

조건문

main.js

```
1 let app = new Vue({
2   el: '#app',
3   data: {
4     product: 'Socks',
5     image: './assets/socks_green.jpg',
6     inventory: 100
7   }
8 });
```

index.html

```
<div class="product-info">
  <h1>{{ product }}</h1>
  <p v-if="inventory > 10">In Stock</p>
  <p v-else-if="inventory > 0">Almost sold out!</p>
  <p v-else>Out of Stock</p>
</div>
```

- 실습 후 inStock:true로 동작하도록 다시 작성

반복문

- 데이터에 details 라는 문자열 배열을 추가
- “80% cotton”, “20% polyester”, “Gender-neutral” 을 배열에 추가
- 재고 항목 아래에 ul로 추가

반복문

main.js

```
1  let app = new Vue({
2    el: '#app',
3    data: {
4      product: 'Socks',
5      image: './assets/socks_green.jpg',
6      inStock: true,
7      details: ['80% cotton', '20% polyester', 'Gender-neutral']
8    }
9  });
```

index.html

```
10 <div id="app">
11   <div class="product">
12     <div class="product-image">
13       
14     </div>
15     <div class="product-info">
16       <h1>{{ product }}</h1>
17       <p v-if="inStock">In Stock</p>
18       <p v-else>Out of Stock</p>
19       <ul>
20         <li v-for="detail in details">{{detail}}</li>
21       </ul>
22     </div>
23   </div>
24 </div>
```

반복문

- 데이터에 variants 라는 배열을 추가
- {variantId:2234, variantColor: “green”}
- {variantId:2235, variantColor: “blue”}
- 두 객체를 배열에 추가
- 상세 정보 (ul) 하단에 div를 이용한 반복문으로 variantColor를 p 태그로 출력
- 이 때 variantId를 key로 사용하도록 함.

반복문

main.js

```
1  let app = new Vue({
2    el: '#app',
3    data: {
4      product: 'Socks',
5      image: './assets/socks_green.jpg',
6      inStock: true,
7      details: ['80% cotton', '20% polyester', 'Gender-neutral'],
8      variants: [
9        {
10         variantId: 2334,
11         variantColor: 'green'
12       },
13       {
14         variantId: 2335,
15         variantColor: 'blue'
16       }
17     ]
18   }
19 });
```

index.html

```
<div class="product-info">
  <h1>{{ product }}</h1>
  <p v-if="inStock">In Stock</p>
  <p v-else>Out of Stock</p>
  <ul>
    <li v-for="detail in details">{{ detail }}</li>
  </ul>
  <div v-for="variant in variants" :key="variant.variantKey">
    <p>{{ variant.variantColor }}</p>
  </div>
</div>
```

이벤트

- 색상 아래쪽에 Add to Cart 버튼을 추가
- 버튼 아래에 다음과 같은 div 추가
- 데이터에 cart 라는 정수형 변수 추가 후 클릭하면 1씩 숫자가 증가하도록 구현: 단, addToCart 라는 함수를 추가하여 구현

```
<button>Add to Cart</button>  
<div class="cart">  
  <p>Cart{{cart}}</p>  
</div>
```

이벤트

main.js

```
1  let app = new Vue({
2    el: '#app',
3    data: {
4      product: 'Socks',
5      image: './assets/socks_green.jpg',
6      inStock: true,
7      details: ['80% cotton', '20% polyester', 'Gender-neutral'],
8      variants: [
9        {
10         variantId: 2334,
11         variantColor: 'green'
12       },
13       {
14         variantId: 2335,
15         variantColor: 'blue'
16       }
17     ],
18     cart: 0
19   },
20   methods: {
21     addToCart() {
22       this.cart += 1
23     }
24   }
25 });
```

index.html

```
15  <div class="product-info">
16    <h1>{{ product }}</h1>
17    <p v-if="inStock">In Stock</p>
18    <p v-else>Out of Stock</p>
19    <ul>
20      <li v-for="detail in details">{{ detail }}</li>
21    </ul>
22    <div v-for="variant in variants" :key="variant.variantKey">
23      <p>{{ variant.variantColor }}</p>
24    </div>
25
26    <button v-on:click="addToCart">Add to Cart</button>
27
28    <div class="cart">
29      <p>Cart{{ cart }}</p>
30    </div>
31  </div>
```

이벤트

- variants 배열에 있는 각 객체에 자신의 이미지 경로를 가리키는 variantImage 변수를 추가
- 화면의 blue/green에 mouseover 이벤트 발생 시 그림을 바꾸도록 구현
- 이벤트 처리는 updateProduct 함수를 이용

이벤트

main.js

```
1  let app = new Vue({
2    el: '#app',
3    data: {
4      product: 'Socks',
5      image: './assets/socks_green.jpg',
6      inStock: true,
7      details: ['80% cotton', '20% polyester', 'Gender-neutral'],
8      variants: [
9        {
10         variantId: 2334,
11         variantColor: 'green',
12         variantImage: './assets/socks_green.jpg'
13       },
14       {
15         variantId: 2335,
16         variantColor: 'blue',
17         variantImage: './assets/socks_blue.jpg'
18       }
19     ],
20     cart: 0
21   },
22   methods: {
23     addToCart() {
24       this.cart += 1
25     },
26     updateProduct(variantImage) {
27       this.image = variantImage
28     }
29   }
30 });
```

index.html

```
<div v-for="variant in variants" :key="variant.variantKey">
  <p @mouseover="updateProduct(variant.variantImage)">{{variant.variantColor}}</p>
</div>
```

Class & Styling

- green, blue 를 그림으로 표시: 반복문이 설정된 div를 다음과 같이 수정

```
<div v-for="variant in variants"  
  :key="variant.variantKey"  
  class="color-box"  
  :style="{backgroundColor: variant.variantColor}"  
  @mouseover="updateProduct(variant.variantImage)">  
</div>
```

- backgroundColor 로 표기하거나 'background-color' 로 표기가 가능

Class & Styling

- 재고가 없는 경우(inStock:false) 버튼을 disabled로 처리하고자 함
- css 파일에 disabledButton이 정의되어 있음. 다음과 같이 스타일을 적용

```
<button v-on:click="addToCart"  
  :disabled="!inStock"  
  :class="{disabledButton:!inStock}">Add to Cart</button>
```

Computed property

- data에 문자열 타입의 brand 변수를 추가한 뒤 제목 부분에 “브랜드 제품” 형태로 출력되도록 title 이라는 computed property를 추가하라.
- html 의 {{ product }} 부분에 title이 출력되도록 수정하라.

Computed property

main.js

```
1 let app = new Vue({
2   el: '#app',
3   data: {
4     brand: 'Vue',
5     product: 'Socks',
6     image: './assets/socks_green.jpg',
7     inStock: false,
8     details: ['80% cotton', '20% polyester', 'Gender-neutral'],
9     variants: [
10      {
11        variantId: 2334,
12        variantColor: 'green',
13        variantImage: './assets/socks_green.jpg'
14      },
15      {
16        variantId: 2335,
17        variantColor: 'blue',
18        variantImage: './assets/socks_blue.jpg'
19      }
20    ],
21    cart: 0
22  },
23  methods: {
24    addToCart() {
25      this.cart += 1
26    },
27    updateProduct(variantImage) {
28      this.image = variantImage
29    }
30  },
31  computed: {
32    title() {
33      return this.brand + ' ' + this.product
34    }
35  }
36 });
```

index.html

```
10 <div id="app">
11   <div class="product">
12     <div class="product-image">
13       
14     </div>
15     <div class="product-info">
16       <h1>{{ title }}</h1>
```

Computed property

- 현재는 그림만 색상에 연동되는데 다른 데이터도 색상에 연동되도록 수정하자.
- data의 image 변수 대신 selectedVariant 정수형 변수를 추가하고 0으로 초기화
- 색상을 출력하는 div의 반복문에 variant in variants 대신 index도 가져오도록 수정

main.js

```
1  ✓ let app = new Vue({  
2    el: '#app',  
3  ✓  data: {  
4    brand: 'Vue',  
5    product: 'Socks',  
6    selectedVariant: 0,
```

index.html

```
<div v-for="(variant, index) in variants"  
  :key="variant.variantKey"  
  class="color-box"  
  :style="{backgroundColor: variant.variantColor}"  
  @mouseover="updateProduct(index)">  
</div>
```

Computed property

- updateProduct 함수 수정 및 computed property로 image 추가

main.js

```
23 methods:{
24   addToCart(){
25     this.cart += 1
26   },
27   updateProduct(index){
28     this.selectedVariant = index
29   }
30 },
31 computed:{
32   title(){
33     return this.brand + ' ' + this.product
34   },
35   image(){
36     return this.variants[this.selectedVariant].variantImage
37   }
38 }
```

Computed property

- data의 각 variant 객체에 variantQuantity 라는 정수형 변수를 추가

main.js

```
9      variants: [
10        {
11          variantId: 2334,
12          variantColor: 'green',
13          variantImage: './assets/socks_green.jpg',
14          variantQuantity: 10
15        },
16        {
17          variantId: 2335,
18          variantColor: 'blue',
19          variantImage: './assets/socks_blue.jpg',
20          variantQuantity: 0
21        }
22      ],
```

Computed property

- inStock을 Computed property로 변경하는 함수 구현

Computed property

- inStock을 Computed property로 변경하는 함수 구현

main.js

```
32     computed:{
33         title(){
34             return this.brand + ' ' + this.product
35         },
36         image(){
37             return this.variants[this.selectedVariant].variantImage
38         },
39         inStock(){
40             return this.variants[this.selectedVariant].variantQuantity
41         }
42     }
```

Component 만들기

- 지금까지 작성한 제품 관련 부분들을 <product> 라는 태그로 사용할 수 있도록
- 컴포넌트를 만들고
- 화면에 출력

Component 만들기

- main.js
- <https://drive.google.com/drive/folders/1Br-qQuoNRPa4zLxCIHjPi4nLYIBMwTKo?usp=sharing> 폴더의 중간작업파일 폴더
- index.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Excercise</title>
6      <link rel="stylesheet" type="text/css" href="styles.css">
7    </head>
8    <body>
9      <div class="nav-bar"></div>
10     <div id="app">
11       <product></product>
12     </div>
13     <script src="https://cdn.jsdelivr.net/npm/vue"></script>
14     <script src="main.js"></script>
15   </body>
16 </html>
```


Cart

- Cart 가 제품마다 있으면 이상하므로 Cart 태그를 product 태그에서 삭제하고 앱 div(app)로 이동
- data의 cart 변수를 Vue instance의 data로 이동
- component 함수의 this.cart +=1 을 삭제
- 제품의 Add to cart 가 동작하도록 Event를 처리한다. (\$emit)

Cart

- index.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Excercise</title>
6      <link rel="stylesheet" type="text/css" href="styles.css">
7    </head>
8    <body>
9      <div class="nav-bar"></div>
10     <div id="app">
11       <div class="cart">
12         <p>Cart({{ cart }})</p>
13       </div>
14       <product @add-to-cart="addToCart"></product>
15     </div>
16     <script src="https://cdn.jsdelivr.net/npm/vue"></script>
17     <script src="main.js"></script>
18   </body>
19 </html>
```

Cart

- 제품이 2종류가 있으므로 단순 카운트만 올리는 것은 부족하다.
- cart 를 배열로 바꾸고 Add to cart 버튼 클릭 시 제품의 Id를 배열에 추가하도록 한다.