

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

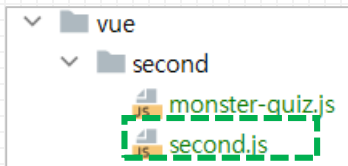
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

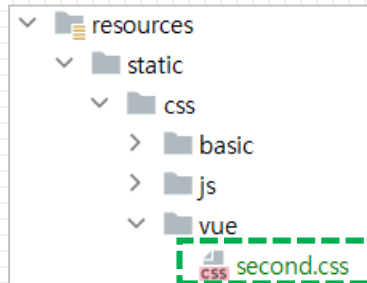
```

7  @Slf4j
8  @Controller
9  public class SecondVueController {
10
11      @GetMapping("/vueForTest")
12      public String getVueForTest () {
13          log.info("VueForTest(): 뷰로 for문을 실행하자.");
14
15          return "/vue/second";
16      }
    
```

.js 위치



.css위치



html

```

1  <!DOCTYPE html>
2  <html lang="en" xmlns:v-on="http://www.w3.org/1999/xhtml" xmlns:v-bind="http://www.w3.org/1999/xhtml">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/vue/second.css">
7  </head>
    
```

```

8  <body>
9      <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
10     <!-- 어떤 동작을 하겠다는 가다를 잡아놓고 -->
11     <div id="app">
12         <p>{{ message }}</p>
13         <ol>
14             <li v-for="item in list">{{ item }}</li>
15         </ol>
16         <button v-on:click="buttonClickTest">클릭해봐!</button>
    
```

```

62 </div>
63 <!-- 이렇게 하라고 오더함 -->
64 <script src="/vue/second/second.js"></script>
65 </body>
66 </html>
    
```

Vue : Array / v-for

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
11 <div id="app">
12   <p>{{ message }}</p>
13   <ol>
14     <li v-for="item in list">{{ item }}</li>
15   </ol>
16   <button v-on:click="buttonClickTest">클릭해봐!</button>
```

For each문처럼 list 배열 값을 item에 넣는다.

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
5     message: '뷰 너무 좋당',
6     initMsg: '양방향 연결이 뭔지 한 번 볼까?',
7     list: ['사과', '바나나', '딸기', '수박', '참외', '포도', '망고', '블루베리', '체리'],
```

⋮

```
20 methods: {
21   // event: 사용자가 어떤 이벤트를 발생시켰는지 알 수 있음
22   buttonClickTest: function (event) {
23     alert('뷰 짱')
24   },
```

버튼을 클릭했을 때 '뷰 짱' 창이 뜬다.

생성된 html

뷰 너무 좋당

1. 사과
2. 바나나
3. 딸기
4. 수박
5. 참외
6. 포도
7. 망고
8. 블루베리
9. 체리

클릭해봐!

Click

localhost:7777 내용:

뷰 짱

확인

Click

Vue : v-model 양방향

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
18 <!-- 양방향 반응형 웹 -->
19 <input v-model="initMsg">
20 <p>{{ initMsg }}</p>
```

생성된 html

양방향 연결이 뭔지 한 번 볼까?

양방향 연결이 뭔지 한 번 볼까?

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
5     message: '뷰 너무 좋당',
6     initMsg: '양방향 연결이 뭔지 한 번 볼까?',
7     list: ['사과', '바나나', '딸기', '수박', '참외', '포도', '망고', '블루베리', '체리'],
8     show: true,
9     num: 3,
10    count: 0,
11    radius: 50,
12    randomNumber: 0,
13    monsterName: '',
14    monsters: [
15      { id: 1, name: '슬라임', hp: 30 },
16      { id: 2, name: '고블린', hp: 50 },
17      { id: 3, name: '카오스 드래곤', hp: 99999999 }
18    ]
19  },
```

적으면 자동으로 = 양방향 바

적으면 자동으로 = 양방향 반응형 웹

Vue : dom의 vue버전

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
22 <!-- dom의 빠른 버전 -->
23 <button v-on:click="show = !show">압!</button>
24 <transition>
25
26 <!-- transition : opacity -->
27 <p v-if="show">invisibility도 쉽게 제어되네?</p>
28 </transition>
```

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
5     message: '뷰 너무 좋당',
6     initMsg: '양방향 연결이 뭔지 한 번 볼까?',
7     list: ['사과', '바나나', '딸기', '수박', '참외', '포도', '망고', '블루베리', '체리'],
8     show: true,
```

CSS

```
/* .v-enter-: 실제 HTML 요소가 삽입되기 이전에 적용되고 한 프레임 후에 제거 됨
   v-enter-active: enter에 대한 활성 및 종료 상태
   HTML 요소가 삽입 되기 이전에 적용됨
   transition 이 완료되면 제거됨 */
/* v-leave-active: v-leave에 대한 활성 및 종료 상태로
   transition이 동작하여 적용된 이후 다시 transition이 동작하면 제거됨 */
/* v-leave-to: transition이 동작되고 transition이 끝나면 제거됨 */

.v-enter-active, .v-leave-active {
  transition: opacity 2s;
}

/* 투명도를 2초동안 Fade In & Fade out */
.v-enter, .v-leave-to {
  opacity: 0;
}
```

투명도 조절을 통해 Fade In & Fade out을 만든다.

생성된 html



Vue : { }

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
29 <p>Hello {{ message }}</p>
30 <p>{{ message.length }}</p>
31 <p>{{ list[3] }}</p>
32 <p>{{ list[num] }}</p>
```

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
5     message: '뷰 너무 좋당',
6     initMsg: '양방향 연결이 뭔지 한 번 볼까?',
7     list: ['사과', '바나나', '딸기', '수박', '참외', '포도', '망고', '블루베리', '체리'],
8   }
9 })
```

list [0] [1] [2] [3] [4] [5] [6] [7] [8]

생성된 html

Hello 뷰 너무 좋당	message
7	message.length
수박	list[3]
수박	list[num]

9 num: 3,

Vue : increment

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
<!-- 클릭을 했을때 몇번 클릭했는지 알아보자 -->  
<button v-on:click="increment">숫자를 카운팅 해보자!</button>  
<p>{{ count }}번 클릭했습니다.</p>
```

js

```
1 var app = new Vue ({  
2   // id랑 맵핑되는 것이 el  
3   el: '#app',  
4   data: {  
5     message: '뷰 너무 좋당',  
6     initMsg: '양방향 연결이 뭔지 한 번 볼까?',  
7     list: ['사과', '바나나', '딸기', '수박', '참외', '포도', '망고', '블루베리', '체리'],  
8     show: true,  
9     num: 3,  
10    count: 0,  
11  },  
12  },  
13  {  
14    // 컴포넌트  
15    // 컴포넌트 이름 : vue-component-name  
16    // 컴포넌트 선택자 : vue-component-name  
17    // 컴포넌트 선택자 : vue-component-name  
18    // 컴포넌트 선택자 : vue-component-name  
19  },  
20  },  
21  {  
22    // 컴포넌트  
23    // 컴포넌트 이름 : vue-component-name  
24    // 컴포넌트 선택자 : vue-component-name  
25    // 컴포넌트 선택자 : vue-component-name  
26    // 컴포넌트 선택자 : vue-component-name  
27  },  
28  },  
29  },  
30  },  
31  },  
32  },  
33  },  
34  },  
35  },  
36  },  
37  },  
38  },  
39  },  
40  },  
41  },  
42  },  
43  },  
44  },  
45  },  
46  },  
47  },  
48  },  
49  },  
50  },  
51  },  
52  },  
53  },  
54  },  
55  },  
56  },  
57  },  
58  },  
59  },  
60  },  
61  },  
62  },  
63  },  
64  },  
65  },  
66  },  
67  },  
68  },  
69  },  
70  },  
71  },  
72  },  
73  },  
74  },  
75  },  
76  },  
77  },  
78  },  
79  },  
80  },  
81  },  
82  },  
83  },  
84  },  
85  },  
86  },  
87  },  
88  },  
89  },  
90  },  
91  },  
92  },  
93  },  
94  },  
95  },  
96  },  
97  },  
98  },  
99  },  
100 },
```

생성된 html

숫자를 카운팅 해보자!

0번 클릭했습니다.

Click

숫자를 카운팅 해보자!

41번 클릭했습니다.

Vue : v-bind 단방향

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

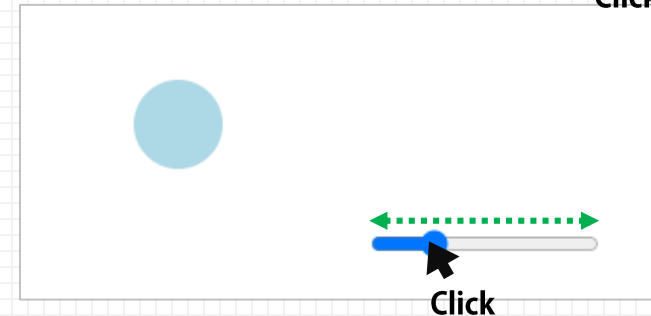
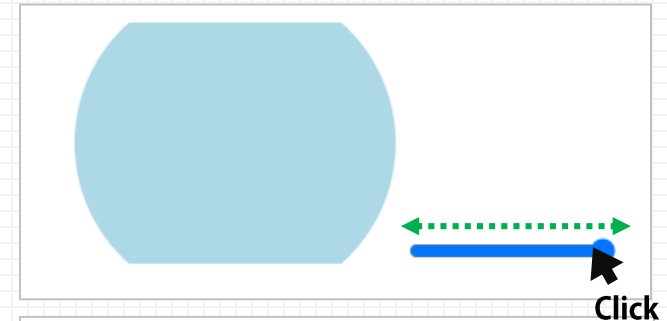
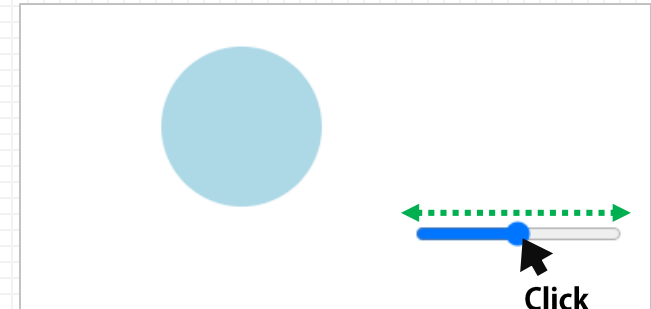
html

```
<!-- v-model: 양방향 , v-bind: 단방향 -->
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <!-- cx, cy 원의 중점 -->
  <circle cx="200" cy="75" v-bind:r="radius" fill="lightblue"/>
</svg>
<input type="range" min="0" max="100" v-model="radius"><br><br><br>
```

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
11    radius: 50,
```

생성된 html



Vue : Math.floor(Math.random())

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
<!-- 랜덤 주사위 게임 -->
<p>{{ randomNumber }}</p>
<button v-on:click="randomGeneration">랜덤 숫자 발생!</button>
```

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
```

```
12   randomNumber: 0, randomGeneration 클릭 전, 초기 숫자값
```

```
59   randomGeneration() {
60     // 1 ~ 10 까지의 난수를 생성합니다.
61     this.randomNumber = Math.floor(Math.random() * 10) + 1;
62   }
63 }
```

생성된 html

0 randomGeneration 클릭 전, 초기 숫자값

랜덤 숫자 발생!

Click

6

랜덤 숫자 발생!

Click

4

랜덤 숫자 발생!

Click

Vue : 게임만들기

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
<!-- 게임처럼 -->
<input v-model="monsterName">
<button v-on:click="addMonster">몬스터 추가하기</button>
<ul>
  <li v-for="(monster, index) in monsters"
    v-bind:key="monster.id"
    v-bind:class="{ strong: monster.hp > 300 }">
    ID.{{ monster.id }} {{ monster.name }} HP.{{ monster.hp }}
    <span v-if="monster.hp > 300">개세다</span>
    <button v-on:click="userAttack(index)">펑타</button>
    <button v-on:click="removeMonster(index)">몬스터 삭제</button>
  </li>
</ul>
</div>
```

Reduce ()

// reduce()를 사용하면 알아서 내부에 있는 요소를 전부 날개로 쪼개서 하나하나 비교하게 됨
// 그래서 function(a,b)에 입력으로 들어가는 a,b는
// 모두 monsters 배열에 있는 각각의 날개 요소들임
// 계속 날개 단위로 끝까지 비교한다면
// 마지막엔 제일 큰 숫자(결국 맨 마지막 녀석)를 얻게 될 것이다.

// reduce는 전체 순회를 끝날때까지 동작이 값이 증발되지 않으며,
// 최종 결과를 얻을때까진 이전 결과를 유지하며 연산이 진행됨

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
13     monsterName: '',
14     monsters: [
15       { id: 1, name: '슬라임', hp:30 },
16       { id: 2, name: '고블린', hp:50 },
17       { id: 3, name: '카오스 드래곤', hp:99999999 }
18     ]
19   },
```

```
addMonster: function() {
```

```
  var max = this.monsters.reduce(function (a,b) {
    return a > b.id ? a : b.id
  }, 0)

  this.monsters.push({
    id: max + 1,
    name: this.monsterName
    hp: 500
  })
},
```

Vue : 게임만들기

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
<!-- 게임처럼 -->
<input v-model="monsterName">
<button v-on:click="addMonster">몬스터 추가하기</button>
<ul>
  For each문과 동일한 역할.
  <li v-for="(monster, index) in monsters"
    v-bind:key="monster.id"
    v-bind:class="{ strong: monster.hp > 300 }">
    ID.{{ monster.id }} {{ monster.name }} HP.{{ monster.hp }}
    <span v-if="monster.hp > 300">개새다</span>
    <button v-on:click="userAttack(index)">평타</button>
    <button v-on:click="removeMonster(index)">몬스터 삭제</button>
  </li>
</ul>
</div>
```

생성된 html

랜덤 숫자 발생!

몬스터 추가하기

- ID.1 슬라임 HP.30
- ID.2 고블린 HP.50
- ID.3 카오스 드래곤 HP.99999999 개새다

js

```
1 var app = new Vue ({
2   // id랑 맵핑되는 것이 el
3   el: '#app',
4   data: {
5
6     monsterName: '',
7     monsters: [
8       { id: 1, name: '슬라임', hp:30 },
9       { id: 2, name: '고블린', hp:50 },
10      { id: 3, name: '카오스 드래곤', hp:99999999 }
11    ]
12  },
13  },
14  },
15  },
16  },
17  },
18  },
19  },
```

```
51 removeMonster: function (index) {
52   // 찾은 인덱스 값에서 1개를 지워라
53   // 즉 인덱스 값의 정보를 지워라
54   this.monsters.splice(index,1)
55 },
56 userAttack: function (index) {
57   this.monsters[index].hp -= 10
58 },
```

Vue : 게임만들기

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

생성된 html

랜덤 숫자 발생! 몬스터 추가하기

- ID.1 슬라임 HP.30
- ID.2 고블린 HP.50
- ID.3 카오스 드래곤 HP.99999999 개새다

addMonster:

랜덤 숫자 발생! 몬스터 추가하기

- ID.1 슬라임 HP.30
- ID.2 고블린 HP.50
- ID.3 카오스 드래곤 HP.99999999 개새다
- ID.4 피카츄 HP.500 개새다
- ID.5 파이리 HP.500 개새다

removeMonster

랜덤 숫자 발생! 몬스터 추가하기

- ID.1 슬라임 HP.30
- ID.2 고블린 HP.50
- ID.3 카오스 드래곤 HP.99999999 개새다
- ID.5 파이리 HP.500 개새다

userAttack

랜덤 숫자 발생! 몬스터 추가하기

- ID.1 슬라임 HP. -20
- ID.2 고블린 HP.30
- ID.3 카오스 드래곤 HP.99998959 개새다
- ID.5 파이리 HP.280

Attack이 -10씩 적용되지만, HP가 - 가 되어도 몬스터가 죽지않는다.

Vue : 게임만들기

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

js

```
64 beforeCreate() {
65   console.log('Vue 객체를 만들기 이전입니다.')
66 },
67 created() {
68   console.log('Vue 객체를 만들었습니다.')
69 },
70 beforeMount() {
71   console.log('HTML 요소를 붙이기 전입니다.')
72 },
73 mounted() {
74   console.log('HTML 요소를 붙입니다.')
75 },
76 beforeUpdate() {
77   console.log('VDOM의 변화를 감지합니다.')
78
79   var i
80   for (i = 0; i < this.monsters.length; i++) {
81     if (this.monsters[i].hp <= 0) {
82       // splice : 배열 요소 삭제
83       this.monsters.splice(i, 1)
84     }
85   }
86 },
87 updated() {
88   console.log('VDOM의 변화를 적용합니다.')
89 },
90 beforeDestroy() {
91   console.log('Vue 객체를 파괴하기 이전입니다..')
92 },
93 destroyed() {
94   console.log('Vue 객체를 파괴하였습니다.')
95 },
96 })
```

크롬 Console

Elements	Console
top	Filter
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	
VDOM의 변화를 감지합니다.	
VDOM의 변화를 적용합니다.	

Attack시, VDOM 변화를 감지/적용
Js 파일 내 beforeUpdate()와
updated()가 attack에 영향을 끼친다.

생성된 html

userAttack

랜덤 숫자 발생!

몬스터 추가하기

• ID.1 슬라임 HP.10

평타

몬스터 삭제

• ID.2 고블린 HP.30

평타

몬스터 삭제

• ID.3 카오스 드래곤 HP.99998959

개썰다

평타

몬스터 삭제

• ID.5 파이어 HP.280

평타

몬스터 삭제

랜덤 숫자 발생!

몬스터 추가하기

• ID.2 고블린 HP.30

평타

몬스터 삭제

• ID.3 카오스 드래곤 HP.99998959

개썰다

평타

몬스터 삭제

• ID.5 파이어 HP.280

평타

몬스터 삭제