



# HomeWork (Not Completed)

사진1

```

if(this.money - tmpSum >= 0) {
    this.money -= tmpSum
}

for(var i = 0; i < this.shopListValue.length; i++) {
    this.myInventory.push({
        itemName: this.shopList[this.shopListValue[i]].itemName,
        effect: this.shopList[this.shopListValue[i]].effect
    })
}

```

← CalcBuy를 통해 MyInventory  
를 가지고 사진2, 3과 같은 코드를  
작성하였다.

사진2

```

<n></n>
<label>
    <input type="checkbox" v-model="inventoryView"
           v-on:click="inventoryList()"/>인벤토리
    <button v-on:click="installWeapon(characterAtk)">장착</button>
</label>
<table border="1">
    <tr v-if="inventoryView" v-for="(list, idx) in myInventory">
        <th align="center" width="20">번호</th>
        <th align="center" width="80">아이템명</th>
        <th align="center" width="320">효과</th>
        <th align="center" width="80">개수</th>
        <th align="center" width="80">데미지</th>
    </tr>
    <tr v-if="!inventoryView" v-for="(list, idx) in myInventory">
        <th align="center" width="20">{{ idx + 1 }}</th>
        <th align="center" width="80">{{ list.itemName }}</th>
        <th align="center" width="320">{{ list.effect.desc }}</th>
        <th align="center" width="80">{{ list.effect.amount }}</th>
        <th align="center" width="40">
            <label>
                <input type="checkbox"
                       v-model="inventoryListValue" v-bind:value="idx">
            </label>
        </th>
    </tr>
</table>

```

여기서 예제에서는 상점부분에서 구매를 막았을  
Click하면 목록에 들어가지 않고 텁만 나오나감.  
그럼에도 불구하고 부터는 정상 작동  
사진 2는 inventoryList class이다.

사진3

```

inventoryList: function(){
    if(!this.inventoryView){
        this.myInventory = []
    }
},
installWeapon: function(characterAtk){
    for(var i=0; i < this.inventoryView.length; i++){
        this.characterAtk += this.inventoryView[i].effect.atk
    }
},

```

또한 장착Click시 공격력 증가  
는 이렇게 하면 되겠다 하였지만  
안되었다.  
beforeUpdate에서 문제해결은 가능하니?

```
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   <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
7   <script src="https://cdn.jsdelivr.net/npm/lodash@4.17.21/lodash.min.js"></script>
8   <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
9   <link href="css/vue/trans.css" rel="stylesheet">
10 </head>
11 <body>  Vue의 el과 동일하게 사용.
12           ↑          Vue data 속성과 input을 번역할 때 사용.
13   <div id="app">    ↳ text 입력은 번역할 때 사용.
14     캐릭터명: <input v-model="characterName"><br>
15     레벨: <input v-model="characterLv"><br>
16     상태: <input v-model="characterStatus"><br>
17     HP: <input v-model="characterHp"><br>
18     ATK: <input v-model="characterAtk"><br>
19     DEF: <input v-model="characterDef"><br>
20
21     <p>캐릭터명: {{ characterName }}&nbsp;&nbsp;&nbsp;
22       경험치: {{ curExp }} / {{ needExp }}</p>
23     <p>레벨: {{ characterLv }}</p>
24     <p>상태: {{ characterStatus }}</p>
25     <p>HP: {{ characterHp }}</p>
26     <p>ATK: {{ characterAtk }}</p>
27     <p>DEF: {{ characterDef }}</p>
28     <p>돈: {{ money }}</p>
29
30     <button v-on:click="addMonster">몬스터 추가</button><br>
31     <button v-on:click="addManyMonster">100마리씩 몬스터 추가</button>
32     <button v-on:click="amaterasu">아마테라스</button>
```

사용자가 입력한 내용을  
즉시<P>태그로 보내는 것.

마우스 클릭 시 실행 기능 메소드 지정.

```

34     <ul>
35         <li v-for="(monster, index) in monsters"
36             v-bind:key="monster.id" → 장식단위에서 순번을 줄기. (순서가 아니라 번호)
37             v-bind:class="{ strong: monster.hp > 300 }">
38                 ID.{{ monster.id }} {{ monster.monsterName }}
39                 HP.{{ monster.hp }} STATUS.{{ monster.status }}
40                 <button v-on:click="doRemove(index)">몬스터 삭제</button>
41                 <span v-if="monster.hp > 300">강함</span>
42                 <span v-if="monster.hp <= monster.morbidity">빈사상태</span>
43                 <button v-on:click="doAttack(index)">공격</button>
44             </li>
45         </ul>
46
47     <h3>상점</h3>
48     <label>
49         <input type="checkbox" v-model="shopView"
50             v-on:click="shuffleShopList()">상점
51             method
52     </label>
53     <button v-on:click="calcBuy()">구매</button>
54     <table border="1">
55         <tr v-if="shopView">
56             <th align="center" width="20">번호</th>
57             <th align="center" width="80">이름</th>
58             <th align="center" width="80">가격</th>
59             <th align="center" width="320">효과</th>
60             <th align="center" width="40">구매</th>
61         </tr>
62         <tr v-if="shopView" v-for="(list, idx) in shopList"
63             v-bind:key="list.itemName" → 상품번호
64             <th align="center" width="20">{{ idx + 1 }}</th>
65             <th align="center" width="80">{{ list.itemName }}</th>
66             <th align="center" width="80">{{ list.price }}</th>
67             <th align="center" width="320">{{ list.effect.desc }}</th>
68             <th align="center" width="40">
69                 <label>
70                     <input type="checkbox"
71                         v-model="shopListValue" v-bind:value="idx">
72                 </label>
73             </th>

```

```
77 <script>
78     var app = new Vue({
79         el: '#app',
80         data: {
81             monsters: [
82                 { id: 1, monsterName: '슬라임', hp: 10, status: 'Default' },
83                 { id: 2, monsterName: '고블린', hp: 30, status: 'Default' },
84                 { id: 3, monsterName: '드래곤', hp: 5000, status: 'Default' }
85             ],
86             monBooks: [
87                 { monsterName: '슬라임', exp: 1, hp: 10, sprMoney: 10 },
88                 { monsterName: '고블린', exp: 2, hp: 20, sprMoney: 20 },
89                 { monsterName: '빨간 슬라임', exp: 5, hp: 10, sprMoney: 40 },
90                 { monsterName: '구울', exp: 10, hp: 50, sprMoney: 100 },
91                 { monsterName: '해골 병사', exp: 20, hp: 100, sprMoney: 200 },
92                 { monsterName: '해골 궁수', exp: 20, hp: 70, sprMoney: 200 },
93                 { monsterName: '해골 기사', exp: 50, hp: 400, sprMoney: 500 },
94                 { monsterName: '고스트', exp: 100, hp: 1000, sprMoney: 700 },
95                 { monsterName: '듀라한', exp: 200, hp: 4000, sprMoney: 1000 },
96                 { monsterName: '리치', exp: 500, hp: 10000, sprMoney: 2000 },
97                 { monsterName: '리치킹', exp: 3000, hp: 100000, sprMoney: 10000 },
98                 { monsterName: '베히모스', exp: 2000, hp: 80000, sprMoney: 5000 },
99                 { monsterName: '베히모스킹', exp: 25000, hp: 800000, sprMoney: 50000 },
100                 { monsterName: '드래곤', exp: 100000, hp: 1000000, sprMoney: 100000 },
101             ],
102             itemBooks: [
103                 { itemName: 'HP Potion I', price: 50, effect:
104                     { desc: "20 hp recover", amount: 200 } },
105                 { itemName: 'HP Potion II', price: 200, effect:
106                     { desc: "100 hp recover", amount: 200 } },
107                 { itemName: 'HP Potion III', price: 600, effect:
108                     { desc: "400 hp recover", amount: 200 } },
109                 { itemName: 'HP Potion IV', price: 1200, effect:
110                     { desc: "800hp recover", amount: 200 } },
111                 { itemName: 'HP Potion V', price: 2000, effect:
112                     { desc: "1200hp recover", amount: 200 } },
113                 { itemName: 'Sword', price: 300000, effect:
114                     { desc: "weapon", atk: 100 } },
115             ]
116         }
117     }
118 
```

```
115     { itemName: 'Iron Sword', price: 1000000, effect:
116         { desc: "weapon", atk: 200 } }
117     ],
118     characterName: '',
119     characterLv: 1,
120     characterStatus: '',
121     characterHp: 100,
122     characterAtk: 9,
123     characterDef: 1,
124     onlyCalcDef: 1,
125     curExp: 0,
126     needExp: 10,
127     monsterName: '',
128     money: 0,
129     shopList: [],
130     shopListValue: [],
131     shopView: false,
132     myInventory: []
133 },
134 methods: {
135     addMonster() {
136         var max = this.monsters.reduce(function(a, b) {
137             return a > b.id ? a : b.id
138         }, 0)
139
140         var lenMax = this.monBooks.length
141         var idx = Math.floor(Math.random() * lenMax)
142
143         this.monsters.push({
144             id: max + 1,
145             monsterName: this.monBooks[idx].monsterName,
146             hp: this.monBooks[idx].hp,
147             status: 'Default'
148         })
149     },
150 }
```

↑ 데이터 셋팅  
상자에 넣고 있어야 합니다  
그럼 A, B면 둘다 가능

```
150 addManyMonster: function() {
151     for(var i = 0; i < 100; i++) { 100개 몬스터를 추가합니다.
152         var max = this.monsters.reduce(function(a, b) {
153             return a > b.id ? a : b.id
154         }, 0)
155         var lenMax = this.monBooks.length
156         var idx = Math.floor(Math.random() * lenMax)
157         this.monsters.push({
158             id: max + 1,
159             monsterName: this.monBooks[idx].monsterName,
160             hp: this.monBooks[idx].hp,
161             status: 'Default'
162         })
163     }
164 },
165 doRemove: function(index) { 인덱스 제거. ↑
166     this.monsters.splice(index, 1)
167 },
168 doAttack: function(index) {
169     this.monsters[index].hp -= this.characterAtk
170 },
171 amaterasu: function() {
172     var i
173     마테라су 공격.
174     for(i = 0; i < this.monsters.length; i++) {
175         this.monsters[i].hp =
176             parseInt(this.monsters[i].hp -
177             this.characterAtk * 1.5) 마테라су 공격
178
179         if(this.monsters[i].hp > 0) {
180             this.monsters[i].status = 'Stun'
181         }
182     }
183 },
```

```
184     shuffleShopList: function() {
185         if(!this.shopView) {
186             this.shopListValue = []
187         }
188
189         for(var i = 0; i < 10; i++) {
190             var randIdx = Math.floor(Math.random() * this.itemBooks.length)
191             this.shopList[i] = this.itemBooks[randIdx]
192         }
193     },
194     calcBuy: function () {
195         var tmpSum = 0
196
197         console.log('calcbuy(): ' + this.shopListValue.length)
198         console.log('shopList len: ' + this.shopList.length)
199
200         for(var i = 0; i < this.shopListValue.length; i++) {
201             console.log(this.shopListValue[i])
202
203             for(var j = 0; j < this.shopList.length; j++) {
204                 console.log('inner loop')
205
206                 if(this.shopListValue[i] == j) {
207                     console.log('match!')
208                     tmpSum += this.shopList[j].price
209                     break
210                 }
211             }
212         }
213
214         if(this.money - tmpSum >= 0) {
215             this.money -= tmpSum
216
217             for(var i = 0; i < this.shopListValue.length; i++) {
218                 this.myInventory.push({
219                     itemName: this.shopList[this.shopListValue[i]].itemName,
220                     effect: this.shopList[this.shopListValue[i]].effect
221                 })
222             }
223         }
224     }
225 }
```

```
223
224     } else {
225         alert('Give me Money')
226     }
227 },
228 beforeCreate: function () {
229     console.log("Before Create: " + this.monsters)
230 },
231 created: function () {
232     console.log("Created: " + this.monsters)
233 },
234 beforeMount: function () {
235     console.log("Before Mount: " + this.monsters)
236 },
237 mounted: function () {
238     console.log("Mounted: " + this.monsters)
239 },
240 beforeUpdate: function() {
241     //console.log("Before Update: " + this.monsters)
242
243     var i
244
245     for(i = 0; i < this.monsters.length; i++) {
246         if(this.monsters[i].hp <= 0) {
247             for(var j = 0; j < this.monBooks.length; j++) {
248                 if(this.monsters[i].monsterName ==
249                     this.monBooks[j].monsterName) {
250                     this.curExp += this.monBooks[j].exp
251                     var tmp = this.monBooks[j].sprMoney
252                     this.money += parseInt(tmp + Math.random() * (tmp / 2))
253                 }
254             }
255             this.monsters.splice(i, 1) → 杀死怪物
256         }
257     }

```

ORIGINALLY before update?  
this!

Σ DROP EXP CRED

→ ~~杀死怪物~~

```
259     var criteria = this.curExp
260
261     for(; this.curExp >= this.needExp; ) {
262         this.curExp = parseInt(this.curExp - this.needExp) →lv 4p 했을 때 끝나게 됩니까.
263
264         this.characterLv += 1
265         this.characterHp += 10
266         this.characterAtk += 2
267         this.onlyCalcDef += 0.4
268         this.characterDef = Math.floor(this.onlyCalcDef)
269
270         if(this.characterLv < 10) {
271             this.needExp = parseInt(this.needExp * 1.1)
272         } else if(this.characterLv < 20) {
273             this.needExp = parseInt(this.needExp * 1.2)
274         } else if(this.characterLv < 30) {
275             this.needExp = parseInt(this.needExp * 1.3)
276         } else if(this.characterLv < 40) {
277             this.needExp = parseInt(this.needExp * 1.4)
278         } else if(this.characterLv < 50) {
279             this.needExp = parseInt(this.needExp * 1.5)
280         } else if(this.characterLv < 90) {
281             this.needExp = parseInt(this.needExp * 1.7)
282         } else if(this.characterLv < 100) {
283             this.needExp = parseInt(this.needExp * 2)
284         }
285     }
286 },
287 updated: function () {
288     console.log("Updated: " + this.monsters)
289 },
290 beforeDestroy: function () {
291     console.log("Before Destroy: " + this.monsters)
292 },
293 destroyed: function () {
294     console.log("Destroyed: " + this.monsters)
295 }
296 }))
```