

Vue의 장점:

정보를 script<=>브라우저 직통으로 연동
controller를 거칠 필요가 없어짐!!

JS에 vue파일(.js) 생성

1.<head>상에서 스크립트 링크 반영 링크 복붙

- <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14"></script>

2.<body>내부

설명:

<HTML>

1) HTML문법인 <div>내부에서 {message}를 호출

<script>->이부분 따로 빼서 호출해주는게 바람직함

1) JS에서 지원해주는 Vue라는 객체를 app로 객체화를 시킴.

2) el=>element 에 id="app"을 연동

3) 내부 data에 message의 내용을 기입

```
<div id="app">
  <h1>{{ message }}</h1>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      message: 'Hello Vue.js'
    }
  })
</script>
```

기본 예제

<HTML>

```
<div id="app">
  <p>이거시 뷰우다: {{ message }}</p>
  <p>{{ message.length }}</p>
  <p>이런식으로 배열 특정 요소 선택도 가능: {{ list[2] }}</p>
  <p>자체 변수로 맵핑도 가능하지롱! {{ list[num] }}</p>
</div>
```

<Script>

```
let app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue.js',
    list: [
      'BMW', 'Audi', 'Benz'
    ],
    num: 1
  }
})
```

<브라우저>

이거시 뷰우다: Hello Vue.js

12

이런식으로 배열 특정 요소 선택도 가능: Benz

자체 변수로 맵핑도 가능하지롱! Audi

세부 문법

- V-on:click

- V-for: "item in list"

item -> 반복되는 배열의 엘리먼트

list -> 원본 데이터 배열

- V-bind:key

- V-model

- (배열).sort((a,b)=>a-b) 가격 내리차순 정렬

- (배열).sort((a,b)=>b-a) 가격 오름차순 정렬

methods : {

priceLow(){

 this.oneroom.sort(function(a,b){

 return a.price - b.price

 });

},

priceHigh(){

 this.oneroom.sort(function(a,b){

 return b.price - a.price

ClickEvent 만들기

V-on:click

<HTML>

```
3
4 <div id="app">
5   <p>{{ count }} 번 클릭했습니다.</p>
6   <button v-on:click="increment">마! 빨리 눌러라!</button>
7 </div>
8
```

<Script>

<브라우저>

0 번 클릭했습니다.

마! 빨리 눌러라!

BlindEvent 만들기(<https://kr.vuejs.org/v2/guide/list.html>)

```

let app = new Vue({
  el: '#app',
  data: {
    count: 0
  },
  methods: {
    increment: function() {
      this.count += 1
    }
  }
})

```

V-blne

```

<li v-for="item in list"
    v-bind:key="item.id">
  ID: {{ item.id }} {{ item.name }} HP: {{ item.hp }}
</li>

```

설명list rendering방식

v-for="item in list" //foreach와 동일

->list에서 요소들을 item에 배치

v-bind:key="item.id"// item의 id를 primarykey로 설정(중복 방지)

<Script>

```

let app = new Vue({
  el: '#app',
  data: {
    list: [
      { id: 1, name: '슬라임', hp: 100 },
      { id: 2, name: '고블린', hp: 200 },
      { id: 3, name: '사슴', hp: 150 },
      { id: 4, name: '리본돼지', hp: 250 }
    ]
  }
})

```

<브라우저>

- ID: 1 슬라임 HP: 100
- ID: 2 고블린 HP: 200
- ID: 3 사슴 HP: 150
- ID: 4 리본돼지 HP: 250

새로운 ID 추가 기능

V-model text는?

```

<div id="app">
  이름: <input v-model="name">
  <button v-on:click="addMonster">몬스터 추가</button>

  <ul>
    <li v-for="item in list"
        v-bind:key="item.id">
      ID: {{ item.id }} {{ item.name }} HP: {{ item.hp }}
    </li>
  </ul>
</div>

```

```

    name: '일단은 고정값: 키메라다!!!'
  },
  methods: {
    addMonster: function () {
      let max = this.list.reduce(function(a, b) {
        return a > b.id ? a : b
      }, 0)

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

```

이름:

- ID: 1 슬라임 HP: 100
- ID: 2 고블린 HP: 200
- ID: 3 사슴 HP: 150
- ID: 4 리본돼지 HP: 250

a가b의 id보다 크냐? 크면 a값을 넣고 작으면 b의 id값을 넣어라
ID 삭제 기능

```

<li v-for="(item, index) in list"
    v-bind:key="item.id">
  ID: {{ item.id }} {{ item.name }} HP: {{ item.hp }}
  <button v-on:click="removeMonster(index)">
    몬스터 삭제
  </button>
</li>

```

index와 몬스터 삭제 버튼 추가

```

removeMonster: function (index) {
  this.list.splice(index, 1)
}

```

이름:

- ID: 1 슬라임 HP: 100
- ID: 2 고블린 HP: 200
- ID: 3 사슴 HP: 150
- ID: 4 리본돼지 HP: 250

ID 수정 기능 (공격 기능)

단계1 버튼을 누를 때 -10씩 체력이 감소하게

단계2 체력이 0이 되면 리스트에서 삭제 되게
deforeUpdate()?

몬스터 추가를 눌렀을 때 다양한 몬스터가 자동으로 추가 되게하기 지금
은 키메라만 나눔

디렉티브(directive)란?

디렉티브를 직역하면 명령,지시이다. 디렉티브는 말 그대로 무엇을 하는 것을 지시하는 것이며 Vue.js의 View영역에서 요소의 기능을 명령하는 역할을 한다.

mission 디렉티브 정리

기본 디렉티브

| 디렉티브 | 설명 |
|-----------|--|
| v-text | innerText와 동일한 기능을 수행하며
태그를 인코딩하여 문자 그대로 보여준다. |
| v-html | v-html innerHTML과 동일한 기능을 수행하며
태그를 파싱하여 화면에 구현한다. |
| v-bind | .attr()과 비슷한 기능을 수행하며
태그의 속성을 동적으로 변경할 때 사용한다. |
| v-model | 양방향 데이터를 공유할 때 사용한다. |
| v-show | css의 display 와 동일한 기능을 수행한다. |
| v-if | if 조건문 구현 |
| v-else | else 조건문 구현 |
| v-else-if | else if 조건문 구현 |
| v-for | for 반복문 구현 |
| v-pre | vue.js가 컴파일되지 않게하며
{{머쉬태시}}도 그대로 보여진다 |
| v-once | vue.js가 처음 한번만 렌더링을 수행하며
데이터가 변경되어도 처음값만 보여준다. |
| v-clock | 컴파일 되지않는 값이 순간
나오는 것을 막고 결과값만 보여준다. |