



# 웹 디자이너의 도전

## Vue.js 따라하기

Vue · tiful Korea  
이승민 | 알고랩

# 누구?



2017 알고랩 **디자이너** 및 개발자

2016 맘톡 **디자이너**

2015 산업**디자인**학과 공학사



```
switch (id) {
    case "pu_history":
        for (var i = 0; i < pu_list.length; i++) {
            if (pu_list[i] === data.SenderAddress && pu_details_list[i] === data.SenderAddressDetails && pu_name_list[i] === data.Sender && pu_phone_list[i] === data.SenderPhone) {
                return;
            }
        }
        str = str + '<tr>';
        str += '    <td>' + data.SenderAddress + '<br>' + data.SenderAddressDetails + '</td>';
        str += '    <td>' + data.Sender + '</td>';
        str += '    <td>' + data.SenderPhone + '</td>';
        str += '    <td><input type="button" value="주소지 입력" onclick="insertAddr(\'';
        str += id + '\', \'' + data.SenderAddress + '\', \'' + data.SenderAddressDetails + '\', \'' + data.Sender + '\', \'' + data.SenderPhone + '\', \'' + data.SenderAddressLegal;
        str += '\', \'' + data.SenderAddressSector + '\', \'' + data.SenderAddressHub + '\', \'' + data.SenderAddressLatitude + '\', \'' + data.SenderAddressLongitude + '\')"></td>';
        str = str + '</tr>';
        pu_list[index] = data.SenderAddress;
        pu_details_list[index] = data.SenderAddressDetails;
        pu_name_list[index] = data.Sender;
        pu_phone_list[index] = data.SenderPhone;
        index++;
        break;
    case "do_history":
        for (var i = 0; i < do_list.length; i++) {
            if (do_list[i] === data.ConsigneeAddress && do_details_list[i] === data.ConsigneeAddressDetails && do_name_list[i] === data.Consignee && do_phone_list[i] === data.ConsigneePhone) {
                return;
            }
        }
        str = str + '<tr>';
        str += '    <td>' + data.ConsigneeAddress + '<br>' + data.ConsigneeAddressDetails + '</td>';
        str += '    <td>' + data.Consignee + '</td>';
        str += '    <td>' + data.ConsigneePhone + '</td>';
        str += '    <td><input type="button" value="주소지 입력" onclick="insertAddr(\'';
        str += id + '\', \'' + data.ConsigneeAddress + '\', \'' + data.ConsigneeAddressDetails + '\', \'' + data.Consignee + '\', \'' + data.ConsigneePhone + '\', \'' + data.ConsigneeAddressLegal;
        str += '\', \'' + data.ConsigneeAddressSector + '\', \'' + data.ConsigneeAddressHub + '\', \'' + data.ConsigneeAddressLatitude + '\', \'' + data.ConsigneeAddressLongitude + '\')"></td>';
        str = str + '</tr>';
        do_list[index] = data.ConsigneeAddress;
        do_details_list[index] = data.ConsigneeAddressDetails;
        do_name_list[index] = data.Consignee;
        do_phone_list[index] = data.ConsigneePhone;
        index++;
        break;
    default:
}
}
```

hardcoding




# 디자이너도 Vue.js를 쓸 수 있다?

디자이너니까 오프라인용 굿즈 **쯤이야!**  
마케팅 콘텐츠  
편집 디자인



# 디자이너도 Vue.js를 쓸 수 있다?

디자이너니까 UX/UI 디자인 **썸이야!**  
웹 퍼블리싱  
Vue.js



# 개발자에게 친절한 디자이너?

CSS는 제가 만질게요.

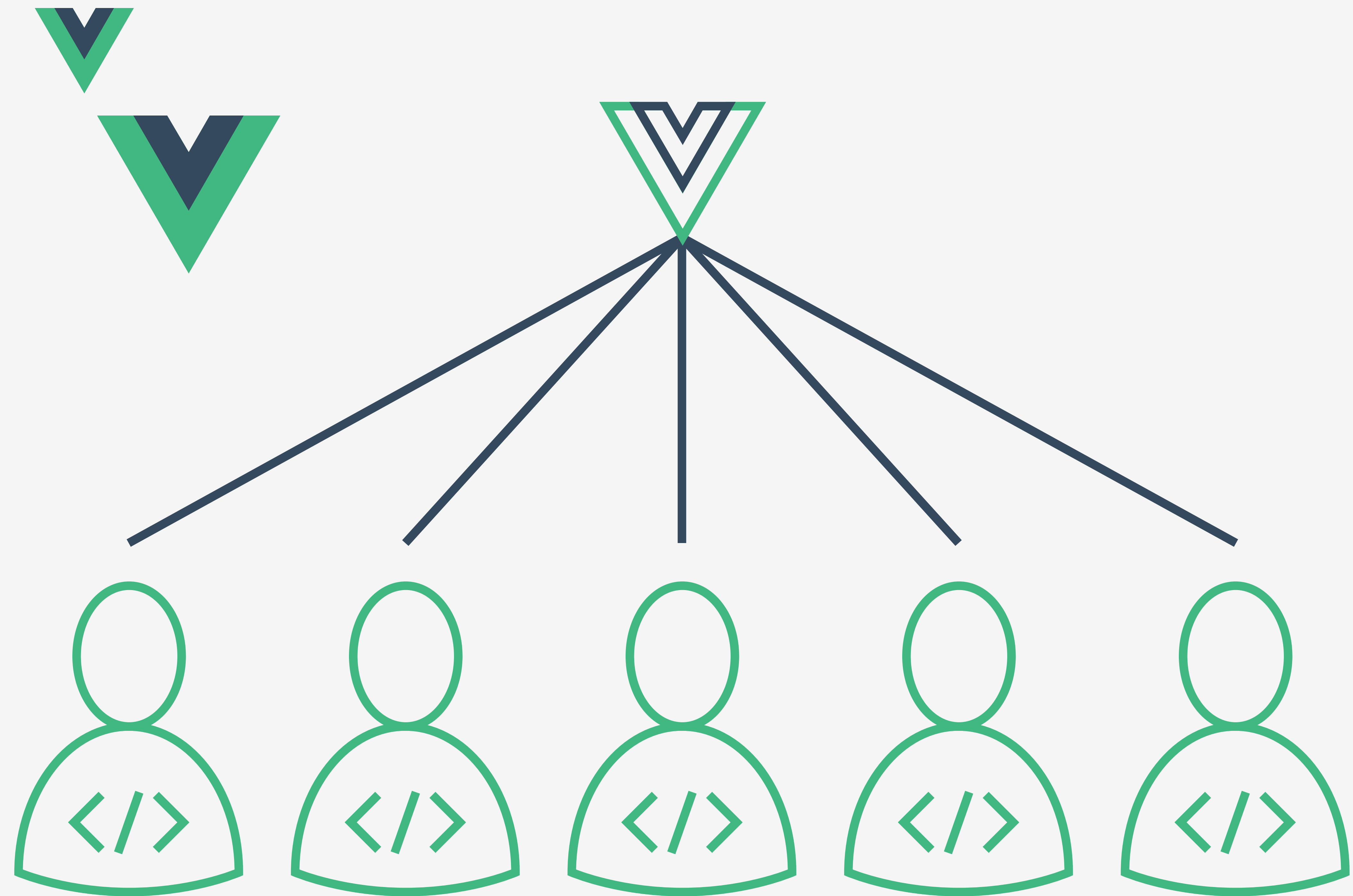
API input과 output은 이렇게 해주세요.







Vue.js  
써봐?







# 왜 jQuery로 만족할 수 없는가?

```
1 <div id="output"></div>
2 <button id="increment">Increment</button>
3 <script>
4   var counter = 0;
5   $(document).ready(function() {
6     var $output = $('#output');
7     $('#increment').click(function() {
8       counter++;
9       $output.html(counter);
10    });
11    $output.html(counter);
12  });
13 </script>
```

jQuery



```
1 <div id="app">
2   <div>{{ counter }}</div>
3   <button v-on:click="increment">Increment</button>
4 </div>
5 <script>
6   new Vue({
7     el: '#app',
8     data: {
9       counter: 0
10    },
11    methods: {
12      increment() {
13        this.counter++;
14      }
15    }
16  });
17 </script>
```

Vue.js





# 왜 React로 만족할 수 없는가?

```
1 render () {
2   let { items } = this.props
3   let children
4   if (items.length > 0) {
5     children = (
6       <ul>
7         {items.map(item =>
8           <li key={item.id}>{item.name}</li>
9         )}
10      </ul>
11    )
12   } else {
13     children = <p>No items found.</p>
14   }
15   return (
16     <div className='list-container'>
17       {children}
18     </div>
19   )
20 }
```

React

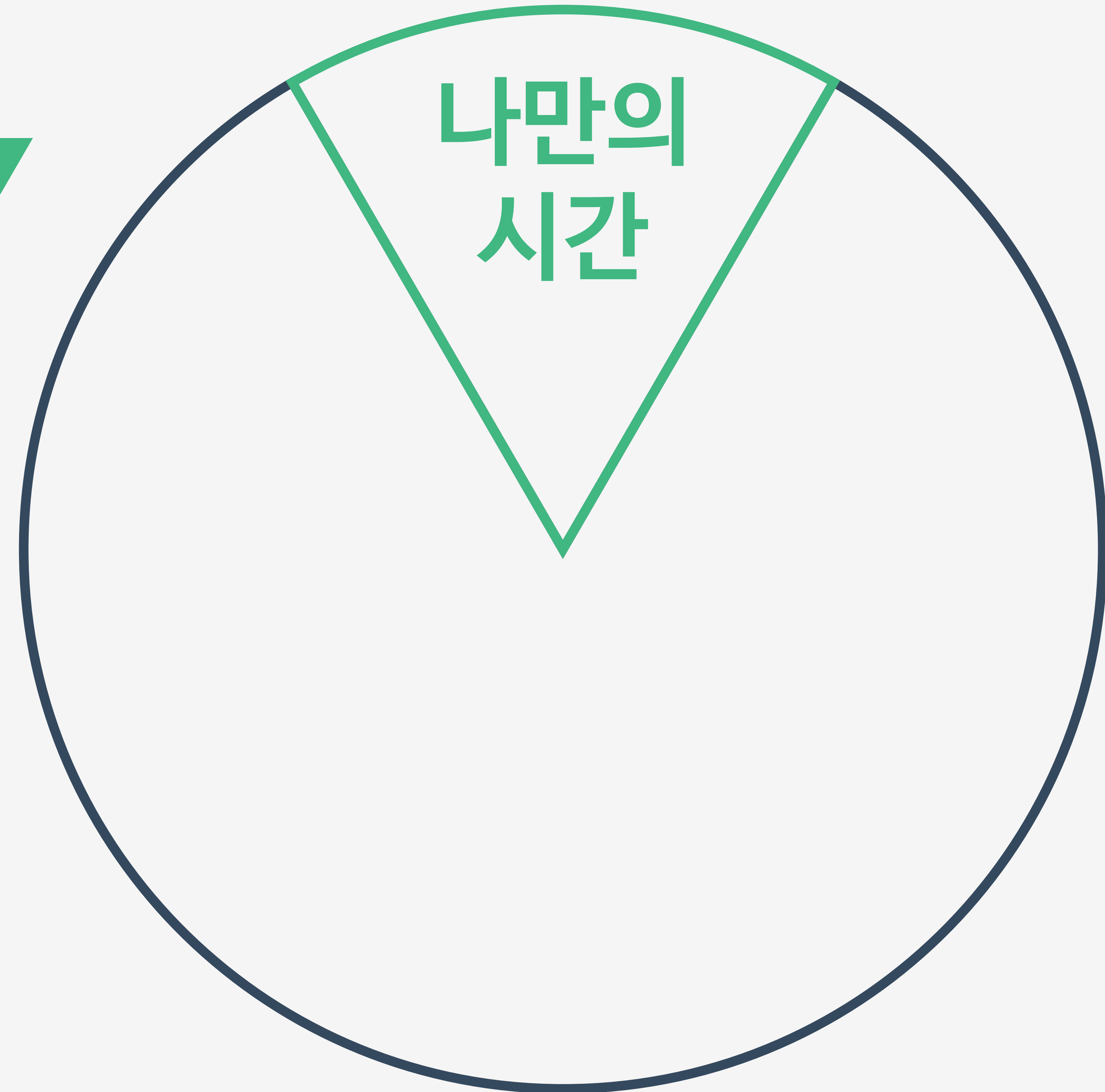
```
1 <template>
2   <div class="list-container">
3     <ul v-if="items.length">
4       <li v-for="item in items" :key="item.id">
5         {{ item.name }}
6       </li>
7     </ul>
8     <p v-else>No items found.</p>
9   </div>
10 </template>
```


Vue.js





잠은?





# 쉽고 간단하게 JavaScript 쓰기?

시간이 없지만 공부 짬이야!









# Vue.js를 setup하는 방법?

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8   <script src="https://unpkg.com/vue"></script>
9 </head>
10 <body>
11   <script src="app.js"></script>
12 </body>
13 </html>
```







# Vue instance를 만드는 방법?

```
1 new Vue({  
2   el: "#app",  
3   data: {  
4     |  
5   }  
6 });  
7
```





# Vue instance data object

```
1  new Vue({  
2    el: "#app",  
3    data: {  
4      playerHealth: 100,  
5      monsterHealth: 100,  
6      gameIsRunning: false  
7    }  
8  });
```







template syntax



# {{ textInterpolation }}

```
<div class="healthbar">  
  <div class="healthbar text-center" style="background-color: g  
    {{ playerHealth }}  
  </div>  
</div>
```





binding inline styles

# v-bind:style, :style

```
<div class="healthbar">
  <div
    class="healthbar text-center"
    style="background-color: green; margin: 0; color: white;"
    :style="{width: playerHealth + '%'}">
    {{ playerHealth }}
  </div>
</div>
```







# conditional rendering

## v-if, v-else

```
<section class="row controls" v-if="!gameIsRunning">
  <div class="small-12 columns">
    <button id="start-game" @click="startGame">시작</button>
  </div>
</section>
<section class="row controls" v-else>
  <div class="small-12 columns">
    <button id="attack" @click="attack">공부</button>
    <button id="special-attack" @click="specialAttack">밤샘</button>
    <button id="heal" @click="heal">낮잠</button>
    <button id="give-up" @click="giveUp">포기</button>
  </div>
</section>
```





template syntax directives

# v-on:click, @click

```
<div class="small-12 columns">  
  <button id="start-game" @click="startGame">시작</button>  
</div>
```








# Vue instance methods

```
methods: {  
  startGame: function() {  
    this.gameIsRunning = true;  
    //initialize healthbars  
    this.playerHealth = 100;  
    this.monsterHealth = 100;  
    //initialize turns  
    this.turns = [];  
  },  
}
```





쉽네?  
공부하면 되지!

시간이 없지만 공부 짬이야!





# Vue.js & JavaScript

```
attack: function() {  
  //player attacks  
  var damage = this.calculateDamage(3, 10);  
  this.monsterHealth -= damage;  
  this.turns.unshift({  
    isPlayer: true,  
    text: 'smilee가 vue.js에게 ' + damage + '의 피해를 입혔습니다.'  
  });  
  
  //check if the player won  
  if (this.checkWin()) {  
    return;  
  }  
  
  this.monsterAttacks();  
},
```





공부만으로는 안돼?  
밤을 새면 되지!

피곤하지만 밤샘 쫓아야!





# Vue.js & JavaScript

```
specialAttack: function() {  
  //player attacks  
  var damage = this.calculateDamage(10, 20);  
  this.monsterHealth -= damage;  
  this.turns.unshift({  
    isPlayer: true,  
    text: 'smilee가 vue.js에게 *' + damage + '*의 피해를 입혔습니다.'  
  });  
  //check if the player won  
  if (this.checkWin()) {  
    return;  
  }  
  this.monsterAttacks();  
},
```





# Vue.js 정복하기? 밤을 새면 되지!

일주일 밤샘 쫓아야... 힘드네. 낮잠을 자볼까?




# Vue.js & JavaScript

```
heal: function() {  
  if (this.playerHealth <= 90) {  
    this.playerHealth += 10;  
  } else {  
    this.playerHealth = 100;  
  }  
  this.turns.unshift({  
    isPlayer: true,  
    text: 'smilee가 ' + 10 + '의 피해를 회복했습니다.'  
  });  
  this.monsterAttacks();  
},
```







# Vue.js 정복하기? 적당히 해!

이렇게 살 바에는 포기가...



# Vue.js & JavaScript

```
giveUp: function() {  
  this.gameIsRunning = false;  
},
```





# Vue.js 정복하기? 끝까지 가보자!



Vue.js 꼭 무찌를거야!





# list rendering v-for

```
<section class="row log" v-if="turns.length > 0">
  <div class="small-12 columns">
    <ul>
      <li v-for="turn in turns" :class="{ 'player-turn': turn.isPlayer, 'monster-turn': !turn.isPlayer}">
        {{ turn.text }}
      </li>
    </ul>
  </div>
</section>
```





# 감사합니다!

ideas007@naver.com

Vue · tiful Korea

**이승민 | 알고랩**