

프론트엔드 개발자를 위한 Vue.js 시작하기

# Vue.js </> 시작하기

## Built-in Directive



한국기술교육대학교  
온라인 평생교육원

## 학습내용

- v-bind, v-if, v-for
- v-on, v-model

## 학습목표

- v-bind, v-if, v-for Directive의 사용 방법에 대해 설명할 수 있다.
- v-on, v-model Directive의 사용 방법에 대해 설명할 수 있다.

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 1 v-bind

- Component의 속성 혹은 Class 값을 바인딩 하기 위한 Directive

## v-bind, v-if, v-for

### 2 class 바인딩

- v-bind을 이용하여 css class 및 style 등록
- v-bind:class="{ className: data }"

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 2 class 바인딩

- data가 true이면 class가 추가되고 false이면 추가되지 않음

```
<div v-bind:class="{ active: isActive }"></div>
.....
data(){
  return {
    isActive: true
  }
}
//출력 : <div class="active"></div>
```

## v-bind, v-if, v-for

### 2 class 바인딩

- 객체 바인딩

```
<div v-bind:class="classObject"></div>
.....
classObject: {
  'active': true,
  'text-danger': true
}
//<div class="active text-danger"></div>
```

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 2 class 바인딩

#### 배열 바인딩

```
<div v-bind:class="[activeClass, errorClass]"></div>
.....
data() {
  return {
    activeClass: 'active',
    errorClass: 'text-danger'
  }
}
//<div class="active text-danger"></div>
```

## v-bind, v-if, v-for

### 2 class 바인딩

#### 3항 연산자 이용 바인딩

```
<div v-bind:class="[isActive ? activeClass : errorClass]"></div>
//<div class="active"></div>
```

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 2 class 바인딩

- 축약형으로 바인딩
- v-bind는 콜론(:)으로 이용 가능함

```
<a v-bind:href="url">google</a>
<a :href="url">google</a>
```

## v-bind, v-if, v-for

### 3 v-if

- 값이 true이면 출력 false이면 출력되지 않음
- v-else, v-else-if와 같이 사용 가능함

```
<div v-if="type === 'A'">A</div>
<div v-else-if="type === 'B'">B</div>
<div v-else-if="type === 'C'">C</div>
<div v-else>Not A/B/C</div>
```

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 4 v-show

- **v-if**와 동일
- display style 값 변경으로 화면에 출력을 제어함
- **v-else**와 같이 사용하지 못함

## v-bind, v-if, v-for

### 4 v-show

- <template> 태그에 사용할 수 없음

```
<div v-show="false">I am Show</div>
//<div style="display: none;">I am Show</div>
```

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 5 v-for

- 반복 출력
- v-for="item in items"
- v-for Directive가 사용된 Element에는 v-bind:key가 선언되어 있어야 함

```
<li v-for="item in items" :key="item">
  {{ item.message }}
</li>
```

## v-bind, v-if, v-for

### 5 v-for

- index 값 추출

```
<li v-for="(item, index) in items" :key="index">
  {{ index }} - {{ item.message }}
</li>
```



# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 5 v-for

- Object Literal를 v-for에 사용하면 객체의 멤버들을 iterator 해 줌

```
<li v-for="value in object" :key="value">
    {{ value }}
</li>
.....
object: {
    firstName: 'John',
    lastName: 'Doe',
    age: 30
}
```

## v-bind, v-if, v-for

### 5 v-for

- key, index 획득 가능
- (value, key, index) In Object

```
<li v-for="(value, key, index) in object" :key="index">
    {{index}} - {{key}} : {{ value }}
</li>
```

# v-bind, v-if, v-for

## v-bind, v-if, v-for

### 5 v-for

#### range

```
<span v-for="n in 2" :key="n">{{ n }} - Hello <br/></span>
```

# v-on, v-model

## v-on, v-model

### 1 v-on

- 이벤트(Event) 바인딩
- v-on:click="someFun"
- 축약으로 @click="someFun" 사용가능

```
<button v-on:click="greet">Greet</button>
.....
methods: {
  greet: function (event) {
    .....
  }
}
```

## v-on, v-model

### 1 v-on

- parameter 전달

```
<button v-on:click="warn('hello', $event)">
  Submit
</button>
.....
methods: {
  warn: function (message, event) {
    if (event) event.preventDefault()
    .....
  }
}
```

# v-on, v-model

## v-on, v-model

### 2 이벤트(Event) Modifier

- 다양한 modifier 등록으로 쉽게 이벤트 제어 가능

```
<form v-on:submit.prevent="onSubmit"></form>
```

## v-on, v-model

### 2 이벤트(Event) Modifier

.stop

event.stopPropagation()

.prevent

event.preventDefault()

.capture

- 이벤트를 capture 하겠다는 의미
- 하위 element에서 이벤트가 발생하여도 capture가 선언된 곳에서 이벤트를 먼저 처리하기 위해서 사용

# v-on, v-model

## v-on, v-model

### 2 이벤트(Event) Modifier

.self

자신에게 발생한 이벤트만  
처리하겠다는 의도

.once

한번만 이벤트 허용

## v-on, v-model

### 2 이벤트(Event) Modifier

- 이벤트 수식을 등록하면서 chain 형태로 여러 개 등록 가능

```
<a v-on:click.stop.prevent="doThat"></a>
```

# v-on, v-model

## v-on, v-model

### 3 키(Key) Modifier

- 키 이벤트 처리시 이벤트가 발생한 key 지정

```
<input @keyup.enter="keyEvent">
```

## v-on, v-model

### 4 v-model

- Two-way Data Binding

```
<input v-model="name" placeholder="input name">
```

# v-on, v-model

## v-on, v-model

### 4 v-model

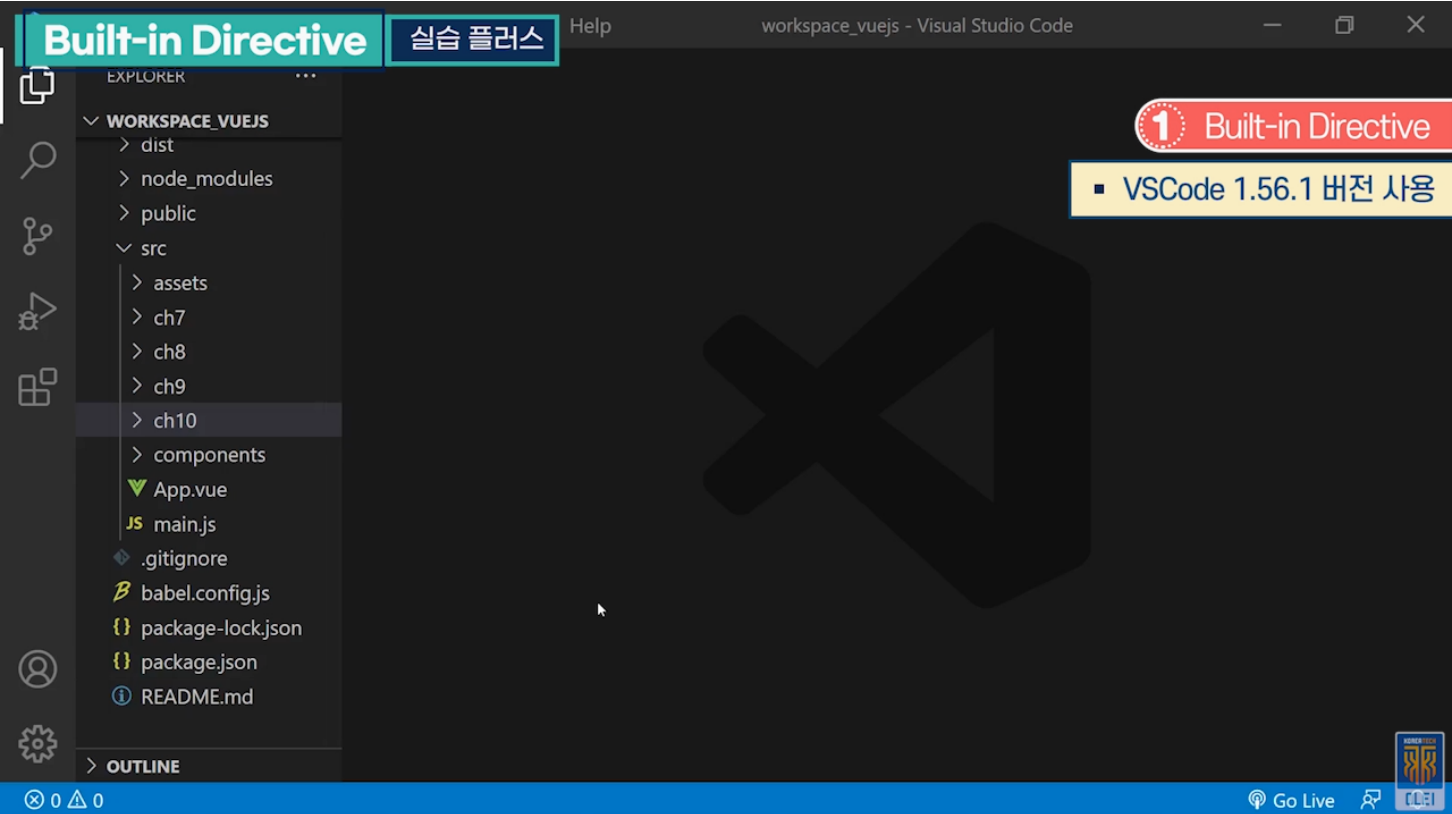
- checkbox의 경우 v-model에 의해 바인딩 되는 데이터는 true/false임
- 여러 개의 checkbox가 있고 check 된 value 값을 문자열 배열로 저장하는 것도 가능

## v-on, v-model

### 4 v-model

```
<input type="checkbox" value="kkang" v-model="checkedNames">  
<input type="checkbox" value="kim" v-model="checkedNames">  
<input type="checkbox" value="lee " v-model="checkedNames">
```

# 실습 플러스



실습단계
src → ch11 서브 폴더 생성
Component 내에서 Built-in Directive 테스트
생성한 폴더 내 Test.vue 파일 생성
script 영역 정의
정의한 데이터를 Built-in Directive로 HTML에 Binding
v-if Directive 테스트에 사용됨
v-for 테스트를 위한 데이터 선언
methods 지정
로그인 타입을 변경할 때 호출되는 함수
Event 시 호출되는 함수 정의
Event Callback 함수 추가 정의
정의한 데이터와 함수를 이용하기 위해 템플릿에서 내장 Directive 사용
v-bind 테스트
데이터 값으로 Binding 하는 style 정의



# 실습 플러스

실습단계
v-bind:를 :으로 축약 가능
v-if 테스트
현재 loginType 데이터가 username일 경우
template을 복사하여 추가로 v-else 정의
Event 추가
v-for 테스트
선언되어 있는 변수 개수 만큼 li Tag 반복됨
한 번 반복되면서 자동으로 들어간 index 값과 데이터 출력
배열 데이터 이외에 object로 v-for 선언 가능
다양한 event 테스트
input Tag 사용하여 KeyEvent 테스트
Two Way Data Binding
이전에 생성한 ch10의 html, main.js, vue.config.js 파일 복사하여 이용
Vue 명령으로 실행
브라우저에 Component 출력
console 창에서 각 event의 차이 확인 가능
첫 번째 버튼, console에 출력되지만 Refresh 상태
두 번째 버튼, Refresh 되지 않음