

# 커스텀 디렉티브

## 소개

코어에 포함된 기본 디렉티브 세트(예: `v-model` 또는 `v-show`) 외에도 Vue를 사용하면 커스텀 디렉티브를 정의할 수 있습니다.

우리는 Vue에서 컴포넌트 기초와 컴포저블이라는 두 가지 형태의 코드 재사용을 도입했습니다. 컴포넌트는 주요 빌딩 블록(building-block)이고, 컴포저블은 상태 저장 로직을 재사용하는 데 중점을 둡니다. 반면에 커스텀 디렉티브는 주로 일반 엘리먼트에 대한 저수준(low-level) DOM 접근과 관련된 로직을 재사용하기 위한 것입니다.

커스텀 디렉티브는 컴포넌트의 생명주기 혹은 포함하는 객체처럼 정의됩니다. 혹은 디렉티브가 바인딩된 엘리먼트를 수신합니다. 다음은 엘리먼트가 Vue에 의해 DOM에 삽입될 때, `<input>` 에 포커스 되는 커스텀 디렉티브 구현의 예제입니다:

```
<script setup>
// 템플릿에서 v-focus로 활성화 가능
const vFocus = {
  mounted: (el) => el.focus()
}
</script>

<template>
  <input v-focus />
</template>
```

vue

포커스 되어야 함

페이지의 다른 곳을 클릭하지 않았다고 가정하면, 위의 인풋은 자동으로 포커스 되어야 합니다. 이 디렉티브는 페이지 로드 시 뿐만 아니라, Vue에서 동적으로 엘리먼트를 삽입할 때도 작동하기 때문에 `autofocus` 속성보다 더 유용합니다.

`<script setup>` 에서 `v` 접두사로 시작하는 모든 camelCase 변수를 커스텀 디렉티브로 사용할 수 있습니다. 위의 예에서 `vFocus` 는 템플릿에서 `v-focus` 로 사용할 수 있습니다.

`<script setup>` 을 사용하지 않는 경우, `directives` 옵션을 사용하여 커스텀 디렉티브를 등록할 수 있습니다:

```
export default {
  setup() {
    /* ... */
  },
  directives: {
    // 템플릿에서 v-focus로 활성화 가능
    focus: {
      /* ... */
    }
  }
}
```

js

앱 수준에서 커스텀 디렉티브를 전역적으로 등록하는 것도 일반적입니다:

```
const app = createApp({})

// 모든 컴포넌트에서 v-focus를 사용할 수 있도록 합니다.
app.directive('focus', {
```

js

```
/* ... */
})
```

#### TIP

커스텀 디렉티브는 원하는 기능을 직접 DOM 조작을 통해서만 달성할 수 있는 경우에만 사용해야 합니다. 가능하면 `v-bind` 와 같은 내장 디렉티브를 사용하여 선언적 템플릿을 사용하는 것이 더 효율적이고 서버 렌더링에 친숙하기 때문입니다.

## 디렉티브 훅

디렉티브를 정의하는 객체는 다음과 같은 여러 훅 기능을 제공할 수 있습니다(모두 선택 사항):

```
const myDirective = {
  // 바인딩된 엘리먼트의 속성 또는
  // 이벤트 리스너가 적용되기 전에 호출됩니다.
  created(el, binding, vnode, prevVnode) {
    // 인자에 대한 자세한 내용은 아래를 참고.
  },
  // 엘리먼트가 DOM에 삽입되기 직전에 호출됩니다.
  beforeMount(el, binding, vnode, prevVnode) {},
  // 바인딩된 엘리먼트의 부모 컴포넌트 및
  // 모든 자식 컴포넌트의 mounted 이후에 호출됩니다.
  mounted(el, binding, vnode, prevVnode) {},
  // 부모 컴포넌트의 updated 전에 호출됩니다.
  beforeUpdate(el, binding, vnode, prevVnode) {},
  // 바인딩된 엘리먼트의 부모 컴포넌트 및
  // 모든 자식 컴포넌트의 updated 이후에 호출됩니다.
  updated(el, binding, vnode, prevVnode) {},
  // 부모 컴포넌트의 beforeUnmount 이후에 호출됩니다.
  beforeUnmount(el, binding, vnode, prevVnode) {},
  // 부모 컴포넌트의 unmounted 전에 호출됩니다.
  unmounted(el, binding, vnode, prevVnode) {}
}
```

js

## 훅 인자

디렉티브 훅에는 다음 인자가 전달됩니다:

`el` : 디렉티브가 바인딩된 엘리먼트입니다. DOM을 직접 조작하는 데 사용할 수 있습니다.

`binding` : 다음 속성을 포함하는 객체입니다.

`value` : 디렉티브에 전달된 값입니다. 예를 들어 `v-my-directive="1 + 1"` 에서 `value`는 `2` 입니다.

`oldValue` : 이것은 `beforeUpdate` 및 `updated` 에서만 사용할 수 있습니다. 값이 변경되었는지 여부에 관계없이 사용 가능합니다.

`arg` : 디렉티브에 전달된 인자(있는 경우). 예를 들어 `v-my-directive:foo` 에서 인자는 `"foo"` 입니다.

`modifiers` : 수식어가 있는 경우 수식어를 포함하는 객체입니다. 예를 들어 `v-my-directive.foo.bar` 에서 수식어 객체는 `{ foo: true, bar: true }` 입니다.

`instance` : 디렉티브가 사용되는 컴포넌트의 인스턴스입니다.

`dir` : 디렉티브를 정의하는 객체

`vnode` : 바인딩된 엘리먼트를 나타내는 기본 `VNode`.

`prevVnode` : 이전 렌더링에서 바인딩된 엘리먼트를 나타내는 `VNode`입니다. `beforeUpdate` 및 `updated` 훅에서만 사용할 수 있습니다.

다음과 같은 디렉티브를 사용한다고 가정한 예제를 살펴봅시다:

```
<div v-example:foo.bar="baz">
```

template

binding 인자는 다음과 같은 형태의 객체입니다:

```
{
  arg: 'foo',
  modifiers: { bar: true },
  value: /* `baz`의 값 */,
  oldValue: /* 업데이트 전 `baz`의 값 */
}
```

js

내장 디렉티브와 유사하게 커스텀 디렉티브 인자는 동적일 수 있습니다. 예를 들어:

```
<div v-example:[arg]="value"></div>
```

template

여기서 디렉티브 인자는 컴포넌트 상태의 `arg` 속성을 기반으로 반응형으로 업데이트됩니다.

참고

`el` 을 제외하고 이러한 인자들은 읽기 전용으로 처리하고 절대 수정해서는 안 됩니다. 혹 간에 정보를 공유해야 하는 경우 엘리먼트의 `dataset`을 통해 공유하는 것이 좋습니다.

## 간단하게 함수로 사용하기

커스텀 디렉티브가 `mounted` 및 `updated` 에 대해 동일한 동작을 갖는 것이 일반적이며, 다른 혹은 필요하지 않습니다. 이러한 경우 디렉티브를 객체가 아닌 함수로 정의할 수 있습니다:

```
<div v-color="color"></div>
```

template

```
app.directive('color', (el, binding) => {
  // 이 함수가 호출되는 시점은 `mounted`와 `updated`입니다.
  el.style.color = binding.value
})
```

js

## 객체를 값으로 전달하기

디렉티브에 여러 값이 필요한 경우, JavaScript 객체 리터럴을 전달할 수도 있습니다. 디렉티브는 모든 유효한 JavaScript 표현식을 사용할 수 있음을 기억하십시오.

```
<div v-demo="{ color: 'white', text: '안녕!' }"></div>
```

template

```
app.directive('demo', (el, binding) => {
  console.log(binding.value.color) // => "white"
  console.log(binding.value.text) // => "안녕!"
})
```

js

# 컴포넌트에서 사용

추천하지 않음

컴포넌트에 커스텀 디렉티브를 사용하는 것은 권장되지 않습니다. 컴포넌트에 여러 개의 루트 노드가 있을 경우 예상치 못한 동작이 발생할 수 있습니다.

컴포넌트에 사용될 때 커스텀 디렉티브는 폴스루 속성과 유사하게 항상 컴포넌트의 루트 노드에 적용됩니다.

<code>&lt;MyComponent v-demo="test" /&gt;</code>	template
<code>&lt;!-- MyComponent 템플릿에서 --&gt;</code>	template
<code>&lt;div&gt; &lt;!-- 여기에 v-demo 디렉티브가 적용됩니다. --&gt; &lt;span&gt;컴포넌트 컨텐츠...&lt;/span&gt; &lt;/div&gt;</code>	

컴포넌트는 잠재적으로 하나 이상의 루트 노드를 가질 수 있습니다. 다중 루트 컴포넌트에 적용될 때, 디렉티브는 무시되고 경고가 발생합니다. 속성과 달리, 디렉티브는 `v-bind="$attrs"` 로 다른 요소에 전달될 수 없습니다.