



폴스루 속성

이 페이지는 이미 [컴포넌트 기본](#)을 읽었다고 가정합니다. 컴포넌트가 처음이라면 먼저 해당 내용을 읽어보세요.

속성 상속

"폴스루 속성"이란 컴포넌트에 전달되었지만, 해당 컴포넌트의 `props`나 `emits`에 명시적으로 선언되지 않은 속성이나 `v-on` 이벤트 리스너를 의미합니다. 일반적인 예로는 `class`, `style`, `id` 속성이 있습니다.

컴포넌트가 하나의 루트 엘리먼트만 렌더링할 때, 폴스루 속성은 자동으로 루트 엘리먼트의 속성에 추가됩니다. 예를 들어, 다음과 같은 템플릿을 가진 `<MyButton>` 컴포넌트가 있다고 가정해 봅시다:

```
<!-- <MyButton>의 템플릿 -->
<button>Click Me</button>
```

template

그리고 부모가 이 컴포넌트를 다음과 같이 사용할 때:

```
<MyButton class="large" />
```

template

최종적으로 렌더링되는 DOM은 다음과 같습니다:

```
<button class="large">Click Me</button>
```

html

여기서 `<MyButton>` 은 `class` 를 허용되는 prop으로 선언하지 않았습니다. 따라서 `class` 는 폴스루 속성으로 간주되어 `<MyButton>` 의 루트 엘리먼트에 자동으로 추가됩니다.



자식 컴포넌트의 루트 엘리먼트에 이미 `class` 나 `style` 속성이 있다면, 부모로부터 상속받은 `class` 및 `style` 값과 병합됩니다. 이전 예시에서 `<MyButton>` 의 템플릿을 다음과 같이 변경한다고 가정해봅시다:

```
<!-- <MyButton>의 템플릿 -->
<button class="btn">Click Me</button>
```

template

그러면 최종적으로 렌더링되는 DOM은 다음과 같이 됩니다:

```
<button class="btn large">Click Me</button>
```

html

v-on 리스너 상속

동일한 규칙이 `v-on` 이벤트 리스너에도 적용됩니다:

```
<MyButton @click="onClick" />
```

template

`click` 리스너는 `<MyButton>` 의 루트 엘리먼트, 즉 네이티브 `<button>` 엘리먼트에 추가됩니다. 네이티브 `<button>` 이 클릭되면 부모 컴포넌트의 `onClick` 메서드가 실행됩니다. 만약 네이티브 `<button>` 에 이미 `v-on` 으로 바인딩된 `click` 리스너가 있다면, 두 리스너가 모두 실행됩니다.

중첩 컴포넌트 상속

컴포넌트가 루트 노드로 또 다른 컴포넌트를 렌더링하는 경우, 예를 들어 `<MyButton>` 을 리팩토링하여 루트로 `<BaseButton>` 을 렌더링한다고 가정해봅시다:

```
<!-- 또 다른 컴포넌트를 단순히 렌더링하는 <MyButton/>의 템플릿 -->
<BaseButton />
```

template

그러면 `<MyButton>` 이 받은 폴스루 속성은 자동으로 `<BaseButton>` 으로 전달됩니다.

참고할 점:

- 전달된 속성에는 `prop`으로 선언된 속성이나, `<MyButton>` 에서 선언된 이벤트의 `v-on` 리스너는 포함되지 않습니다. 즉, 선언된 `prop`과 리스너는 `<MyButton>` 에서 "소비"됩니다.
- 전달된 속성은 `<BaseButton>` 에서 `prop`으로 선언되어 있다면 `prop`으로 받아질 수 있습니다.



속성 상속 비활성화

컴포넌트가 속성을 자동으로 상속받지 않도록 하려면, 컴포넌트 옵션에서 `inheritAttrs: false` 를 설정할 수 있습니다.

3.3 버전부터는 `<script setup>` 에서 `defineOptions` 를 직접 사용할 수도 있습니다:

```
<script setup>
defineOptions({
  inheritAttrs: false
})
// ...setup 로직
</script>
```

vue

속성 상속을 비활성화하는 일반적인 시나리오는 속성을 루트 노드가 아닌 다른 엘리먼트에 적용해야 할 때입니다. `inheritAttrs` 옵션을 `false` 로 설정하면 폴스루 속성을 어디에 적용할지 완전히 제어할 수 있습니다.

이 폴스루 속성들은 템플릿 표현식에서 `$attrs` 로 직접 접근할 수 있습니다:

```
<span>폴스루 속성: {{ $attrs }}</span>
```

template

`$attrs` 객체에는 컴포넌트의 `props` 나 `emits` 옵션에 선언되지 않은 모든 속성(예: `class` , `style` , `v-on` 리스너 등)이 포함됩니다.

몇 가지 참고 사항:

`prop`과 달리, 폴스루 속성은 JavaScript에서 원래의 케이스를 유지하므로, `foo-bar` 와 같은 속성은 `$attrs['foo-bar']` 로 접근해야 합니다.

`@click` 과 같은 `v-on` 이벤트 리스너는 `$attrs.onClick` 아래의 함수로 객체에 노출됩니다.

이전 섹션의 `<MyButton>` 컴포넌트 예시를 사용해보면, 실제 `<button>` 엘리먼트를 스타일링을 위해 추가적인 `<div>` 로 감싸야 할 때가 있습니다:

```
<div class="btn-wrapper">
  <button class="btn">Click Me</button>
</div>
```

template

`class` 나 `v-on` 리스너와 같은 모든 폴스루 속성을 바깥쪽 `<div>` 가 아니라 내부 `<button>` 에 적용하고 싶습니다. 이를 위해 `inheritAttrs: false` 와 `v-bind="$attrs"` 를 사용할 수 있습니다:



```
    <button class="btn" v-bind="$attrs">Click Me</button>
  </div>
```

`v-bind` 에 인자가 없을 때는 객체의 모든 속성을 대상 엘리먼트의 속성으로 바인딩합니다.

다중 루트 노드에서의 속성 상속

하나의 루트 노드를 가진 컴포넌트와 달리, 여러 루트 노드를 가진 컴포넌트는 자동 폴스루 속성 동작이 없습니다. `$attrs` 가 명시적으로 바인딩되지 않으면 런타임 경고가 발생합니다.

```
<CustomLayout id="custom-layout" @click="changeValue" />
```

template

`<CustomLayout>` 이 다음과 같은 다중 루트 템플릿을 가진 경우, Vue는 폴스루 속성을 어디에 적용해야 할지 확신할 수 없으므로 경고가 발생합니다:

```
<header>...</header>
<main>...</main>
<footer>...</footer>
```

template

`$attrs` 가 명시적으로 바인딩되면 경고가 사라집니다:

```
<header>...</header>
<main v-bind="$attrs">...</main>
<footer>...</footer>
```

template

자바스크립트에서 폴스루 속성 접근하기

필요하다면, `<script setup>` 에서 `useAttrs()` API를 사용하여 컴포넌트의 폴스루 속성에 접근할 수 있습니다:

```
<script setup>
import { useAttrs } from 'vue'
```

vue



`<script setup>` 을 사용하지 않는 경우, `attrs` 는 `setup()` 컨텍스트의 속성으로 노출됩니다:

```
export default {js
  setup(props, ctx) {
    // 폴스루 속성은 ctx.attrs로 노출됩니다
    console.log(ctx.attrs)
  }
}
```

여기서 `attrs` 객체는 항상 최신 폴스루 속성을 반영하지만(성능상의 이유로) 반응형이 아닙니다. 변경 사항을 감지하기 위해 `watcher`를 사용할 수 없습니다. 반응성이 필요하다면 `prop`을 사용하세요. 또는, 각 업데이트마다 최신 `attrs` 로 부수 효과를 수행하려면 `onUpdated()` 를 사용할 수 있습니다.

[🔗](#) [GitHub](#)에서 이 페이지 편집

[< Previous](#)

컴포넌트 `v-model`

[Next >](#)

슬롯