폴스루 속성

이 페이지에서는 컴포넌트 기초를 이미 읽었다고 가정합니다. 컴포넌트를 처음 사용하는 경우, 그 문서를 먼저 읽으십시오.

속성 상속

"폴스루(fallthrough: 대체) 속성"은 컴포넌트에 전달되는 속성 또는 v-on 이벤트 리스너 이지만, 이것을 받는 컴포넌트의 **props** 또는 **emits**에서 명시적으로 선언되지 않은 속성입니다. 일반적인 예로는 class , style , id 속성이 있습니다.

컴포넌트가 싱글 루트 엘리먼트를 렌더링하면 폴스루 속성이 루트 엘리먼트의 속성에 자동으로 추가됩니다. 예를 들어 <MyButton> 템플릿이 있는 컴포넌트가 있다고 가정합니다:

template

```
<!-- <MyButton>의 템플릿 -->
<button>클릭하기</button>
```

그리고 이 컴포넌트를 다음과 같이 사용하는 부모:

template

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

최종 렌더링된 DOM은 다음과 같습니다:

html

```
<button class="large">클릭하기</button>
```

여기서 <MyButton> 은 class 를 허용되는 프로퍼티로 선언하지 않았습니다. 따라서 class 는 폴스루 어트리뷰트로 처리되어 <MyButton> 의 루트 엘리먼트에 자동으로 추가됩니다.

class 와 style 의 병합

자식 컴포넌트의 루트 엘리먼트에 이미 class 또는 style 속성이 있는 경우, 상위 엘리먼트에서 상속된 class 또는 style 값과 병합됩니다. 이전 예에서 <MyButton> 의 템플릿을 다음과 같이 변경한다면:

template

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

그러면 최종 렌더링된 DOM은 다음과 같습니다:

html

```
<button class="btn large">클릭하기</button>
```

v-on 리스너 상속

v-on 이벤트 리스너에도 동일한 규칙이 적용됩니다:

template

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

click 리스너는 <MyButton> 의 루트 엘리먼트인 <button> 엘리먼트에 추가됩니다. <button> 을 클릭하면 부모 컴포넌트의 onClick 메서드가 트리거됩니다. <button> 에 이미 v-on 으로 바인딩된 click 리스너가 있는 경우 두 리스너가 모두 트리거됩니

다.

중첩된 컴포넌트 상속

컴포넌트가 다른 컴포넌트를 루트 노드로 렌더링하는 경우, 예를 들어 <MyButton> 을 리팩터링하여 <BaseButton> 을 루트로 렌더링합니다:

template
```
<!-- 다른 컴포넌트 하나를 렌더링하는 <MyButton/> 템플릿 -->
<BaseButton />
```

그러면 <MyButton> 이 수신한 폴스루 속성이 자동으로 <BaseButton> 으로 전달됩니다.

참고:

전달되는 속성이 <MyButton> 에서 props 로 선언되었거나 emit 으로 등록된 핸들러일 경우, <BaseButton> 으로 전달되지 않습니다. <MyButton> 에서 명시적으로 선언되어 "사로잡혔기(consumed)" 때문입니다.

그러나 전달되는 속성이 <MyButton> 의 템플릿에서 다시 선언된 경우, props 나 핸들러로 허용될 수 있습니다.

속성 상속 비활성화

만약 컴포넌트가 속성을 자동으로 상속받지 않도록 하려면, 컴포넌트의 옵션에서 inheritAttrs: false 를 설정하면 됩니다.

3.3 버전부터는 <script setup> 에서 직접 **defineOptions** 를 사용할 수도 있습니다.

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

속성 상속을 비활성화하는 일반적인 시나리오는 루트 노드 이외의 다른 엘리먼트에 속성을 적용해야 하는 경우입니다. inheritAttrs 옵션을 false 로 설정하면 폴스루 속성을 적용해야 하는 위치를 완전히 제어할 수 있습니다.

이러한 폴스루 속성은 템플릿 표현식에서 $attrs 로 직접 접근할 수 있습니다:

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

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

참고:

props와 달리 폴스루 속성은 JavaScript에서 원 표기를 유지하므로 foo-bar 와 같은 속성은 $attrs['foo-bar'] 로 접근해야 합니다.

@click 과 같은 v-on 이벤트 리스너는 $attrs.onClick 속성에 노출 됩니다.

이전 섹션의 <MyButton> 컴포넌트 예제 사용 - 때로는 스타일 지정을 위해 실제 <button> 엘리먼트를 로 추가 래핑해야 할 수도 있습니다:

template
```
<div class="btn-wrapper">
  <button class="btn">클릭하기</button>
</div>
```

여기서 class 및 v-on 리스너와 같은 폴스루 속성이 외부 가 아닌 내부 <button> 에 적용되기를 원할 수 있습니다. inheritAttrs: false 로 설정하고, v-bind="$attrs" 로 이를 구현할 수 있습니다:

template
```
<div class="btn-wrapper">
  <button class="btn" v-bind="$attrs">클릭하기</button>
</div>
```

인자없는 **v-bind** 는 객체의 모든 속성을 대상 엘리먼트의 속성으로 묶는다는 것을 기억하세요.

다중 루트 노드에서 속성 상속

단일 루트 노드가 있는 컴포넌트와 달리 여러 루트 노드가 있는 컴포넌트에는 자동 속성 폴스루 동작이 없습니다. $attrs 가 명시적으로 바인딩되지 않은 경우 런타임 경고가 발행됩니다.

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

<CustomLayout> 에 다음과 같은 다중 루트 템플릿이 있는 경우, Vue가 폴스루 속성을 적용할 위치를 확신할 수 없기 때문에 경고가 표시됩니다:

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

$attrs 가 명시적으로 바인딩된 경우, 경고가 표시되지 않습니다:

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

JavaScript에서 폴스루 속성 접근하기

필요한 경우 useAttrs() API를 사용하여 <script setup> 에서 컴포넌트의 폴스루 속성에 접근할 수 있습니다:

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

const attrs = useAttrs()
</script>
```

<script setup> 을 사용하지 않으면 attrs 가 setup() 컨텍스트의 속성으로 노출됩니다:

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

여기서 attrs 객체는 항상 최신 폴스루 속성을 반영하지만 (성능상의 이유로) 반응하지 않습니다. 감시자를 사용하여 변경 사항을 관찰할 수 없습니다. 반응성이 필요하다면 prop를 사용하세요. 또는 onUpdated() 를 사용하여 업데이트할 때마다 최신 attrs 로 부작용을 수행할 수 있습니다.