

컴포넌트 등록

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

Vue 컴포넌트는 템플릿을 발견할 때, Vue가 구현 할 수 있도록 위치를 "등록"해야 합니다. 컴포넌트를 등록하는 방법에는 전역 및 로컬의 두 가지가 있습니다.

전역 등록

`.component()` 메서드를 사용하여 현재 **Vue** 앱에서 컴포넌트를 전역으로 사용할 수 있도록 할 수 있습니다:

```
import { createApp } from 'vue'

const app = createApp({})

app.component(
  // 등록될 이름
  'MyComponent',
  // 구현체
  {
    /* ... */
  }
)
```

js

SFC를 사용하는 경우, 가져온 `.vue` 파일을 등록합니다:

```
import MyComponent from './App.vue'

app.component('MyComponent', MyComponent)
```

js

`.component()` 메서드는 다음과 같이 연결할 수 있습니다:

```
app
  .component('ComponentA', ComponentA)
  .component('ComponentB', ComponentB)
  .component('ComponentC', ComponentC)
```

js

전역으로 등록된 컴포넌트는 앱 내의 모든 컴포넌트 템플릿에서 사용할 수 있습니다:

```
<!-- 이것은 앱 내부의 모든 컴포넌트 내에서 작동합니다. -->
<ComponentA/>
<ComponentB/>
<ComponentC/>
```

template

이는 모든 자식 컴포넌트에도 적용됩니다. 즉, 이 세 컴포넌트 모두 서로의 내부에서도 사용될 수 있습니다.

로컬 등록

편리하지만 전역 등록에는 몇 가지 단점이 있습니다:

전역 등록은 빌드 시스템이 사용하지 않는 컴포넌트를 제거하는 것(일명 "tree-shaking")을 방지합니다. 컴포넌트를 전역적으로 등록했지만 결국 앱의 어느 곳에서도 사용하지 않으면 최종 번들에 계속 포함됩니다.

전역 등록은 대규모 앱에서 의존 관계를 덜 명확하게 만듭니다. 자식 컴포넌트를 사용하는 부모 컴포넌트에서 자식 컴포넌트의 구현을 찾기가 어렵습니다. 이것은 너무 많은 전역 변수를 사용하는 것과 유사하므로 장기적인 유지 관리 측면에 영향을 줄 수 있습니다.

로컬 등록은 등록된 컴포넌트의 가용성 범위를 현재 컴포넌트로만 제한합니다. 의존 관계를 더 명시적으로 만들고 트리-쉐이킹에 더 친숙합니다.

SFC를 사용할 때, `<script setup>` 로 가져온 컴포넌트는 등록절차 없이 로컬로 사용할 수 있습니다:

```
<script setup>
import ComponentA from './ComponentA.vue'
</script>

<template>
  <ComponentA />
</template>
```

vue

`<script setup>` 이 아닌 경우 `components` 옵션을 사용해야 합니다:

```
import ComponentA from './ComponentA.js'

export default {
  components: {
    ComponentA
  },
  setup() {
    // ...
  }
}
```

js

`components` 객체의 각 속성에 대한 키는 컴포넌트의 이름으로 등록되고, 값은 컴포넌트의 구현이 포함됩니다. 위의 예는 ES2015 속성 약칭을 사용하고 있으며 다음과 동일합니다:

```
export default {
  components: {
    ComponentA: ComponentA
  }
  // ...
}
```

js

로컬로 등록된 컴포넌트는 자식 컴포넌트에서 **사용할 수 없습니다**. 이 경우 `ComponentA` 는 현재 컴포넌트에서만 사용할 수 있으며, 자식 컴포넌트에서 사용할 수 없습니다.

컴포넌트 이름 표기법

가이드 문서에서는 컴포넌트를 등록할 때 PascalCase 이름을 사용합니다. 이유는 이렇습니다:

PascalCase 이름은 유효한 JavaScript 식별자입니다. 이렇게 하면 JavaScript에서 컴포넌트를 더 쉽게 가져오고 등록할 수 있습니다. IDE의 자동 완성 기능도 지원합니다.

`<PascalCase />` 는 기본 HTML의 템플릿 엘리먼트가 아닌 Vue 컴포넌트임을 더 명확하게 합니다. 또한 Vue 컴포넌트를 사용자 정의 엘리먼트(웹 컴포넌트)와 구별합니다.

이것은 SFC 또는 문자열 템플릿으로 작업할 때 권장되는 스타일입니다. 그러나 **in-DOM** 템플릿 구문 분석 주의 사항에서 설명한 것처럼 PascalCase 태그는 in-DOM 템플릿에서 사용할 수 없습니다.

운 좋게도 Vue는 PascalCase를 사용하여 등록된 컴포넌트에 대한 kebab-case 태그 해석을 지원합니다. 이것은 MyComponent 로 등록된 컴포넌트가 <MyComponent> 또는 <my-component> 를 통해 템플릿에서 참조될 수 있음을 의미합니다. 이를 통해 템플릿 소스에 관계없이 동일한 JavaScript 컴포넌트 등록 코드를 사용할 수 있습니다.