

Vue 앱 만들기

앱 인스턴스

모든 Vue 앱은 `createApp` 함수를 사용하여 새로운 **앱 인스턴스**를 생성하는 것으로 시작합니다:

```
import { createApp } from 'vue'

const app = createApp({
  /* 최상위 컴포넌트 옵션 */
})
```

js

최상위 컴포넌트

우리가 `createApp` 에 전달하는 객체는 사실 컴포넌트입니다. 모든 앱에는 다른 컴포넌트를 자식으로 포함할 수 있는 "최상위 컴포넌트"가 필요합니다.

싱글 파일 컴포넌트를 사용하는 경우 일반적으로 다른 파일에서 루트 컴포넌트를 가져옵니다:

```
import { createApp } from 'vue'
// 싱글 파일 컴포넌트에서 최상위 컴포넌트 앱을 가져옵니다.
import App from './App.vue'

const app = createApp(App)
```

js

이 가이드의 많은 예제에는 단일 컴포넌트만 필요하지만 대부분의 실제 앱은 중첩되고 재사용 가능한 컴포넌트 트리로 구성됩니다. 예를 들어 Todo 앱의 컴포넌트 트리는 다음과 같을 수 있습니다:

```
App (최상위 컴포넌트)
├── TodoList
│   ├── TodoItem
│   │   ├── TodoDeleteButton
│   │   └── TodoEditButton
└── TodoFooter
    ├── TodoClearButton
    └── TodoStatistics
```

가이드의 뒷부분에서는 여러 컴포넌트를 함께 정의하고 구성하는 방법에 대해 설명합니다. 그 전에는 단일 컴포넌트 내부에서 일어나는 일에 초점을 맞추겠습니다.

앱 마운트하기

앱 인스턴스는 `.mount()` 메서드가 호출될 때까지 아무 것도 렌더링하지 않습니다. "컨테이너"가 될 실제 DOM 엘리먼트 또는 셀렉터 문자열을 인자로 필요합니다.

```
<div id="app"></div>
```

html

```
app.mount('#app')
```

js

앱의 최상위 컴포넌트 콘텐츠는 컨테이너 엘리먼트 내에서 렌더링됩니다. 컨테이너 엘리먼트 자체는 앱의 일부로 간주되지 않습니다.

`.mount()` 메서드는 반드시 앱의 환경설정 및 에셋(asset)이 모두 등록 완료된 후에 호출되어야 합니다. 또한 에셋 등록 메서드와 달리 반환 값은 앱 인스턴스가 아닌 최상위 컴포넌트 인스턴스입니다.

DOM 내부의 최상위 컴포넌트 템플릿

빌드 과정 없이 Vue를 사용할 때, 마운트 컨테이너 내부에 직접 최상위 컴포넌트의 템플릿을 작성할 수 있습니다:

```
<div id="app">
  <button @click="count++">{{ count }}</button>
</div>
```

html

```
import { createApp } from 'vue'
```

js

```
const app = createApp({
  data() {
    return {
      count: 0
    }
  }
})
```

```
app.mount('#app')
```

최상위 컴포넌트에 아직 `template` 옵션이 없으면 Vue는 자동으로 컨테이너의 `innerHTML` 을 템플릿으로 사용합니다.

앱 환경설정

앱 인스턴스는 몇 가지 앱 레벨의 옵션을 구성할 수 있는 `.config` 객체를 노출합니다. 예를 들어 모든 자식 컴포넌트에서 에러를 캡처하는 앱 레벨의 에러 핸들러를 정의합니다.

```
app.config.errorHandler = (err) => {
  /* 에러 처리 */
}
```

js

앱 인스턴스는 앱 범위의 에셋을 등록하기 위한 몇 가지 방법도 제공합니다. 예를 들어 컴포넌트 등록:

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

js

이렇게 하면 `TodoDeleteButton` 을 앱 어디에서나 사용할 수 있습니다. 컴포넌트 및 기타 에셋 유형에 대한 등록은 가이드의 뒷부분에서 설명합니다. **API** 앱에서 앱 인스턴스 API의 전체 목록을 찾아볼 수도 있습니다.

앱을 마운트하기 전에 앱의 모든 환경설정을 적용했는지 확인하세요!

멀티 앱 인스턴스

앱 인스턴스는 동일한 페이지 내 하나로 제한되지 않습니다. `createApp` API를 사용하면 여러 Vue 앱이 동일한 페이지에 공존할 수 있으며, 각각은 구성 및 전역 자산에 대한 고유한 범위를 갖습니다.

```
const app1 = createApp({  
  /* ... */  
})  
app1.mount('#container-1')  
  
const app2 = createApp({  
  /* ... */  
})  
app2.mount('#container-2')
```

js

서버에서 렌더링 된 대규모 페이지의 HTML을 개선하고 특정 부분을 제어하기 위해 Vue가 필요한 경우, 페이지 전체를 단일 Vue 앱 인스턴스로 마운트하지 마십시오. 대신 여러 개의 작은 앱 인스턴스를 만들고 담당해야 하는 엘리먼트에 마운트합니다