

생명 주기 훅

각 Vue 컴포넌트 인스턴스는 생성될 때 일련의 초기화 단계를 거칩니다. 예를 들어, 데이터 감시를 설정하고, 템플릿을 컴파일하고, 인스턴스를 DOM에 마운트하고, 데이터가 변경되면 DOM을 업데이트해야 합니다. 그 과정에서 생명 주기 훅(lifecycle hooks)이라 불리는 함수도 실행하여, 특정 단계에서 개발자가 의도하는 로직이 실행될 수 있도록 합니다.

생명 주기 훅 등록하기

예를 들어 `onMounted` 훅은 컴포넌트가 초기 렌더링 및 DOM 노드 생성이 완료된 후 코드를 실행하는 데 사용할 수 있습니다:

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

onMounted(() => {
  console.log('컴포넌트가 마운트 됐습니다.')
})
</script>
```

vue

인스턴스 생명 주기의 여러 단계에서 호출되는 다른 훅도 있으며, 가장 일반적으로 사용되는 것은 `onMounted`, `onUpdated`, `onUnmounted` 가 있습니다.

`onMounted` 를 호출하면, Vue는 등록된 콜백 함수를 현재 활성 컴포넌트 인스턴스와 자동으로 연결합니다. 이를 위해서는 컴포넌트 설정 중에 이러한 훅은 **동기적으로** 등록해야 합니다. 예를 들어 다음과 같이 하지 마십시오:

```
setTimeout(() => {
  onMounted(() => {
    // 작동하지 않습니다.
  })
}, 100)
```

js

이 훅이 `setup()` 또는 `<script setup>` 내에 배치된 코드 순서에 영향을 받는다는 것을 의미하는 것이 아닙니다. `onMounted()` 는 호출 스택이 동기식이며, `setup()` 내에서 시작되는 경우에만 외부 함수를 실행하는 방식으로 사용할 수 있습니다.

```
import { onMounted } from 'vue'

export default {
  setup() {
    // onMounted() 혹은 setup 함수 내에서 사용하는 경우에만
    // 마치 외부의 함수를 호출한 것처럼 작성할 수 있습니다.
    //
    // 이렇게 작성된 onMounted() 혹은
    // 컴포넌트가 마운트 된 이후 콜백 함수를 실행하지만
    // `this`를 통해 컴포넌트 인스턴스에 접근할 수 없습니다.
    onMounted(function() {
      // `this`를 통해 컴포넌트 인스턴스에 접근할 수 없습니다.
      console.log('onMounted가 호출 되었습니다:', this)
    })
  },

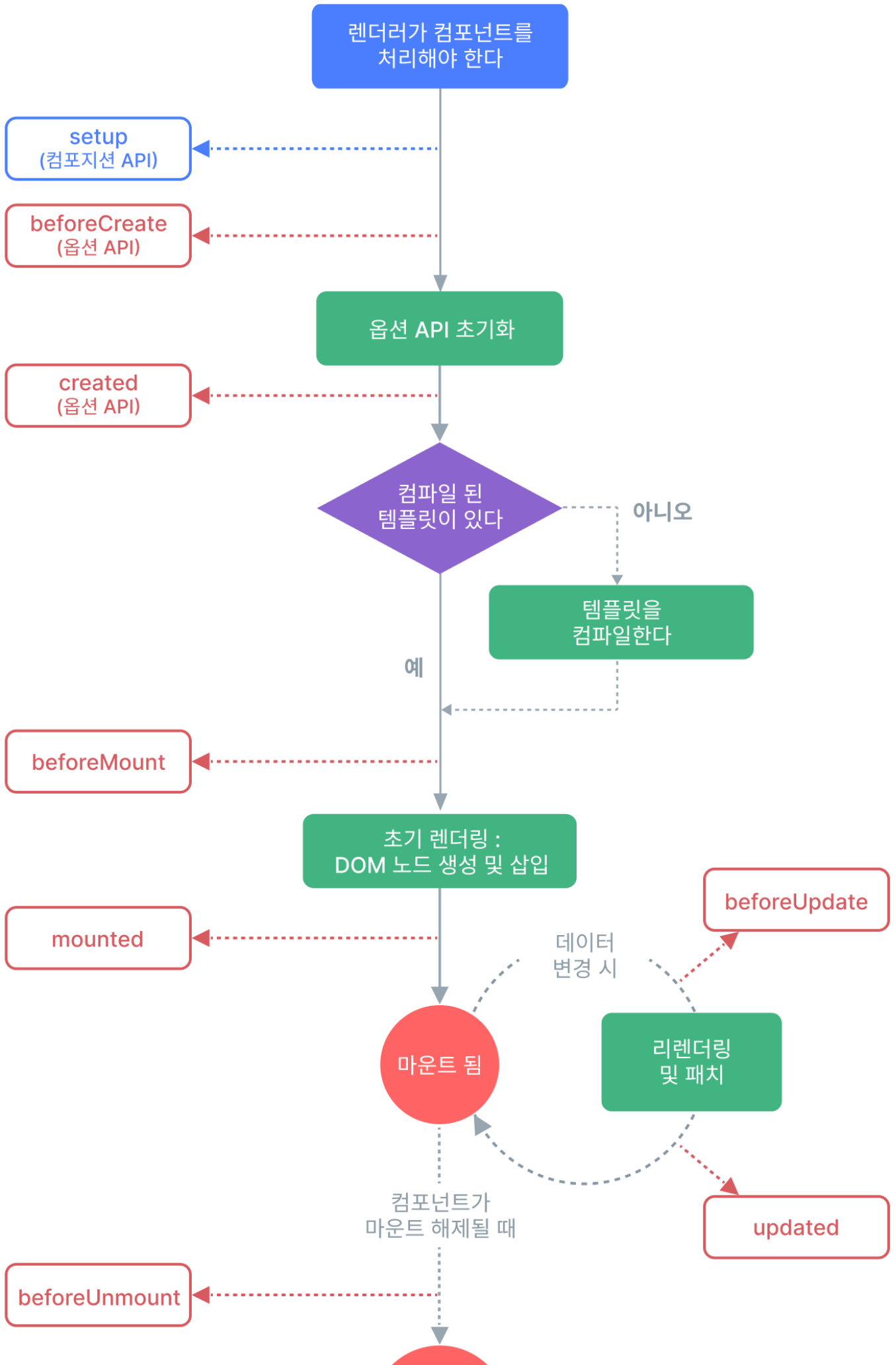
  // 컴포넌트가 마운트 된 후,
  // 옵션 API 방식의 mounted() 훅을 실행.
  mounted() {
    // `this`를 통해 컴포넌트 인스턴스에 접근할 수 있습니다.
    console.log('mounted()가 호출 되었습니다:', this)
  }
}
```

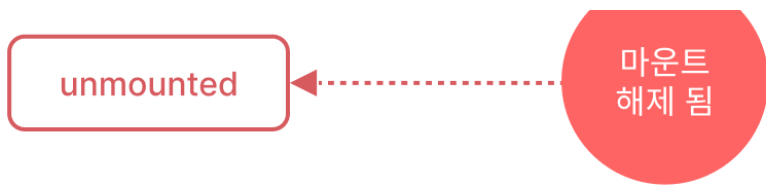
js

}
}

생명 주기 표

다음은 인스턴스 생명 주기에 대한 표입니다. 지금 진행 중인 모든 것을 완전히 이해할 필요는 없지만, 더 많이 배우고 구축함에 따라 유용한 참고 자료가 될 것입니다.





생명 주기 혹은 모든 종류와 사용 사례에 대한 자세한 내용은 생명 주기 **API**를 참조하세요.