

Lifecycle Hooks

Lifecycle hooks are special functions that let you run code at specific stages of a component's life from creation to destruction. Vue provides a set of lifecycle hooks that let you tap into these stages.

Lifecycle Diagram

```
beforeCreate → created → beforeMount → mounted  
→ beforeUpdate → updated  
→ beforeUnmount → unmounted
```

Example :

```
<template>  
  <div>  
    <h2>Count: {{ count }}</h2>  
    <button @click="count++">Increment</button>  
  </div>  
</template>  
  
<script setup>  
import { ref, onBeforeMount, onMounted, onBeforeUpdate, onUpdated,  
onBeforeUnmount, onUnmounted } from 'vue'  
  
const count = ref(0)  
  
// onBeforeMount: Just before the template is mounted to DOM  
onBeforeMount(() => {  
  console.log('onBeforeMount: DOM not mounted yet')  
})  
  
// onMounted: After the component is inserted into the DOM  
onMounted(() => {  
  console.log('onMounted: Component is in the DOM now')  
})  
  
// onBeforeUpdate: Called before DOM updates when reactive data changes  
onBeforeUpdate(() => {  
  console.log('onBeforeUpdate: Reactive value changed, DOM will update soon')  
})  
  
// onUpdated: Called after the DOM updates  
onUpdated(() => {  
  console.log('onUpdated: DOM updated')  
})  
  
// onBeforeUnmount: Called before the component is removed from DOM  
onBeforeUnmount(() => {
```

```
    console.log('onBeforeUnmount: Component will be removed')
  })

  // onUnmounted: Called after component is removed
  onUnmounted(() => {
    console.log('onUnmounted: Component has been removed')
  })
</script>
```

1. beforeCreate()

Not available in Composition API. It's implicitly part of setup. So we don't use beforeCreate() directly.

2. created()

Not available in Composition API. Again, all code inside