Chapter 5 – Vue3 Lifecycle Hooks

Asst.Prof. Dr. Umaporn Supasitthimethee
ผศ.ดร.อุมาพร สุภสิทธิเมธี

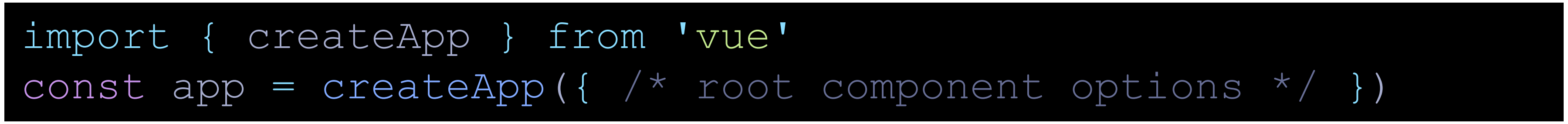
https://vuejs.org/guide/essentials/lifecycle.html#lifecycle-hooks

Lifecycle Hooks

- Each component instance goes through a series of initialization steps when it's created
- For example,
    - it needs to set up data observation
    - compile the template
    - mount the instance to the DOM
    - update the DOM when data changes
- Along the way, it also runs pre-defined functions called *lifecycle hooks*, giving users the opportunity to add their own code at specific stages.

Creating a Vue Application

- The `createApp` function takes a root component `(App.vue)` as an argument and returns a Vue app instance.
- The Vue app instance returned by the `createApp` is also called an application context object.

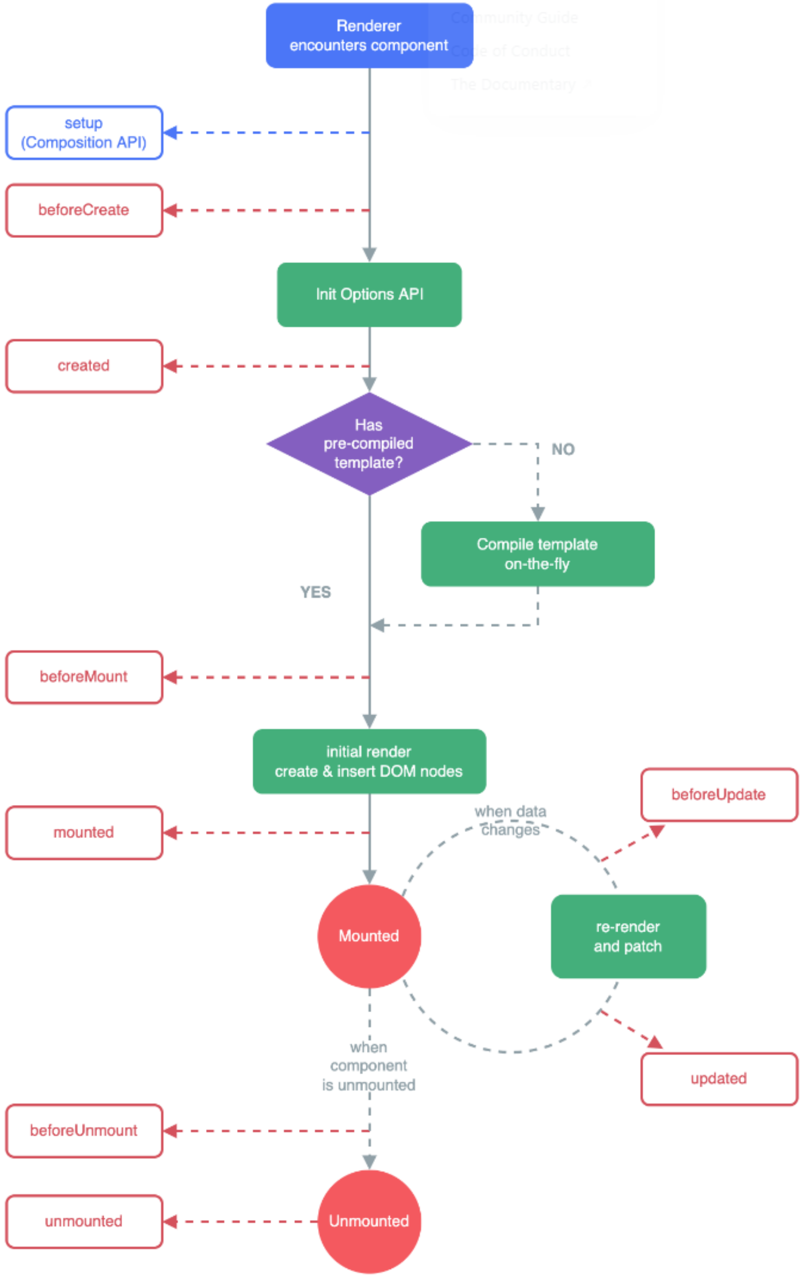


```
import { createApp } from 'vue'
const app = createApp({ /* root component options */ })
```

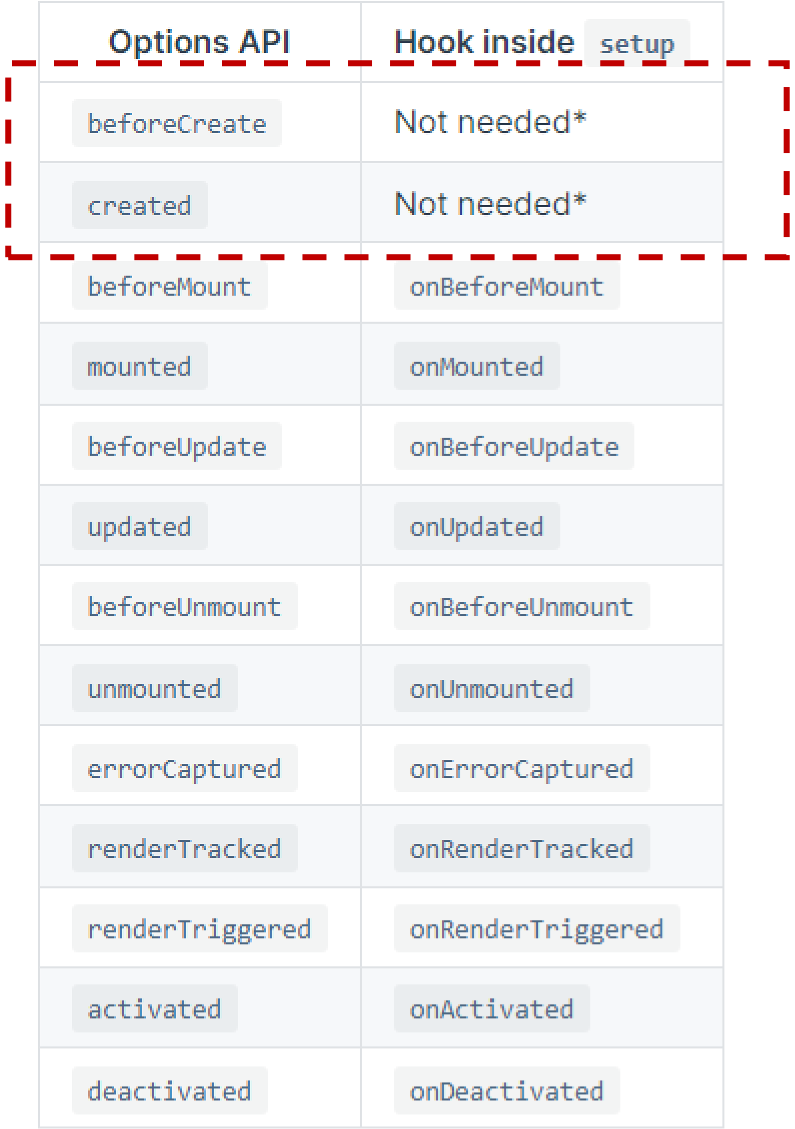
- An application instance won't render anything until its `.mount()` method is called.

https://vuejs.org/guide/essentials/application.html#creating-a-vue-application

Lifecycle Hooks Diagram



Because setup is run around the beforeCreate and created lifecycle hooks, you do not need to explicitly define them.



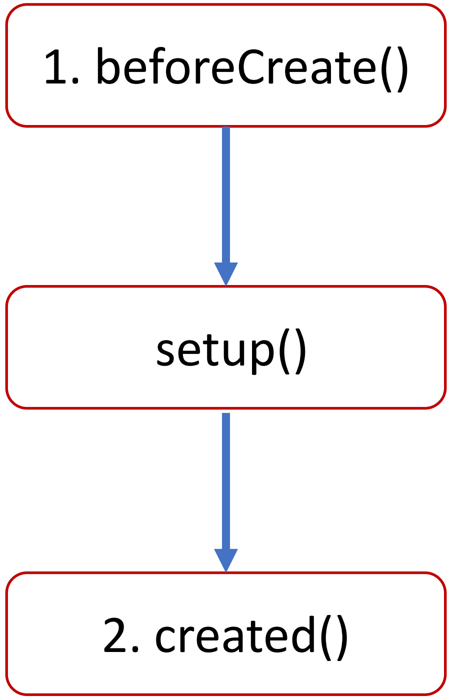
Options API	Hook inside `setup`
`beforeCreate`	Not needed*
`created`	Not needed*
`beforeMount`	`onBeforeMount`
`mounted`	`onMounted`
`beforeUpdate`	`onBeforeUpdate`
`updated`	`onUpdated`
`beforeUnmount`	`onBeforeUnmount`
`unmounted`	`onUnmounted`
`errorCaptured`	`onErrorCaptured`
`renderTracked`	`onRenderTracked`
`renderTriggered`	`onRenderTriggered`
`activated`	`onActivated`
`deactivated`	`onDeactivated`

These functions accept a callback that will be executed when the hook is called by the component

https://vuejs.org/guide/essentials/lifecycle.html#lifecycle-diagram

Composition API on Created Hooks

beforeCreate and created are not needed when using the composition API



1. beforeCreate: called immediately after a new Vue instance is initialized, before computed properties, watchers, events, data, etc., are not set up.

2. created: called after the instance has been created. Since instance is created we can set up data observation, computed properties, methods, watch/event callbacks. But we can not do DOM operation since the component is not mounted.

3. **onBeforeMount:** after template compiled, dynamic placeholders and interpolations are removed and replaced with the values. The template and the styles are all compiled here, but the DOM cannot be manipulated yet.
4. **onMounted:** called when component is mounted**,** show the HTML elements with defined content by Vue app on the screen
5. **onBeforeUpdate:** called when reactive data has changed, before the DOM is re-rendered.
6. **onUpdated:** called when reactive data has changed, and the DOM has been re-rendered. Mounted does not reached again in updated because template is not unmounted.
7. **onBeforeUnmount():** called right before a component instance is unmounted. At this stage the instance is still fully functional.
8. **onUnmounted():** called after a component instance has been unmounted. When this hook is called, all directives of the component instance have been unbound, all event listeners have been removed, and all child component instances have also been unmounted.

https://www.vuemastery.com/courses/vue-3-essentials/lifecycle-hooks/

Lifecycle Hooks Functions

```
function onBeforeMount(callback: () => void): void
```

```
function onMounted(callback: () => void): void
```

```
function onBeforeUpdate(callback: () => void): void
```

```
function onUpdated(callback: () => void): void
```

```
function onBeforeUnmount(callback: () => void): void
```

```
function onUnmounted(callback: () => void): void
```