

# Vue.js Lifecycle and Template Refs

Topics : [Vue](#)

Written on [January 11, 2024](#)

In Vue.js, the lifecycle of a component refers to the series of events that occur from the creation to destruction of the component. Each Vue component goes through various lifecycle hooks during its existence. Some of the key lifecycle hooks include:

1. **beforeCreate:** This hook is called synchronously right before the instance is created. At this point, data observation, events, and computed properties are not set up yet.
2. **created:** The created hook is called after the instance is created. At this point, the component has been initialized, and you have access to the reactive data and events.
3. **beforeMount:** This hook is called just before the component is about to be mounted to the DOM.
4. **mounted:** The mounted hook is called after the component has been mounted to the DOM. It is often used for initial setup that requires access to the DOM.
5. **beforeUpdate:** This hook is called when the data changes, but before the DOM is updated.
6. **updated:** The updated hook is called after a data change causes the component to re-render and the DOM is updated.
7. **beforeDestroy:** This hook is called just before a component is destroyed. It is used for cleanup tasks, such as removing event listeners or timers.
8. **destroyed:** The destroyed hook is called after a component is destroyed. It is used for cleanup that needs to happen after the component is removed.

Now, regarding template refs, they are used to get a reference to a DOM element or a child component in the template. You can use template refs with the `ref` attribute. Here's a simple example:

```
<template>
```

```
<div>
<input ref="myInput" v-model="inputValue" />
<button @click="focusInput">Focus Input</button>
</div>
</template>
```

```
<script>
export default {
  data() {
    return {
      inputValue: ""
    };
  },
  methods: {
    focusInput() {
      // Accessing the input element using the template ref
      this.$refs.myInput.focus();
    }
  }
};
</script>
```

In this example, the `ref="myInput"` attribute is added to the input element, creating a template ref named `myInput`. In the `focusInput` method, `this.$refs.myInput` is used to get a reference to the input element and then call the `focus` method on it.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)