

## **mixins**

<https://ua.vuejs.org/api/options-composition.html#mixins>  
<https://ua.vuejs.org/guide/reusability/composables.html>

**Mixins** - це спосіб додавання повторюваного коду до компонентів Vue. Об'єкт Mixin може містити будь-які параметри компонента. Коли компонент використовує Mixin, всі параметри в Mixin будуть об'єднані з параметрами компонента

## Приклад

```
const myMixin = {  
  data() {  
    return {  
      message: 'Hello from Mixin!'  
    }  
  }  
}
```

Властивість буде  
додано у розділ  
**data**

```
Vue.createApp({  
  mixins: [myMixin],  
  data() {  
    return {  
      name: 'Vue 3'  
    }  
  },  
  template: `  
    <div>  
      <h1>{{ message }}</h1>  
      <p>{{ name }}</p>  
    </div>  
  `,  
}).mount('#app')
```

Ще один приклад

```
// MyMixin.js
export const MyMixin = {
  data() {
    return {
      mixinData: 'Це дані з mixin!'
    };
  },
  methods: {
    sayHello() {
      console.log('Привіт з mixin!');
    }
  }
};
```

```
// MyComponent.vue
<template>
  <div>  <h1>{{ mixinData }}</h1>
  <button @click="sayHello">Натискайте мене</button>
  </div>
</template>
<script>
  import { MyMixin } from './MyMixin';
  export default {
    mixins: [MyMixin],
    data() {
      return {
        componentData: 'Це дані з компоненту!'
      }
    },
    methods: {
      greet() { console.log('Привіт з компоненту!') },
      mounted() {
        this.sayHello(); // Викликаємо метод із mixin
        this.greet();   // Викликаємо метод із компоненту
      }
    }
  }
</script>
```

Недоліки:

- **Незрозуміле джерело властивостей:** при використанні багатьох міксинів стає незрозуміло, яка властивість екземпляра впроваджується яким міксином, що ускладнює відстеження реалізації та розуміння поведінки компонента. Ось чому ми також рекомендуємо використовувати шаблон "референція + деструктуризація" для композиційних функцій: це робить джерело властивості зрозумілим у споживаючих компонентах.
- **Колізії просторів імен:** кілька міксинів від різних авторів потенційно можуть зареєструвати однакові ключі властивостей, спричиняючи колізії просторів імен. За допомогою композиційної функції ви можете перейменувати деструктуровані змінні, якщо є конфліктні ключі від різних композиційних функцій.
- **Неявний зв'язок крос-міксинів:** кілька міксинів, які повинні взаємодіяти один з одним, повинні покладатися на спільні ключі властивостей, що робить їх неявно зв'язаними. За допомогою композиційних функцій значення, повернуті однією композиційною функцією, можна передати в інший як аргументи, як і у звичайні функції.

Не рекомендовано використовувати міксини у Vue 3 !!!