

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu cách thức hoạt động của Reactive State trong Vue.js.
- ✓ Sử dụng các hàm `ref()` và `reactive()` để quản lý trạng thái.
- ✓ Hiểu rõ các loại Data Binding bao gồm: one-way binding, two-way binding, class binding, và style binding.

NỘI DUNG

PHẦN I:

Bài 1 (2 điểm): Quản lý trạng thái với `ref()`

Yêu cầu:

1. Tạo trạng thái reactive cho số lượng sản phẩm trong giỏ hàng.
2. Hiển thị số lượng sản phẩm và cung cấp hai nút để tăng hoặc giảm số lượng, không cho phép giảm dưới 0.

Hướng dẫn

Bước 1: Tạo 1 file mới trong thư mục **Labs/Lab-2** đặt tên **Bai1.vue**

```
<template>
<div class="container">
  <h1>Quản lý giỏ hàng với ref()</h1>
  <p>Số lượng sản phẩm: {{ productCount }}</p>
  <button @click="increaseCount">Tăng</button>
  <button @click="decreaseCount" :disabled="productCount === 0">Giảm</button>
</div>
</template>
<script setup>
  import { ref } from "vue";
  const productCount = ref(0);
  const increaseCount = () => {
    productCount.value++;
  };
  const decreaseCount = () => {
    if (productCount.value > 0) {
      productCount.value--;
    }
  };
</script>
```

Bước 2: Nhúng file **Bai1.vue** vào **App.vue**.

```
<script setup>
  import Bai1 from "../Labs/Lab-2/Bai1.vue";
</script>
<template>
  <Bai1 />
</template>
```

Bài 2 (2 điểm): Quản lý trạng thái với Reactive()

Yêu cầu:

1. Tạo trạng thái phức tạp cho thông tin người dùng (tên, tuổi, email) và cập nhật thông tin qua form.
2. Kiểm tra tính hợp lệ trước khi cập nhật dữ liệu.

Hướng dẫn

Bước 1: Tạo file mới trong thư mục **Labs/Lab-2** đặt tên **Bai2.vue**.

```
<template>
<div class="container">
<h1>Quản lý thông tin người dùng với reactive()</h1>
<form @submit.prevent="submitForm">
  <label>Tên: <input v-model="user.name" required /></label><br />
  <label>Tuổi: <input type="number" v-model="user.age" required /></label><br />
  <label>Email: <input v-model="user.email" required /></label><br />
  <button type="submit">Cập nhật</button>
</form>
<p>Tên: {{ user.name }}</p>
<p>Tuổi: {{ user.age }}</p>
<p>Email: {{ user.email }}</p>
</div>
</template>

<script setup>
import { reactive } from "vue";
const user = reactive({
  name: "",
  age: null,
  email: "",
});
const submitForm = () => {
  if (user.name && user.email) {
    alert("Thông tin hợp lệ");
  } else {
    alert("Vui lòng nhập đầy đủ thông tin");
  }
};
</script>
```

Bước 2: Nhúng file **Bai2.vue** vào **App.vue**.

PHẦN II:

Bài 3 (3 điểm) : One-way và Two-way Data Binding

Yêu cầu:

1. Tạo input sử dụng one-way binding và two-way binding.
2. Thay đổi giá trị thông qua two-way binding.

Hướng dẫn:

Bước 1: Tạo file mới trong thư mục **Labs/Lab-2** đặt tên **Bai3.vue**.

```
<template>
<div class="container">
  <h1>One-way và Two-way Binding</h1>
  <p>One-way binding: <input :value="message" readonly /></p>
  <p>Two-way binding: <input v-model="message" /></p>
  <p>Giá trị hiện tại: {{ message }}</p>
</div>
</template>
<script setup>
  import { ref } from "vue";
  const message = ref("Xin chào!");
</script>
```

Bước 2: Nhúng file **Bai4.vue** vào **App.vue**.

Bài 4 (3 điểm) : Class và Style Binding

Yêu cầu:

1. Thay đổi màu và kích thước văn bản dựa trên class và style binding.
2. Kiểm tra điều kiện cho kích thước văn bản (không cho phép kích thước nhỏ hơn 10px hoặc lớn hơn 50px).

Hướng dẫn:

Bước 1: Tạo file mới trong thư mục **Labs/Lab-2** đặt tên **Bai4.vue**.

```
<template>
<div class="container">
  <h1 :class="{ active: isActive }" :style="{ fontSize: fontSize + 'px' }">Văn bản có thể thay đổi</h1>
  <button @click="isActive = !isActive">Bật/Tắt màu chữ</button>
  <label>
    >Chọn kích thước chữ:
    <input type="number" v-model="fontSize" />
  </label>
  <p v-if="fontSize < 10 || fontSize > 50">Kích thước không hợp lệ!</p>
</div>
```

```
</template>
<script setup>
  import { ref } from "vue";
  const isActive = ref(false);
  const fontSize = ref(16);
</script>
<style scoped>
.active { color: red;}
</style>
```

Bài 5: Giảng viên cho thêm

**** Giảng viên có thể cho các bài tập phù hợp phục vụ assignment.**

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện

trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM

--- Hết ---