

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng thao tác với:

- ✓ Data binding và Reactivity trong VueJS
- ✓ Classes Binding (Liên kết các lớp)
- ✓ Binding inline styles (Ràng buộc kiểu nội tuyến)

NỘI DUNG

PHẦN I: Data binding và Reactivity trong VueJS

Bài 1 (2 điểm): Làm quen cơ bản với sự kiện và binding động

Tạo một nút bấm (button) và khi bấm vào nút, hiển thị một thông báo được cập nhật.

Yêu cầu:

- ✚ Tạo một biến message với giá trị mặc định.
- ✚ Khi bấm nút, giá trị của message sẽ được thay đổi thành một thông điệp mới.

Gợi ý tham khảo:

```
<template>
  <div class="p-5">
    <p>{{ message }}</p>
    <button class="btn btn-dark" @click="updateMessage">Thay đổi thông điệp</button>
  </div>
</template>

<script setup>
import { ref } from 'vue';

const message = ref('Thông điệp ban đầu');
const updateMessage = () => {
  message.value = 'Thông điệp đã được thay đổi!';
};
</script>
```

Bài 2 (2 điểm): Two-way Data Binding với v-model

Tạo một input nhập vào giá trị bất kỳ và hiển thị giá trị của trường nhập liệu đó ngay bên dưới theo thời gian thực.

Yêu cầu:

- ✚ Sử dụng v-model để binding giá trị từ input đến biến userInput.
- ✚ Hiển thị giá trị của biến userInput dưới thẻ <p>.

Gợi ý tham khảo:

```
<template>
  <div>
    <input v-model="userInput" placeholder="Nhập dữ liệu vào đây" />
    <p>Giá trị bạn nhập: {{ userInput }}</p>
  </div>
</template>

<script setup>
import { ref } from 'vue';

const userInput = ref('FPL');
</script>
```

Bài 3 (2 điểm): Nhập và hiển thị thông tin cá nhân

Tạo một form nhập vào các trường thông tin bao gồm: Họ và tên, Tuổi, Địa chỉ Email.

Yêu cầu:

- ✚ Sử dụng v-model để liên kết dữ liệu giữa các input fields và các biến reactive trong Vue.
- ✚ Hiển thị trực tiếp những thay đổi khi người dùng nhập thông tin vào form.

Thông tin cá nhân

Họ và Tên:

Nguyen Van A

Tuổi:

18

Email:

anv@example.com

Thông tin đã nhập:

Họ và Tên: Nguyen Van A

Tuổi: 18

Email: anv@example.com

Gợi ý tham khảo:

```
<template>
  <div class="col-sm-8">
    <h1>Thông tin cá nhân</h1>
    <form>
      <div class="mb-3">
        <label>Họ và Tên:</label>
        <input v-model="userInfo.name" type="text" class="form-control"
placeholder="Nhập họ và tên" />
      </div>
      <div class="mb-3">
        <label>Tuổi:</label>
        <input v-model="userInfo.age" type="number" class="form-control"
placeholder="Nhập tuổi" />
      </div>
      <div class="mb-3">
        <label>Email:</label>
        <input v-model="userInfo.email" type="email" class="form-control"
placeholder="Nhập email" />
      </div>
    </form>
  </div>
```

```

    <!-- Hiển thị thông tin ngay sau khi nhập -->
    <div class="info-display mt-4 p-3 bg-dark text-white rounded">
      <h2>Thông tin đã nhập:</h2>
      <p><strong>Họ và Tên:</strong> {{ userInfo.name }}</p>
      <p><strong>Tuổi:</strong> {{ userInfo.age }}</p>
      <p><strong>Email:</strong> {{ userInfo.email }}</p>
    </div>
  </div>
</template>





<script setup>
import { reactive } from 'vue';

// Khởi tạo đối tượng thông tin người dùng sử dụng reactive để tạo reactivity
const userInfo = reactive({
  name: '',
  age: null,
  email: ''
});
</script>

```

PHẦN II: Thao tác với Class và Style binding

Bài 4 (4 điểm): Xây dựng ứng dụng tạo và hiển thị danh sách bài viết gồm các yêu cầu sau:

-  Tạo bài đăng blog mới (tiêu đề, tác giả, nội dung).
-  Hiển thị danh sách các bài viết đã đăng.
-  Cho phép thay đổi màu sắc, kiểu dáng của mỗi bài viết thông qua việc sử dụng class binding và inline style binding.
-  Cấu trúc dự án
 - **App.vue**: Thành phần chính của ứng dụng.
 - **PostList.vue**: Thành phần hiển thị danh sách các bài viết.
 - **CreatePost.vue**: Thành phần cho phép tạo bài viết mới.

Hoặc sinh viên có thể tổ chức cấu trúc dự án như sau:

- **App.vue**: Thành phần chính của ứng dụng.
- **Post.vue**: Thành phần tạo và hiển thị danh sách các bài viết.

Tạo bài viết mới

Danh sách bài viết

Bật Mí 8 Nguyên Tắc Phối Màu Quần Áo Sành Điệu

Nguyễn Văn Tèo

Bạn có thể phối áo màu xanh lá, quần màu kem và giày màu tím theo cách phối màu kiểu tam giác. Hoặc có thể phối các set đồ cùng ton lạnh hoặc nóng ...

Hướng dẫn:

1. Tạo mã code tại [CreatePost.vue](#). (Phần hướng dẫn này khuyến khích sinh viên tìm hiểu về **emit** trong Vuejs)

```
<template>
  <div>
    <h2>Tạo bài viết mới</h2>
    <input v-model="title" placeholder="Tiêu đề bài viết" />
    <input v-model="author" placeholder="Tên tác giả" />
    <textarea v-model="content" placeholder="Nội dung bài viết"></textarea>
    <button @click="submitPost">Đăng bài</button>
  </div>
</template>
```

```
<script setup>
import { ref } from 'vue'

// Sử dụng defineEmits để emit sự kiện
const emit = defineEmits(['add-post'])

const title = ref('')
const content = ref('')
const author = ref('')

function submitPost() {
  if (title.value && content.value && author.value) {
    const newPost = {
      title: title.value,
      content: content.value,
      author: author.value
    }

    // Gửi sự kiện 'add-post' với dữ liệu của bài viết
    emit('add-post', newPost)

    // Reset fields
    title.value = ''
    content.value = ''
    author.value = ''
  }
}
</script>
```

```
<style>
input,
textarea {
  display: block;
  width: 100%;
  margin: 10px 0;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #ddd;
}

button {
  padding: 10px 20px;
  background-color: #3498db;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
</style>
```

2. Tạo mã code tại **PostList.vue**

Dưới đây có sử dụng **v-for** để duyệt qua danh sách bài viết post. Sinh viên có thể tìm hiểu cơ bản trước về **v-for**

Thay đổi màu sắc, kiểu dáng của mỗi bài viết thông qua việc sử dụng class binding và inline style binding.

```
<template>
  <div>
    <h2>Danh sách bài viết</h2>
    <div v-for="(post, index) in posts" :key="index" :class="{ 'highlighted': post.title.length > 20 }"
      :style="{ backgroundColor: post.backgroundColor, color: post.textColor }" class="post">
      <h3 class="text-success">{{ post.title }}</h3>
      <h4 class="text-danger">{{ post.author }}</h4>
      <p>{{ post.content }}</p>
    </div>
  </div>
</template>

<script setup>
import { defineProps } from 'vue';
// định nghĩa props
defineProps({
  posts: {
    type: Array,
    required: true,
  }
});
</script>

<style>
.post {
  padding: 20px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 10px;
}

.highlighted {
  border: 2px solid lightgray;
}
</style>
```

3. Khai báo tại **App.vue**

```

<template>
  <div id="app">
    <h1>Ứng dụng Blog nhỏ với Vue.js</h1>
    <CreatePost @add-post="addPost" />
    <PostList :posts="posts" />
  </div>
</template>

<script setup>

import { ref } from 'vue'
import CreatePost from './components/CreatePost.vue'
import PostList from './components/PostList.vue'

const posts = ref([])

function addPost(post) {
  posts.value.push(post)
}
</script>

<style>
#app {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
}
</style>

```

**** Giảng viên có thể cho thêm 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 ---