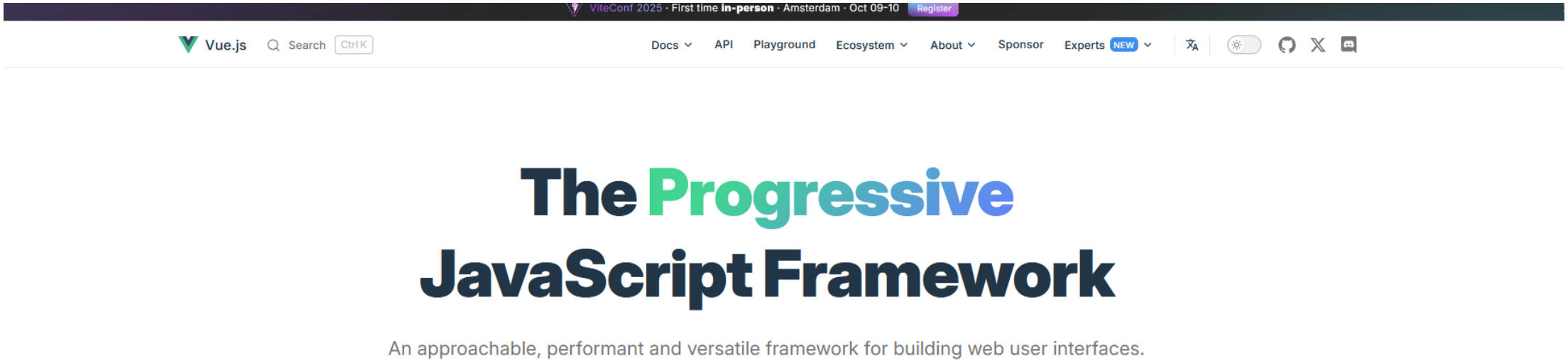
VUE.js



Irene Ioannidou

VUE.js

What is it?

Progressive Component-based JavaScript Framework for UI

VUE.js

Component-based

Apps are made up of reusable pieces (components).

VUE.js

Reactive data binding

UI updates automatically when data changes.

VUE.js

Progressive

You can use it just for parts of a web page, or for full apps.

VUE.js

Ecosystem

Works well with Vue Router (navigation), Pinia/Vuex (state management), and Vue CLI / Vite (project setup).

VUE.js

How to run?

1. Directly in the Browser (Simple Way)

2. With a Build Tool (Node.js)

VUE.js

How to run?

1. Directly in the Browser (Simple Way)

```
<script src="https://cdn.jsdelivr.net/npm/vue@3"></script>
```

VUE.js

How to run?

2. With a Build Tool
(Node.js)

npm create vite@latest my-vue-app -- --template vue

cd my-vue-app

npm install

VUE.js

How it runs

1.const { createApp } = Vue;  const createApp = Vue.createApp;

When you load Vue from a <script> tag (CDN), a **global Vue object** becomes available.

Destructuring: pulls out the createApp function from the Vue object, so you can use it directly.

2. createApp ->makes a new Vue application instance.

.mount("#app") -> tells Vue “take control of the element with id=app and render stuff there.”

VUE.js

How it runs

3. In Vue, when you declare variables inside data() (or ref() in composition API), Vue makes them reactive.

```
data() {
  return {
    color: ""   // <-- reactive variable
  }
}
```

4. When color changes → Vue automatically updates the DOM where it's used.

***** No need for **document.querySelector** or **innerText** manually (like in VanillaJS).

VUE.js

Example

```
<div id="app">
  <h1>Your favorite color: {{ color || "unknown" }}</h1>
  <input v-model="color" placeholder="Type your favorite color" />
</div>

<script>
const { createApp } = Vue;

createApp({
  data() {
    return {
      color: ""   // variable "color"
    }
  }
}).mount("#app");
</script>
```

color is the **reactive variable** inside Vue's data.

|| is fallback text if the variable is empty

Binding (v-model="color") **links the input** box to **color.**

Interpolation ({{ color || "unknown" }}) **displays** the **value** of color **in** the **DOM.**

VUE.js

Summary

```
<h1>Your favorite color: {{ color || "unknown" }}</h1>
<input v-model="color" placeholder="Type your favorite color" />
```

- **Reactive variable** color lives in Vue's data.
- **Binding** (v-model="color") links the input box to color.
- **Interpolation** ({{ color || "unknown" }}) displays the value of color in the DOM.
- When the user types → color updates → interpolation re-renders → page updates.

VUE.js

Summary

Reactive variable: special variable Vue watches (like color).

Interpolation ({{ }}): shows the variable in the page.

Binding (v-model): keeps inputs and variables in sync.

VUE.js

EXTRA

```
<p v-if="color === 'blue'">
Blue is the color of the sky! ☁
</p>
```

VUE.js

EXTRA

== (loose equality)

Compares values after converting types if needed.

"5" == 5 → ✅ true (string "5" is converted to number).

0 == false → ✅ true (number 0 is treated as false).

=== (strict equality)

Compares value AND type.

"5" === 5 → ❌ false (string is not the same type as number).

0 === false → ❌ false (different types).

"blue" === "blue" → ✅ true (same type and value).