

Core Features

1. Declarative Rendering -> HTML Template based JS State
2. Reactivity -> Vue tracks JS State und updated DOM

Usage

- Enhancing static HTML without a build step
- Embedding as Web Components on any page
- Single-Page Application (SPA)
- Fullstack / Server-Side-Rendering (SSR)
- Jamstack / Static-Site-Generation (SSG)
- Targeting desktop, mobile, WebGL or even the terminal

Single File Components

```
<script>
export default {
  data() {
    return {
      count: 0
    }
  }
}
</script>

<template>
  <button @click="count++">Count is: {{ count }}</button>
</template>

<style scoped>
button {
  font-weight: bold;
}
</style>
```

Options API

```
<script>
export default {
  // Properties returned from data() becomes reactive state
  // and will be exposed on `this`.
  data() {
    return {
      count: 0
    }
  },

  // Methods are functions that mutate state and trigger updates.
  // They can be bound as event listeners in templates.
  methods: {
    increment() {
      this.count++
    }
  },

  // Lifecycle hooks are called at different stages
  // of a component's lifecycle.
  // This function will be called when the component is mounted.
  mounted() {
    console.log(`The initial count is ${this.count}.`)
  }
}
</script>

<template>
  <button @click="increment">Count is: {{ count }}</button>
</template>
```

There functions to declare the logic of the component like **data, methods, mounted**

Composition API

With **script setup** Vue enables direct use of Top-Level Data in template

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

// reactive state
const count = ref(0)

// functions that mutate state and trigger updates
function increment() {
  count.value++
}

// lifecycle hooks
onMounted(() => {
  console.log(`The initial count is ${count.value}.`)
})
</script>

<template>
  <button @click="increment">Count is: {{ count }}</button>
</template>
```

Für Calatena empfehle ich die Composition API da sie mächtiger ist und effzienter und strukturierte in Anwendung

npm init vue@latest -> Initialisiert das neue Projekt

npm install -> installiert alle Dependencies

npm run dev

npm run build -> für Production