Vue.js

Team Members

Diego Antonio Pi Arteaga

diego.antonio.pi.19@ull.edu.es

José Ángel Portillo García

jose.portillo.11@ull.edu.es

Álvaro Pérez Ramos

alvaro.perez.ramos.23@ull.edu.es

INDEX

- ❑ How Popular is Vue Today?
- ❑ What is Vue?
- ❑ Initial Config
- ❑ Vue Usage Diagram
  - ❑ Rendering
  - ❑ Components
  - ❑ Routing
  - ❑ State Management
  - ❑ Build System
- ❑ APIs
- ❑ Vue directives
- ❑ Figuras
- ❑ Lissajous curves

How Popular is Vue Today?

One of the top frameworks alongside React and Angular

Used by Alibaba, Xiaomi, GitLab, Behance, etc

Large community + wide support

205k+ GitHub stars

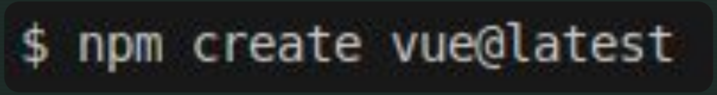
What is VUE?

Developed by **Evan You**, open source

Ideal for building **Single Page Applications**

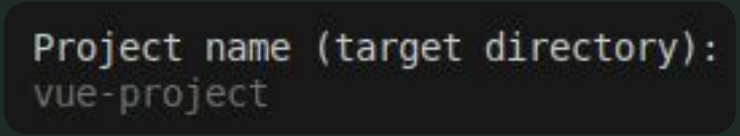
Initial configuration

1. Initialize a new Project:

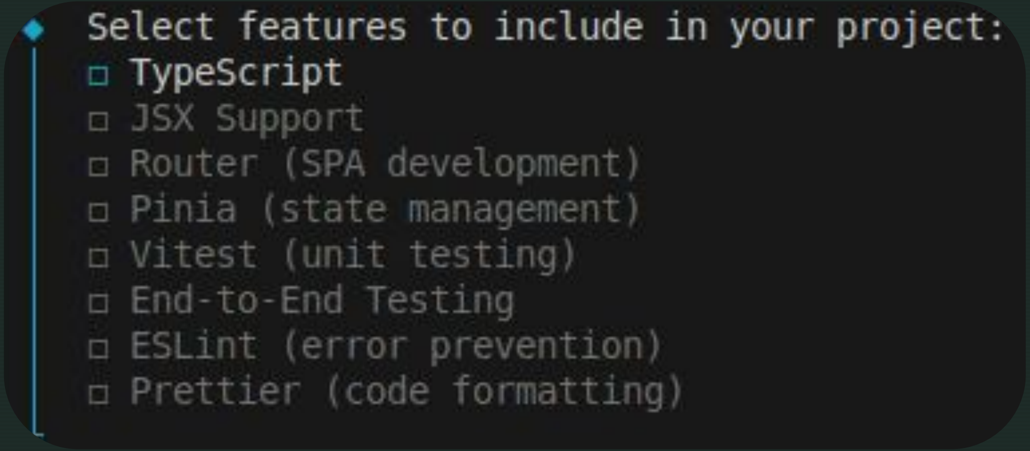


```
$ npm create vue@latest
```

2. Choose a Project Name
   and select the features you want:

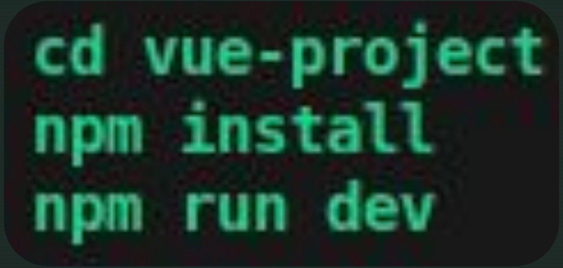


```
Project name (target directory):
vue-project
```



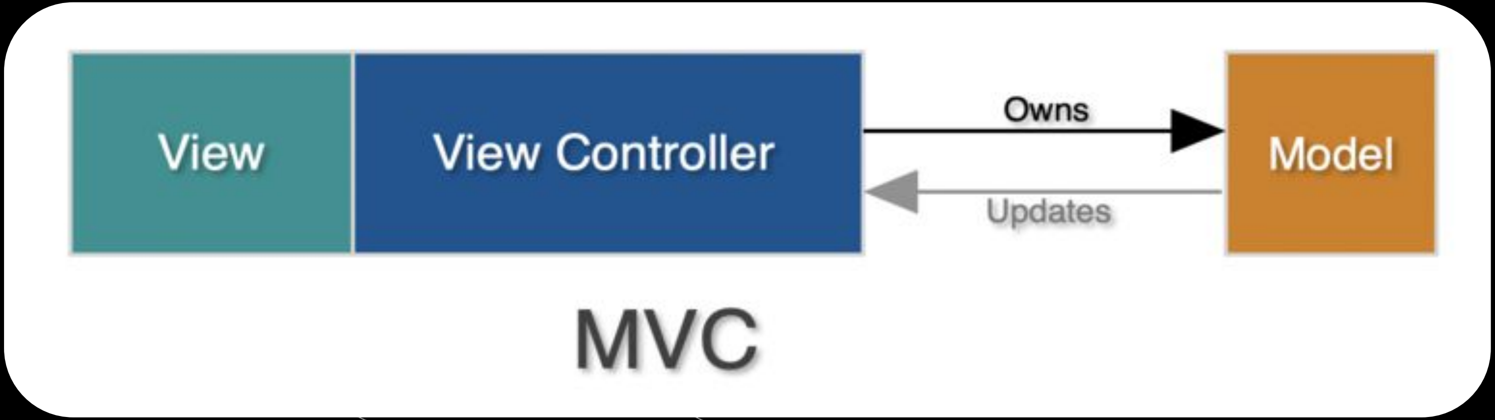
```
◆  Select features to include in your project:
│  □ TypeScript
│  □ JSX Support
│  □ Router (SPA development)
│  □ Pinia (state management)
│  □ Vitest (unit testing)
│  □ End-to-End Testing
│  □ ESLint (error prevention)
│  □ Prettier (code formatting)
└
```

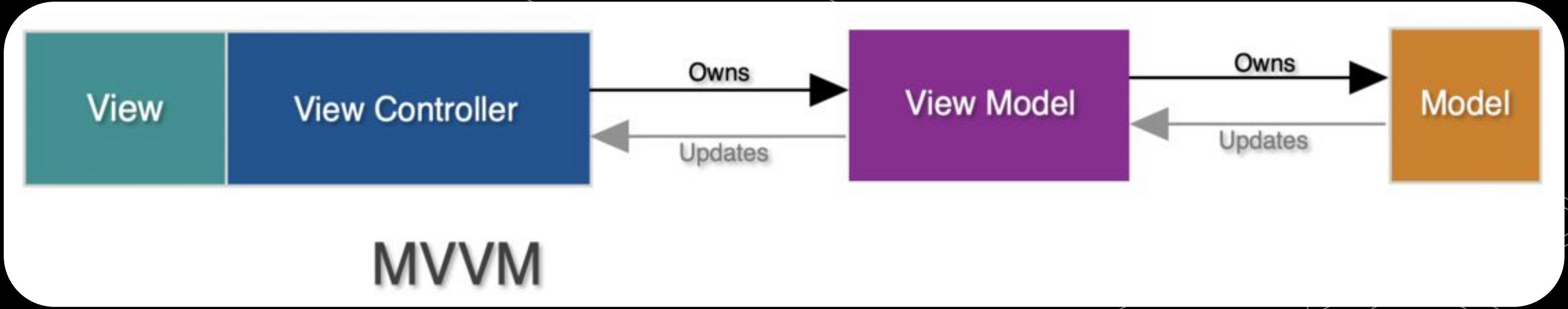
3. Lastly to try the default program use:



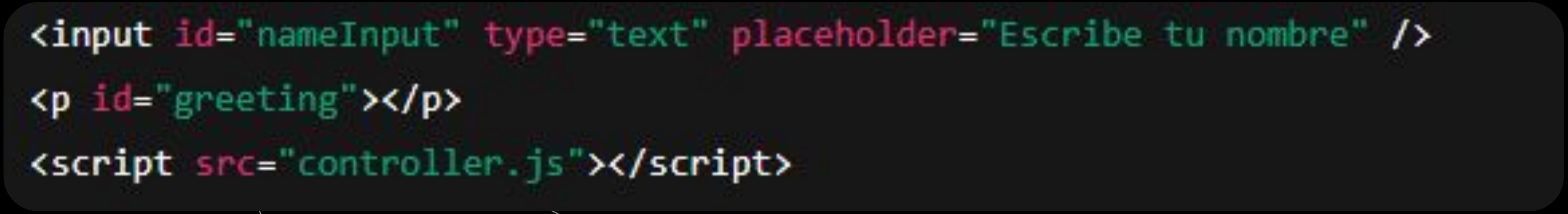
```
cd vue-project
npm install
npm run dev
```

Pattern Comparison

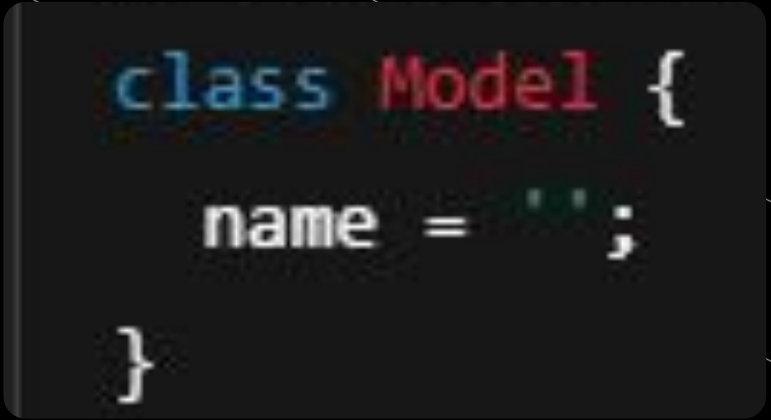




Pattern Comparison (MVC)

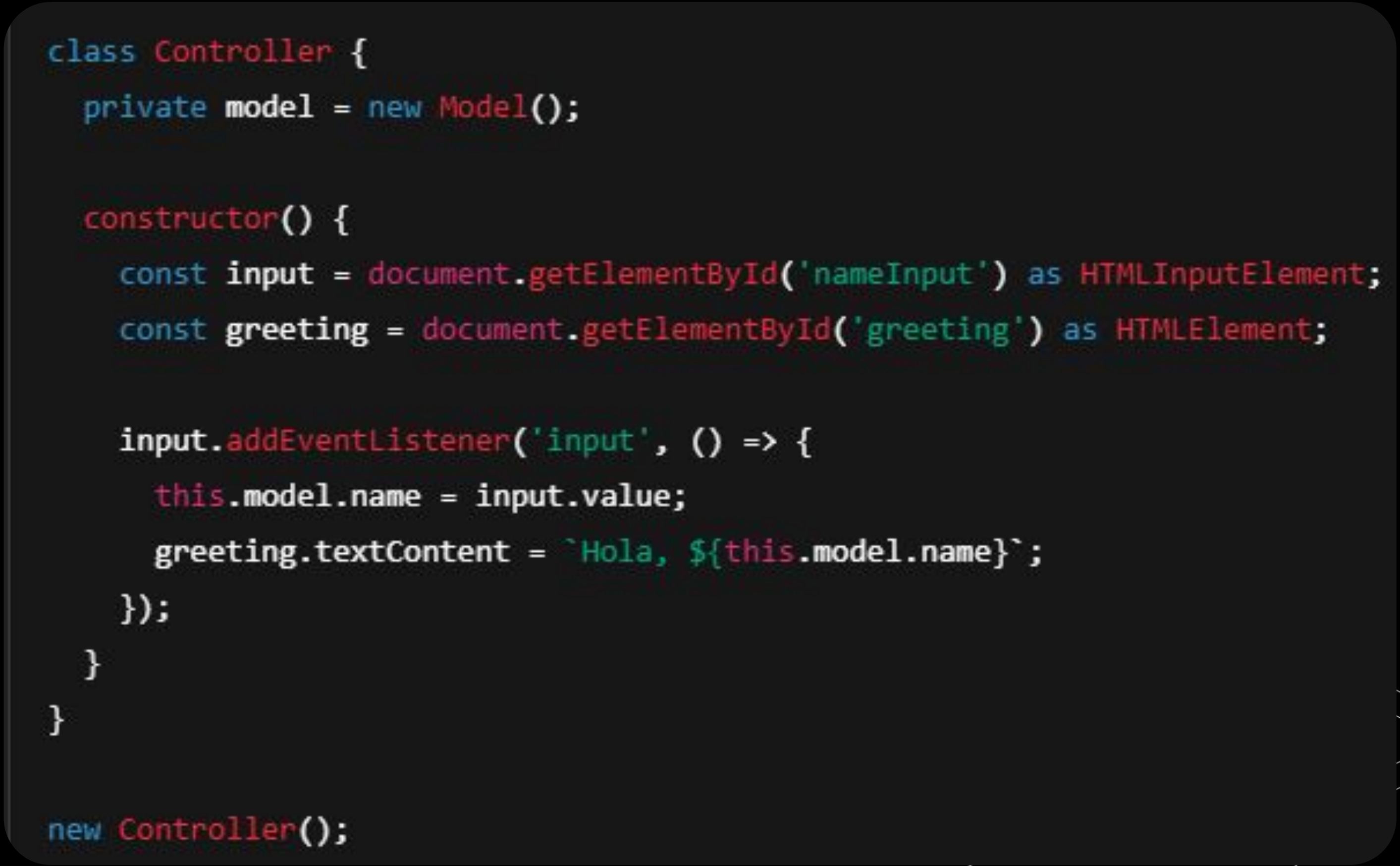


```html
<input id="nameInput" type="text" placeholder="Escribe tu nombre" />
<p id="greeting"></p>
<script src="controller.js"></script>
```



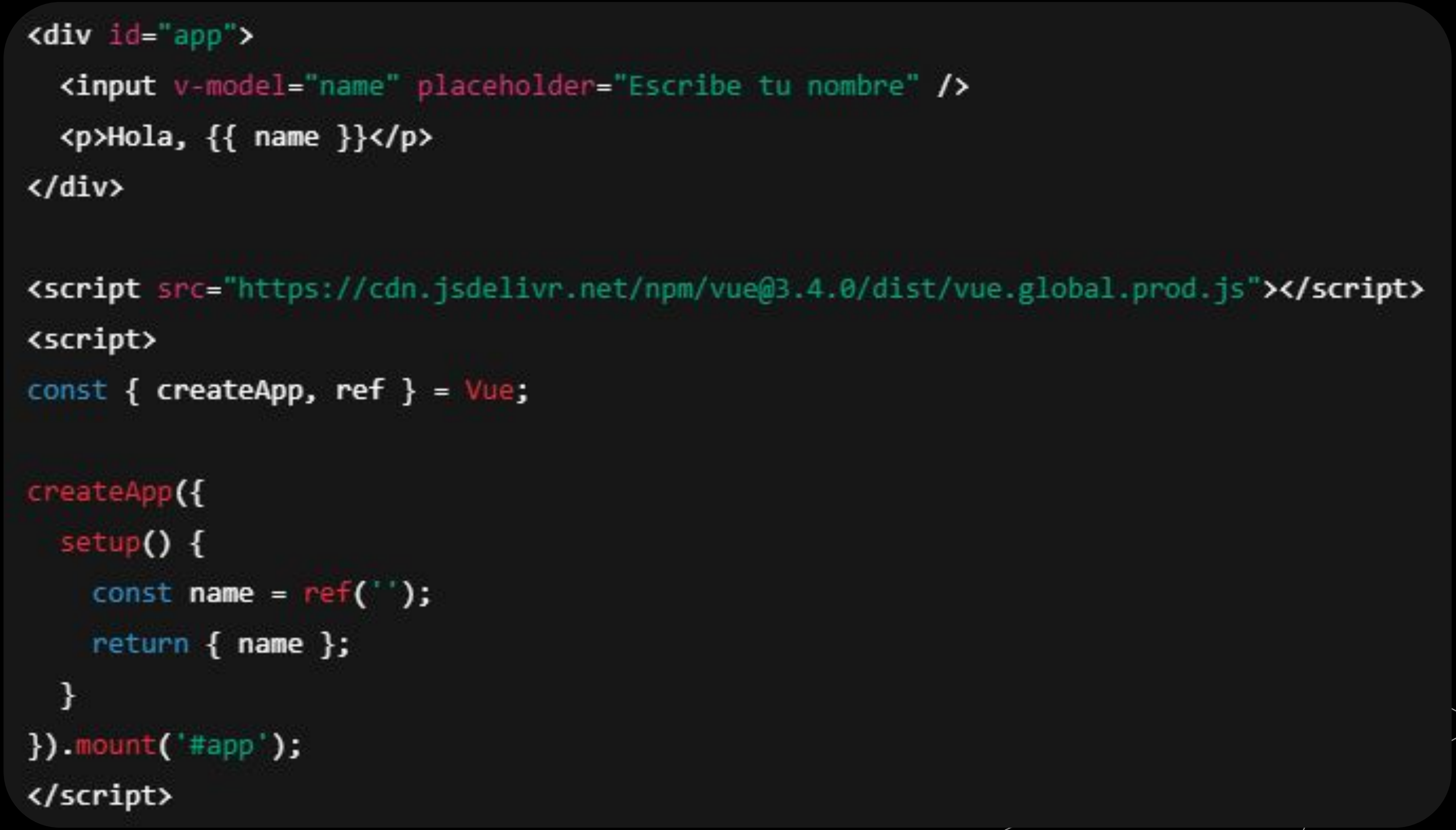
```js
class Model {
  name = '';
}
```

Pattern Comparison (MVC)



```
class Controller {
  private model = new Model();

  constructor() {
    const input = document.getElementById('nameInput') as HTMLInputElement;
    const greeting = document.getElementById('greeting') as HTMLElement;

    input.addEventListener('input', () => {
      this.model.name = input.value;
      greeting.textContent = `Hola, ${this.model.name}`;
    });
  }
}

new Controller();
```

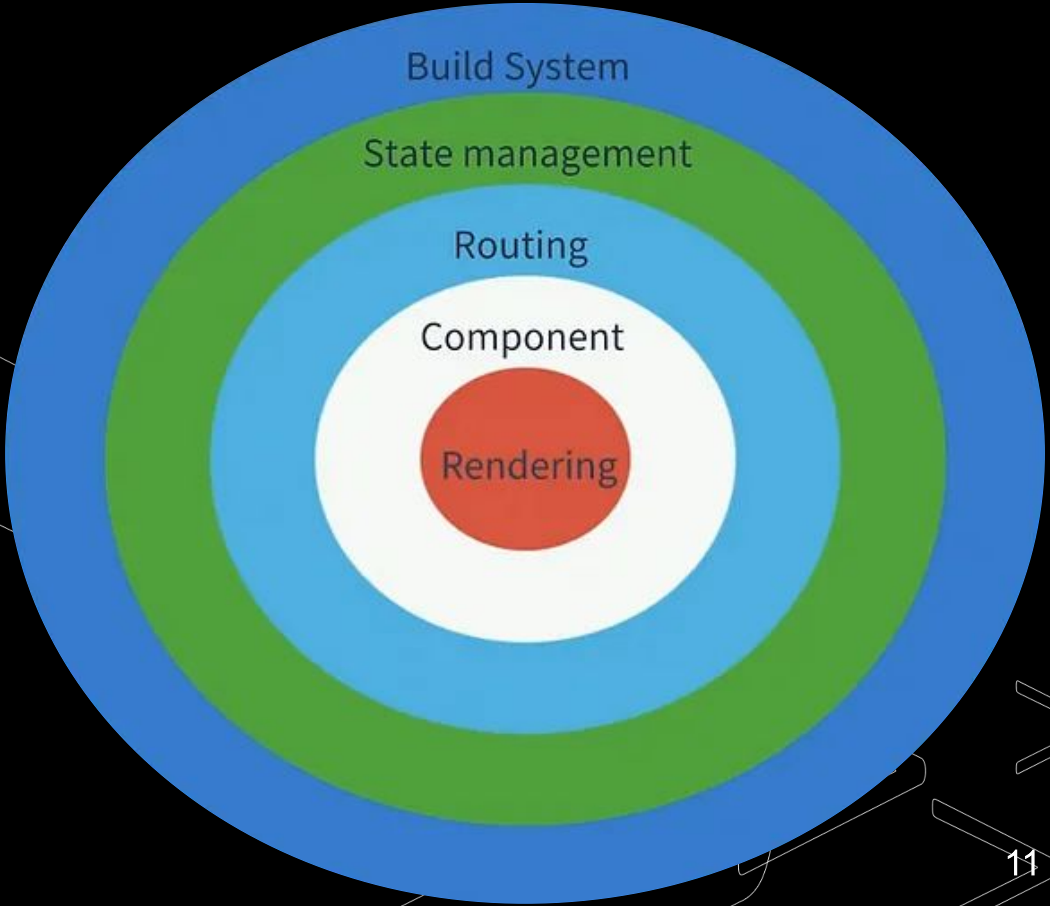
Pattern Comparison (MVVM)



```html
<div id="app">
  <input v-model="name" placeholder="Escribe tu nombre" />
  <p>Hola, {{ name }}</p>
</div>


<script src="https://cdn.jsdelivr.net/npm/vue@3.4.0/dist/vue.global.prod.js"></script>
<script>
const { createApp, ref } = Vue;

createApp({
  setup() {
    const name = ref('');
    return { name };
  }
}).mount('#app');
</script>
```

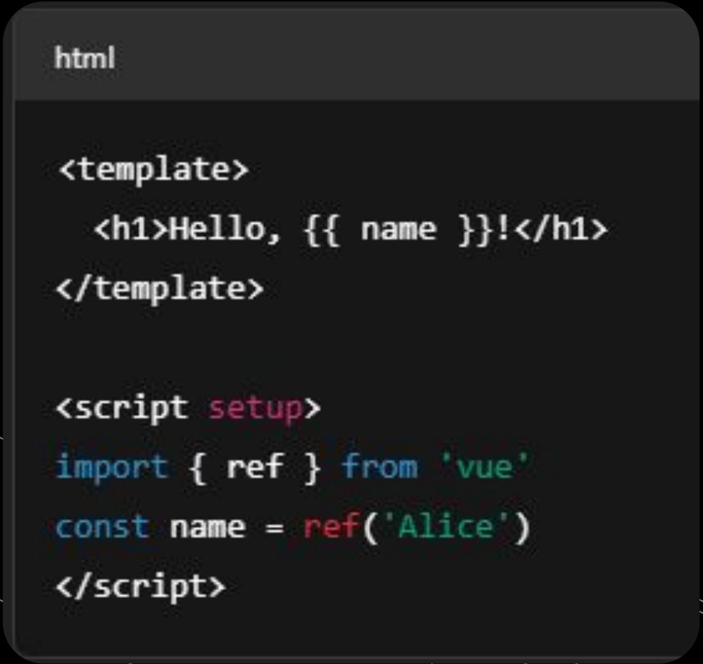
Vue Usage Diagram

A **progressive JavaScript framework** for building user interfaces



Rendering

Rendering in Vue.js refers to how the framework converts your data and components into DOM elements shown in the browser



```html
<template>
  <h1>Hello, {{ name }}!</h1>
</template>

<script setup>
import { ref } from 'vue'
const name = ref('Alice')
</script>
```

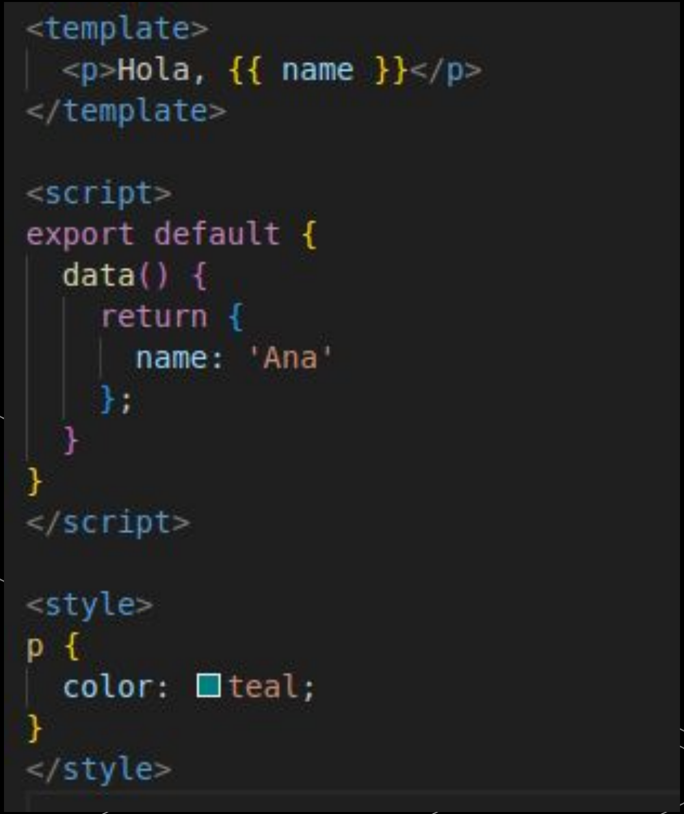
Components

Reusable piece of code containing:

Template: Contains the HTML to view.

Script: Contains data and the logic of the program.

Style: Local styles for the component.



```
<template>
  <p>Hola, {{ name }}</p>
</template>

<script>
export default {
  data() {
    return {
      name: 'Ana'
    };
  }
}
</script>

<style>
p {
  color: teal;
}
</style>
```

Vue directives

¿What are they?

Special attributes in HTML (v-attribute) and provide reactive functionality to the DOM.

What are they for?

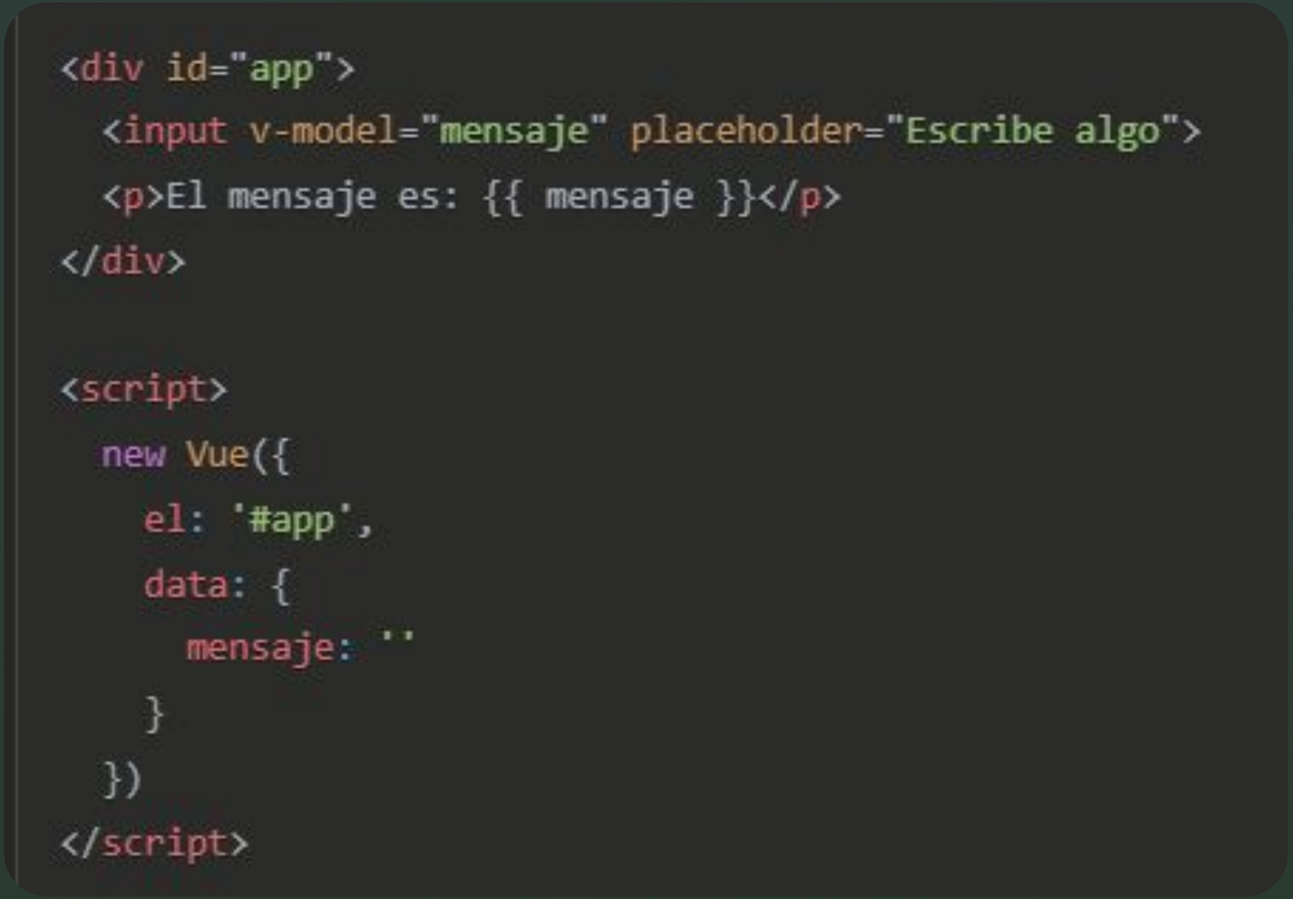
- They connect HTML to application data.
- They automatically update the view when data changes.
- They simplify common programming tasks.

Advantages

- Cleaner, more declarative code.
- Less manual manipulation of the DOM.
- Easier development.
- Better maintenance.

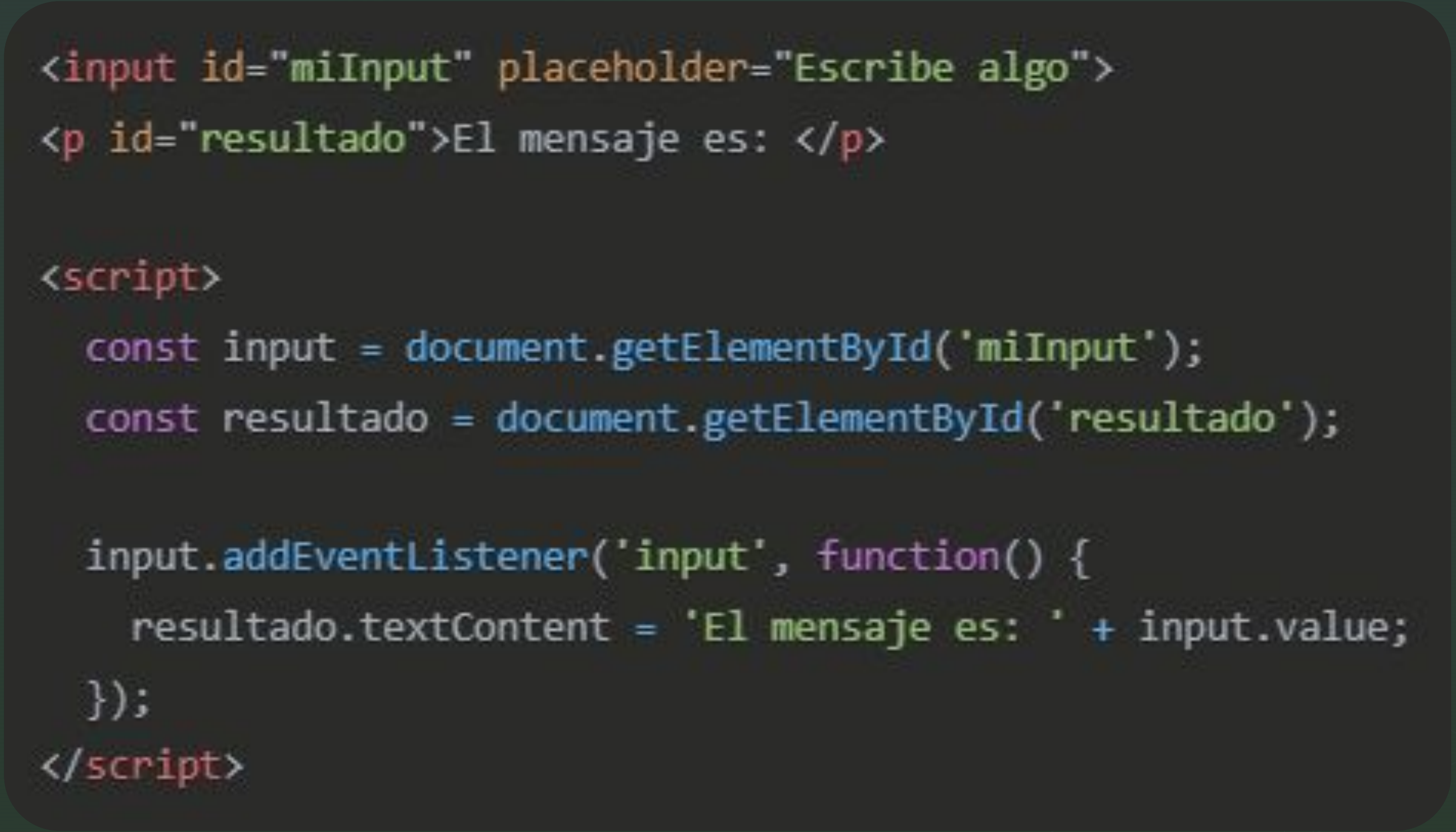
Built-in Directives | Vue.js
Custom Directives | Vue.js

v-model (Bidirectional link of data) (Vue)



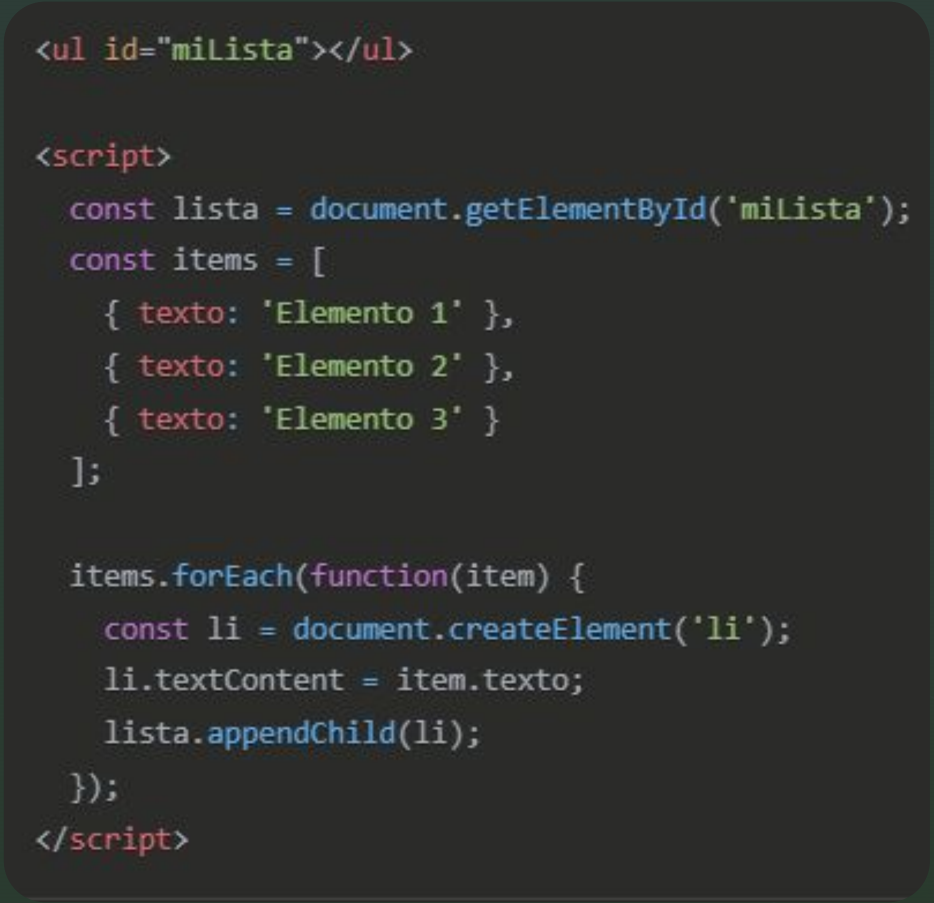
```
<div id="app">
  <input v-model="mensaje" placeholder="Escribe algo">
  <p>El mensaje es: {{ mensaje }}</p>
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      mensaje: ''
    }
  })
</script>
```

v-model (Bidirectional link of data) (No-Vue)



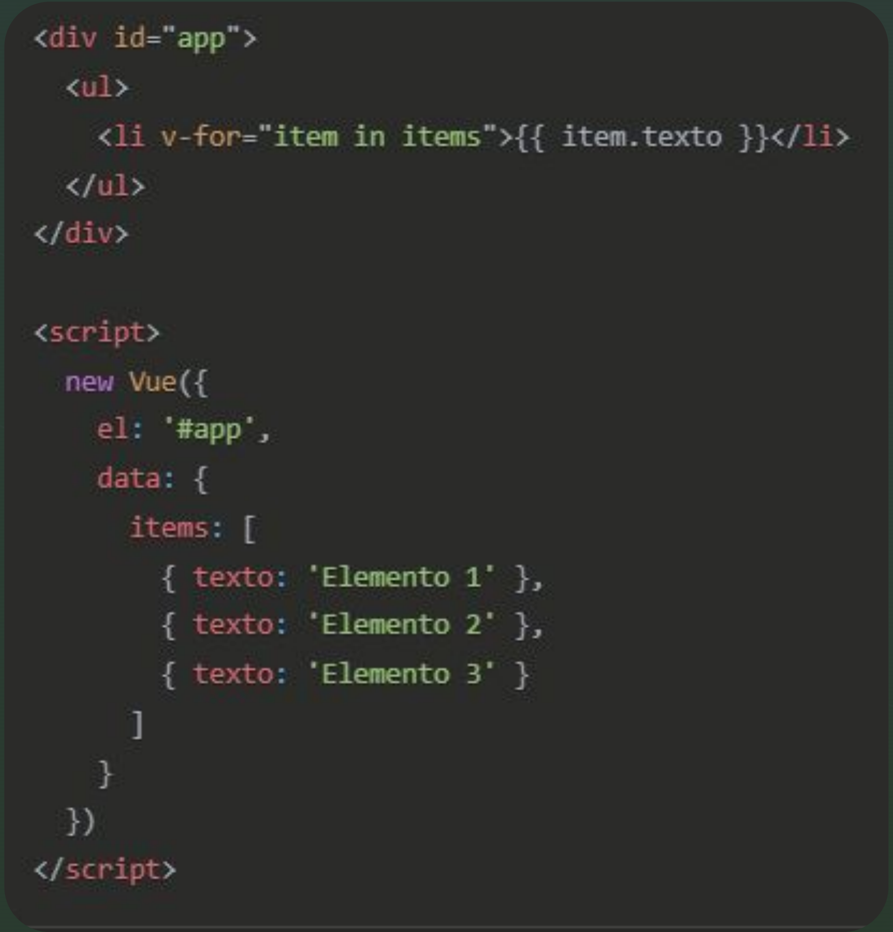
```
<input id="miInput" placeholder="Escribe algo">
<p id="resultado">El mensaje es: </p>


<script>
  const input = document.getElementById('miInput');
  const resultado = document.getElementById('resultado');

  input.addEventListener('input', function() {
    resultado.textContent = 'El mensaje es: ' + input.value;
  });
</script>
```

v-for (Lists render) (No-Vue)



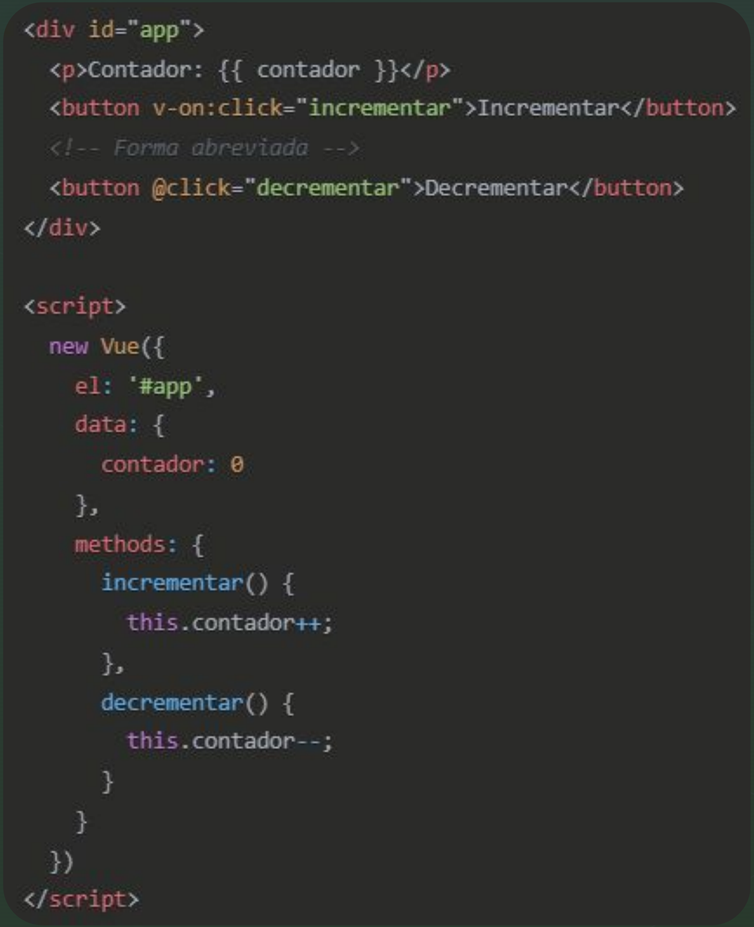
```
<ul id="miLista"></ul>

<script>
  const lista = document.getElementById('miLista');
  const items = [
    { texto: 'Elemento 1' },
    { texto: 'Elemento 2' },
    { texto: 'Elemento 3' }
  ];

  items.forEach(function(item) {
    const li = document.createElement('li');
    li.textContent = item.texto;
    lista.appendChild(li);
  });
</script>
```

v-for (Lists render) (Vue)



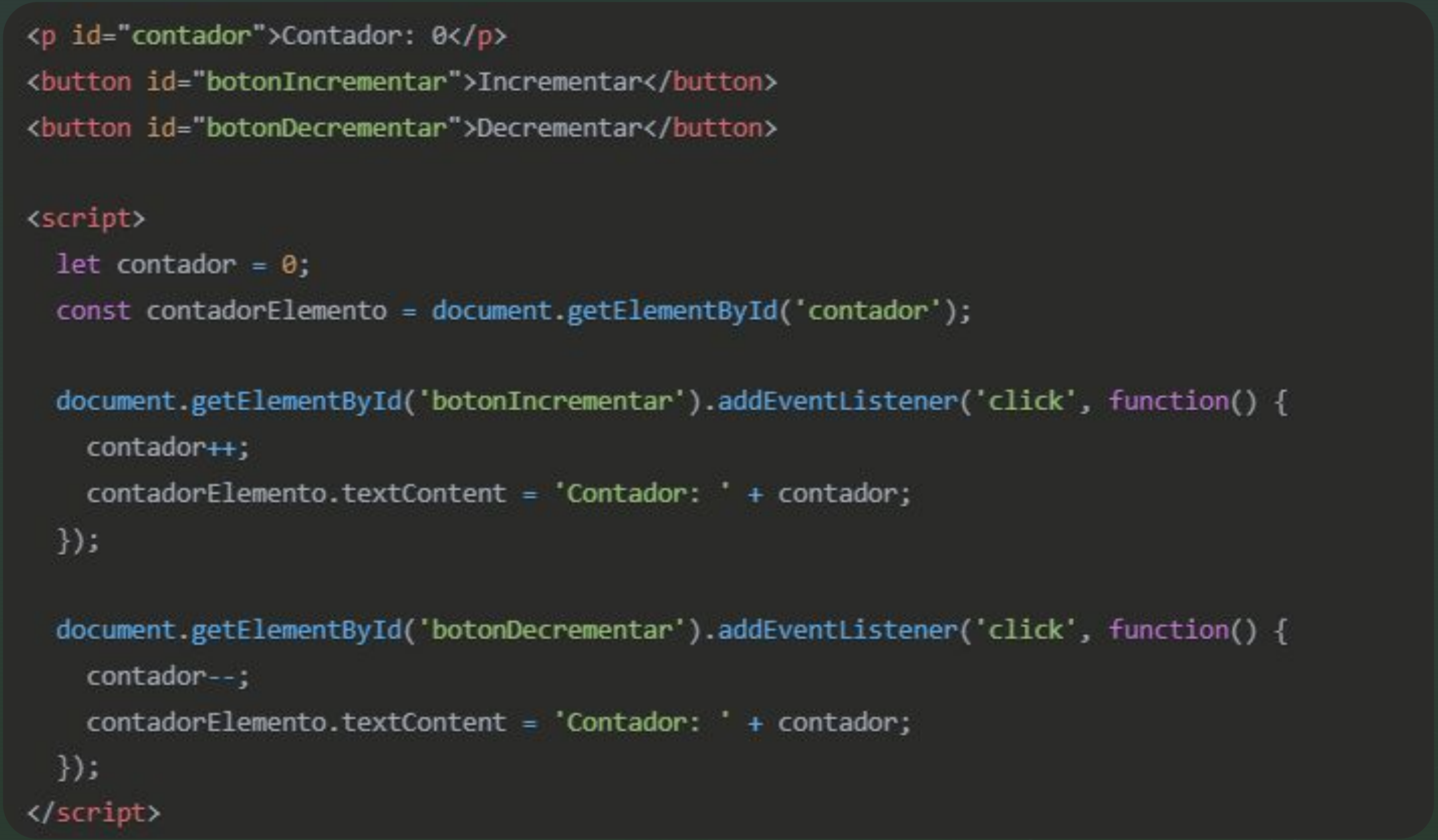
```
<div id="app">
  <ul>
    <li v-for="item in items">{{ item.texto }}</li>
  </ul>
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      items: [
        { texto: 'Elemento 1' },
        { texto: 'Elemento 2' },
        { texto: 'Elemento 3' }
      ]
    }
  })
</script>
```

v-on (Events Management) (Vue)



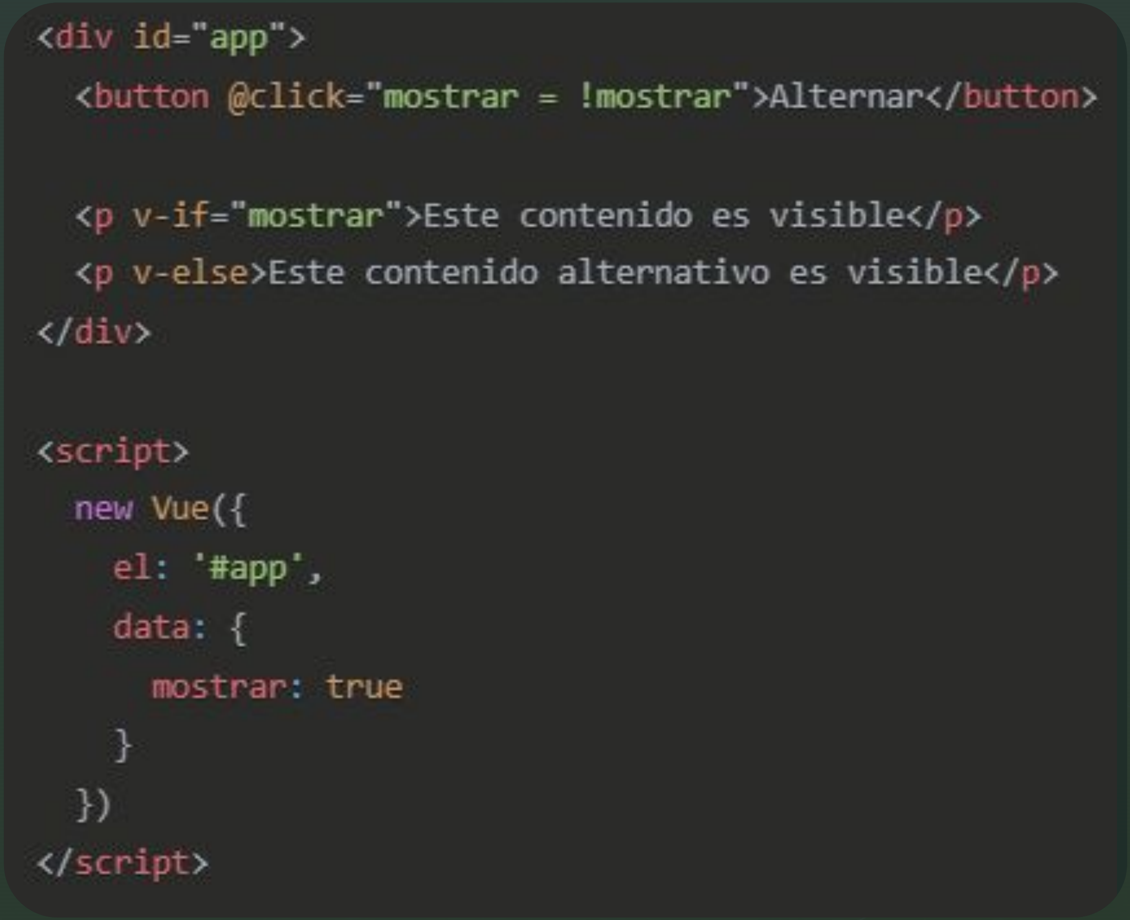
```
<div id="app">
  <p>Contador: {{ contador }}</p>
  <button v-on:click="incrementar">Incrementar</button>
  <!-- Forma abreviada -->
  <button @click="decrementar">Decrementar</button>
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      contador: 0
    },
    methods: {
      incrementar() {
        this.contador++;
      },
      decrementar() {
        this.contador--;
      }
    }
  })
</script>
```

v-on (Events Management) (No-Vue)



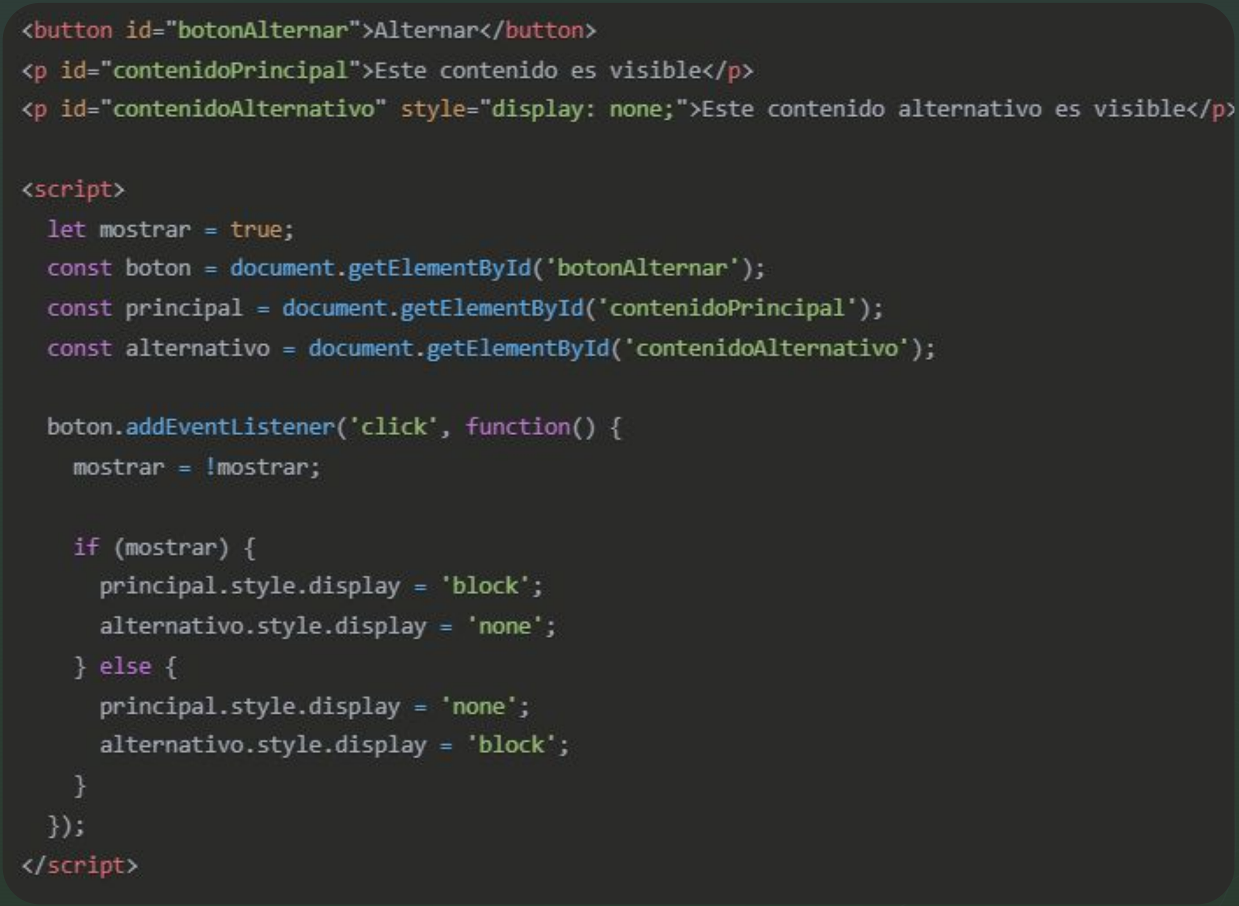
```
<p id="contador">Contador: 0</p>
<button id="botonIncrementar">Incrementar</button>
<button id="botonDecrementar">Decrementar</button>

<script>
  let contador = 0;
  const contadorElemento = document.getElementById('contador');

  document.getElementById('botonIncrementar').addEventListener('click', function() {
    contador++;
    contadorElemento.textContent = 'Contador: ' + contador;
  });

  document.getElementById('botonDecrementar').addEventListener('click', function() {
    contador--;
    contadorElemento.textContent = 'Contador: ' + contador;
  });
</script>
```

v-if / v-else (condicional render) (Vue)



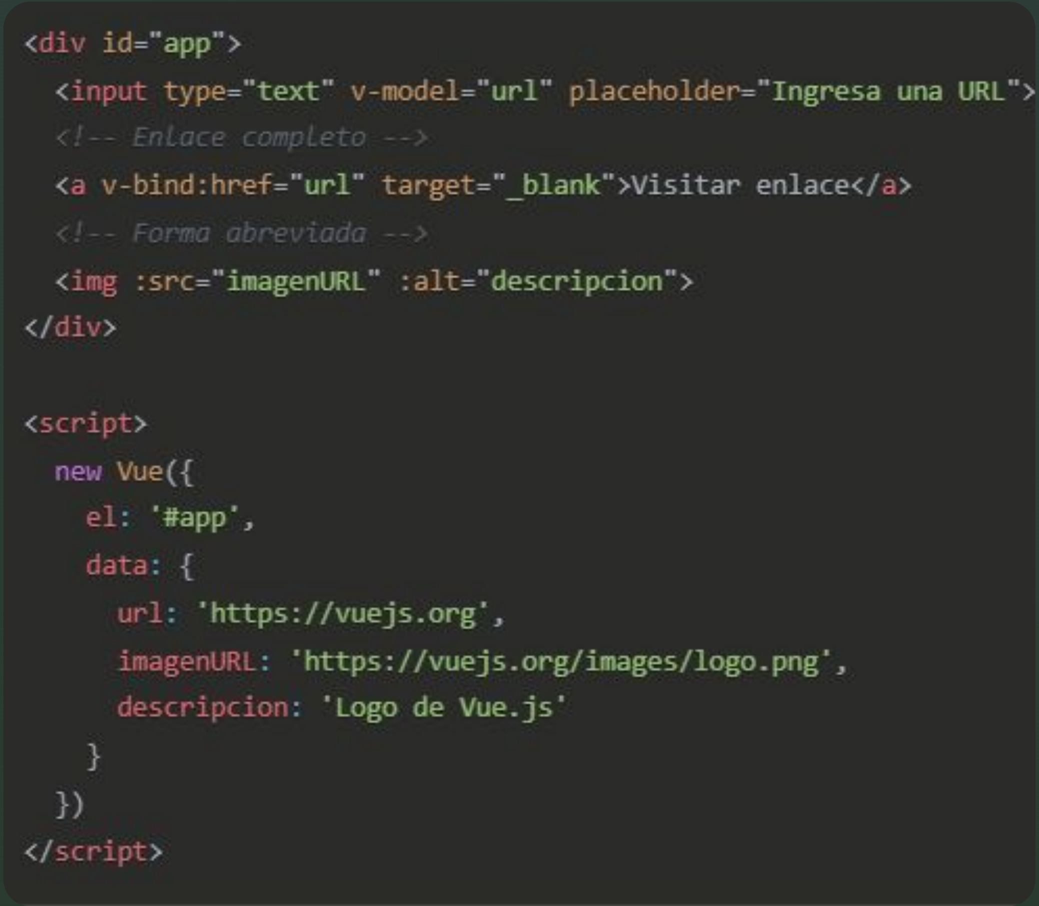
```
<div id="app">
  <button @click="mostrar = !mostrar">Alternar</button>

  <p v-if="mostrar">Este contenido es visible</p>
  <p v-else>Este contenido alternativo es visible</p>
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      mostrar: true
    }
  })
</script>
```

v-if / v-else (Conditional render) (No-Vue)



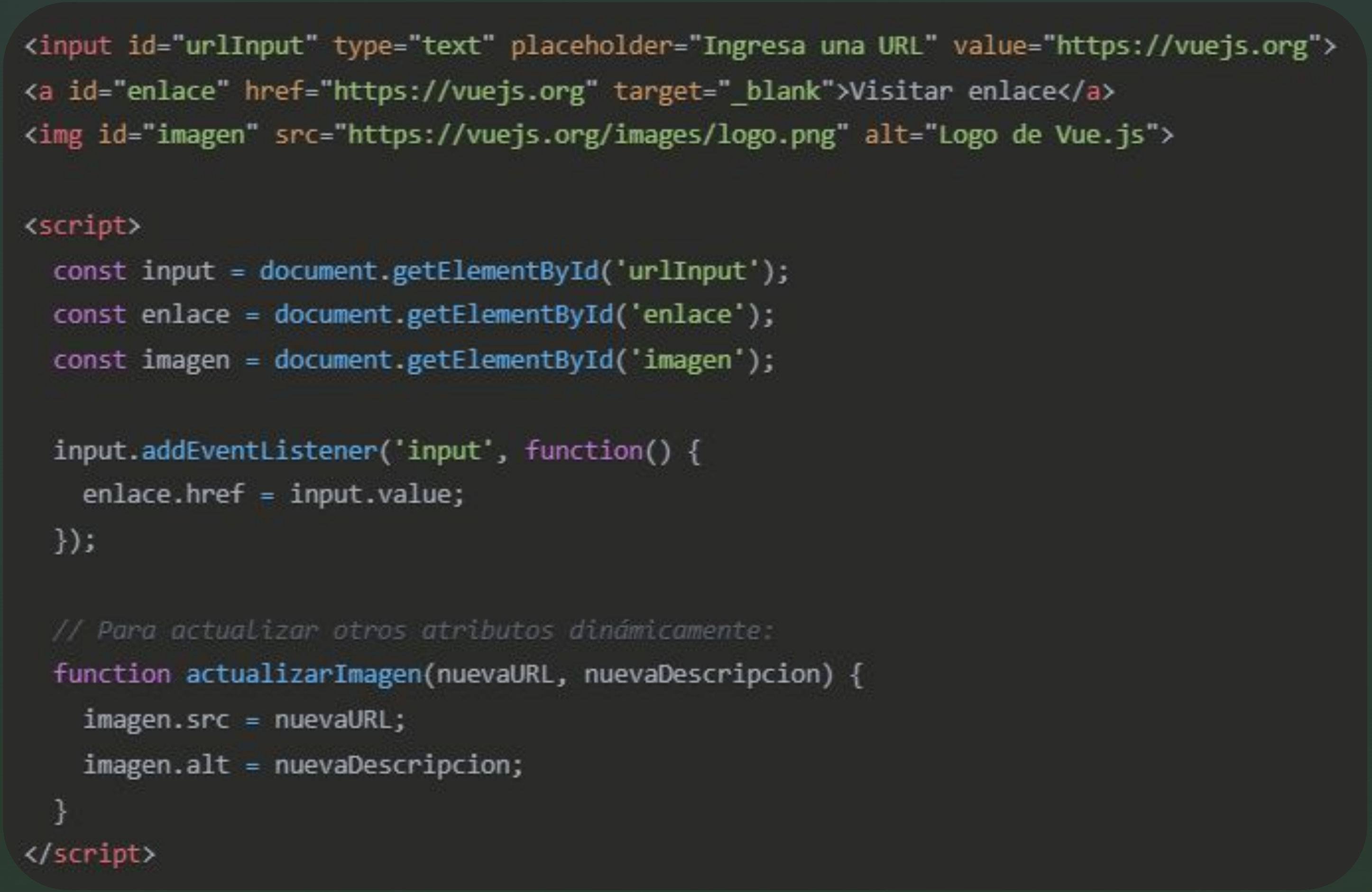
```
<button id="botonAlternar">Alternar</button>
<p id="contenidoPrincipal">Este contenido es visible</p>
<p id="contenidoAlternativo" style="display: none;">Este contenido alternativo es visible</p>

<script>
  let mostrar = true;
  const boton = document.getElementById('botonAlternar');
  const principal = document.getElementById('contenidoPrincipal');
  const alternativo = document.getElementById('contenidoAlternativo');

  boton.addEventListener('click', function() {
    mostrar = !mostrar;

    if (mostrar) {
      principal.style.display = 'block';
      alternativo.style.display = 'none';
    } else {
      principal.style.display = 'none';
      alternativo.style.display = 'block';
    }
  });
</script>
```

v-bind (Attribute link) (Vue)



```
<div id="app">
  <input type="text" v-model="url" placeholder="Ingresa una URL">
  <!-- Enlace completo -->
  <a v-bind:href="url" target="_blank">Visitar enlace</a>
  <!-- Forma abreviada -->
  <img :src="imagenURL" :alt="descripcion">
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      url: 'https://vuejs.org',
      imagenURL: 'https://vuejs.org/images/logo.png',
      descripcion: 'Logo de Vue.js'
    }
  })
</script>
```

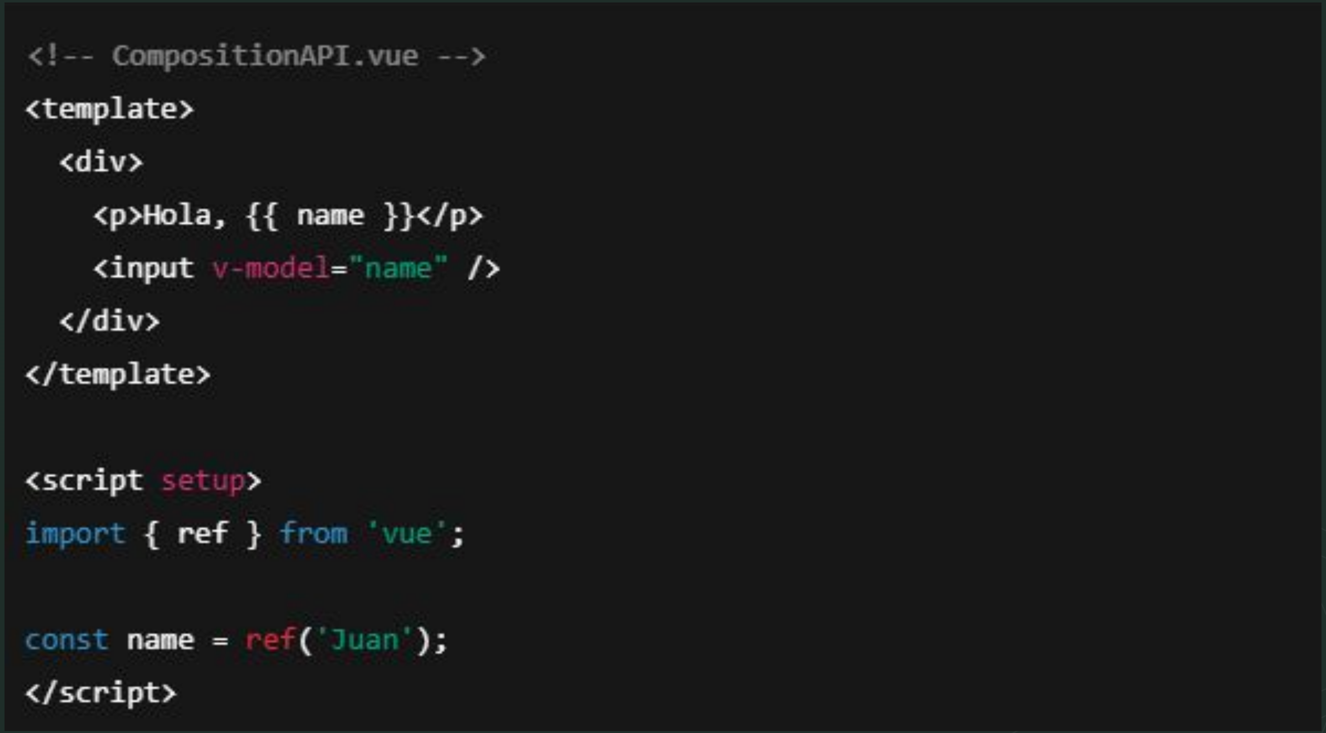
v-bind (Attribute link) (No-Vue)



```
<input id="urlInput" type="text" placeholder="Ingresa una URL" value="https://vuejs.org">
<a id="enlace" href="https://vuejs.org" target="_blank">Visitar enlace</a>
<img id="imagen" src="https://vuejs.org/images/logo.png" alt="Logo de Vue.js">

<script>
  const input = document.getElementById('urlInput');
  const enlace = document.getElementById('enlace');
  const imagen = document.getElementById('imagen');

  input.addEventListener('input', function() {
    enlace.href = input.value;
  });

  // Para actualizar otros atributos dinámicamente:
  function actualizarImagen(nuevaURL, nuevaDescripcion) {
    imagen.src = nuevaURL;
    imagen.alt = nuevaDescripcion;
  }
</script>
```

APIs (Composition)

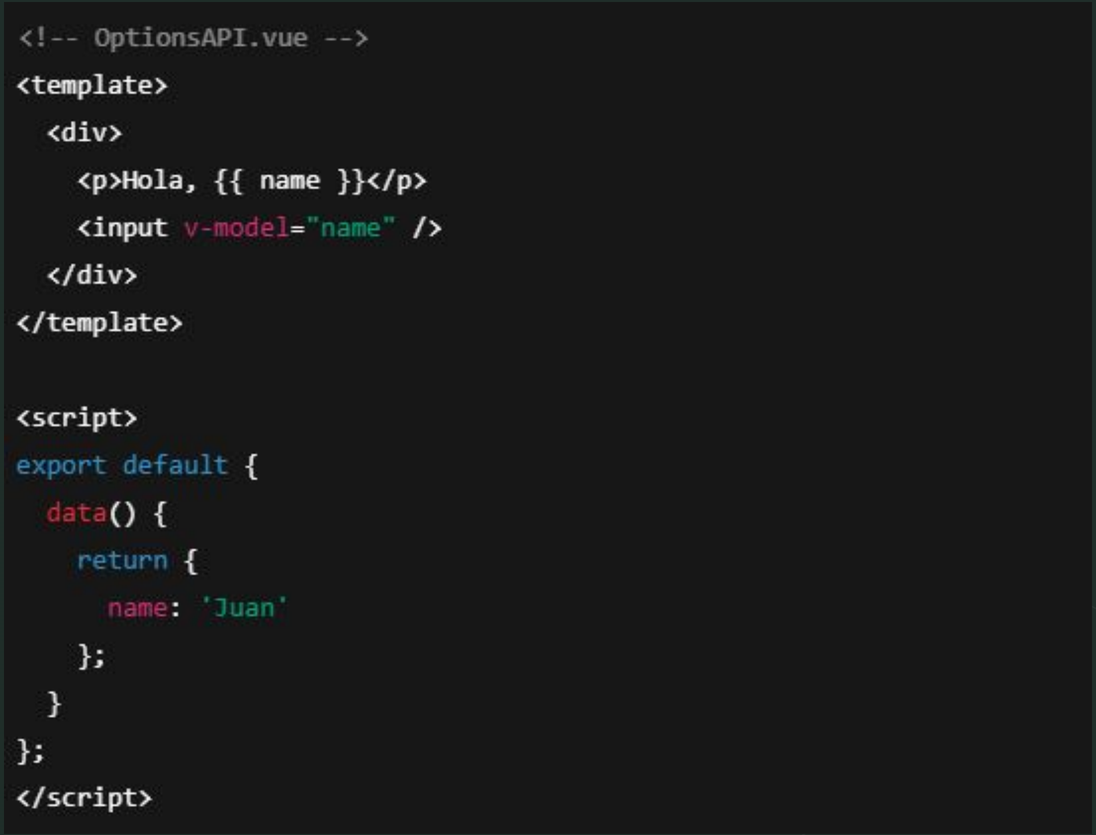
More powerful, function-based



```
<!-- CompositionAPI.vue -->
<template>
  <div>
    <p>Hola, {{ name }}</p>
    <input v-model="name" />
  </div>
</template>

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

const name = ref('Juan');
</script>
```

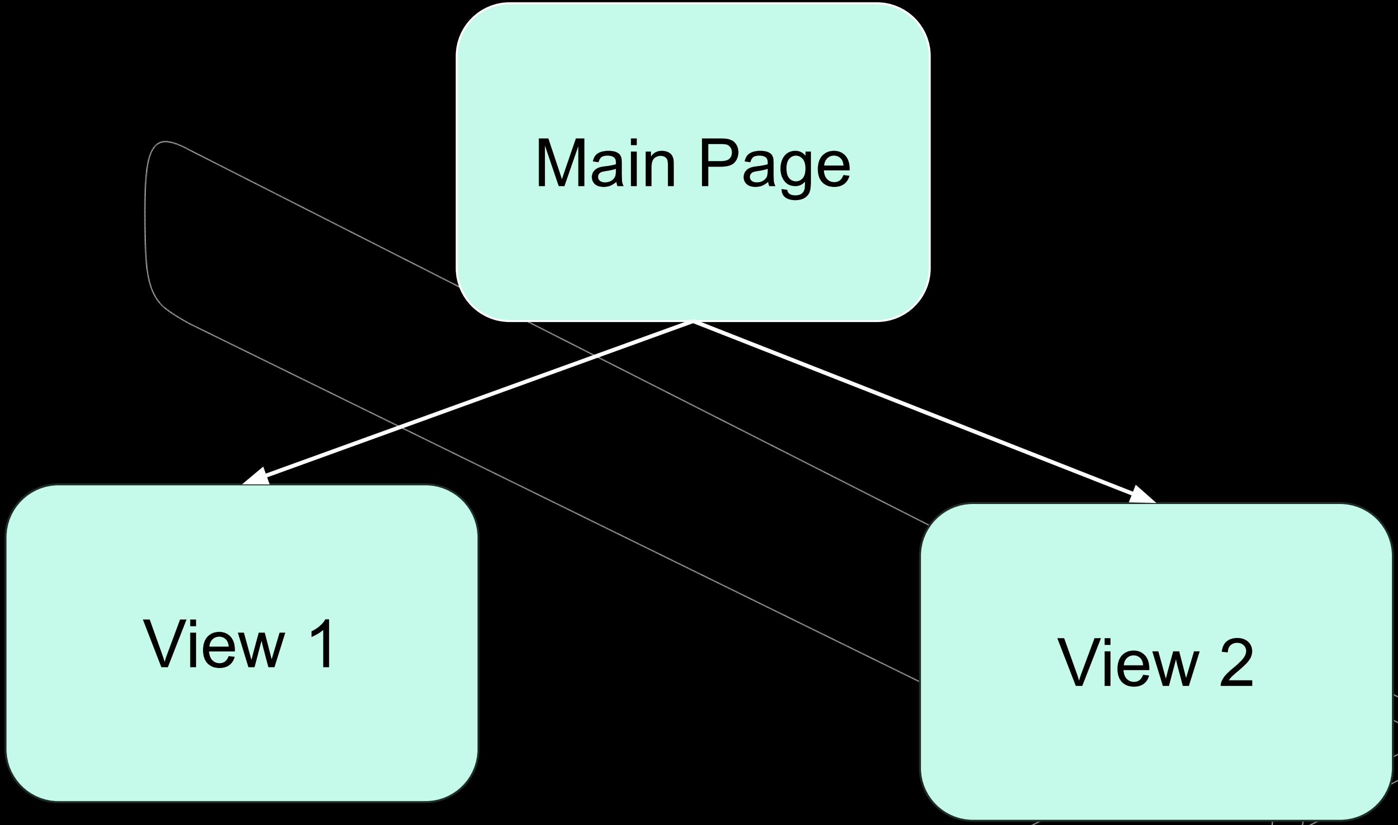
APIs (Options)

classic, object-based

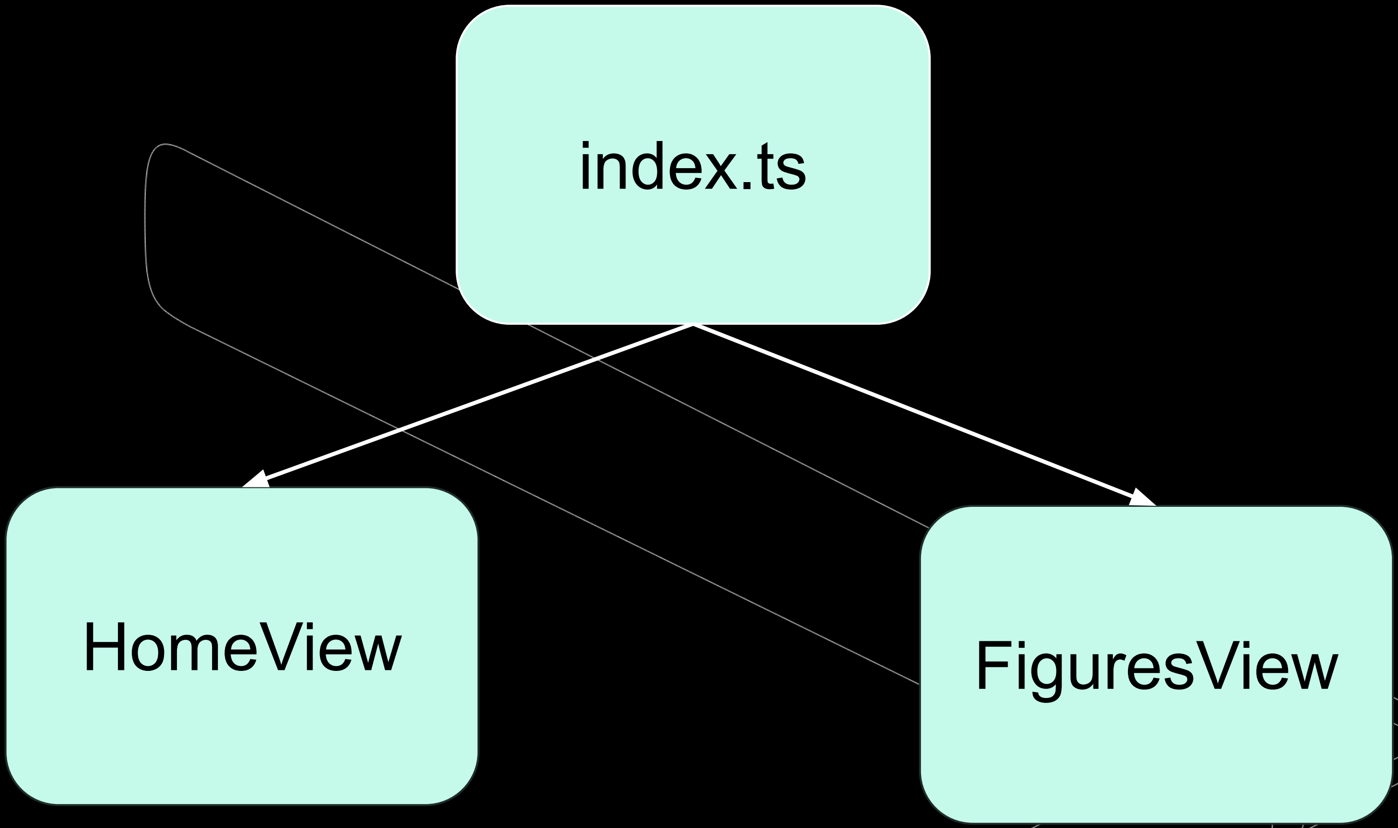


```
<!-- OptionsAPI.vue -->
<template>
  <div>
    <p>Hola, {{ name }}</p>
    <input v-model="name" />
  </div>
</template>

<script>
export default {
  data() {
    return {
      name: 'Juan'
    };
  }
};
</script>
```

Routing

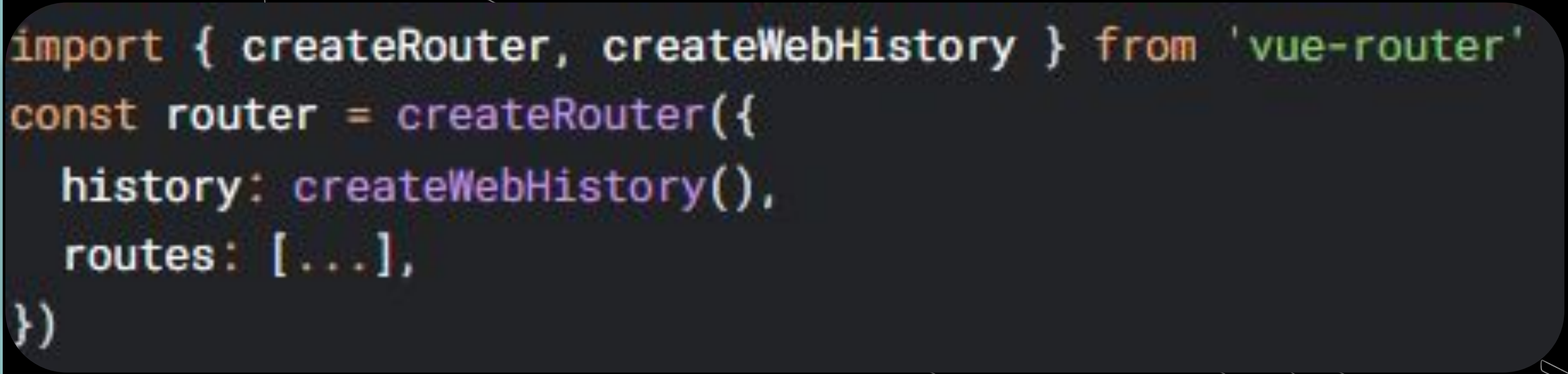


Routing



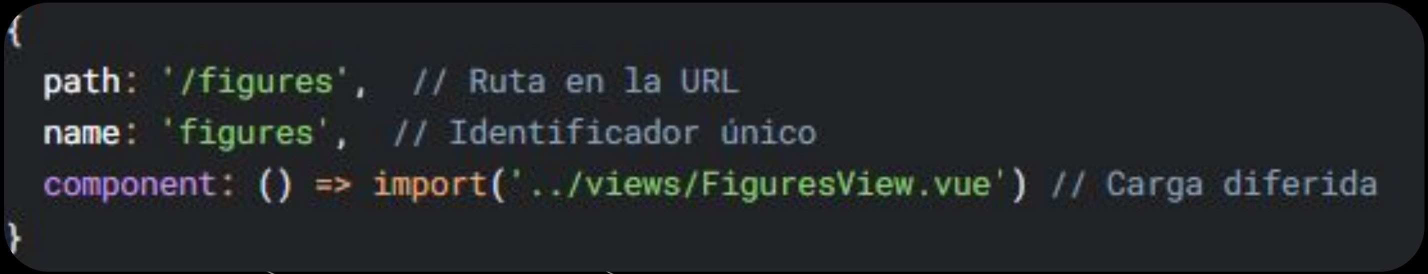
Routing

Creating the router and the web history

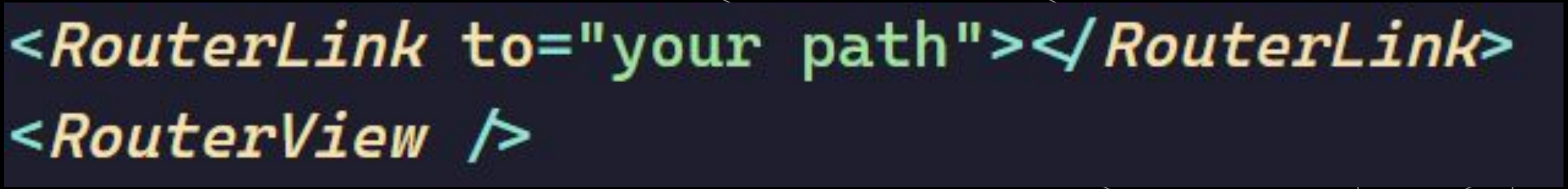


```
import { createRouter, createWebHistory } from 'vue-router'
const router = createRouter({
  history: createWebHistory(),
  routes: [...],
})
```

Routing



```
{
  path: '/figures',  // Ruta en la URL
  name: 'figures',  // Identificador único
  component: () => import('../views/FiguresView.vue') // Carga diferida
}
```



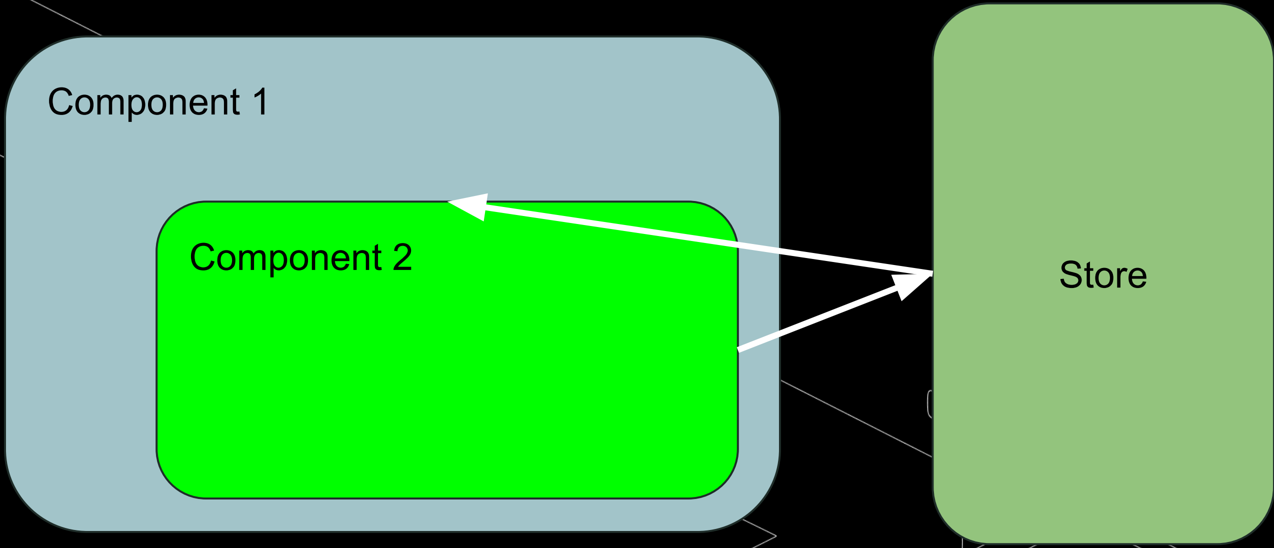
```
<RouterLink to="your path"></RouterLink>
<RouterView />
```

State Management

Pinia The intuitive store for Vue.js

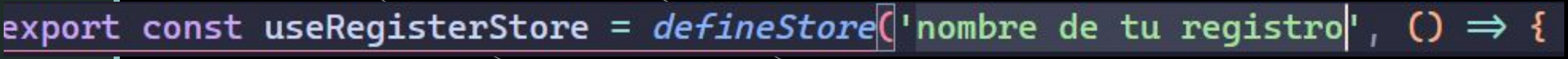
State Management

Pinia storage the states of the page



State Management

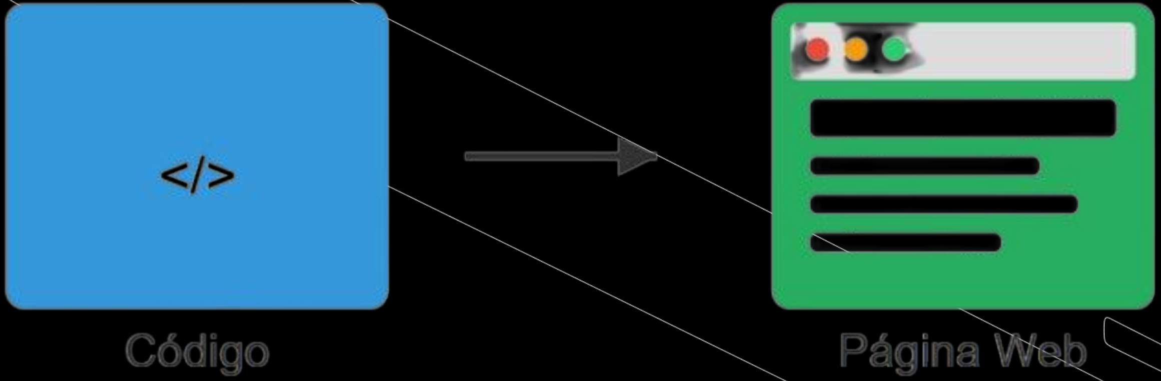
An example to how to start your own storage function



```
export const useRegisterStore = defineStore('nombre de tu registro', () => {
```

Build System

The **build system** is a set of tools that transforms your source code (Vue files, modern JavaScript, styles, etc.) into optimized, production-ready assets.



Build System

What it does:

- Compiles .vue files and modern JavaScript
- Bundles and minifies files for faster loading
- Manages dependencies
- Supports features like hot module replacement

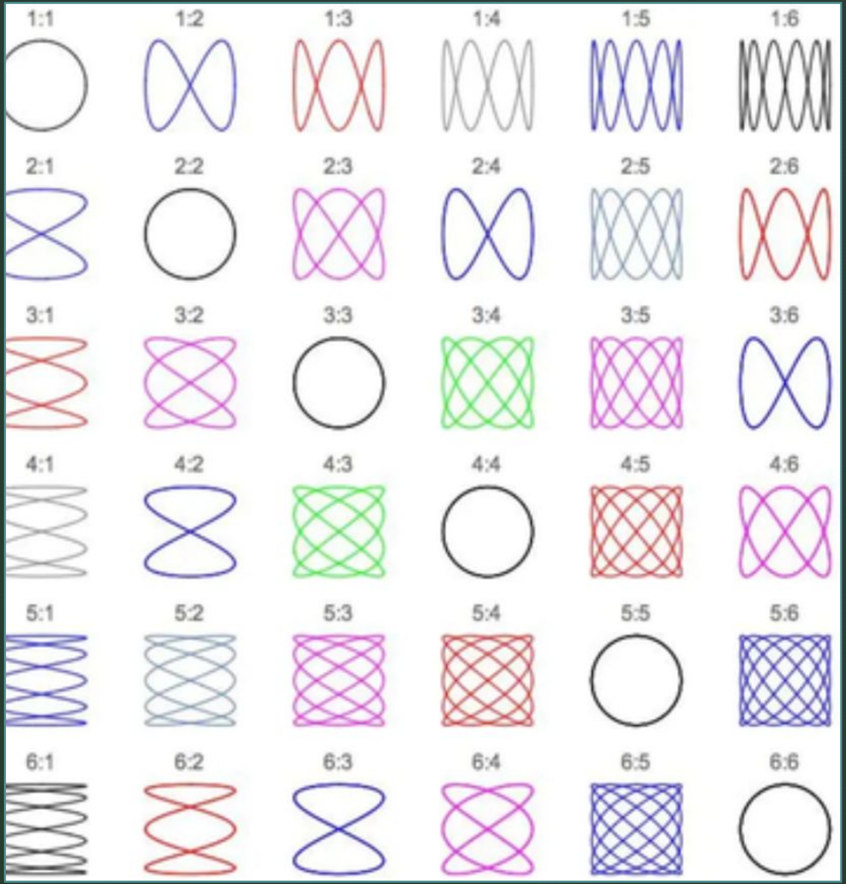
Build System

Common Tools:

- **Vite** (modern, fast default tool for Vue)
- **Webpack** (older, still in use)

Figures

Lissajous curves



Bibliografía

- Examples | Vue.js
- ¿Qué es Vue.JS? - Blog de Código Facilito
- Estructura de carpetas de VueJS - Javascript en español
- Qué es Vue JS y qué lo diferencia de otros frameworks | OpenWebinars
- <script setup> | Vue.js
- Composables | Vue.js
- ¿Qué es Vue.JS? - Blog de Código Facilito
- Vue.JS : The Progressive Framework!