

# Vue.js

UN  
FRAMEWORK  
VELOCE E  
VERSATILE PER  
CREARE  
INTERFACCE  
WEB





# Che cos'è Vue.js?

Vue.js è un framework JavaScript progressivo utilizzato per costruire interfacce utente. A differenza di altri framework monolitici, Vue è progettato per essere adottato in modo incrementale. Ciò significa che puoi iniziare integrandolo in parti di un progetto già esistente oppure utilizzarlo per creare nuove applicazioni da zero. Vue si concentra principalmente sul livello dell'interfaccia utente e offre un ecosistema che supporta la gestione dello stato, il routing e la costruzione di applicazioni complesse.





# Storia di Vue.js

**Creatore:** Evan You

**Rilasciato:** Febbraio 2014

Vue.js è stato sviluppato da Evan You, un ex ingegnere di Google che ha lavorato al progetto AngularJS. Desiderando creare qualcosa di più leggero e flessibile, Evan ha iniziato a lavorare su Vue nel 2013, rilasciando la prima versione pubblica nel febbraio 2014. Vue ha rapidamente guadagnato popolarità grazie alla sua semplicità, alla curva di apprendimento dolce e alla potente reattività.





## Caratteristiche Principali

- **Reattività:** Vue.js utilizza un sistema di reattività basato su un oggetto Proxy per tracciare le dipendenze e aggiornare automaticamente la DOM quando i dati cambiano.
- **Componenti:** Vue consente di suddividere l'interfaccia utente in componenti riutilizzabili e modulari. I componenti possono essere combinati per creare applicazioni complesse mantenendo il codice organizzato e facile da gestire.
- **Facilità di Integrazione:** Vue può essere utilizzato per migliorare progetti esistenti aggiungendo funzionalità interattive senza dover riscrivere completamente il codice esistente.





# PROXY

Oggetto Proxy: Vue 3 utilizza Proxy, una funzionalità di ECMAScript 2015, per intercettare le operazioni sui dati. Quando si definiscono proprietà dati in un componente Vue, queste vengono avvolte da Proxy. Ogni volta che si accede o si modifica una proprietà, Vue può monitorare queste operazioni grazie ai Proxy.





# Prima App

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>My first Vue page</title>
</head>
<body>

  <div id="app">
    {{ message }}
  </div>

  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

  <script>
    const app = Vue.createApp({
      data() {
        return {
          message: "Hello World!"
        }
      }
    })

    app.mount('#app')

  </script>
</body>
</html>
```





# Direttive

- V-bind: collega un attributo in un elemento html a un dato di vue
- V-if: crea un elemento html dipendentemente da una condizione (v-show)
- V-for: crea una lista di tag partendo da un array in vue utilizzando un ciclo for
- V-on: collega un evento di un tag html a un metodo presente in vue
- V-model: crea un doppio collegamento html tag e variabile, utilizzato soprattutto in form e input





# V-bind

```
<div id="app">
  
</div>

<script>
new Vue({
  el: '#app',
  data: {
    imageUrl: 'https://vuejs.org/images/logo.png'
  }
});
</script>
```







# V-if

```
<div id="app">
  <p v-if="isVisible">This is conditionally rendered.</p>
  <button @click="toggleVisibility">Toggle Visibility</button>
</div>

<script>
new Vue({
  el: '#app',
  data: {
    isVisible: true
  },
  methods: {
    toggleVisibility() {
      this.isVisible = !this.isVisible;
    }
  }
});
</script>
```





# V-for

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

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





# V-on

```
<div id="app">
  <button v-on:click="sayHello">Click me</button>
</div>

<script>
new Vue({
  el: '#app',
  methods: {
    sayHello() {
      alert('Hello, Vue!');
    }
  }
});
</script>
```





# V-model

```
<div id="app">
  <input v-model="message" placeholder="Type something">
  <p>The message is: {{ message }}</p>
</div>

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

