

Vue

Les directives



Ce que sont les directives

En Vue, les directives constituent les structures de contrôle qui vont être exécutées pour gérer la construction du HTML de manière dynamique

Il en existe principalement 5:

- v-bind
- v-if
- v-for
- v-on
- v-model



Utiliser les directives : V-BIND

V-BIND permet de relier un attribut html aux données de manière réactive

```
<body>
  <main id="app">... </main>
  <div id="app-2">
    <span v-bind:title="message">
      Hover your mouse over me for a few seconds
      to see my dynamically bound title!
    </span>
  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script>
<script>
  var app = new Vue({...})

  var app2 = new Vue({
    el: '#app-2',
    data: {
      message: 'You loaded this page on ' + new Date().toLocaleString()
    }
  })
</script>
</body>
```



Utiliser les directives : V-IF

V-IF permet de faire du templating conditionnel

```
<body>
  <main id="app">...</main>
  <div id="app-2">...</div>

  <div id="app-3">
    <span v-if="seen">{{ message }}</span>
    <span v-else> {{ messageKo }}</span>
  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script>
<script>
  var app = new Vue({...})
  var app2 = new Vue({...})

  var app3 = new Vue({
    el: '#app-3',
    data: {
      seen: true,
      message: 'Enregistrement OK!',
      messageKo: 'Enregistrement KO!'
    }
  })
</script>
</body>
```



Utiliser les directives : V-FOR

V-FOR permet de faire du templating bouclé

```
<body>
  <main id="app">...</main>
  <div id="app-2">...</div>
  <div id="app-3">...</div>

  <div id="app-4" >
    <ol>
      <li v-for="todo in todos">
        {{ todo.text }}
      </li>
    </ol>
  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script>
<script>
  var app = new Vue({...})
  var app2 = new Vue({...})
  var app3 = new Vue({...})

  var app4 = new Vue({
    el: '#app-4',
    data: {
      todos: [
        { text: 'Learn JavaScript' },
        { text: 'Learn Vue' },
        { text: 'Build something awesome' }
      ]
    }
  })
</script>
</body>
```



Utiliser la directive V-ON et les méthodes

V-ON permet d'écouter des événements sur des éléments HTML.

L'appel de ces événements va déclencher une méthode de l'application ou du composant Vue.

Ces méthodes doivent être déclarées comme propriété de l'application ou du composant Vue.

C'est-à-dire au même niveau que la propriété « data »



Utiliser la directive V-ON et les méthodes

```
<body>
  <main id="app">...</main>
  <div id="app-2">...</div>
  <div id="app-3">...</div>
  <div id="app-4">...</div>

  <div id="app-5">
    <p>{{ message }}</p>
    <button @click="reverseMessage">Go!</button>
  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script>
<script>
  var app = new Vue({...})
  var app2 = new Vue({...})
  var app3 = new Vue({...})
  var app4 = new Vue({...})

  var app5 = new Vue({
    el: '#app-5',
    data: {
      message: 'Hello Vue.js!'
    },
    methods: {
      reverseMessage: function() {
        this.message = this.message.split('').reverse().join('')
      }
    }
  })
</script>
</body>
```



Exemple : todo basique avec des événements

```
<body>
<div id="app-6">
  <ol>
    <li v-for="(todo, index) in todos">
      {{ todo.text }}
      <button v-
on:click="supprimerTodo(index)">Supprimer</button>
    </li>
  </ol>
  <input v-on:keyup.13="ajouterTodo" v-model="message" />
  <button v-on:click="ajouterTodo">Ajouter todo</button>
</div>
```

```
var app6 = new Vue({
  el: "#app-6",
  data: {
    todos: [],
    message: "Hello Vue!"
  },
  methods: {
    ajouterTodo: function () {
      this.todos.push({ text: this.message });
    },
    supprimerTodo: function (index) {
      this.todos.splice(index, 1);
    }
  }
});
</script>
</body>
```

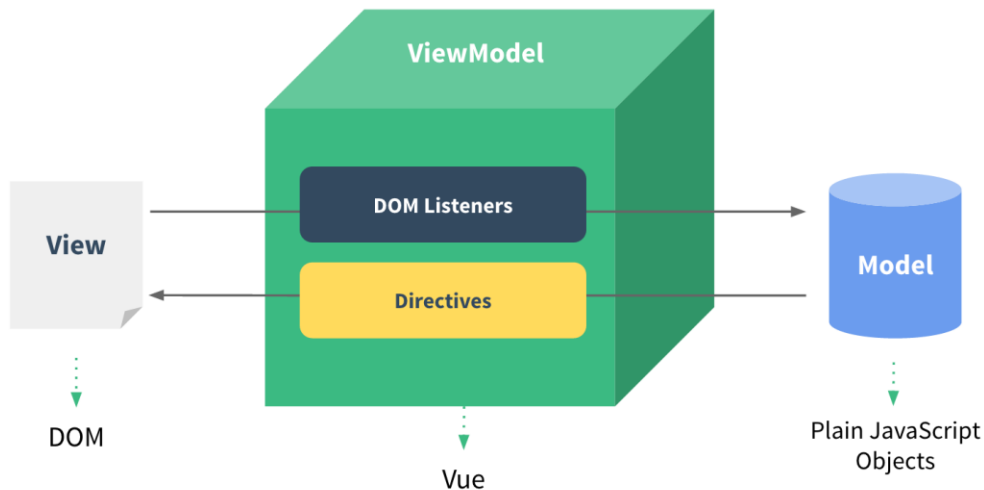

Vue

Les formulaires



Utiliser les directives : V-MODEL

V-MODEL met en place le two-way binding qui permet à un composant de mettre à jour les données qu'il doit afficher, depuis son affichage même





Utiliser les directives : V-MODEL

La directive V-Model: nous sert à interagir avec la valeur des inputs d'un formulaire Vue.

Il permet la communication bilatérale entre un composant vue et les données qui lui sont relatives

C'est une forme de V-BIND couplé à un V-ON:INPUT

```
<input v-model="something">
```



Équivaut à



```
<input  
  v-bind:value="something"  
  v-on:input="something = $event.target.value"  
>
```



Utiliser les directives : V-MODEL

Vous trouverez ci-dessous le code d'un formulaire basique en Vue.js

<https://pastebin.com/s35qFb3q>

Exercice :

- Nav-bar