

Jean-Frédéric VINCENT

# Créez une application web avec Vue.js



# Vue JS

FrameWork **FrontEnd** JavaScript



## Pourquoi Vue JS ?

- SPA **S**ingle **P**age **A**pplication
  - Les utilisateurs ne chargent une page web qu'une seule fois.
  - le lancement de nouvelles pages est instantanée
  - Les contenus de la page sont mis à jour simplement avec AJAX
- Economie de code : Plus de ~~DOM~~

# Vue JS



Quel différence avec les autres framework ?  
React ? Angular ? Polymer ?

Un **CDN** et c'est parti !

- Plutôt que d'installer un projet avec le **CLI** Command Line Interface
  - 20 000 fichiers
  - 500 Mo
- un simple CDN suffit !

# Le CDN

```
<html>
<head>
  <title>Document</title>
</head>
<body>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
</body>
</html>
```

# Le partie HTLM la vue

cibler une <div> pour afficher pour communiquer avec les données et les méthodes

```
<!-- ... -->  
<body>  
  <div id="app"></div>  
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>  
</body>  
<!-- ... -->
```

# Le partie JS

## Création d'un objet Vue à partir de la Classe VUE

```
<!-- ... -->
<body>
  <div id="app"></div>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  <script>
    const app = new Vue();
  </script>
</body>
<!-- ... -->
```

# Le partie JS

Relier la balise (element en anglais) **div** et l'objet Vue

```
<!-- ... -->
<body>
  <div id="app"></div>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  <script>
    const app = new Vue( {el: '#app' } );
  </script>
</body>
<!-- ... -->
```

# Le partie JS

Les différentes parties de notre objet Vue

```
el: '#app' ,
```

```
data: { // ... },
```

 Le **data Store** : les données de l'application

```
computed: { // ... },
```

 les valeurs calculées

```
methods: { // ... },
```

 les méthodes ou fonctions



# Les moustaches coté JS et vue HTML

```
<h1> {{ titre }} </h1>
```

```
data: {  
  titre = 'Mon titre',  
  couleur = 'vert'  
},
```

Vue JS

# Utilisation du data store

Démonstration

# Data Store : les tableaux

```
<ul>  
  <li v-for="fruit in tab">  
    {{ fruit }}  
  </li>  
</ul>
```

```
data: {  
  tab = [ 'kiwi', 'poire', 'kiwi' ]  
},
```

Vue JS

# Utilisation du data store

Démonstration

# Data Store : saisie de données

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

```
data: {  
  info : "",
```

# Data Store : les methodes

```
<button v-on:click="affiche()"> GO </button>
```

```
<button @click="affiche()"> GO2 </button>
```

```
methods: {  
  affiche(){  
  },  
},
```

Vue JS

# Lecture de données

## Démonstration

# Index du tableau ?

```
<p v-for="(item, index) in fruits">  
  {{ index }} - {{item}}  
</p>
```