



Prérequis

- Essentiel
 - HTML
 - CSS
 - Javascript
- Recommandé
 - Node.js
 - POO
 - Ajax

Installation

- Extension chrome/firefox pour débbugger [DevTools](#)
- CDN

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

- NPM

```
npm install vue
```

- [Vue CLI](#)

```
npm install -g @vue/cli
```

```
vue ui
```

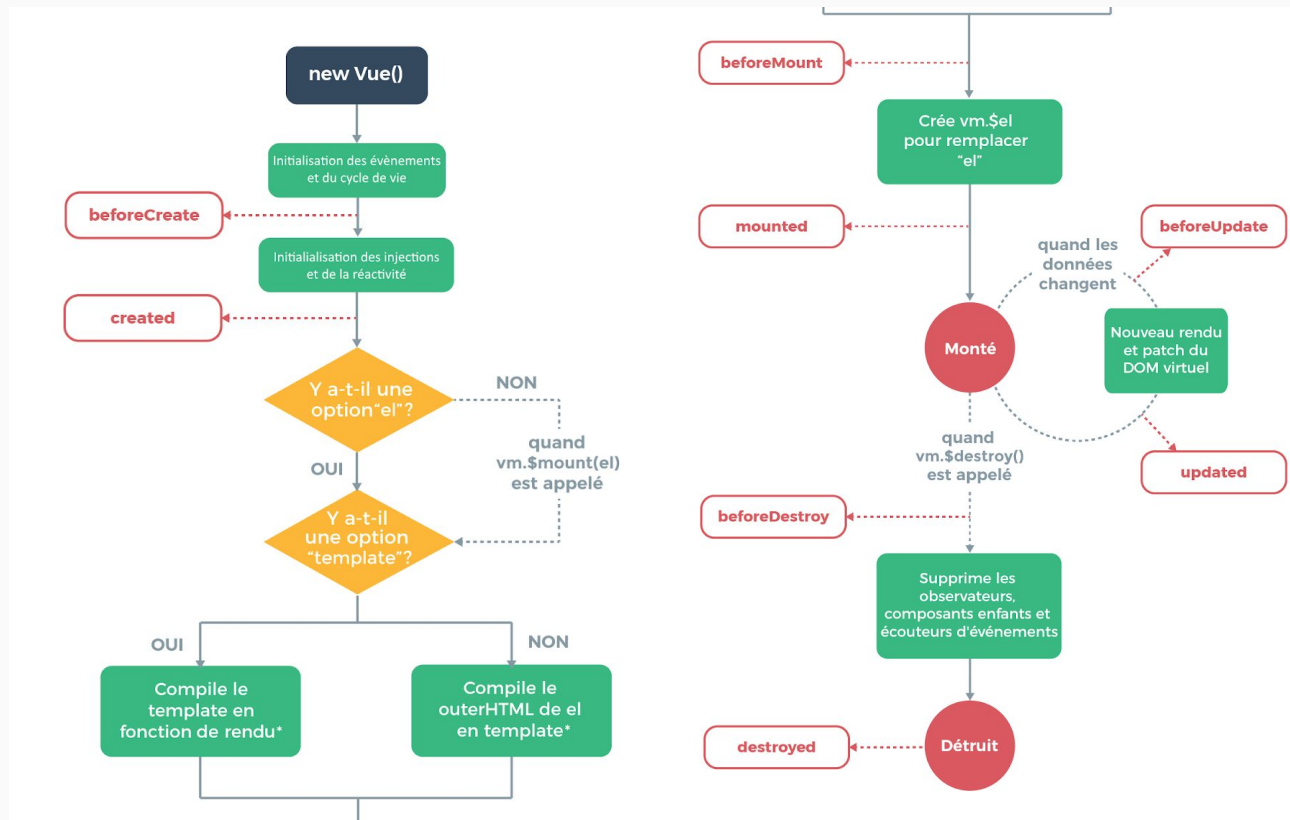
L'instance de vue & Lifecycle

```
var vm = new Vue({  
  // options  
})
```

Instance racine

- └─ TodoList
 - └─ TodoItem
 - └─ DeleteTodoButton
 - └─ EditTodoButton
 - └─ TodoListFooter
 - └─ ClearTodosButton
 - └─ TodoListStatistics

```
new Vue({  
  created: function () {  
    console.log('created')  
  }  
})
```



Data & Methods

```
new Vue({
  data: {
    livres: [
      { titre: `Peter Pan`, auteur: `James Matthew Barrie` },
      { titre: `La stratégie ender`, auteur: `Orson Scott Card` }
    ]
  },
  created: function() {
    this.listLivres()
  },
  methods: {
    listLivres: function() {
      for (livre of this.livres) {
        console.log(`${livre.titre} de ${livre.auteur}`)
      }
    }
  }
})
```

Computed & Watch

```
new Vue({  
  data: {  
    message: `Bonjour`  
  },  
  computed: {  
    reversedMessage: function() {  
      return this.message.split('').reverse().join('')  
    }  
  },  
  watch: {  
    message: function (newMessage, oldMessage) {  
      console.log(reversedMessage)  
    }  
  }  
})
```

Interpolation, Directives & Shorthands

```
<span>Message : {{ msg }}</span>  
<span v-once>Ceci ne changera jamais : {{ msg }}</span>
```

```
<p>String HTML entre accolades : {{ rawHTML }}</p>  
<p>String HTML avec directive v-html : <span v-html="rawHTML"></span></p>
```

```
<div v-bind:id="divId"></div>  
<button v-bind:disabled="btnDisabled">Valider</button>
```

```
{{ (btnDisabled) ? `Ok ${divId}` : `ko` }}  
{{ var a = 12; }} // NON!
```

```
v-(bind|if|on|for...)(:attribut|[attributDynamique])(.modificateur)=(expression)  
v-bind:id="divId", v-if="seen", v-on:click="method", v-on:submit.prevent="method"
```

```
:id="divId"  
@click="method"
```

Class & Styles

```
<label :style="{color: labelColor, `font-size`: `${labelSize}px`}">Nom</label>  
<label :style="labelStyle">Nom</label>
```

```
data: {  
  labelColor: `blue`,  
  labelSize: 22,  
  labelStyle: {  
    color: `blue`,  
    fontSize: 22  
  }  
}
```

```
<input type="text" :class="{ 'is-valid': validate }" />  
data: { validate: true }
```

```
<input type="text" :class="class" />  
data: { class: { 'is-valid': true } }
```


Conditional v-if

```
<template v-if="expression">
  <p>Expression true</p>
  <p>Paragraphe 2</p>
</template>
<p v-else-if="expression == null">Expression undefined</p>
<p v-else>Expression false</p>
```

```
<template v-if="loginType == 'username'">
  <label>Username</label>
  <input type="text" key="username" />
</template>
<template v-else>
  <label>Email</label>
  <input type="email" key="email" />
</template>
```

```
<p v-show="expression">Expression true</p>
```

Attention ne pas combiner la directive v-if avec v-for sur le même DOM!

Loop v-for

```
<ul>
  <li v-for="livre (in|of) livres|10" :key="livre.titre">{{ livre.titre|livre }}</li>
</ul>
```

```
<ul>
  <li v-for="(livre, index) (in|of) livres" :key="...">{{ message }} - {{ index }} - {{
livre.titre }}</li>
</ul>
```

```
<ul>
  <li v-for="value|(value, name|(, index)) (in|of) objLivre" :key="...">{{ index }} - {{ name }} :
{{ value }}</li>
</ul>
```

```
data: {
  message: `Accessible`,
  livres: [
    { titre: `Peter Pan`, auteur: `James Matthew Barrie` },
    { titre: `La stratégie ender`, auteur: `Orson Scott Card` }
  ],
  objLivre: { titre: `Titre du livre`, auteur: `Auteur du livre` }
}
```

Events

```
<button v-on:click|@click="counter++">Nb click {{counter}}</button>
```

```
data: {  
  counter: 0  
}
```

```
<button @click="minCounter">Method minCounter</button>  
<button @click="sayHi('Peter', $event)">Say Hi</button>
```

```
methods: {  
  minCounter: function(event) {  
    this.counter--;  
    console.log(event)  
  },  
  sayHi: function(name, event) {  
    alert(`Salut ${name}`);  
    console.log(event)  
  }  
}
```

@click(.modificateur)="expression" avec modificateur : stop, prevent, capture, self, once, passive, code de touche, touche système, souris [voir documentation](#)

Form binding v-model

```
<input type="text" v-model="msg" />
<textarea v-model="description"></textarea>
<input type="checkbox" v-model="checked" />
<input type="radio" v-model="radioPick" value="Un" />
<input type="radio" v-model="radioPick" value="Deux" />
<input type="checkbox" v-model="checkeds" value="A" name="grouped" />
<input type="checkbox" v-model="checkeds" value="B" name="grouped" />
<input type="checkbox" v-model="checkeds" value="C" name="grouped" />
<select v-model="selected"><option disabled value="">Choisir...</option>[...]</select>
<select multiple v-model="selecteds"><option disabled value="">Choisir...</option>[...]</select>
```

```
data: {
  msg: '',
  description: '',
  checked: false,
  radioPick: '',
  checkeds: [],
  selected: '',
  selecteds: []
}
```

Avec modificateurs : `.lazy`, `.number`, `.trim` [voir documentation](#).