

VUE

VUE

DURE TOTALE DU MODULE : 21H00

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Affichage de contenu HTML (directive v-html)

Dans des applications web classique (HTML,CSS,JS), il est possible et pratique d'utiliser la technique des Template Strings, l'idée consiste à ce qu'une variable contienne une chaîne de caractères (String) dans laquelle on écrit du code HTML (Template), on affiche ensuite cette variable dans la page HTML.

De fait nous avons vu comment afficher des variables et nous pourrions être tenté d'utiliser la même technique en VUE (afficher dans le Template une variable de type String qui contient du code HTML).

Mais ... le code HTML ne sera pas interprété par le navigateur et c'est une bonne chose, VUE intègre de base une mesure de sécurité contre les attaques XSS (Cross Site Scripting), à la place VUE propose une directive (v-html).

[SOLUTION BookListV1.3\(GitHub\)](#)

[SOLUTION BookListV1.3\(StackBlitz\)](#)

Fichier JS

```
Vue.createApp({
  data(){
    return {
      unTemplateLivre : "<h1>Les mémoires de Steven Seagal</h1>"
    }
  }
}).mount('#bookListApp');
```

Fichier HTML

```
<h1 class="text-center">Utilisation de la directive v-html</h1>
<hr>
<div id="bookListApp" class="card">
  <!-- Avec String interpolation {} le code HTML n'est pas interprété par le
navigateur -->
  <p class="text-center">{{unTemplateLivre}}</p>
  <!-- Directive v-html pour afficher du code HTML contenu dans un variable
String -->
  <p v-html="unTemplateLivre" class="text-center"></p>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Rendu

Utilisation de la directive v-html

```
<h1>Les mémoires de Steven Seagal</h1>
```

Les mémoires de Steven Seagal

Lier Des attributs HTML (directive v-bind)

La directive v-bind va permettre de relier n'importe quel attribut html (href, src, alt, title, etc..) à une variable ou expression JS.

PS : la directive v-bind étant une des plus utilisées on a une notation raccourcie :

v-bind :href peut s'écrire de cette façon :href

[SOLUTION BookListV1.4\(GitHub\),](#)

[SOLUTION BookListV1.4\(StackBlitz\)](#)

Fichier JS

```
Vue.createApp({
  data(){
    return {
      leLien : "https://www.google.com",
    }
  }
}).mount('#bookListApp');
```

Fichier HTML

```
<h1 class="text-center">Utilisation de la directive v-bind</h1>
<hr>
<div id="bookListApp" class="card">
  <!-- Ici on relie l'attribut href avec notre variable leLien -->
  <a v-bind:href="leLien">LIEN DYNAMIQUE</a>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

TP : DataBinding


Créer une nouvelle application qui affiche et gère les données d'un profil utilisateur.

TP à rendre au format Zip ou lien github

L'application doit afficher les données de l'utilisateur suivantes :

- L'image
- Le nom
- L'âge
- Son nombre fétiche (un nombre aléatoire)
- Le résultat d'une fonction qui fait l'âge +10
- Un input texte pré-rempli avec le nom

TP DataBinding - Profile User



Mario

45 ans

age + 10 : 55 ans

NB Random : 0.3148920573285465

[SOLUTION TP USERPROFILE \(GitHub\)](#)

[SOLUTION TP USERPROFILE \(Stackblitz\)](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Réagir à des évènements DOM

Maintenant que nous avons vu comment afficher des données via l'interpolation `{{ }}`, et lier des attributs HTML à des données via le binding, la prochaine étape pour construire des applications réactives, consiste à pouvoir réagir aux évènements du DOM (le click sur un bouton, le clavier, la souris de l'utilisateur...).

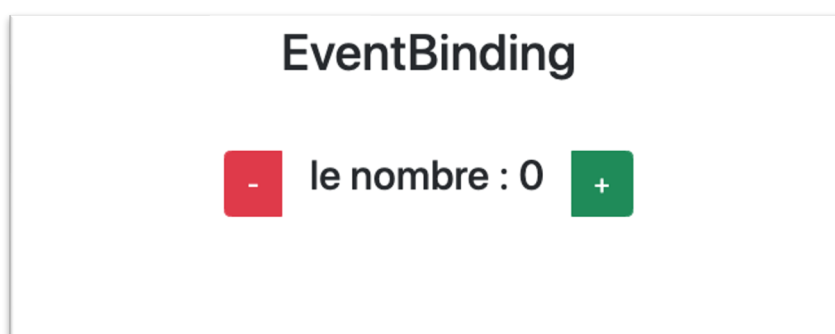
En Javascript (classique) pour réagir à ces évènements, on utilise la fonction `addEventListener`, VUE quant à lui propose la directive `v-on`, que l'on utilise côté HTML comme un attribut.

Afin d'illustrer l'utilisation de cette directive nous allons mettre en place une application qui augmente ou réduit un nombre.

PS : la directive `v-on` étant une des plus utilisées on a une notation raccourcie :
`v-on :click` peut s'écrire de cette façon `@click`

Exercice : setUp Application pour EventBinding

Mettre en place une nouvelle application VUE dans laquelle on affiche une donnée qui est un nombre entre 2 boutons + et -



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier HTML :

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button type="button" class="btn btn-danger"></button>
    <h3 class="mx-3">le nombre : {{leNombre}}</h3>
    <button type="button" class="btn btn-success">+</button>
  </div>
</div>
```

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      leNombre : 0
    };
  },
  methods: {
  }
}).mount('#monApp');
```

Nous allons maintenant mettre en place la directive v-on directement sur les boutons pour réagir au click. (En js classique il faudrait stocker le bouton dans une variable, utiliser un AddEventListener dessus, dans ce AddEventListener décrire une fonction). L'approche de VUE est donc plus directe.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Utilisation de la directive v-on + expressions JS

Pour mettre en place le système de compteur de l'application, nous allons placer 2 directives v-on :click, (après v-on : on précise le nom de l'évènement auquel on veut réagir)

Dans un v-on nous pouvons écrire des expressions JS (un calcul, une condition ternaire). Mais si le code JS se complexifie nous adopteront plutôt l'approche qui consiste à ce qu'un v-on exécute une fonction.

Exemple de v-on :click sur les boutons :

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button v-on:click="leNombre= leNombre-1" type="button" class="btn btn-
danger">--</button>
    <h3 class="mx-3">le nombre : {{leNombre}}</h3>
    <button v-on:click="leNombre++" type="button" class="btn btn-
success">+</button>
  </div>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Utilisation de la directive v-on + fonction

L'approche la plus commune consiste à utiliser v-on et des fonctions que l'on écrit côté JS.

Dans l'exemple ci-après nous allons créer 2 nouvelles fonctions augmenter() et réduire(), elle seront appelées par la directive v-on.

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      leNombre : 0
    };
  },
  methods: {
    augmenter(){
      this.leNombre ++;
    },
    reduire(){
      this.leNombre --;
    }
  }
}).mount('#monApp');
```

Fichier HTML : Nous avons le choix de faire référence à la fonction (reduire) ou d'exécuter la fonction (augmenter()) Généralement on va favoriser le fait de faire référence à la fonction (on utilise pas de paramètres)

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button v-on:click="reduire" type="button" class="btn btn-danger">-</button>
  </div>
  <h3 class="mx-3">le nombre : {{leNombre}}</h3>
  <button v-on:click="augmenter()" type="button" class="btn btn-success">+</button>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Utilisation de la directive v-on + fonction avec des paramètres

Dans cet exemple nous allons modifier nos fonctions côté JS de manière à ce qu'elles prennent en compte un paramètre.

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      leNombre : 0
    };
  },
  methods: {
    augmenter(num){
      this.leNombre += num;
    },
    reduire(num){
      this.leNombre -= num;
    }
  }
}).mount('#monApp');
```

Fichier HTML : on exécute les fonctions en leur passant un paramètre

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button v-on:click="reduire(10)" type="button" class="btn btn-danger">-
  </button>
  <h3 class="mx-3">le nombre : {{leNombre}}</h3>
  <button v-on:click="augmenter(20)" type="button" class="btn btn-
  success">+</button>
  </div>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

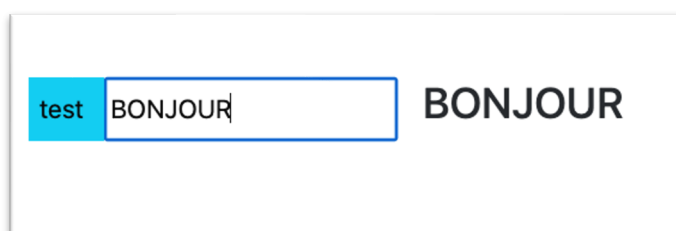
VUE

Utilisation de la directive v-on + fonction avec paramètre pour capter un évènement (Native Event Object)

Nous avons vu comment réagir à un évènement (le click sur un bouton), mais dans certains cas d'utilisation il peut être nécessaire de capter l'évènement pour utiliser ses propriétés.

Par exemple dans un évènement de scroll il y a des propriétés correspondant à la position du scroll, dans un évènement click nous pouvons retrouver les propriétés correspondantes aux coordonnées du click.

Dans l'exemple suivant nous mettons en place un bouton ainsi qu'un champ de texte dans le Template HTML, puis côté JS nous allons écrire des fonctions permettant de capter ces évènements.



Fichier HTML :

Le 1er v-on permettra de capter l'évènement click, le second capter l'évènement input (quand l'utilisateur tape au clavier dans le champ de texte)

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <h3 class="mx-3">le nombre : {{leNombre}}</h3>
    <button v-on:click="capterEventClick($event)" type="button" class="btn btn-info">test</button>
    <input v-on:input="capterEventInput($event)" type="text">
    <h3 class="mx-3">{{uneString}}</h3>
  </div>
</div>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordnatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      leNombre : 0,
      uneString: 'Hello'
    };
  },
  methods: {
    capterEventClick(event){
      console.log(event)
    },
    capterEventInput(event){
      console.log(event);
      this.uneString = event.target.value
    }
  }
}).mount('#monApp');
```

En mettant en place des fonctions dans lesquelles on capte les événements observons les console log qui sont générés lorsque l'on click et lorsqu'on écrit dans l'input.

Dans la fonction capterInputEvent on récupère la valeur contenu dans l'évènement input pour l'assigner à la variable uneString.

Propriété de l'évènement Click (gauche) et Propriété de l'évènement input(droite)

```
bookList(1.6).js:19
▼ PointerEvent {isTrusted: true, pointerId: 1,
  width: 1, height: 1, pressure: 0, ...} ⓘ
  ▶ delegateTarget: document
    isTrusted: true
    altKey: false
    altitudeAngle: 1.5707963267948966
    azimuthAngle: 0
    bubbles: true
    button: 0
    buttons: 0
```

```
bookList(1.6).js:22
InputEvent {isTrusted: true, data: 'd', isComposing: false, inputType: 'insertText', dataTransfer: null, ...} ⓘ
  isTrusted: true
  bubbles: true
  cancelBubble: false
  cancelable: false
  composed: true
  currentTarget: null
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Utilisation des Event Modifiers

Pour aller plus loin avec l'utilisation de la directive v-on, VUE propose des modifiers sur les event, qui vont permettre de modifier un évènement.

Pour illustrer cela, nous allons reprendre notre application de compteur, dans laquelle nous allons rajouter un formulaire contenant un input de type text ainsi qu'un bouton

Dans les applications web classique un comportement par défaut du navigateur avec un formulaire consiste à recharger la page lorsqu'on soumet ce formulaire, nous allons donc mettre en place un modifier permettant d'éviter ce comportement par défaut.

Nous allons également mettre en place des modifiers sur les click des boutons pour réagir au click gauche et droit de la souris.

Fichier HTML :

Sur les boutons : v-on :click.left (réagir au click gauche de la souris), v-on :click.right (réagir au click droit de la souris)

Sur le formulaire : v-on :submit.prevent (pour empêcher le comportement par défaut qui recharge la page)

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button v-on:click.left="reduire(10)" type="button" class="btn btn-danger"></button>
    <h3 class="mx-3">le nombre : {{leNombre}}</h3>
    <button v-on:click.right="augmenter(20)" type="button" class="btn btn-success"></button>
    <button v-on:click="capterEventClick($event)" type="button" class="btn btn-info">test</button>
    <form v-on:submit.prevent>
      <input v-on:input="capterEventInput($event)" type="text">
      <button>Envoyer</button>
    </form>
    <h3 class="mx-3">{{uneString}}</h3>
  </div>
</div>
```

[SOLUTION\(GitHub\)](#) / [SOLUTION \(StackBlitz\)](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

TP: EventBinding

TP à rendre au format Zip ou lien github

Dans cette application nous devons :

- 1 input sur lequel on va réagir à l'événement keyup pour afficher ce qui est tapé au clavier dans un paragraphe
- 1 input sur lequel on va réagir à l'événement keyup pour afficher ce qui est tapé au clavier dans un paragraphe (l'affichage s'effectue UNIQUEMENT si l'on appuie sur la touche esc)
- 1 bouton qui déclenche une alerte.

TP EventBinding

hhhhhh

hhhhhh

fff

fff

Alerte

127.0.0.1:5502 indique
ALERTE GENERALE

OK

hhhhhh

fff

fff

Alerte

[SOLUTION\(GitHub\)](#)

[SOLUTION\(StackBlitz\)](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

2 Way DataBinding (EventBinding + DataBinding) directive v-model

Précédemment nous avons abordé le fait de lier des attributs HTML et réagir à des événements, dans certains cas d'utilisations impliquant des input / textarea (souvent des formulaires), la directive v-model va permettre de lier directement (1 seule directive à utiliser) une variable (JS) (côté Model) avec le contenu (la value) d'un input / textarea (côté Vue).

Exercice v-model

Mettre en place une application dans laquelle nous avons :

°Un input qui utilise :

- v-on pour surveiller les touches du clavier et déclenche une fonction afficherUserName dans laquelle on capte l'évènement pour retrouver ce qui est tapé dans l'input (event.target.value) que l'on assigne à nameUser

- v-bind pour relier la value de cet input avec la variable nameUser

°Un affichage de la variable nameUser dans un paragraphe

°Un autre input sur lequel on utilise un v-model pour le lier à une variable userName

°Un affichage de la variable userName dans un paragraphe

[SOLUTION\(GitHub\)](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier HTML :

```
<div id="monApp" class="container mt-4 mb-4 p-3">
  <div class="mb-3">
    <p>v-on + v-bind</p>
    <input v-on:keyup="afficherNameUser" v-bind:value="nameUser" type="text"
class="form-control" >
  </div>
  <div class="mb-3">
    <p>{{nameUser}}</p>
  </div>
  <div class="mb-3">
    <p>v-model</p>
    <!-- modèle MVC vs MVVM -->
    <input v-model="userName" type="text" class="form-control" >
  </div>
  <div class="mb-3">
    <p>{{userName}}</p>
  </div>
</div>
```

Avec des éléments HTML type inputs, v-model est plus direct.

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return {
      nameUser: '', userName: '';
    },
    methods: {
      afficherNameUser(event) {
        this.nameUser = event.target.value;
      }
    }
  });

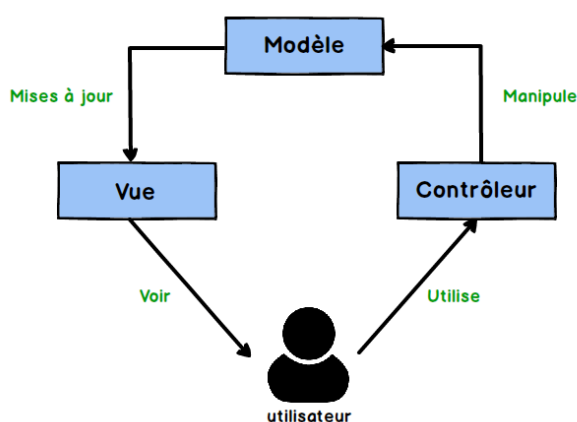
app.mount('#monApp');
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

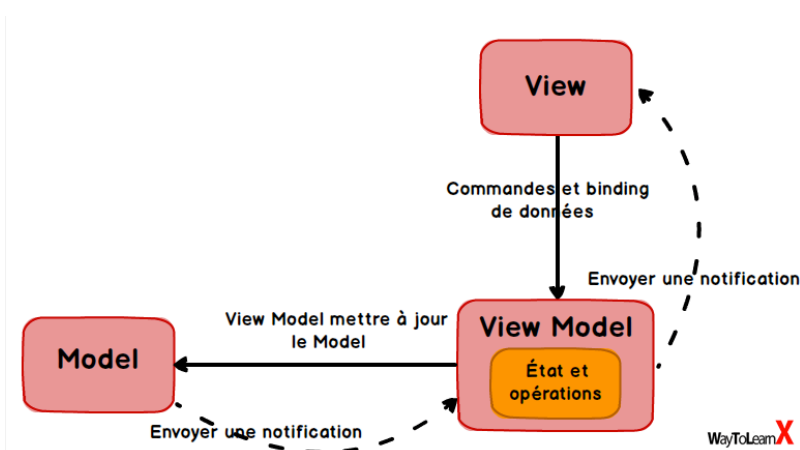
VUE

Design Pattern MVC vs MVVM

Cette directive v-model permet de mettre en évidence une approche différentes dans la construction d'application web, la plupart des applications web « classiques » utilisent un Design Pattern (architecture) MVC (Model, View, Controller) :



Avec v-model nous avons un Design Pattern MVVM :



<https://waytolearnx.com/2020/06/difference-entre-mvc-et-mvvm.html>

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Utilisation des computed Properties

Au cours du développement d'une application VueJS, (plus complète), nous allons gérer plusieurs fonctions, plusieurs données, on va très vite se retrouver dans des situations où sur la même page nous devons réagir à plusieurs événements et exécuter des fonctions, cela va engendrer des interférences entre ces différentes fonctions que l'on utilise. Dans l'exercice suivant nous allons volontairement créer ce problème.

Exercice : Bug événements + fonctions (interférences)

Dans une application :

- Mettre en place un système de compteur
- Une fonction dans laquelle on manipule une variable userName (on exécute cette f° directement dans un paragraphe côté HTML):
 - o Un console log qui affiche « fonction exécutée »
 - o Si userName est vide alors on retourne « test »
 - o Sinon on retourne « un autre test »



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier HTML :

```
<h1 class="text-center">v-on + fonctions (PB ?)</h1>
<div id="monApp" class="container mt-4 mb-4 p-3">
  <div class="mb-3">
    <button v-on:click="augmenter(20)" type="button" class="btn btn-success mt-3">+</button>
    <button v-on:click.left="reduire(10)" type="button" class="btn btn-danger mt-3">-</button>
  </div>
  <div class="mb-3">
    <p>{{afficherNameUser()}}</p>
  </div>
  <div class="mb-3">
    <p>{{leNombre}}</p>
  </div>
</div>
```

Nous avons plusieurs fonctions qui sont exécutées sur la même page HTML

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return {
      leNombre:0,nameUser: ''};
  },
  methods: {
    afficherNameUser() {
      console.log('fonction exécutée');
      if(this.nameUser === ''){
        return 'test'
      }
      else{
        return 'autre Test';
      }
    },
    augmenter(num){
      this.leNombre += num;
    },
    reduire(num){
      this.leNombre -= num;
    },
  }
});
app.mount('#monApp');
```

Dans l'application il y a un problème : lorsque l'on clique sur les boutons qui interagissent avec le nombre, le console log (qui se situe pourtant dans une tout autre fonction) est déclenché.

Le problème vient du fait que nous exécutons directement dans le template la fonction afficherNameUser :

```
<p>{{afficherNameUser()}}</p>
```

Pour des raisons de performance (réactivité) Vue va en temps réel surveiller le DOM, à chaque interaction VueJs va recharger le DOM en arrière-plan (DOM Virtuel) et de fait va ré-exécuter les fonctions. (<https://la-cascade.io/comprendre-le-virtual-dom/>)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Afin de solutionner ce problème, VueJs intègre une alternative à methods pour gérer les fonctions, les computed properties.

Mise en place :

modification côté HTML : on fait référence à la fonction (pas de parenthèse pour exécuter)(comme une property (une variable))

```
<p>{{afficherNameUser}}</p>
```

Modification côté JS : on déplace la fonction dans un objet computed (même utilisation que methods)

```
const app = Vue.createApp({
  data() {
    return {
      leNombre:0,nameUser: ''};
  },
  methods: {
    augmenter(num){
      this.leNombre += num;
    },
    reduire(num){
      this.leNombre -= num;
    },
  },
  computed:{
    afficherNameUser() {
      console.log('fonction executée');
      if(this.nameUser === ''){
        return 'test'
      }
      else{
        return 'autre Test';
      }
    },
  },
});
app.mount('#monApp');
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

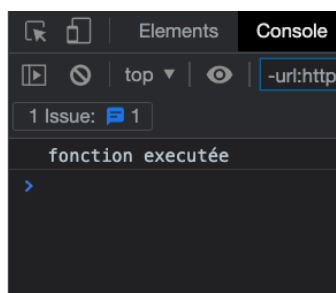
Quand une fonction est en mode computed, VueJS va gérer les dépendances (les variables) utilisées par cette fonction, c'est-à-dire que la fonction ne sera exécutée QUE s'il y a des modifications sur les variables dont elle dépend.

v-on + fonctions (PB ?)



test

130



Maintenant la fonction d'affichage n'est plus déclenchée par les fonctions du compteur.

[LIEN\(GitHub\)](#)

Utilisation des Watchers

Fréquemment dans les applications, nous avons besoin de surveiller les changements d'une seule variable uniquement, pour cela VueJS propose un système de watcher, dans un objet watch (même utilisation que computed et methods), nous allons définir une fonction qui porte le même nom qu'une de nos variable et VueJS va automatiquement capter les changements de valeur de cette variable. Voyons cela en reprenant le système de compteur.

Fichier HTML :

```
<div id="monApp" class="container mt-4 mb-4 p-3">
  <div class="mb-3">
    <button v-on:click="leNombre++" type="button" class="btn btn-success">+</button>
    <p>{{leNombre}}</p>
  </div>
```

On affiche la variable leNombre

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return { leNombre:0 };
  },
  watch:{
    leNombre(value) {
      if(value ==7){
        this.leNombre = 99
      }
    },
  },
});
app.mount('#monApp');
```

Vue Capte automatiquement les changements de valeur (value) de cette variable



[Lien GitHub](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

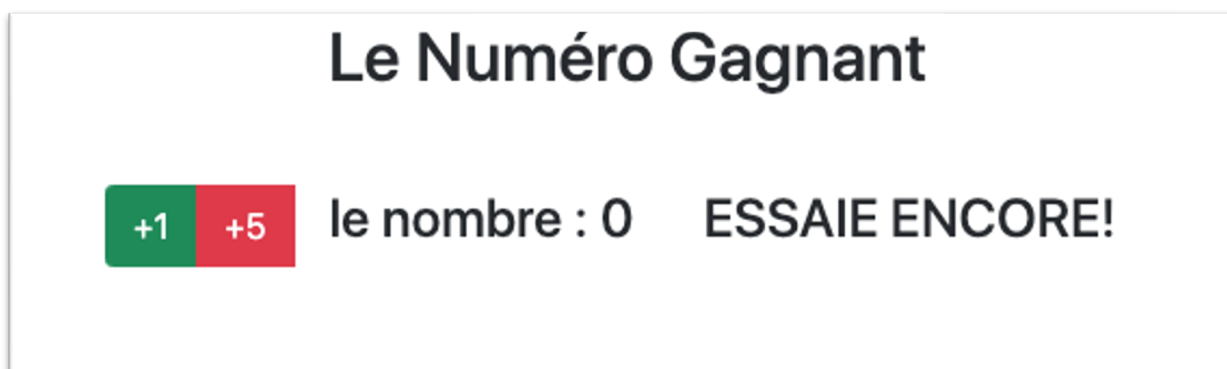
TP : Réactivité

Côté Template :

- Créer une Nouvelle application VUE dans laquelle on a une page qui affiche un système de compteur
- On aura 2 boutons qui permettront d'ajouter 5 et 1 à un nombre mystère
- Un paragraphe dans lequel on affichera le nombre mystère

Dans le <p>, tant que le nombre mystère n'est pas 7, on affiche « Essaie Encore », si le nombre mystère est égal à 7, on affiche le nombre mystère, si le nombre mystère dépasse 7, on affiche « Tu as dépassé le Nb Mystère »

Mettre en place un watcher sur le nombre affiché, au bout de 5 secondes, on le ré initialise à 0



[SOLUTION \(GitHub\)](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		