

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

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;
      }
    }
  });

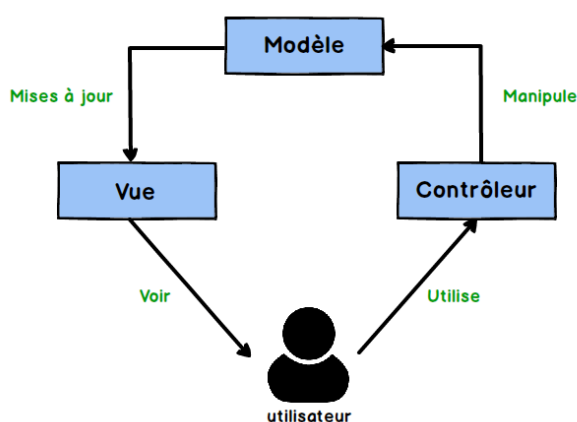
app.mount('#monApp');
```

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 : 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		

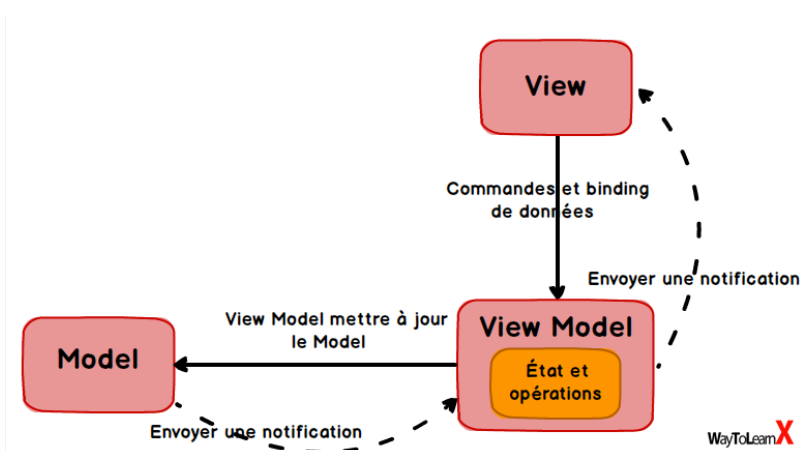
VUE

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 :
Mathieu Mithridate	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

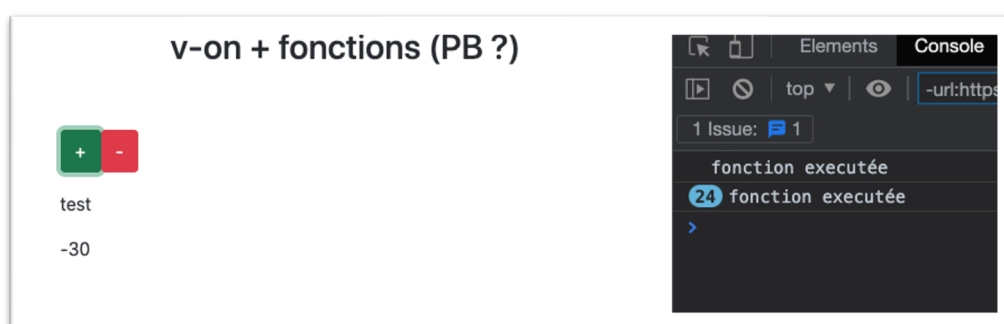
Utilisation des computed Properties

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):
 - Un console log qui affiche « fonction exécutée »
 - Si userName est vide alors on retourne « test »
 - 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 : 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

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

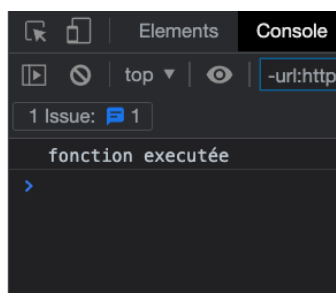
Quand une fonction est en mode computed, VueJS va gérer les dépendances (les variables) utilisées par cette fonction, c'est-à-dire que la fonction ne sera exécutée QUE s'il y a des modifications sur les variables dont elle dépend.

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

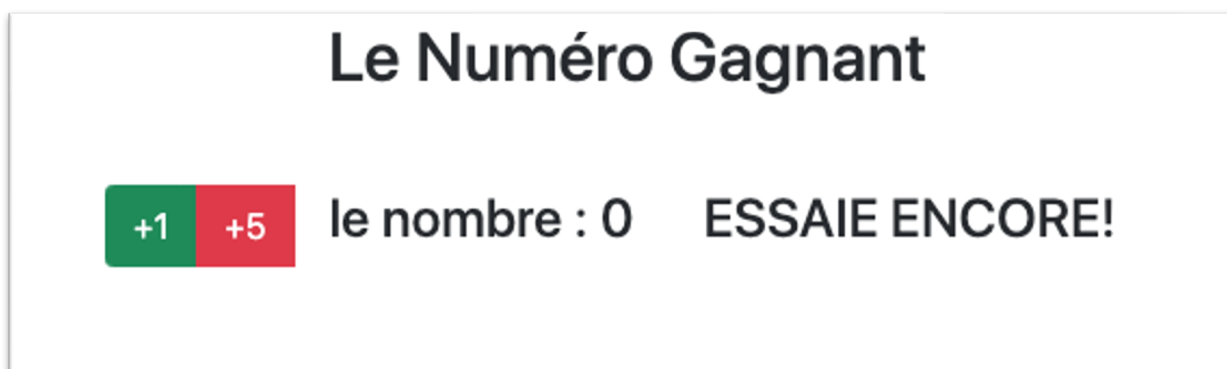
TP : Réactivité

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		

VUE

Gestion des styles dynamiquement

Une autre utilisation de la directive v-bind va permettre de gérer dynamiquement le style des éléments HTML. Techniquement tous les éléments HTML peuvent avoir un attribut style.

```
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>
```

Style Dynamique (inline)

On peut utiliser v-bind pour lier l'attribut style d'un élément avec une variable ou une expression JS.

v-bind:style="{nomPropriétéCSS: duCodeJS}"

Par exemple : on peut définir 2 data côté JS

```
data() {
  return {
    activeColor: 'red',
    myFont: 30
  }
}
```

Et les lier au style (plus précisément aux propriétés CSS color et fontSize).

```
<div :style="{ color: activeColor, fontSize: myFont + 'px' }"></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

Exercice : Style Dynamique (inline)

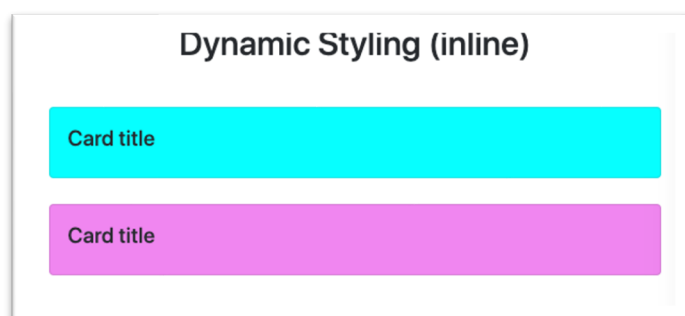
Dans une app Vue le click sur une div va appliquer un style :

Côté JS :

- On a 2 variables selectCard1 et selectCard2 par exemple, initialisées à false
- 1 fonction selectionCard qui prend en paramètre uneCard, dans laquelle on va faire :
 - o Si uneCard égal 1 on passe selectCard1 à true
 - o Si uneCard égal 2 on passe selectCard2 à true
 - o !!! Bonus : faire en sorte : quand on reclick sur la div cela rechange la couleur

Côté HTML :

- Mettre en place 2 div (type card, box, etc..)
- Sur la 1^{ère} div :
 - o on réagit au click et déclenche la fonction selectionCard en passant le chiffre 1 en paramètre
 - o avec un v-bind de style, modifier le backgroundColor, dans une condition ternaire :
 - si selectCard1 est défini on retourne 'red' sinon 'blue'
- Sur la 2^{ème} div
 - o on réagit au click et déclenche la fonction selectionCard en passant le chiffre 2 en paramètre
 - o avec un v-bind de style, modifier le backgroundColor, dans une condition ternaire :
 - si selectCard2 est défini on retourne 'green' sinon 'violet'



SOLUTION

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		