

## 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Table des matières

Objectifs pédagogiques :	5
Présentation et installation de VUE.....	6
Présentation de VUE :.....	6
Installation de VUE (mode CDN) :.....	9
Installation de VUE (mode CLI) .....	9
Création d'une première application vue (mode cdn) .....	12
Réalisation d'une application web TaskList en Javascript .....	12
Exercice : Créer une TaskList en JS.....	12
Application web TaskList en VUE.....	14
Notions / Concept de base de VUE.....	16
Connexion d'une App VUE avec un Template HTML et affichage de données .....	17
Exercice : Nouvelle App VUE qui affiche une variable .....	17
Exercice : Afficher des variables de différents type.....	18
Utilisation des methods.....	20
Exercice : Utilisation des methods.....	20
Affichage de contenu HTML (directive v-html).....	22
Lier Des attributs HTML (directive v-bind).....	23
TP : DataBinding .....	24
Réagir à des évènements DOM .....	25
Exercice : setUp Application pour EventBinding.....	25
Utilisation de la directive v-on + expressions JS .....	27
Utilisation de la directive v-on + fonction.....	28
Utilisation de la directive v-on + fonction avec des paramètres .....	29

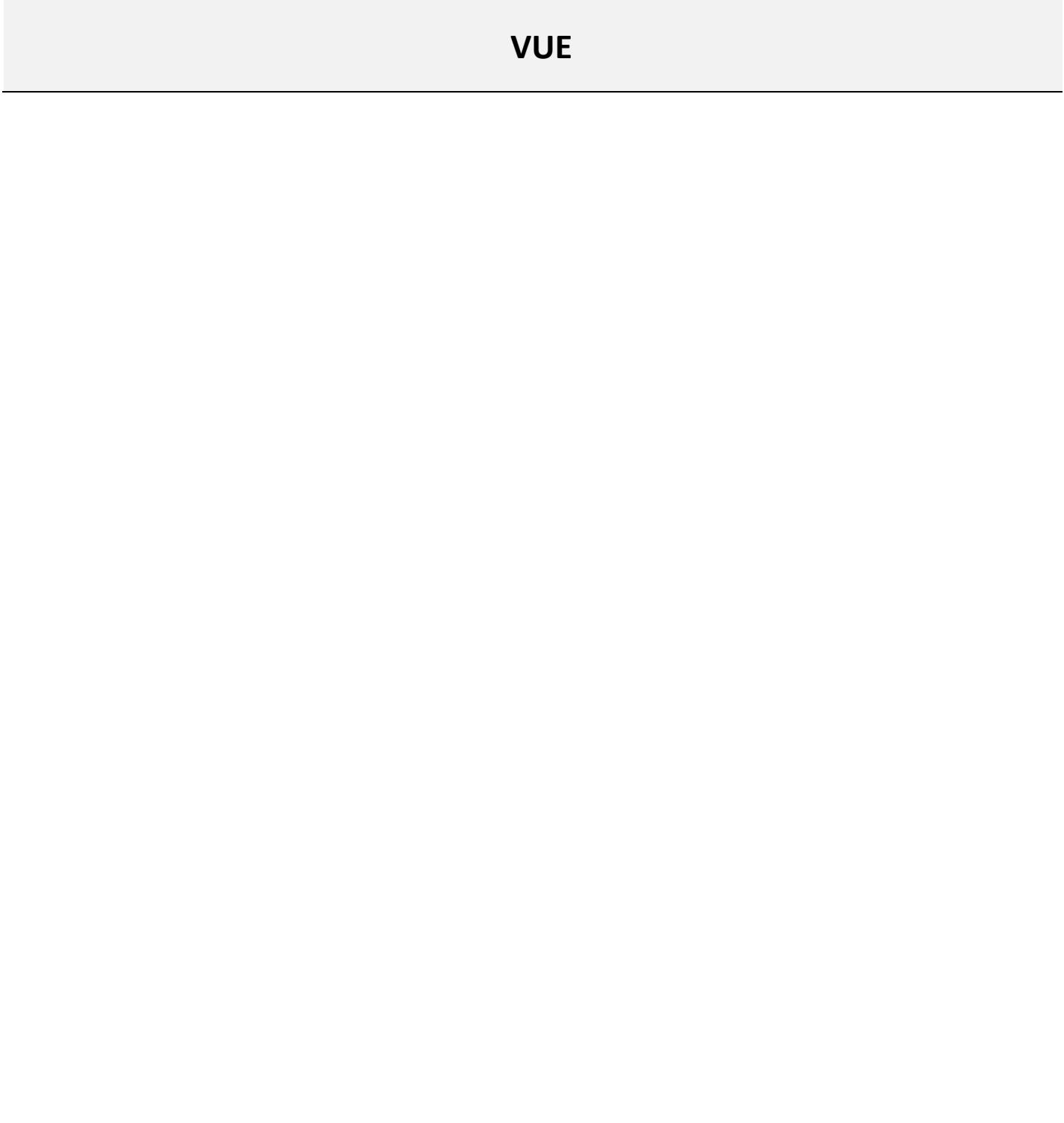
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux		Toute reproduction, représentation, diffusion ou rediffusion, totale ou 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) .....	30
Utilisation des Event Modifiers.....	32
TP: EventBinding.....	33
2 Way DataBinding (EventBinding + DataBinding) directive v-model .....	34
Exercice v-model.....	34
Design Pattern MVC vs MVVM .....	36
Utilisation des computed Properties .....	37
Exercice : Bug évènements + fonctions (interférences) .....	37
Utilisation des Watchers.....	41
TP : Réactivité .....	43
Gestion des styles dynamiquement .....	44
Style Dynamique (inline).....	44
Exercice : Style Dynamique (inline). ....	45
Style Dynamique (class CSS) .....	48
Exercice : Style Dynamique (class CSS) .....	48
TP : Style Dynamique .....	51
Conditional Rendering (directive v-if). ....	52
Exercice : Conditional Rendering .....	53
Directive v-show .....	56
List Rendering (Directive v-for).....	57
Exercice : List Rendering .....	58
Exercice : List Rendering + index .....	61
TP : Conditional / List Rendering .....	64
TP : Jeu Tour Par Tour .....	65

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



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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Objectifs pédagogiques :

Être capable de comprendre les enjeux des Frameworks

Être capable d'initialiser et paramétriser un projet grâce à un Frameworks front-end

Être capable de réaliser des interface utilisateur grâce à un Frameworks front-end

~~Être capable de mettre en place un système de Routing (url)~~

Être capable de gérer des requêtes HTTP (BDD)

Être capable de connecter une application à une base de données

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Présentation et installation de VUE

Présentation de VUE :



**Qu'est-ce que VUE ?**

VUE est un **Framework Javascript** permettant de construire des applications **Front End** interactives, et **réactives** plus facilement.

**Framework Javascript** : ensemble de librairie(s) tierce(s) qui met à disposition des utilitaires, des fonctionnalités et des bonnes pratiques sur comment construire des applications Javascript

**Javascript** : Un langage de programmation supporté par tous les navigateurs.

Permet de manipuler les éléments HTML, les données dans une page web.

**Front End** : « Ce que l'utilisateur voit » HTML + CSS + Javascript.

**Réactives** : L'App est capable de réagir aux actions de l'utilisateur et mettre à jour ce qui s'affiche à l'écran. ==> Le Look et le Feeling d'une appli mobile.

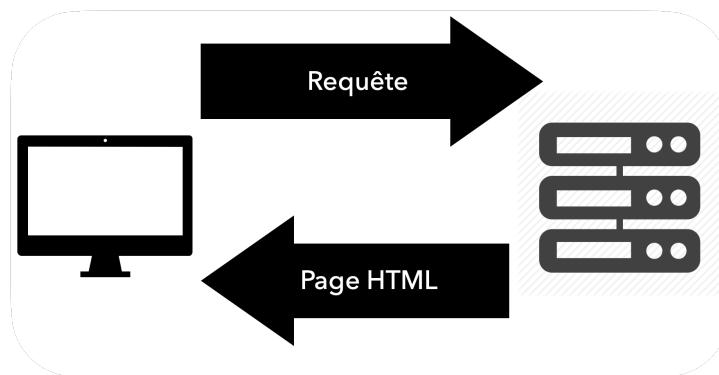
# « À la différence des autres Frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque est concentré uniquement sur la partie vue, et il est vraiment simple de l'intégrer avec d'autres bibliothèques ou projets existants. D'un autre côté, Vue est tout à fait capable de faire tourner des applications web monopages quand il est couplé avec des outils modernes et des bibliothèques complémentaires. »

Vue.js a été créé à l'origine par une seule personne, Evan You, ancien ingénieur de Google ayant notamment travaillé avec Angular.js puis sur le framework Meteor. Après avoir expérimenté une mécanique de réactivité qu'il jugeait plus intéressante que celle d'Angular, il décide de publier ses premiers résultats en juillet 2013. Cinq ans plus tard, Vue.js est le 3ème projet sur Github en nombre de stars et Evan enchaîne les conférences à travers le monde.

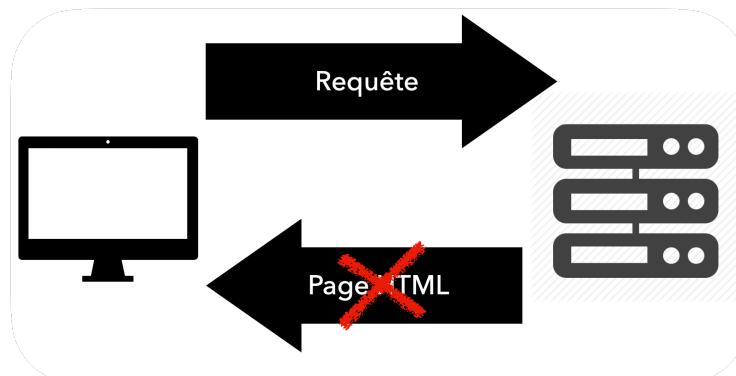
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale 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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Généralement, dans des applications web « classiques », les interactions avec le serveur (click sur un bouton pour envoyer un message par exemple) entraînent un rechargement de la page.



L'arrivée de Javascript a permis d'éviter ces rechargements de pages, dans le sens où JS s'exécute dans le navigateur sur la page qui est chargée, JS va pouvoir manipuler les éléments du DOM de la page.



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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Sur l'aspect technique, il est tout à fait possible de construire n'importe quelle application web en se basant sur du Javascript, sans recourir à l'utilisation d'un Framework (Angular, Vue, React, Svelte ...)

Mais :

- Nous devrons écrire tout le code de l'application (On réinvente la roue ?)
- Risque de réaliser un code non optimisé comprenant des erreurs, des bugs (difficilement maintenable / scalable).
- Travailler en équipe sera plus difficile, tout le monde ne connaît pas votre manière de coder.

L'utilisation d'un Framework permet de répondre à ces problématiques, il propose un ensemble de composants réutilisables, un cadre de travail avec des règles à respecter ainsi que de bonnes pratiques.

VUE adopte une approche différente des autres frameworks (monolithique), Vue se définit comme un framework progressif, on peut choisir d'utiliser de deux manières :

L'approche plutôt en mode widget (maquette gauche) (Vue contrôle 1 ou plusieurs parties spécifiques d'une application), cela permet de facilement intégrer et utiliser vue dans des projets existants.

L'approche en mode « monolithique » (maquette droite), on gère toute une application en VUE



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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Installation de VUE (mode CDN) :

En mode CDN, nous n'allons pas Installer VUE, nous allons directement l'utiliser depuis son server d'hébergement.

<https://vuejs.org/guide/quick-start.html#without-build-tools>

Il suffira d'intégrer cette balise dans nos pages HTML :

```
<script src="https://unpkg.com/vue@3"></script>
```

### Installation de VUE (mode CLI)

(Nous mettrons en place cette technique dans la seconde partie du cours)

En mode CLI (Command Line Interface), nous pouvons créer des projets via des lignes de commande (terminal) entièrement gérés en VUE.

Cette méthode d'installation passe par des outils de gestion de package (il en existe plusieurs mais le plus commun est l'outil NPM qui fait partie de NodeJS)

<https://nodejs.org/en/>

Une fois NodeJS installé vous pouvez utiliser son outil de gestionnaire de package NPM pour installer dans vos projets tout type de librairies, Framework... et notamment VUE CLI

Dans un terminal, la commande ci-dessous va installer VUE CLI sur votre machine :

- npm install -g @vue/cli

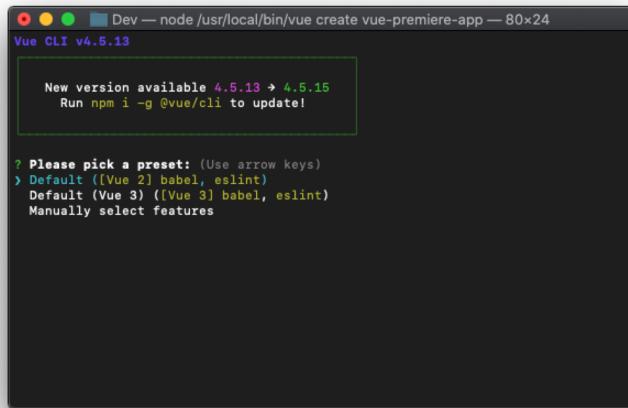
Le « -g » va permettre d'installer VUE CLI globalement sur votre machine vous pourrez par la suite utiliser des commandes pour créer des projets VUE dans n'importe quel dossier de votre machine.

Pour créer et initialiser un nouveau projet VUE :

- Se déplacer dans le terminal dans un dossier (avec la commande cd)
- Et créer une nouvelle application avec la commande :  
vue create leNomDeVotreApplication

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE



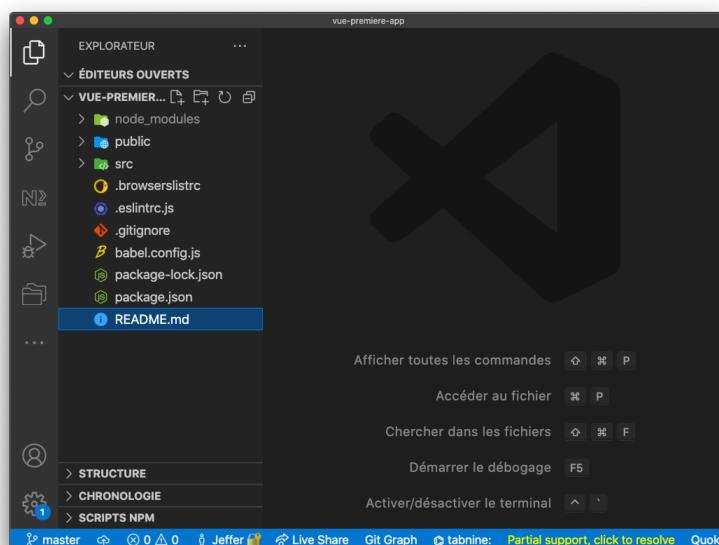
```
Vue CLI v4.5.13

New version available 4.5.13 → 4.5.15
Run npm i -g @vue/cli to update!

? Please pick a preset: (Use arrow keys)
> Default ([Vue 2] babel, eslint)
Default (Vue 3) ([Vue 3] babel, eslint)
Manually select features
```

Le CLI de VUE permet de créer et initialiser directement tout un projet, nous pouvons choisir entre Vue2, Vue3 (La dernière version de Vue que nous utiliserons) et la possibilité de configurer manuellement un projet.

Le CLI va donc créer un nouveau dossier pour votre projet (que vous pouvez ouvrir dans VSCode) avec tout un ensemble de fichiers de base.



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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Dossier node\_modules : c'est dans ce dossier que sont installées toutes les packages, dépendances du projet (exemple si l'on choisit d'installer Bootstrap dans notre projet, les fichiers de Bootstrap se retrouveront dans node\_modules)

Dossier public : ou l'on retrouve la page index.html, la seule page de l'application (qui affichera un composant).

Dossier src : c'est dans ce dossier que l'on développera nos composants, de base toute l'application vue possède un composant racine le fichier App.vue, ainsi qu'un autre composant HelloWorld.vue

Fichier eslint : le fichier de configuration de l'analyseur de code

Fichier babel : le fichier de configuration de Babel (un compilateur JS), son rôle est de traduire des fichiers .vue en JavaScript interprétable directement par les navigateurs

Fichier package.json : le fichier qui va lister les dépendances du projet.

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

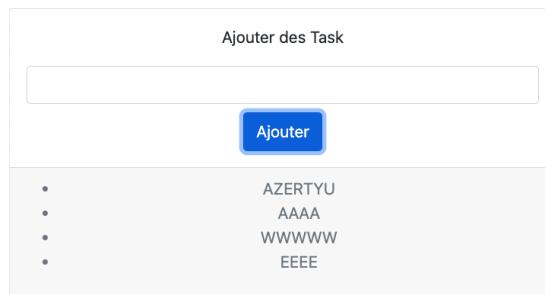
### Création d'une première application vue (mode cdn)

Pour illustrer plus explicitement l'avantage d'utiliser un Framework nous allons créer une application en Javascript (classique), puis nous allons voir les différences avec cette même application en VUE.

#### Réalisation d'une application web TaskList en Javascript

Exercice : Créer une TaskList en JS

**Task List (JS)**



Ajouter des Task

Ajouter

- AZERTYU
- AAAA
- WWWWW
- EEEE

Créer une page HTML (liée à un fichier JS) qui comporte :

- 1 input
- 1 button
- 1 <ul>

En JS :

- 1 variable leInput récupère l'élément input
- 1 variable leButton récupère l'élément button
- 1 variable laList récupère l'élément ul
- 1 fonction ajouterTask dans laquelle :
  - o on créer un élément <li>
  - o on assigne à la propriété.textContent (ou innerText) du <li>, la value de leInput
  - o on place le <li> dans laListe avec la fonction append()

[SOLUTION taskListJS \(StackBlitz\)](#)

[SOLUTION taskListJS \(GitHub\)](#)

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Code JS de l'application :

```
//On stock les éléments HTML dans des variables
const leButton = document.querySelector('button');
const leinput = document.querySelector('input');
const laListe = document.querySelector('ul');

function ajouterTask(){
    //Dans inputContent on stock ce que l'utilisateur tape dans l'input
    const inputContent = leinput.value;
    //Création d'un élément <li></li>
    const listElement = document.createElement('li');
    //Complétion du texte de l'élément <li></li>
    listElement.textContent = inputContent;
    //Ajout du <li></li> dans la <ul></ul> (laListe)
    laListe.append(listElement);
    //Bonus : après avoir ajouté une Task, on vide l'input
    leinput.value = '';
}

leButton.addEventListener('click', ajouterTask);
```

Avec Javascript, nous pouvons le constater, il est nécessaire de prendre le contrôle de tous les éléments du DOM.

L'utilisation de Vue va simplifier la syntaxe JS en permettant d'avoir des interactions directement dans la partie Template (côté 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Application web TaskList en VUE

#### Fichier JavaScript (mode VUE)

```
//Création d'une instance Vue c'est notre application
Vue.createApp({
//Dans une fonction data on va retourner un objet qui contiendra les variables de
l'application
//Toutes les data seront contenues dans this
  data(){
    return {
      tasks: [],
      valeurDeInput:''
    }
  },
// Dans cet Objet methods, on va écrire nos fonctions
  methods:{
    ajouterTask(){
      this.tasks.push(this.valeurDeInput);
      this.valeurDeInput = '';
    }
  }
// L'application est placée sur la balise HTML qui possède l'id app
}).mount('#app');
```

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Page HTML (body)

```

16  <!-- On place un id sur la partie du template que l'on veut contrôler avec VUE (ici vue a le contrôle sur le body) -->
17  <body id="app">
18    <h1 class="text-center">Task List (VUE)</h1>
19    <div class="card text-center">
20      <div class="card-body">
21        <p class="card-text">Ajouter des Task</p>
22        <!-- la directive v-model, sur l'input va permettre de le connecter à une variable JS -->
23        <input v-model="valeurDeInput" class="form-control" type="text" />
24        <!-- la directive v-on:click, sur le button permet de réagir au click et exécuter la fonction ajouterTask côté JS -->
25        <button v-on:click="ajouterTask" type="button" class="btn btn-primary mt-2">Ajouter</button>
26      </div>
27      <div class="card-footer text-muted">
28        <ul>
29          <!-- la directive v-for, sur la <li> va permettre de répéter l'affichage de <li>
30          <!-- autant de fois que des données sont présentes dans le tableau tasks (côté JS) -->
31          <!-- cette syntaxe en double accolades {{uneTask}} permet d'afficher une variable(JS) directement -->
32          <li v-for="uneTask in tasks">{{uneTask}}</li>
33        </ul>
34      </div>
35    </div>
36    <!-- script pour contacter VUE -->
37    <script src="https://unpkg.com/vue@next"></script>
38    <script src="taskList.vue.js"></script>
39    <!-- Option 1: Bootstrap Bundle with Popper -->
40    <script
41      src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
42      integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
43      crossorigin="anonymous"
44    ></script>
45  </body>
```

Via son système de directives (v-on, v-model, v-for, v-if) Vue permet une interaction plus directe entre les variables et fonctions du fichier JS et le Template 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Notions / Concept de base de VUE

L'exemple précédent (application taskList) nous donne un aperçu des principaux concepts et règles de VUE.

Côté Javascript, une application VUE comportera **toujours** des fonctions et objets de base :

- La fonction **createApp()** : va permettre de créer l'application VUE.
- La fonction **data()** : qui retournera toujours un objet qui dans lequel on pourra définir les variables de l'application
- L'objet **methods : {}** : dans lequel on va écrire les fonctions de l'application
- La fonction **mount()** : qui va permettre de placer notre application sur une balise du Template html.

Côté HTML, VUE nous permet d'utiliser le système de directive (v-on, v-model, v-for ...), utilisables comme des attribut HTML ainsi qu'une syntaxe pour afficher directement des données {{leNomDuneVariable}}.

VUE facilite donc le développement d'application Web en proposant une interaction enrichie côté Template ce qui simplifie le code Javascript.

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Connexion d'une App VUE avec un Template HTML et affichage de données

Exercice : Nouvelle App VUE qui affiche une variable

En se basant sur l'exemple de l'application taskList, créer une nouvelle application VUE (1 page HTML + 1 fichier JS).

Dans le fichier JS : en respectant la structure d'une application VUE, définir une variable (un nombre ou une chaîne de caractères).

Dans le fichier HTML : en utilisant la syntaxe {{ }} afficher cette variable.

**SOLUTION BookList(StackBlitz)**

[SOLUTION BookList\(GitHub\)](#)

Fichier JS :

```
//Création d'une instance Vue c'est notre application
Vue.createApp({
//Dans une fonction data on va retourner un objet qui contiendra les variables de
l'application
  data(){
    return {
      unLivre : "Les mémoires de Steven Seagal"
    }
  }
// L'application est montée sur la balise HTML qui possède l'id bookListApp
}).mount('#bookListApp');
```

Fichier HTML :

```
<h1 class="text-center">■ Book List</h1>
<!-- Ici la donnée unLivre ne s'affiche pas (Vue n'a pas le ctrl ici) -->
<p class="text-center">{{unLivre}}</p>
<hr>
<!-- Ici Vue a le ctrl sur toute la card qui a cet id -->
<div id="bookListApp" class="card">
  <p class="text-center">{{unLivre}}</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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Exercice : Afficher des variables de différents type

Dans cette même application VUE, rajouter des variables de différents type et les afficher côté Template Html.

[SOLUTION BookList\(1.1\)\(StackBlitz\)](#)

[SOLUTION BookList\(1.1\)\(GitHub\)](#)

Fichier JS

```
//Création d'une instance Vue c'est notre application
Vue.createApp({
//Dans une fonction data on va retourner un objet qui contiendra les variables de
l'application
  data(){
    return {
      unLivre : "Les mémoires de Steven Seagal",
      unTableau : ['du texte', 99],
      unNombre : 1234567890,
      unObjet : {
        name:'COOL',
        tel: 060503030
      }
    }
  }
// L'application est montée sur la balise HTML qui possède l'id bookListApp
}).mount('#bookListApp');
```

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier HTML:

```
<h1 class="text-center">■ Book List</h1>
    <!-- Ici la donnée unLivre ne s'affiche pas (Vue n'a pas le ctrl ici) -->
    <p class="text-center">{{unLivre}}</p>
    <hr>
    <div id="bookListApp" class="card">
        <p class="text-center">{{unLivre}}</p>
        <p class="text-center">{{unNombre}}</p>
        <p class="text-center">{{unTableau}}</p>
        <p class="text-center">{{unObjet}}</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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Utilisation des methods

Comme nous l'avons abordé dans le projet TaskList, nous pouvons définir des fonctions pour notre application, elles seront écrites au sein de l'objet methods : {}, nous pouvons bien entendu utiliser plusieurs methods dans une application vue.

Il est également possible d'exécuter des methods directement dans le Template HTML, en utilisant la syntaxe {{ nomDeMaFonction() }}.

Nous allons mettre en pratique l'utilisation de methods au travers d'un exercice.

#### Exercice : Utilisation des methods

En se basant sur le projet TaskList, dans une application VUE :

Côté JS :

- Dans methods : {}, rajouter une fonction dans laquelle :
  - o Créer une variable qui est un nombre random (en utilisant Math.random())
  - o Si ce nombre random est supérieur à 0 .5, on retourne une chaîne de caractères (au choix)
  - o Sinon on retourne une autre chaîne de caractères (au choix)

Côté HTML :

- En utilisant la syntaxe {{ }}, exécuter la fonction au sein d'un paragraphe

[SOLUTION BookListV1.2\(GitHub\)](#)

[SOLUTION BookListV1.2\(StackBlitz\)](#)

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      unLivre : "Les mémoires de Steven Seagal"
    }
  },
  methods: {
    afficherLivreRandom (){
      const nombreRandom = Math.random();
      if(nombreRandom > 0.5){
        return "Le livre de La Jungle"
      }
      else{
        return "Le Livre de bobba Fette"
      }
    }
  },
}).mount('#bookListApp');
```

Fichier HTML :

```
<h1 class="text-center">Utilisation des methods</h1>
<hr>
<div id="bookListApp" class="card">
  <p class="text-center">{{afficherLivreRandom()}}</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		<i>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.</i>
	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		<i>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.</i>
	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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

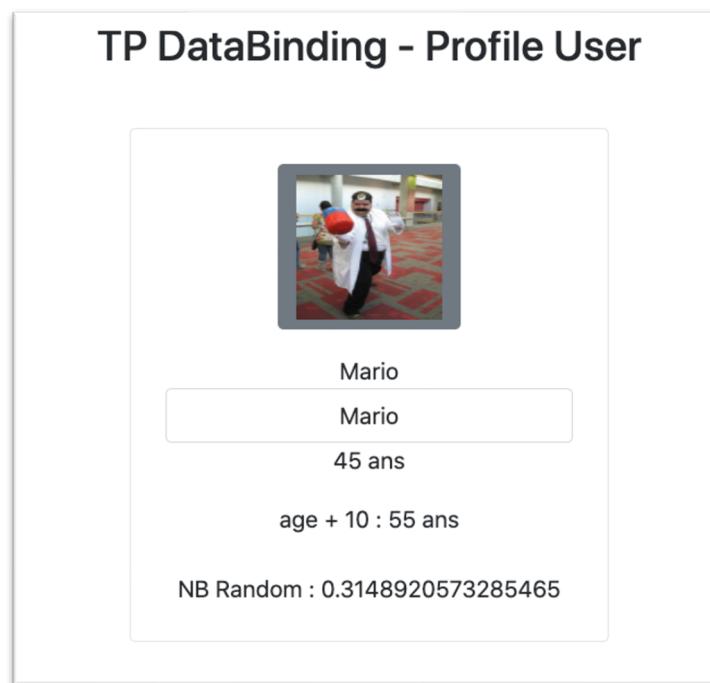
### TP : DataBinding

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



[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Réagir à des événements 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 événements du DOM (le click sur un bouton, le clavier, la souris de l'utilisateur...).

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

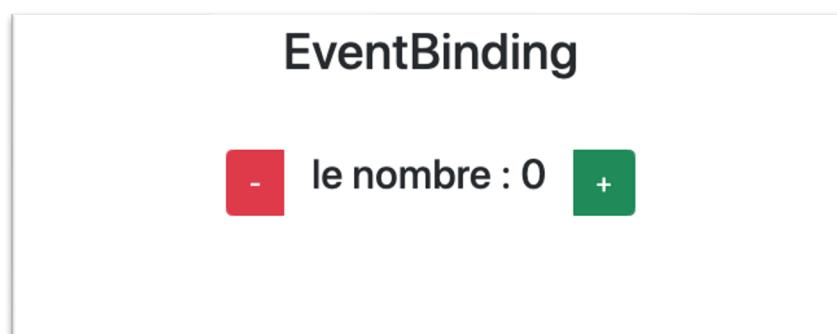
Afin d'illustrer l'utilisation de cette directive nous allons mettre en place une application qui augmente ou réduit un nombre.

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 : set up 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		<i>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.</i>
	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		<i>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.</i>
	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		<i>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.</i>
	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>
    <h3 class="mx-3">le nombre : {{leNombre}}</h3>
    <button v-on:click="augmenter()" 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		<i>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.</i>
	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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

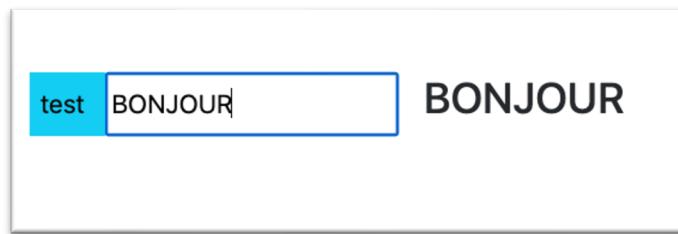
## VUE

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 capturer l'évènement click, le second capturer 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 : Coordinatrice Filière Syst. & Réseaux		<i>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.</i>
	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 évènements 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 value contenu dans l'évènement input pour l'assigner à la variable uneString.

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

<pre>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</pre>	<pre>bookList(1.6).js:22 InputEvent {isTrusted: true, data: 'd', isCo- mposing: false, inputType: 'insertText', dat- aTransfer: null, ...} ⓘ   isTrusted: true   bubbles: true   cancelBubble: false   cancelable: false   composed: true   currentTarget: null</pre>
--	---

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

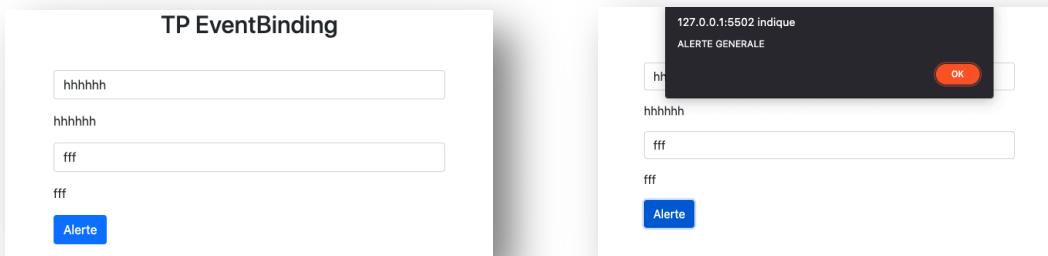
## VUE

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'event keyup pour afficher ce qui est tapé au clavier dans un paragraphe
- 1 input sur lequel on va réagir à l'event 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.



[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		<i>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.</i>
	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 évènements, 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		<i>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.</i>
	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;
    }
  }
});

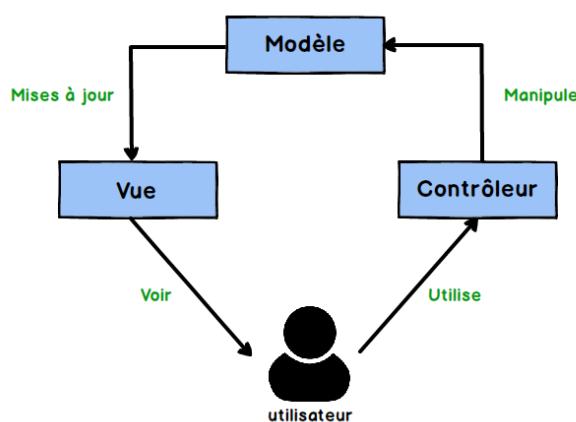
app.mount('#monApp');
```

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

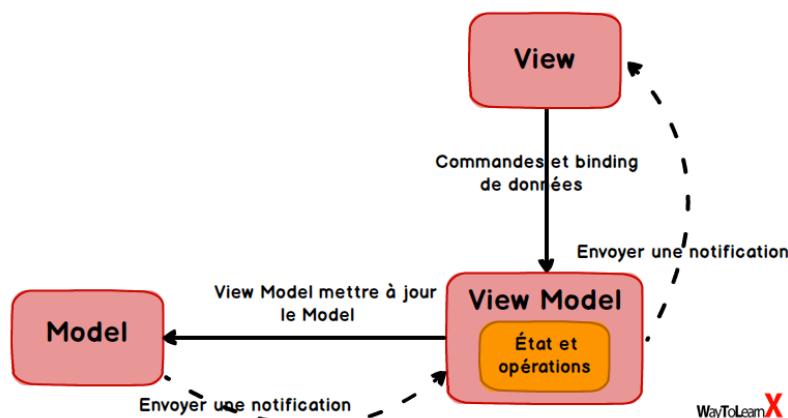
## VUE

### 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

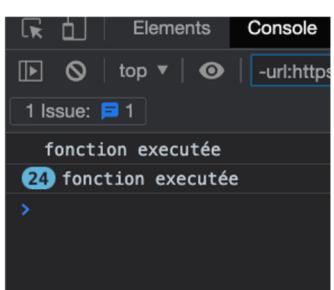
Au cours du développement d'une application VueJS, (plus complète), nous allons gérer plusieurs fonctions, plusieurs données, on va très vite se retrouver dans des situations où sur la même page nous devons réagir à plusieurs évènements et exécuter des fonctions, cela va engendrer des interférences entre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

#### Exercice : Bug évènements + 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 »

**v-on + fonctions (PB ?)**



The screenshot shows a browser's developer tools console with the following output:

```

Elements Console
top ▾ 1 issue: 1
fonction executée
24 fonction executée
>

```

To the left, there is a small interface with a green '+' button and a red '-' button, and some text: "test" and "-30".

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		<i>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.</i>
	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		<i>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.</i>
	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>
```

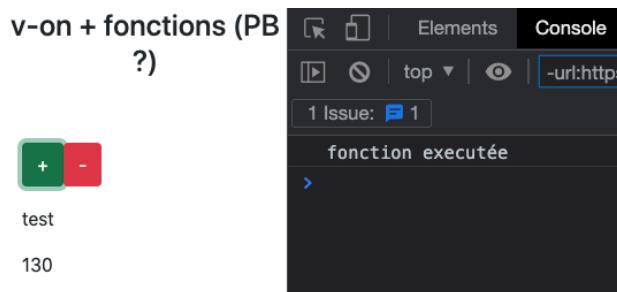
Modification côté JS : on déplace la fonction dans un objet computed (même utilisation que methods)

```
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		<i>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.</i>
	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.



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		<i>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.</i>
	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		<i>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.</i>
	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

# Le Numéro Gagnant

+1
+5
le nombre : 0    **ESSAIE ENCORE!**

[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		<i>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.</i>
	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'attributs 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Exercice : Style Dynamique (inline)

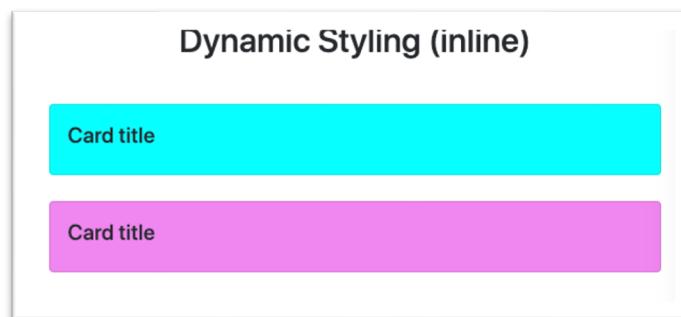
Dans une app Vue le click sur une div va appliquer un style :

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 recharge la couleur

Côté HTML :

- Mettre en place 2 div (type card, box,etc..)
- Sur la 1<sup>ère</sup> 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<sup>ème</sup> 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier HTML :

```
<div id="monApp">
  <div class="row row-cols-1 row-cols-md-3 g-4 m-3">
    <div class="col">
      <div class="card" @click="selectionCard(1)"
           :style="{backgroundColor: selectCard1 ? 'red':'aqua' }">
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card" @click="selectionCard(2)"
           :style="{backgroundColor: selectCard2 ? 'green':'violet' }">
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
        </div>
      </div>
    </div>
  </div>
</div>
```

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return { selectCard1:false,selectCard2:false };
  },
  methods: {
    selectionCard(uneCard){
      if(uneCard ===1){
        this.selectCard1=true;
      }
      if(uneCard ===2){
        this.selectCard2=true;
      }
    }
  },);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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

D'autres cas d'utilisation de v-bind + style, on peut relier à l'attribut style d'un élément HTML, un objet JS contenant des propriétés CSS (c'est une utilisation très similaire à du CSS au final, à voir...)

Petit Rappel : En JS on manipule des propriété CSS avec la notation CamelCase tandis qu'en CSS normal on utilise la notation kebab-case (font-size en CSS vs fontSize en JS)

```
data() {
  return {
    styleObject: {
      color: 'red',
      fontSize: '13px'
    }
  }
}
```

```
<div :style="styleObject">Cool</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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Style Dynamique (class CSS)

Si nous avons beaucoup de modification CSS à appliquer dynamiquement sur nos éléments, il sera plus pratique de passer par des classes que nous définissons au préalable en CSS qui seront activées ou non (true, false)

**v-bind:class="{nomDeMaClasse: duCodeJS(true / false), nomDeMaSecondeClasse :false}"**

```
<div v-bind:class="{nomDeMaClasse: true, nomDeMaSecondeClasse :false}"></div>
```

### Exercice : Style Dynamique (class CSS)

Dans une app Vue, mettre en place 2 div (type card, box,etc..) :

- Quand on click sur la 1<sup>ère</sup> div cela applique une class CSS si une variable selectCard1 est définie
- Quand on click sur la 2<sup>ère</sup> div cela applique une autre class CSS si une variable selectCard2 est définie

### Dynamic Styling (class)



[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier CSS ou balise <style> :

```
<style>
    .class1{
        background-color: mediumslateblue;
    }
    .class2{
        background-color: chartreuse;
    }
</style>
```

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return {
      selectCard1:false,
      selectCard2:false
    };
  },
  methods: {
    selectionCard(uneCard){
      if(uneCard ===1){
        // this.selectCard1=true;
        this.selectCard1=!this.selectCard1;

      }
      if(uneCard ===2){
        // this.selectCard2=true;
        this.selectCard2=!this.selectCard2;

      }
    }
  },
  });
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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier HTML :

```
<div id="monApp">
  <div class="row row-cols-1 row-cols-md-3 g-4 m-3">
    <div class="col">
      <div class="card" @click="selectionCard(1)"
           :class="{class1: selectCard1}">
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card" @click="selectionCard(2)"
           :class="{ 'class2': selectCard2}">
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
        </div>
      </div>
    </div>
  </div>
</div>
```

Autre alternative utilisant un array, dans le cas où l'on veut v-bind l'attribut classe en combinant une classe statique (une classe que nous avons déjà défini en CSS) et la modification dynamique d'une propriété de style.

Exemple : ou une classe CSS « maClasse » serait déjà dans un fichier CSS.

```
<div :class="['maClasse', {backgroundColor:selectCard1}]"
@click="selectionCard(1)"></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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

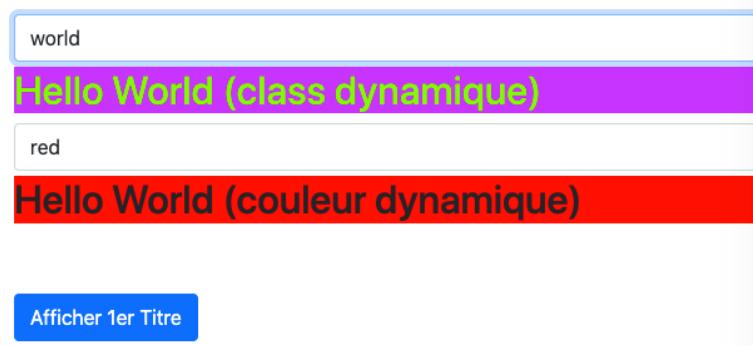
## VUE

### TP : Style Dynamique

Créer une application Vue dans laquelle on a 2 inputs, 2 paragraphes (ou titre etc ...), 1 bouton

- Le bouton doit permettre d'afficher ou non le 1 er paragraphe (ou titre)
- Dans le 1<sup>er</sup> input si on tape « hello » cela applique une classe « maClasseHello » en CSS, si on tape « world » cela applique une classe « maClasseWorld » en CSS
- Sur le 2<sup>er</sup> input on doit pouvoir taper une couleur (red, green, yellow etc...) et cette couleur doit s'appliquer au background color du second paragraphe (ou titre).

### TP:Dynamic Styling



The screenshot shows a Vue.js application interface. At the top, there are two input fields: one containing "world" and another containing "red". Below these inputs are two paragraphs. The first paragraph, which contains the text "Hello World (class dynamique)", is styled with a purple background. The second paragraph, which contains the text "Hello World (couleur dynamique)", is styled with a red background. At the bottom of the interface is a blue button labeled "Afficher 1er Titre".

[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Conditional Rendering (directive v-if)

La directive v-if peut se définir comme étant une directive structurelle, c'est-à-dire que son utilisation va modifier la structure du DOM, nous allons pouvoir afficher ou non n'importe quel élément HTML, selon une expression JS (variable, fonction, condition...) qui renvoi true ou false.

Exemple : ici sur ce h1 le v-if vérifie si une variable « userConnected » est définie, si c'est le cas le titre est affiché sur la page, si la variable n'est pas définie (ou à false) le titre ne sera pas affiché.

```
<h1 v-if="userConnected">Bienvenue cher utilisateur</h1>
```

Dans le cas où une condition v-if renverrai false, donc l'élément HTML ne serait pas affiché, nous pouvons utiliser en combinaison de v-if, la directive v-else pour afficher un autre élément :

Si la variable userConnected est définie (renvoi true), on affiche « bienvenue cher utilisateur » sinon on affiche « oh non »

```
<h1 v-if="userConnected">Bienvenue cher utilisateur</h1>
<h1 v-else>Oh non 😢</h1>
```

Pour aller plus loin il existe aussi la directive v-else-if :

Ici on affiche 3 div en fonction de la variable typeUser Sinon une div « veuillez-vous connecter »

```
<div v-if="typeUser === 'admin'>
  Bienvenue Admin
</div>
<div v-else-if="typeUser === 'visiteur'>
  Bienvenue visiteur
</div>
<div v-else-if="typeUser === 'premium'>
  Bienvenue cher membre premium
</div>
<div v-else>
  Veuillez vous connecter
</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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Exercice : Conditional Rendering

Mettre en place une application WatchList (une liste de films), dans laquelle :

On va gérer un tableau mesFilms (initialisé à vide) et une chaîne de caractères nomFilm.

1 input doit permettre d'ajouter 1 film au tableau mesFilms

1 titre qui affiche nomFilm , ce titre ne doit s'afficher QUE si la longueur de notre tableau est ≠ de 0

Sinon on affiche un paragraphe « Pas encore de Films dans votre Liste »

Bonus : Essayer d'inspecter le paragraphe dans le navigateur quand on ajoute un film

**Conditional Rendering**

WatchList 🎬 (Films à voir)

**Ajouter à votre Liste**

Pas Encore de Films dans votre Liste ?

**Conditional Rendering**

WatchList 🎬 (Films à voir)

**Ajouter à votre Liste**

**Mission Alcatraz**

[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return {
      nomFilm:'',
      mesFilms:[]
    };
  },
  methods: {
    ajouterFilm(){
      this.mesFilms.push(this.nomFilm);
    }
  },
});
app.mount('#monApp');
```

Fichier HTML :

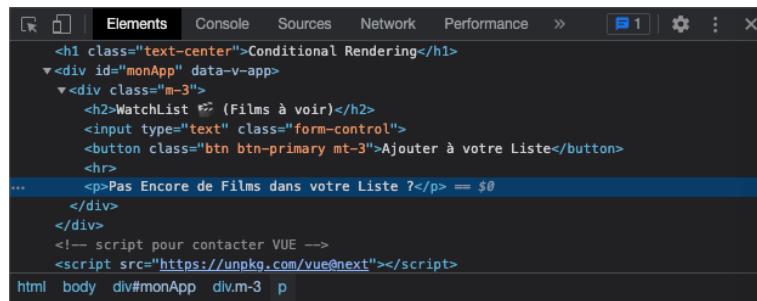
```
<h1 class="text-center">Conditional Rendering</h1>
<div id="monApp">
  <div class="m-3">
    <h2>WatchList 

```

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

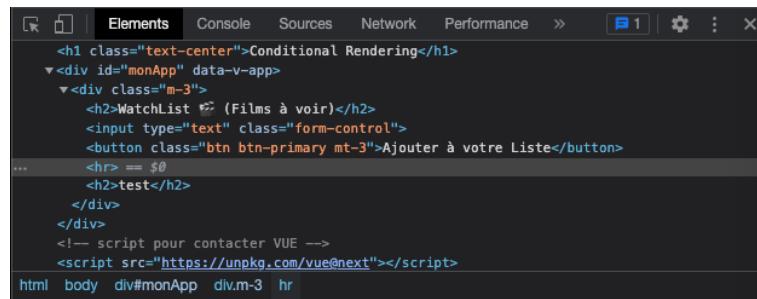
## VUE

Si l'on observe bien avec l'inspecteur d'éléments du navigateur :



```
<h1 class="text-center">Conditional Rendering</h1>
<div id="monApp" data-v-app>
  <div class="m-3">
    <h2>WatchList (Films à voir)</h2>
    <input type="text" class="form-control">
    <button class="btn btn-primary mt-3">Ajouter à votre Liste</button>
    <hr>
    ... == $0
  </div>
</div>
<!-- script pour contacter VUE -->
<script src="https://unpkg.com/vue@next"></script>
```

Après avoir ajouté quelque chose dans le tableau :



```
<h1 class="text-center">Conditional Rendering</h1>
<div id="monApp" data-v-app>
  <div class="m-3">
    <h2>WatchList (Films à voir)</h2>
    <input type="text" class="form-control">
    <button class="btn btn-primary mt-3">Ajouter à votre Liste</button>
    <hr> == $0
    <h2>test</h2>
  </div>
</div>
<!-- script pour contacter VUE -->
<script src="https://unpkg.com/vue@next"></script>
```

On constate que le DOM se remet à jour en temps réel et que l'élément paragraphe est bien Supprimé du 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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

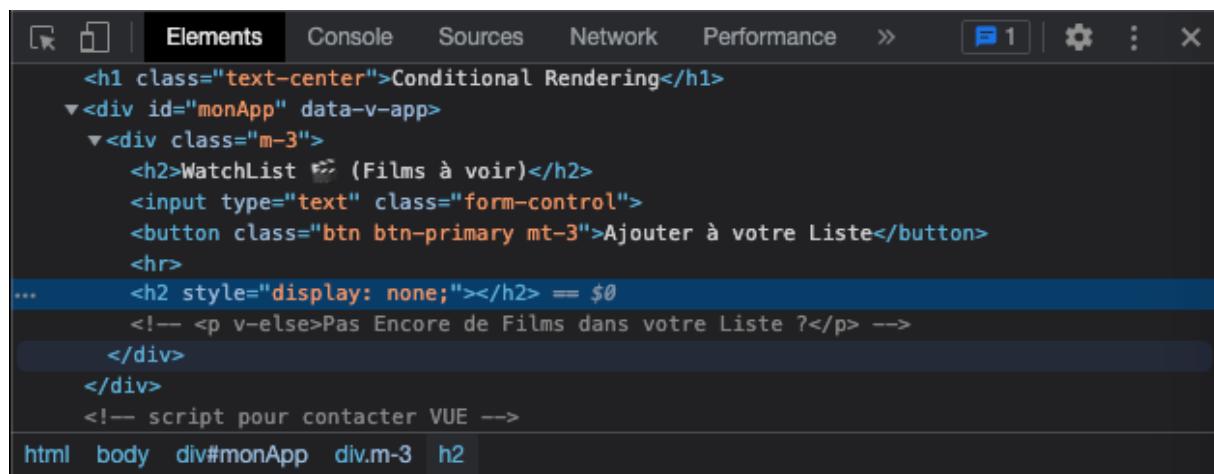
### Directive v-show

Pour de plus rares cas d'utilisations, la directive v-show va s'utiliser comme v-if (sauf qu'on ne peut pas la combiner avec v-else). L'objectif est le même c'est-à-dire d'afficher ou non un élément HTML, SAUF QUE, avec v-show cet élément sera toujours présent dans le DOM (en mode display :none).

Exemple en reprenant l'exercice précédent :

```
<div id="monApp">
    <div class="m-3">
        <h2>WatchList 📽️ (Films à voir)</h2>
        <input v-model="nomFilm" type="text" class="form-control">
        <button @click="ajouterFilm()" class="btn btn-primary mt-3">Ajouter à votre
Liste</button>
        <hr>
        <h2 v-show="mesFilms.length !== 0">{{nomFilm}}</h2>
        <!-- <p v-else>Pas Encore de Films dans votre Liste ?</p> -->
    </div>
</div>
```

Dans l'inspecteur d'éléments du navigateur :



Nous allons privilégier l'utilisation de v-if la plupart du temps.

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### List Rendering (Directive v-for)

Autre directive structurelle (qui va pouvoir modifier la structure du DOM), la directive v-for, qui en combinaison avec toutes les directives vues précédemment (v-bind, v-on, v-if) va permettre d'entièrement automatiser l'affichage de Template HTML avec des données dynamiques.

La directive v-for va s'utiliser afin de répéter l'affichage d'un élément HTML (paragraphe, titre, liste, une div, tout un Template, etc....).

Exemple : Ici côté JS on aurait déjà défini une variable « maListeDeDonnées » qui serait un tableau par exemple : de la même manière que for ... in ou for ... of en JS, on va pouvoir définir une variable temporaire pour parcourir cette liste de données.

```
<p v-for="uneVarTemporaire in maListeDeDonnées">{{uneVarTemporaire}} </p>
```

On retrouve également la notion de scope (la portée de la variable), puisque la variable temporaire ne sera utilisable qu'à l'intérieur du v-for.

```
<div v-for="uneVarTemporaire in maListeDeDonnées">
  <p>{{uneVarTemporaire}}</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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Exercice : List Rendering

En se basant sur l'exercice précédent, mettre en place une application VueJS, dans laquelle

1 input permet d'ajouter quelque chose (le nom d'un film) à une liste (mes films à voir)

L'objectif est de répéter l'affichage des données contenues dans la liste.

#### List Rendering

WatchList 📽 (Films à voir)

Ajouter à votre Liste

Aucun Films à Voir ?

On ajoute des films à la liste des films à voir,  
on répète l'affichage (ici des éléments de liste) des films que l'on ajoute

#### List Rendering

WatchList 📽 (Films à voir)

Ajouter à votre Liste

- Mission Alcatraz
- Piège à grande vitesse
- Piège en haute mer
- Avatar 3

[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier JS :

```
const app = Vue.createApp({
  data() {
    return {
      nomFilm:'',
      mesFilms:[]
    };
  },
  methods: {
    ajouterFilm(){
      this.mesFilms.push(this.nomFilm);
    }
  },
});
app.mount('#monApp');
```

Fichier HTML :

```
<h1 class="text-center">List Rendering</h1>
<div id="monApp">
  <div class="m-3">
    <h2>WatchList 📽️ (Films à voir)</h2>
    <input v-model="nomFilm" type="text" class="form-control" />
    <button @click="ajouterFilm()" class="btn btn-primary mt-3">
      Ajouter à votre Liste
    </button>
    <hr />
    <h2 v-if="mesFilms.length === 0">Aucun Films à Voir ?</h2>
    <ul v-else>
      <li v-for="unFilm in mesFilms">{{unFilm}}</li>
    </ul>
  </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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Une autre utilisation un peu différente de v-for consiste à renseigner une seconde variable temporaire pour parcourir le tableau, cela va permettre d'afficher l'index des cases du tableau parcouru.

Exemple : ici la boucle va parcourir le tableau mesFilms, et va temporairement ranger chaque case dans une variable temporaire « unFilm », avant le IN du for...in, je rajoute une seconde variable unIndex qu je peux afficher dans le paragraphe :

```
<li v-for="(unFilm,unIndex) in mesFilms">{{unIndex}} - {{unFilm}}</li>
```

Rendu : unIndex représente l'indice de chaque case du tableau.

### List Rendering

#### WatchList 🎬 (Films à voir)

SUPER

Ajouter à votre Liste

- 0 - SUPER
- 1 - SUPER
- 2 - SUPER
- 3 - SUPER
- 4 - SUPER
- 5 - SUPER
- 6 - SUPER
- 7 - SUPER
- 8 - SUPER

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Exercice : List Rendering + index

En se basant sur l'exercice précédent (watchList), mettre en place ceci :

Quand on clique sur un des éléments rajoutés dans la liste, ça le supprime du tableau.

### List Rendering

#### WatchList 📽 (Films à voir)

SUPER 3

Ajouter à votre Liste

- 0 - SUPER 1
- 1 - SUPER 2
- 2 - SUPER 3

On a ajouté les films super 1, super 2 et super 3

En cliquant sur le film super 2, cela le supprime du tableau

### List Rendering

#### WatchList 📽 (Films à voir)

SUPER 3

Ajouter à votre Liste

- 0 - SUPER 1
- 1 - SUPER 3

[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Fichier JS : on rajoute une fonction supprimer qui prend un paramètre (quel film nous voulons supprimé), dans cette fonction on utilise la fonction JS splice().

```
const app = Vue.createApp({
  data() {
    return { nomFilm: '', mesFilms: [] };
  },
  methods: {
    ajouterFilm(){
      this.mesFilms.push(this.nomFilm);
    },
    supprimerFilm(unIndex){
      this.mesFilms.splice(unIndex,1);
    }
  },
  app.mount('#monApp');
```

Fichier HTML : on utilise l'index de la boucle for pour le passer en paramètre de la fonction supprimer.

```
<div id="monApp">
  <div class="m-3">
    <h2>WatchList 📽️ (Films à voir)</h2>
    <input v-model="nomFilm" type="text" class="form-control" />
    <button @click="ajouterFilm()" class="btn btn-primary mt-3">
      Ajouter à votre Liste
    </button>
    <hr />
    <h2 v-if="mesFilms.length === 0">Aucun Films à Voir ?</h2>
    <ul v-else>
      <li v-for="(unFilm,unIndex) in mesFilms"
v-on:click="supprimerFilm(unIndex)">{{unIndex}} - {{unFilm}}</li>
    </ul>
  </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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Une dernière méthode pour identifier précisément un élément parmi la liste des éléments qui se répète dans un v-for, consiste à faire un v-bind de l'attribut key pour le lier à la variable temporaire utilisée dans le v-for.

```
<li v-for="(unFilm,unIndex) in mesFilms"
    v-bind:key="unFilm"
    v-on:click="supprimerFilm(unIndex)">
    {{unIndex}} - {{unFilm}}</li>
```

Cette méthode s'utilisera davantage en mettant en place une communication avec une base de données (des listes de données avec une notion de clé - valeur) comme une table SQL, ou un Objet JSON (qui ont des identifiant uniques).

Dans ce cadre ci, plutôt que de se baser sur l'index du v-for (identifiant de tableau : 0,1,2,3, etc..), nous allons préférer lier à l'attribut key des éléments qui sont répétés, l'identifiant unique de la BDD.

(Exemple d'identifiant unique généré par Firebase : -Mw2BWaG802zINFiwLsj).

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### TP : Conditional / List Rendering

Créer une application VueJS dans laquelle on a une page qui affiche :

- 1 input
- 1 bouton Ajouter
- 1 liste (de courses, de films, au choix)
- 1 bouton « masquer Liste »
- Le bouton ajouter permet d'ajouter ce qui est tapé dans l'input dans la liste affiché en dessous
- Le bouton Masquer liste permet d'activer ou pas l'affichage de la liste des éléments
- On peut cliquer sur un élément de la liste afin de le supprimer
- Le message du bouton « masquer liste » doit se modifier en « afficher Liste » selon si la liste est masquée ou pas

**TP : Conditional /List Rendering**

La liste des courses 🛍

Ajouter à votre Liste de Courses

- 0 - AZERTYUIOP
- 1 - WXCVBN
- 2 - QSDFGHJKLM

Masquer Liste

**TP : Conditional /List Rendering**

La liste des courses 🛍

Ajouter à votre Liste de Courses

Afficher Liste

[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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### TP : Jeu Tour Par Tour

(Travail à faire en groupe)

Créer une application VueJS qui sera un jeu d'affrontement au tour par tour (thème au choix), notre personnage (l'utilisateur) devra rivaliser avec un adversaire(l'ordinateur).

L'application devra gérer des barres de vie

Notre personnage aura à disposition les actions suivantes :

- Attaque
- Attaque Magique /spéciale /coup critique
- Se Soigner
- Abandonner

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Étape 1 (init des data + f° d'attaque)

Données : la vie du player, la vie de l'adversaire

Methods : attackAdversaire permet au joueur de lancer une attaque, inflige des dégâts à la vie de l'adversaire

Methods : attackPlayer permet à l'adversaire de lancer une attaque, inflige des dégâts à la vie du joueur (l'adversaire frappe plus fort que le joueur)

PS : pour faire du tour par tour dans la f° attackAdversaire on va appeler la f° attackPlayer

Étape 2 mettre à jour les barres de vie :

- Lier la largeur de la barre de vie de l'adversaire avec son niveau de vie
- Lier la largeur de la barre de vie du joueur avec son niveau de vie

(Attention aux interférences dans les fonctions avec plusieurs évènements)

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Étape 3 Attack magique /spéciale :

- Idem que l'attack normale mais inflige plus de dégâts
- L'attack Magique n'est disponible (le bouton est actif) SEULEMENT tous les 3 tours
- Gérer une data qui stock le round en cours (currentround) initialisé à 0 et les actions attackAdversaire et specialAttackAdversaire vont incrémenter le round en cours

Étape 4 Se soigner :

- Une fonction de soin doit permettre de remettre une quantité de vie
-  il faut s'assurer que la vie du joueur ne dépasse pas 100
- Bien sûr se soigner coute un round dans un jeu au tour par tour
- Aussi à chaque fois que l'on se soigne, l'adversaire lance une attaque

Étape 5 créer une fin :

- Préparer un template HTML de game Over qui s'affiche si on a un winner  
Dans ce template : on affiche un titre en fonction du winner si c'est l'adversaire on affiche « PERDU » Sinon Si le winner c'est le joueur on affiche « VICTOIRE »  
autrement on affiche « égalité »
- Gérer une data winner initialisée à null
- Surveiller le niveau de vie du joueur, SI la vie du joueur est inférieure ou = à 0 ET la vie de l'adversaire est inférieure ou = à 0 alors le winner c'est « aucun » ou « égalité » SINON SI la vie du joueur est inférieure ou = à 0 alors le winner c'est l'adversaire
- Surveiller le niveau de vie de l'adversaire, SINON SI la vie de l'adversaire est inférieure ou = à 0 alors le winner c'est le joueur.

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		<i>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.</i>
	Marc CECCALDI Coordinateur Filière Développement		