

VUE

VUE

DURE TOTALE DU MODULE : 21H00

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

VUE

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.

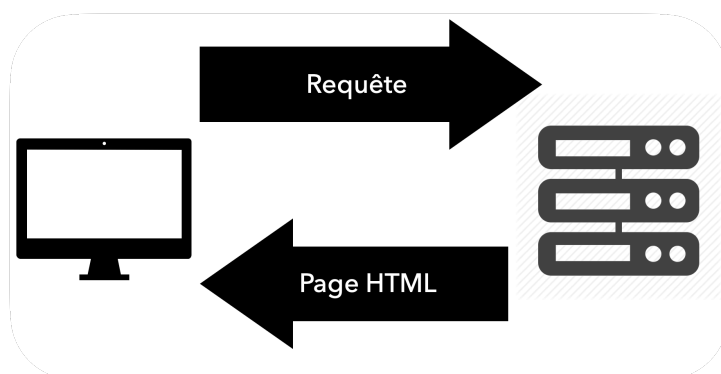
« À la différence des autres Frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque est concentré uniquement sur la partie vue, et il est vraiment simple de l'intégrer avec d'autres bibliothèques ou projets existants. D'un autre 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.

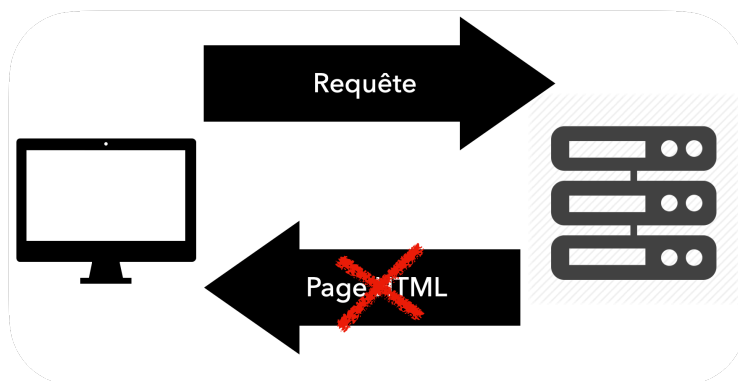
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou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

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	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

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 :

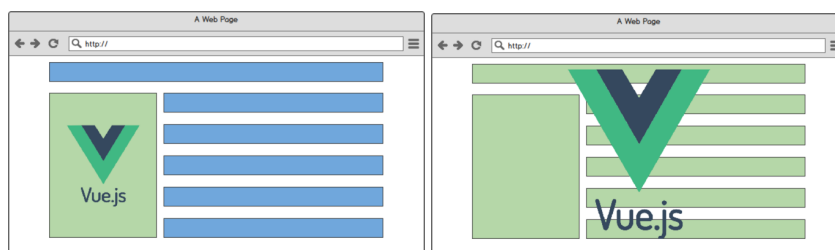
- Nous devons écrire tout le code de l'application (On réinvente la roue ?)
- Risque de réaliser un code non optimisé comprenant des erreurs, des bugs (difficilement maintenable / scalable).
- Travailler en équipe sera plus difficile, tout le monde ne connaît pas votre manière de coder.

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	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

Installation de VUE (mode CDN) :

En mode CDN, nous n'allons pas Installer VUE, nous allons directement l'utiliser depuis son serveur 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`

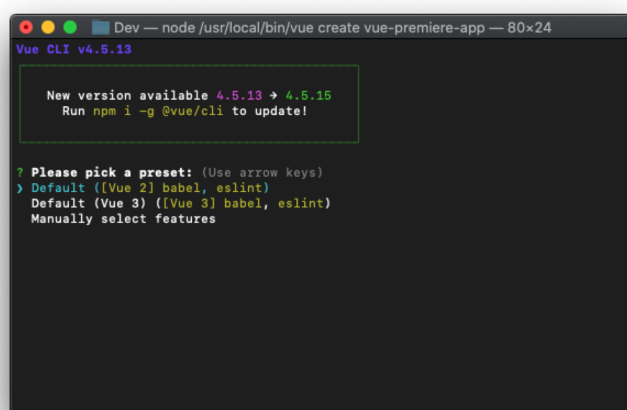
Le « -g » va permettre d'installer VUE CLI globalement sur votre machine vous pourrez par la suite utiliser des commandes pour créer des projets VUE dans n'importe quel dossier de votre machine.

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`

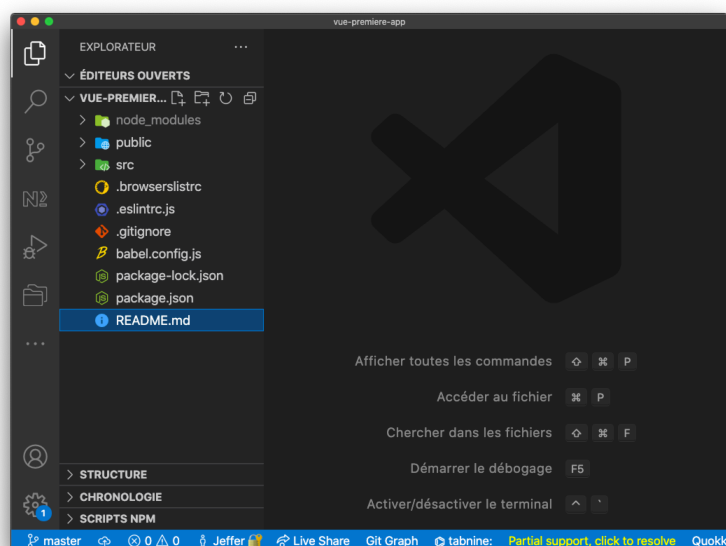
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou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



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	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

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	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

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

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 crée un élément
 - o on assigne à la propriété textContent (ou innerText) du , la valeur de leInput
 - o on place le 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	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

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	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

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	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

Page HTML (body)

```

16  <!-- On place un id sur la partie du template que l'on veut ctrl avec VUE (ici vue a le ctrl 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 data 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-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+0MhuP+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	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

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 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	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

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 {
      unLivres : "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 unLivres ne s'affiche pas (Vue n'a pas le ctrl ici) -->
<p class="text-center">{{unLivres}}</p>
<hr>
<!-- Ici Vue a le ctrl sur toute la card qui a cet id -->
<div id="bookListApp" class="card">
  <p class="text-center">{{unLivres}}</p>
</div>
```

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

VUE

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	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"><img alt="book icon" data-bbox="340 245 355 255"/> 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	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 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 chaine de caractères (au choix)
 - o Sinon on retourne une autre chaine 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	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Fichier JS :

```
Vue.createApp({
  data(){
    return {
      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	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

Affichage de contenu HTML (directive v-html)

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

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

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

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

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

Fichier JS

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

Fichier HTML

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

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

VUE

Rendu

Utilisation de la directive v-html

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

Les mémoires de Steven Seagal

Lier Des attributs HTML (directive v-bind)

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

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

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

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

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

Fichier JS

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

Fichier HTML

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

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

VUE

TP : DataBinding


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

TP à rendre au format Zip ou lien github

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

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

TP DataBinding - Profile User



Mario

Mario

45 ans

age + 10 : 55 ans

NB Random : 0.3148920573285465

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

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

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

VUE

Réagir à des évènements DOM

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

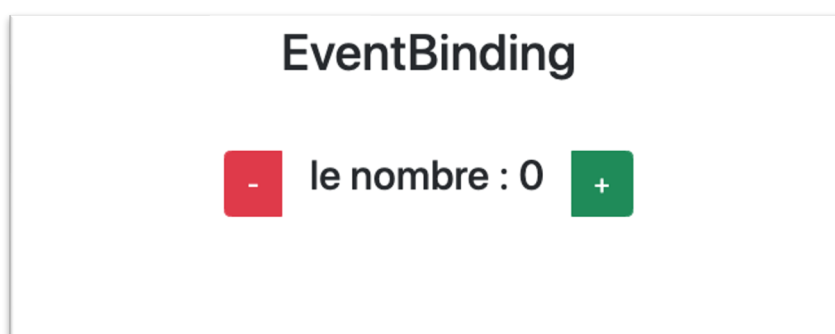
En Javascript (classique) pour réagir à ces évènements, on utilise la fonction `addEventListener`, VUE quant à lui propose la directive `v-on`, que l'on utilise côté HTML comme un attribut.

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

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

Exercice : setUp Application pour EventBinding

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



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

VUE

Fichier HTML :

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

Fichier JS :

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

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

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

VUE

Utilisation de la directive v-on + expressions JS

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

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

Exemple de v-on :click sur les boutons :

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

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

VUE

Utilisation de la directive v-on + fonction

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

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

Fichier JS :

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

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

```
<div id="monApp" class="container mt-4 mb-4 p-3 d-flex justify-content-center">
  <div class="btn-group" role="group" aria-label="Basic mixed styles example">
    <button v-on:click="reduire" type="button" class="btn btn-danger">-</button>
    <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	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

VUE

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

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

Fichier JS :

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

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

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

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

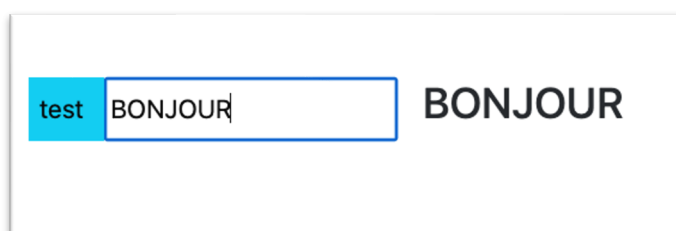
VUE

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

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

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

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



Fichier HTML :

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

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

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

VUE

Fichier JS :

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

En mettant en place des fonctions dans lesquelles on capte les 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 valeur contenu dans l'évènement input pour l'assigner à la variable uneString.

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

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

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

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

VUE

Utilisation des Event Modifiers

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

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

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

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

Fichier HTML :

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

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

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

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

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