**VUE**

**Dure totale du module : 21H00**

Table des matières

[Objectifs pédagogiques : 5](#_Toc109634132)

[Présentation et installation de VUE 6](#_Toc109634133)

[Présentation de VUE : 6](#_Toc109634134)

[Installation de VUE (mode CDN) : 9](#_Toc109634135)

[Installation de VUE (mode CLI) 9](#_Toc109634136)

[Création d’une première application vue (mode cdn) 12](#_Toc109634137)

[Réalisation d’une application web TaskList en Javascript 12](#_Toc109634138)

[Exercice : Créer une TaskList en JS 12](#_Toc109634139)

[Application web TaskList en VUE 14](#_Toc109634140)

[Notions / Concept de base de VUE 16](#_Toc109634141)

[Connexion d’une App VUE avec un Template HTML et affichage de données 17](#_Toc109634142)

[Exercice : Nouvelle App VUE qui affiche une variable 17](#_Toc109634143)

[Exercice : Afficher des variables de différents type 18](#_Toc109634144)

[Utilisation des methods 20](#_Toc109634145)

[Exercice : Utilisation des methods 20](#_Toc109634146)

[Affichage de contenu HTML (directive v-html) 22](#_Toc109634147)

[Lier Des attributs HTML (directive v-bind) 23](#_Toc109634148)

[TP : DataBinding 24](#_Toc109634149)

[Réagir à des évènements DOM 25](#_Toc109634150)

[Exercice : setUp Application pour EventBinding 25](#_Toc109634151)

[Utilisation de la directive v-on + expressions JS 27](#_Toc109634152)

[Utilisation de la directive v-on + fonction 28](#_Toc109634153)

[Utilisation de la directive v-on + fonction avec des paramètres 29](#_Toc109634154)

[Utilisation de la directive v-on + fonction avec paramètre pour capter un évènement (Native Event Object) 30](#_Toc109634155)

[Utilisation des Event Modifiers 32](#_Toc109634156)

[TP: EventBinding 33](#_Toc109634157)

[2 Way DataBinding (EventBinding + DataBinding) directive v-model 34](#_Toc109634158)

[Exercice v-model 34](#_Toc109634159)

[Design Pattern MVC vs MVVM 36](#_Toc109634160)

[Utilisation des computed Properties 37](#_Toc109634161)

[Exercice : Bug évènements + fonctions (interférences) 37](#_Toc109634162)

[Utilisation des Watchers 41](#_Toc109634163)

[TP : Réactivité 43](#_Toc109634164)

[Gestion des styles dynamiquement 44](#_Toc109634165)

[Style Dynamique (inline) 44](#_Toc109634166)

[Exercice : Style Dynamique (inline) 45](#_Toc109634167)

[Style Dynamique (class CSS) 48](#_Toc109634168)

[Exercice : Style Dynamique (class CSS) 48](#_Toc109634169)

[TP : Style Dynamique 51](#_Toc109634170)

[Conditional Rendering (directive v-if) 52](#_Toc109634171)

[Exercice : Conditional Rendering 53](#_Toc109634172)

[Directive v-show 56](#_Toc109634173)

[List Rendering (Directive v-for) 57](#_Toc109634174)

[Exercice : List Rendering 58](#_Toc109634175)

[Exercice : List Rendering + index 61](#_Toc109634176)

[TP : Conditional / List Rendering 64](#_Toc109634177)

[TP : Jeu Tour Par Tour 65](#_Toc109634178)

# Objectifs pédagogiques :

Être capable de comprendre les enjeux des Frameworks

Être capable d'initialiser et paramétrer 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

# Notions / Concept de base de 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 chaine 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)](https://github.com/jefff404/bookList-vue/tree/v1.3-exemple-v-html)

[SOLUTION BookListV1.3(StackBlitz)](https://stackblitz.com/edit/web-platform-hbqtka?file=index.html)

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>

Rendu

