**VUE**

**Dure totale du module : 21H00**

Table des matières

[Objectifs pédagogiques : 3](#_Toc111651105)

[Notions / Concept de base de VUE 4](#_Toc111651106)

[Connexion d’une App VUE avec un Template HTML et affichage de données 5](#_Toc111651107)

[Exercice : Nouvelle App VUE qui affiche une variable 5](#_Toc111651108)

[Exercice : Afficher des variables de différents type 6](#_Toc111651109)

# 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

L’exemple précèdent (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 noter 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.

## 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 chaine de caractères).

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

SOLUTION BookList(StackBlitz)

[SOLUTION BookList(GitHub)](https://github.com/jefff404/bookList-vue)

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>

### 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)](https://stackblitz.com/edit/web-platform-hbqtka?file=index.html)

[SOLUTION BookList(1.1)(GitHub)](https://github.com/jefff404/bookList-vue/tree/v1.1-afficher-variables)

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');

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>