**VUE**

**Dure totale du module : 21H00**

Table des matières

[Objectifs pédagogiques : 3](#_Toc111661691)

[Notions / Concept de base de VUE 4](#_Toc111661692)

[Utilisation des Watchers 4](#_Toc111661693)

# 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

### Utilisation des Watchers

Fréquemment dans les applications, nous avons besoin de surveiller les changement 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

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](https://github.com/jefff404/bookList-vue/tree/v1.10-watchers)