**VUEJS**

**Basics** :

<https://alithya.udemy.com/course/vuejs-2-the-complete-guide/>

Page officiel

<https://vuejs.org/guide/quick-start>

la façon la plus simple d’ajouter vue est de copier le script de démarrage suivant (depuis la page off)

A black background with orange and blue text

Description automatically generated

Javascript vs Vuejs (au mount on donne l’id du composant qu’on veut manipuler) et la var s’appelle toujours **data** et **methods** pour gérer l’interaction des events du user

A computer screen shot of a program code

Description automatically generated

Coté html en plus de la balise script ajouté

A computer screen with text and images

Description automatically generated

Format sur vscode : File - Preference – KeyBoard shortcuts et rechecher le shortcut (ex : format document)

Interpolation : la capture inclus html et vue js(ici il prend la chaine de caract et l’affiche dans le html grâce {{ }}

A screen shot of a computer program

Description automatically generated

v-bind pour setter une prop dynamiquement ex :

A screen shot of a computer

Description automatically generated

Define method : comme le nom data, le nom **methods** est obligatoire pour déclarer les différentes méthodes. Data et methods sont au même niveau.

A computer screen shot of text

Description automatically generatedcall 

Data : toutes les données retournée par data sont accessible dans la partie methods grâce **à this.xxxxx**

Pour interpréter le html (disable pour cause de sécurité) : use **v-html**

 

Display img : 

Bind input 

Edit event on button : **v-on**:click=”xxx” ou v-on:mouseleave

Ex avec ou sans ( ), methodes declares dans **methods** A black background with orange and white text

Description automatically generated

Méthode avec param : 

Afficher ce qui est saisie dans un input avec v-on :**input** A black background with white text

Description automatically generated

Name est déclaré en haut dans la partie data avec et = à chaine vide A black background with white text and symbols

Description automatically generated

La même méthode avec 2 args  A black background with white text

Description automatically generatedastName ;

Pour ne pas recharger la page tout le temps après un click par ex :

A black background with white text

Description automatically generated et A close up of text

Description automatically generated

Ou mieux encore A black background with white text

Description automatically generated et A black background with colorful text

Description automatically generated

Pour dire que seul le click droit est autorisé 

Valider seulement le input après touche enter

A black background with orange and white text

Description automatically generated et A computer screen with text and symbols

Description automatically generated

Pour fixer la valeur de départ même si counter change de val 

Binding value : pour par ex effacer le text saisie dans un input

A computer screen with text and symbols

Description automatically generated et A black background with white text

Description automatically generated

Mieux encore avec le 2 way binding **v-model**

A computer code with text

Description automatically generated with medium confidence

Pour ne pas ex une methode si une partie de la page change faire: Computed Properties même niveau que data

A computer code on a black background

Description automatically generatedcomputed,

appel sans( ) 

utilisé **methods** si la methode doit etre exec qq soit ce qui change dans la page. Sinon use **computed** pour des raisons de performance.

**Watchers** : **watch** comme methods et computed

A black screen with white text

Description automatically generated use case 

Ex qui retourne le nom complet saisie dans 2 input : ici computed est plus efficient pour cet expleA computer code with many colorful text

Description automatically generated with medium confidence

Les 2 input A black background with white text

Description automatically generated

Exple pour watch : le counter ne depasse pas 50 A black background with white text and colorful text

Description automatically generated

Est aussi optimale pour l’envoi de requête http, de timer

Le meme ex que ci-dessus mais remis a 0 après 2 sec A screen shot of a computer code

Description automatically generated

On peut remplacer **v-on:** par **@** ex v-on:click par @click

**Dynamic Styling**:

Lors qu’on clique sur le div on change la couleur de bordure A black screen with white text

Description automatically generated

Et coté js

A screen shot of a computer code

Description automatically generated

Le meme ex avec du css

A black background with white text

Description automatically generated A screen shot of a computer program

Description automatically generated

Meme rendu demo est le style par defaut, et on ajoute la classe active si le box est selectionné

A screen shot of a computer

Description automatically generated

La meme chose avec computed (pour des cas avec des tests plus compliqués)

A black background with white text

Description automatically generated 

On peut aussi setter en array 

Exo : le css de user1 et user2 est derriere

A screen shot of a computer code

Description automatically generated A screen shot of a computer screen

Description automatically generated

Mettre du style selon la saisie : myBackground est initialisé dans data à ‘’



**Rendered Content Conditionaly :**

Pour ne pas afficher une partie use **v-if** 

**v-else** Doit être contigu a la partie ayant le v-if pour foncyionner  (on a aussi **v-else-if**)

v-show est comme v-if mais avec v-show est présent alors que sa visibilité est none(css). Tandis que v-if enléve le composant. A screen shot of a computer

Description automatically generated

v.if est plus performant. Sauf on a besoin de bcp changé la visibilité d’un élément

v-for : pour itérer 

on peut aussi avoir l’index de l’item (ça commence à 0) 

pour parcourir un objet (il peut aussi avoir un index après 

remove element d’une liste A black and blue background with white text

Description automatically generated et 

Pour éviter que le click sur la zone de text supprime l’élément faire **@click.stop** A screen shot of a computer

Description automatically generated

**v-bind:key** est equivalent à **:key** (toujours utiliser **key** avec v-for pour éviter des surprises et on lui donne une valeur unique comme l’id)

on peut avoir une fonction en dehors du createApp  et l’appeler avec getRandom sans this

barre de progression  ou le mettre dans computed A computer screen shot of text

Description automatically generated et 

disabled  la methode est dans computed A black background with white text

Description automatically generated

Ici on surveille l’évolution du score dans le watch, winnner est initialisé a null dans data A computer screen with text and images

Description automatically generated A computer screen with text

Description automatically generated pour le use case

Push ajoute un element à la fin, unshift au début ex :

A screenshot of a computer screen

Description automatically generated use case log--player et log--monster sont des cssA screen shot of a computer screen

Description automatically generated

rendu A screenshot of a game

Description automatically generated

On peut aussi utilisé ref pour setter des valeurs de input comme dans v-model ou @input

 refs avec s 

**Lifecyle vue** : on peut les créer n’importe où même niveau que data, methods …

* beforeCreated : on peut y mettre par ex des req http
* created : comme le précédent ; la vue n’est pas encore monté
* beforeMount : la vue n’est toujours pas visible
* mounted : à ce niveau on voit la vue
* beforeUpdate : on voit la vue mais pas la modif
* updated : ici on voit la vue et sa modif
* beforeUnmount : avant de killer l’app (on peut le provoqué avec app.unmount() mais c rare)
* unmounted : l’app est enlevé et on ne voit + la page

**Components** :

Le problème : si on travaille sur une liste d’éléments qui a un bouton show/hide détails pour chaque élément si on clique sur un bouton il s’exécute sur tous les autres. C la que les composants entrent en scène comme ce que l’on fait avec angular.

**Vue CLI**:

Installer cli avec : npm install -g @vue/cli

Pour créer une app : **vue** **create** vue-first-app

Pour lancer l’application : npm run serve

Dans src, main.js sera maintenant le point d’entrée du programme

A screen shot of a computer

Description automatically generated

Et App.vue est un fichier .vue avec une partie template (html), script (js ou vue) et style pour le css. Ex basique

A screen shot of a computer screen

Description automatically generated

Pour créer un new component : dans src/components créer un fichier .vue

A screen shot of a computer screen

Description automatically generated et import A screen shot of a computer code

Description automatically generated ensuite l’utiliser dans App.vue A computer screen with text

Description automatically generated

**Component Communication :**

Avec props

A black background with orange text

Description automatically generateddans son templateA screen shot of a computer

Description automatically generated

Et au niveau de App.vue A black background with white text and numbers

Description automatically generated

Props est immutable, pour contourner ça il y a 2 façons de faire dont : déclarer dans data

 comme on ne peut changer directement isFavorite on met sa valeur dans ce new champ et c’est ce champ qu’on va manipuler  et afficher dans le template. Avec cette methode on ne change pas vraiment la donnée initiale du parent (juste la copie)

Au lieu de lui donner un tableau, on peut donner au props un objet plus complex avec des validationsA computer screen shot of code

Description automatically generated

Si on ne donne pas de nom par ex on aura un warning sur la console car c un champ requis.

On peut boucler ave v for au niveau du parent aussi et enlever les data de l’enfant

A screen shot of a computer code

Description automatically generated

Communication du component à son parent : ex du isFavorite

ToggleFavorite devient A black background with white text

Description automatically generated ici on donne l’id de l’element concerné. Id est dans props

Dans le parent App.vue A computer screen shot of a computer code

Description automatically generatedet la méthode

**A computer screen shot of a computer code

Description automatically generated**

Pour juste des besoins d’informations , (log et dire qu’on a tel emit dans le component) A computer screen shot of text

Description automatically generateddeclaration simple et avec validation

**Delete friend** :

Dans le component enfant qui reçoit et affiche la liste  ou bien créer une fonction qui a la ligne this.$emits('delete') et @click= ‘delete’

Et ajouter dans la liste des emits la new methode pour info

Dans la vue qui a la liste ajouter A black background with white text

Description automatically generated pour gérer l’envoie du del fait par le fils

La fonction en question 

**Injection** : ça se passe entre parent (ancêtre) et fils toujours. Ca marche comme les props mais les données passent entre plusieurs composants.

Elever props 

Ajouter une zone provide même niveau que data A screen shot of a computer code

Description automatically generated

Use case enlever props et remplacer par inject A screenshot of a computer program

Description automatically generated

Pour ne pas dupliquer les données écrire provide comme data cad A black background with white text

Description automatically generated

Ex qui ajoute un element a la liste après 3s A screen shot of a computer

Description automatically generated

Au lieu de faire emit et props pn peut le mettre dans provide et l’injecter après

A screen shot of a computer

Description automatically generatedsimple dans composant A black background with text and symbols

Description automatically generated with medium confidence

Use case :  A computer screen shot of text

Description automatically generated

**NB** : props et event doivent être privilégier à la place de provide et inject.

Resumé : props : parent -> fils, event : child -> parent (emit)

**Plus loin avec les components** :

Quand on fait app.component(‘xxxx-yy’, Xxxxx) ce component sera disponible au niveau de tous les autres components même s’il n’en ont pas besoin

Et donc pour un import localisé dans la partie script faire l’import et ajouter une partie components où le declarerA black screen with white text

Description automatically generated

Ou bien faire toujours l’import A black background with text

Description automatically generated et use case (self closing ne marche pas avec le tiret -)

Style scoped : par défaut le css définit dans style est partagé. Pour éviter ce side effect ajouter scoped sur style 

**SLOT** : Pour passer du code html avec son css à un template. Ex suivant : la classe de base

A screen shot of a computer program

Description automatically generated import global dans main.js et use caseA screen shot of a computer code

Description automatically generated

Quand on a plusieurs slot, il faut les nommer pour les différencier (ou laisser juste un sans nom)

A computer screen with white text and red text

Description automatically generated use case A screen shot of a computer program

Description automatically generated ou avec le mot defaut

A screen shot of a computer code

Description automatically generated

Slot avec contenu par défaut : si on l’appel sans lui donner le contenu, on aura le contenu par défaut

A computer screen with text and images

Description automatically generateduse case : ici la partie header n’est pas def A screen shot of a computer

Description automatically generateddonc on aura l’affichage par défaut

Pour ne pas afficher une case vide lors de l’inspection faire A computer screen with text and images

Description automatically generated

**v-slot:header** est equivalent à **#header**