

Tutoriel à Vue JS

Durant ce tutoriel nous allons mettre en pratique plusieurs notions fondamentales de Vue JS afin de pouvoir aborder au mieux le projet qui suivra.

Préparation du TP

Rendez-vous à l'url ci-dessous pour créer votre propre dépôt. Ensuite clonez le en local via Git.

URL dépôt tutoriel : <https://classroom.github.com/a/1TDRmUjd>

Une fois le projet cloné vous possédez un dossier contenant plusieurs fichiers :

- `index.html` : c'est le point d'entrée de votre application, il contiendra les inclusions de scripts et l'importation de votre application Vue JS.
- `app.js` : il s'agit du fichier dans lequel vous écrirez votre code javascript au format Vue JS.
- `characters.json` : un fichier de données qui vous sera utile pour l'un des exercices.
- `css`: le dossier contient une fiche css par défaut, vous pourrez ajouter les autres ou compléter celle-ci.

La propriété *data*

Cette propriété permet de stocker des données et gérer ce que l'on appelle l'état de notre application.

app.js

Commencer par ajouter une data *user* avec la valeur par défaut votre nom.

index.html

Du côté de l'HTML, dans la `<div class="user">` vous aller afficher cette data dans un `<h2>`. En précisant que vous êtes connecté.

La directive v-model

L'intérêt d'utiliser des data dans notre HTML est qu'elles peuvent être mise à jour instantanément sans rechargement de la page. Cela peut se faire via les directives.

index.html

On va rajouter, dans la `<div class="user">`, un champ input où l'on indiquera un nom et notre `<h2>` devra se mettre à jour en fonction du nom indiqué dans l'input. Pour cela utiliser la directive *v-model*.

Remarque: Amusez vous à inspecter la zone `<h2>` lorsque vous changez le nom. Vous verrez quelque chose d'intéressant.

La directive v-for

app.js

Créer une data *characters* qui sera un tableau et contiendra la valeur par défaut suivante:

```
[ 'Mario', 'Link', 'Samus', 'Donkey Kong' ]
```

index.html

Maintenant grâce à la directive `v-for` parcourez votre tableau et dans la `<div class="list">` afficher le résultat avec une liste ``.

Travailler avec des données plus complexes

Essayons d'intégrer des données plus complexes à notre boucle.

app.js

Les données contenues dans le tableau *character* seront celles récupérées dans le fichier *data/character.json*.

index.html

Adapter ensuite votre `v-for` pour qu'il puisse gérer des objets. Vous afficherez l'attribut *name* dans un `<h3>` et l'attribut *portrait* dans une balise ``. Ces deux balises seront contenues dans une balise ``.

Les méthodes et la directive v-on

En plus des données notre instance de vue va pouvoir contenir des méthodes.

app.js

Nous allons créer la méthode *add()*. Pour commencer cette méthode fera seulement un `console.log('Ajout réussi')`.

index.html

Rajouter dans votre liste, en dessous de la balise `` une balise `<button>Ajouter</button>`. Ce bouton utilisera la directive `v-on` afin de déclencher la méthode *add()* lors d'un **click**.

Vérifier dans votre console si le message s'affiche correctement.

Ajouter des personnages dans une équipe

On veut désormais qu'au clic d'un personnage de la liste ce dernier s'ajoute à une autre liste qui sera notre équipe.

app.js

On va implémenter notre méthode *add()*. Supprimer le `console.log()`. La méthode devra prendre en paramètre le personnage reçu et l'ajouter dans un nouveau tableau *team*.

index.html

Le but est que lors du click sur le bouton, on envoie en paramètre le personnage sur lequel on a cliqué. Il faut aussi créer une nouvelle liste dans la `<div class="team">` qui contiendra la liste des personnages choisis. Cette liste aura exactement la même forme que celle de la liste des personnages mais affichera l'équipe choisie à la place.

La directive v-if

Depuis le début la div permettant l'affichage de l'équipe s'affiche. Ce n'est pas très esthétique on préférerait n'afficher ce bloc de code uniquement si la data *team* contient au moins une valeur.

index.html

A l'aide de la directive v-if, faites en sorte que le block de la div `<div class="team">` ne s'affiche que s'il y a au moins un élément dans le tableau *team*.