1. **The vue Instance**

**Le but :** On va voir comment utilisé Vue pour afficher des data sur le web.

Nous avons donc deux fichier index.html et main.js.

**Le problème :** Nous voulons voir le contenu de la variable javascript sur le web.

**Code de départ :**

Une image contenant texte

Description générée automatiquement

Pour utiliser vue.js nous allons commencer par importer le CDN qui va contenir toutes les dépendances du framework et mettre le lien dans le fichier index.html.

* <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

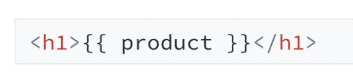
Dans le main.js nous allons créer une instance vue qui va faire office de cœur de l’application.

Une image contenant texte

Description générée automatiquement

Notre instance peu donc prendre plusieurs paramètre :

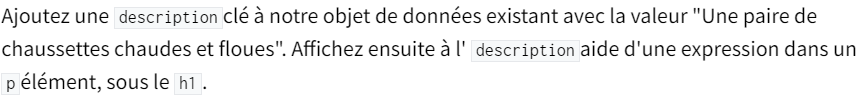
* Le premier params permet de relier l’instance vue à un élément du DOM
* Le deuxième params montre qu’il y a une place pour les données à partir de data. On va pouvoir y a accéder a partir de l’expression product.



On utilise l’expression Product afin d’afficher les éléments stocker dedans. On a besoin des doubles accolades

Vue JS est reactif cela veut dire que a chaque mise à jour des données elle seront automatiquement changer en temps réelle sur l’application web.

**Exercice numéro 1 :**



1. **Liaison d’attributs**

Le but est d’utilisé la liaison d’attributs pour afficher une image et lui attacher un texte alternatif en fonction des valeurs des données de notre instance.

Nous avons rajouté un fichier css.

**Code de départ :**

Une image contenant texte

Description générée automatiquement

Nous avons créé un div pour l'image du produit et les informations sur le produit, afin de les styliser avec Flexbox.



Notez que nous avons ajouté une imagesource à nos données.

**Problème :** Nous voulons faire apparaitre dynamiquement une image dans nos données et que la mise à jour se fasse automatiquement sur la page web.

**Solution :** Pour lier la valeur de l’image de notre objet de donnée à src dans notre balaise img nous utilisons la directive v-bind de vue.



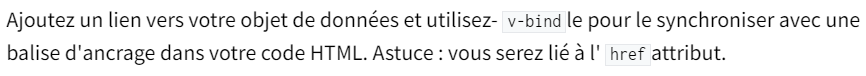
On peut vouloir ajoute un texte à cette image, on va donc ajouter une expression dans notre objet de donnée et ensuite utiliser v-bind pour la liaison. Chaque fois que les données image et altText changent, ces données mises à jour restent liées aux attributs src et alt

Au lieux d’écrire v-bind:src= ‘image’ nous pouvons écrire :src=’image’

Une image contenant texte

Description générée automatiquement

**Exercice numéro 2 :**



1. **Rendu conditionnel**

Dans se cours nous allons voir comment afficher conditionnellement des éléments avec vue.

**But** : Le but est d’afficher un texte indiquant si notre produit est en stock ou non, en fonction de nos données.

En stock, En cours d’épuisement et en rupture de stock.

Code de départ :

Une image contenant texte

Description générée automatiquement

Tous d’abord nous allons créer une variable booléenne que nous appellerons inStock.

Ensuite nous allons créer deux paragraphe afin d’afficher le message en fonction de la condition.

Nous aurons deux paragraphe avec la valeur ‘in Stock’ et une autre avec la valeur ‘out of stock’.

Dans la balise paragraphe on met notre condition.

* v-if= ‘’inStock’’
* v-else if ‘’’’
* v-else

If la valeur boolean = true alors on affiche inStock sinon on affiche outOfStock.

**Other example :**



Une image contenant texte

Description générée automatiquement

**v-show** est plus performante que l’insertion et la suppression avec répétées d’un élément avec v-if / v-else.

**Exercice numéro 3 :**

Ajoutez une propriété onSale aux données du produit qui est utilisée pour restituer conditionnellement une étendue qui indique « On Sale ! »

1. **Rendu de la liste**

Dans cette leçon, nous allons apprendre à afficher et manipuler des listes sur nos pages web.

But : Nous voulons afficher la liste des détails de notre produit.

Nous avons un tableau détails, avec 3 valeurs.

**Problème :** Nous voulons que notre page affiche les détails. Comment peut-on parcourir ce tableau pour afficher ses données.

**Solution :** Nous avons une nouvelle directive Vue pour parcourir un tableau, il permet de boucler sur le tableau et de rendre les données à partir de celui-ci.

Une image contenant texte

Description générée automatiquement

Dans la directive on a l’alias détail au singulier, ensuite le mot in et le tableau en question. ET enfin on affiche les données avec l’expression detail.

Code de départ :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

**Itérer sur des objets :**

**Problème :** La page produit que nous créons doit pouvoir afficher différentes versions du même produit, en fonction d'un tableau dans nos données de variants. Comment itérerions-nous dans ce tableau d'objets pour afficher ses données ?

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

**Exercice numéro 4 :**

Ajoutez un tableau de sizes aux données et utilisez v-for pour les afficher dans une liste.

1. Gestion des événements

Dans cette leçon, nous allons apprendre à écouter les événements du DOM que nous pouvons utiliser pour déclencher des méthodes.

But : Nous voulons avoir un bouton qui incrémente le nombre d’article dans notre panier.

**Une image contenant texte

Description générée automatiquementCode de départ :Une image contenant texte

Description générée automatiquement**

**Problématic :** Nous avons besoin d’un bouton pour écouter les événements de clic dessus, puis déclencher une méthode lorsque ce clic se produit, afin d’incrémenter le total de notre panier.

Dans un premier temps, on ajoute une nouvelle propriété de data pour notre attribut cart.

* Cart : 0,

Une image contenant texte

Description générée automatiquement

Dans le code HTML, on créer un panier avec sa valeur à l’intérieur.



Nous venons d’ajouter un bouton qui va permettre d’incrémenter le card.

Il est mieux de passer par une fonction pour séparer la logique métier et le front-end.



Une image contenant texte

Description générée automatiquement

Nous avons créé une fonction anonyme afin de mettre la logique métier dedans. Cette méthode est appeler au moment d’un click suite a la directive de vue (v-on :click= ‘addToCart’)

Une image contenant texte

Description générée automatiquement

**Exercice 5 :**

Créez un bouton et une méthode pour décrémenter la valeur de .cart

1. Reliure de classe et de style

Dans cette leçon, nous allons apprendre à styliser dynamiquement notre code HTML en liant des données à l’attribut style d’un élément, ainsi qu’à sa classe.

But : Notre premier objectif dans cette leçon est d’utiliser nos variantes de couleurs pour styliser background-color les divs. Puisque nos variantes de couleurs sont « vert » et « bleu », nous voulons un div background-color avec un vert et un div background-color avec un bleu .

Code de départ :

Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement

**Problème** : Dans la leçon précédente, nous avons créé un gestionnaire d’événements qui met à jour l’image du produit en fonction de la balise p survolée. Au lieu d’imprimer la couleur de la variante dans une balise p, nous voulons utiliser cette couleur pour définir le style d’un div background-color. De cette façon, au lieu de survoler le texte d’une balise p, nous pouvons survoler les carrés colorés, ce qui mettrait à jour l’image du produit pour qu’elle corresponde à la couleur qui a été survolée.

Une image contenant texte

Description générée automatiquement

**Solution** : Tout d’abord, ajoutons une classe color-box à notre div, ce qui lui donne une largeur, une hauteur et une marge. Puisque nous imprimons toujours « vert » et « bleu » sur la page, nous pouvons utiliser ces chaînes de couleurs variantes et les lier à notre attribut de style, comme suit

**Problème** : Nous avons ceci dans nos données



Lorsque la valeur boolean est à false nous devons interdire le click par les utilisateurs car il n’y a plus de produit disponible. Pour désactiver le bouton nous allons utilisé disabled.

**Solution :**

Une image contenant texte

Description générée automatiquement

Nous pouvons utiliser la liaison d’attribut pour chaque fois qu’il est faux.

Cependant il faut changer la couleur, au moment où il est désactivé. Pour cela nous pouvons lié l’attribut isStock avec une class a chaque fois qu’il sera faux afin de changer la couleur du bouton.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

**Exercice 6 :**

Lorsque la valeur inventory = 0, liez une classe text-decoration : line-through à la balise p « Out of Stock » qui s’ajoute à cet élément.

1. Propriétés calculées
2. Composants
3. Communiquer les évènements
4. Formulaires et modèles en V
5. Onglets