

# Filtres & Directives

*Transformons nos variables dans la vue !*

Animé par Mazen Gharbi

# Qu'est-ce qu'un filtre ?

▷ Les filtres vont permettre d'appliquer des formatages à nos variables ;

- › Uppercase une chaîne de caractère ;
- › N'afficher que les 5 premiers éléments d'un tableau ;
- › etc.

▷ Voici un exemple de filtre :

```
{{ a | capitalize }}
```


▷ Ce qui est fort, c'est que l'on peut chaîner nos filtres :

```
{{ a | lowercase | capitalize }}
```

# Filtre global

- ▷ Dans l'exemple précédent, le scope du filtre était local, donc non accessible au reste de l'application ;
- ▷ Pour partager un filtre, on peut passer par les mixins ou utiliser l'objet Vue directement :

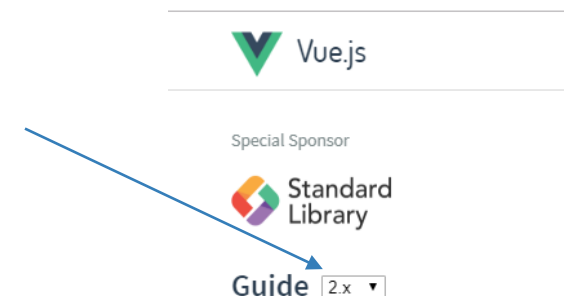
Instance vue globale



```
Vue.filter('json', function (value) {  
    return JSON.stringify(value);  
});
```

# Différence v1 & v2

- ▷ Plusieurs **break changes** entre VueJS 1 & 2
- ▷ Les filtres ne peuvent plus être appliqués dans un v-for
- ▷ Disparition de `filterBy` / `orderBy` (cf [ici](#))
  - › Préférez l'utilisation des « computed properties » avec la librairie `lodash`
- ▷ L'objectif a été d'alléger un maximum la librairie
  - › Mué..
- ▷ *Ne pas confondre les docs :*



# Directives

- ▷ Une directive est un attribut appliqué sur un élément du DOM ;
- ▷ Permet de le mettre en forme / lui attribuer un comportement ;
- ▷ De nombreuses directives sont fournis par Vue
  - › v-model, v-bind, v-on etc.
- ▷ Parfois, vous aurez besoin d'appliquer des comportements spécifiques
  - › Pour ce faire, il va être nécessaire de créer nos propres directives

# Custom Directive

▷ Créons une directive qui appliquera le background rouge aux éléments du DOM

```
APP.component('Home', {  
  directives: {  
    'custom-background': (el, binding) => {  
      el.style.backgroundColor = 'red';  
    }  
  },  
});
```

```
APP.directive('custom-background', (el, binding) => {  
  el.style.backgroundColor = 'red';  
});
```



<p v-custom-background>Un background rouge !</p>

# Les Hooks, encore eux !

Nom	Quand est-il appelé ?
created	Appelé quand les attributs et écouteurs d'événements sont appliqués
beforeMount	Est appelée une fois que la directive est attachée à l'élément. Fait office de constructeur
mounted	Se produit quand l'élément est inséré dans le DOM parent
beforeUpdate	Survient un peu avant que l'élément soit mis à jour
updated	Réagit une fois que le composant et les enfants ont été mis à jour
unmounted	Advient une fois la directive supprimée

▷ Ces hooks sont représentés par des fonctions

▷ Chacune de ces fonctions prendra en paramètre **l'élément DOM**, le **binding** et le **VNode**

- › **Binding** : Objet contenant les paramètres passés au hook (*read-only*)
- › **VNode** : Nous permet d'interagir avec le node dans le « Virtual DOM » (*read-only*)

# Mise en place

```
Vue.directive("hook-example", {  
  created(el, binding, vnode) {  
    console.log("bind");  
  },  
  mounted(el, binding, vnode) {  
    console.log("inserted");  
  },  
  updated(el, binding, vnode) {  
    console.log("updated");  
  },  
  unmounted(el, binding, vnode, oldVnode) {  
    console.log("componentUpdated");  
  }  
});
```



# Passage de paramètres

▷ Il existe plusieurs méthodes pour envoyer des paramètres à une directive

```
<div v-example-1="'red'">  
  Directive 1  
</div>  
  
<div v-example-2:blue>  
  Directive 2  
</div>  
  
APP.directive("example-1", {  
  beforeMount(el, binding, vnode) {  
    el.style.backgroundColor = binding.value;  
  }  
});  
  
APP.directive("example-2", {  
  beforeMount(el, binding, vnode) {  
    el.style.backgroundColor = binding.arg;  
  }  
});
```

# Modificateurs d'événements

- ▷ Il est également possible d'appliquer puis récupérer des modificateurs d'événements

```
APP.directive("example-1", {  
  beforeMount(el, binding, vnode) {  
    el.style.backgroundColor = binding.value;  
  
    if(binding.modifiers.white) {  
      el.style.color = 'white';  
    }  
  }  
});
```

```
<div v-example-1.white="'red'">  
  Directive avec modificateur  
</div>
```

# Cas pratiques

```
APP.directive("sticky", function(el, binding, vnode) {  
  el.style.position = "fixed";  
  
  if (binding.modifiers.bottom) {  
    el.style.bottom = 0;  
  } else if (binding.modifiers.top) {  
    el.style.top = 0;  
  }  
  
  if (binding.modifiers.right) {  
    el.style.right = 0;  
  } else if (binding.modifiers.left) {  
    el.style.left = 0;  
  }  
});
```

---

`<div v-sticky.right.bottom>`Cette div sera collé en bas à droite !`</div>`

# Questions