

Angular 4

ngFor

vertika.org by Jean Garutti -- Angular 4 -- 03/09/2017 -- 1

info@vertika.org

ngFor



- Simple, enfantin
- Et vraiment fort

Jean Garutti – vertika.org

by Jean Garutti -- Angular 4 -- 2

info@vertika.org

Je répète, je répète



by Jean Garutti -- Angular 4 -- 3

ngFor

- C'est une directive qui produit un élément de tableau,
 - itérant à travers chaque élément à l'intérieur.

by Jean Garutti

-- Angular 4 --

-- 4

ngFor, classes multiples

- Pour des classes multiples, vous pouvez créer un tableau et mettre ces classes *à l'intérieur* pour appliquer à votre liste itérative.
- Vous pouvez ensuite combiner cela avec ***ngFor**

by Jean Garutti

-- Angular 4 --

-- 5

ngFor

1. `<div *ngFor = "let whiskey of whiskeyCat" (click) = "setSelectedWhiskey(whiskey)" class="product">`
- 2.

by Jean Garutti

-- Angular 4 --

-- 6

ngFor

```

2.
3.     
4.     <h1> {{whiskey.whiskeyName}} </h1>
5.     <h1> {{whiskey.price}} </h1>
6.     {{whiskey.description}}
7.     {{whiskey.rating}}
8.     <span> <a (click) ="selectWhiskey(whiskey)"
    class ="ion-edit"> </a> </span>
9. </div>

```

by Jean Garutti

-- Angular 4 --

-- 7

let

- Il fait partie de la syntaxe du modèle Angular.
- Il crée une variable locale qui peut être référencée n'importe où dans ce modèle spécifique.

by Jean Garutti

-- Angular 4 --

-- 8

let

- Vous créez l'**index i** et il le conserve comme référence au **whisky** actuel.
- Tout cela se fait sous le couvert de cette directive.

by Jean Garutti

-- Angular 4 --

-- 9

let

- Et si vous vouliez styliser le premier et le dernier élément de votre directive **ngFor** ?

Let le premier et le dernier élément

- Ceci est relativement simple en utilisant **let** de nouveau pour affecter les premiers éléments à la variable **first**
- Et **last** pour la dernière variable.

Let

1. ``
2. `<li *ngFor="let item of items;
let i = index;
let firstItem = first;
let lastItem = last" [ngClass]="{
active: firstItem }">`
3. `{{ i }} {{ firstItem }} {{
lastItem }} {{ item }}`
4. ``
5. ``

*ngFor et iterables

- Que faire si vous avez un tableau
 - ou un *ensemble* d'objets ?
- Comment les afficher dans un modèle?
- La directive ***ngFor** est pratique.
- Voici comment afficher un tableau :

by Jean Garutti

-- Angular 4 --

-- 13

Interpolation, d'un Array simple

```

1. @Component({
2.   // Other component properties removed.
3.   template: `
4.     <h1>Hey guys!</h1>
5.     <ul>
6.       <li *ngFor="let arr of myArr">{{ arr }}
7.     </li>
8.   </ul>
9.   `,
10. })

11. export class AppComponent {
12.   myArr = ['him', 'hers', 'yours', 'theirs'];
13. }

```

by Jean Garutti

-- Angular 4 --

-- 14

*ngIf

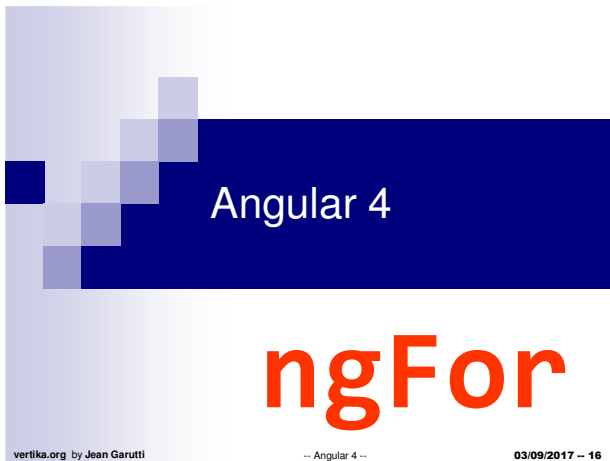
- Si vous voulez afficher un **Template** donné seulement si une expression correspond à une certaine condition.
- Dans ces cas, vous utiliserez la directive ***ngIf**.
- exemple :

```
<li *ngIf = "myArr"> Ouais, j'existe. </li>
```

by Jean Garutti

-- Angular 4 --

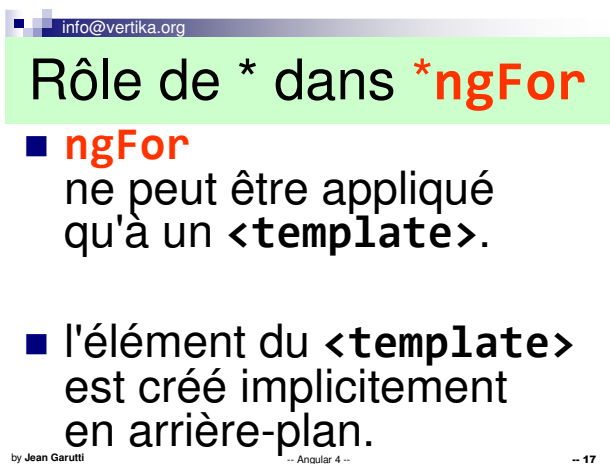
-- 15



Angular 4

ngFor

vertika.org by Jean Garutti -- Angular 4 -- 03/09/2017 -- 16

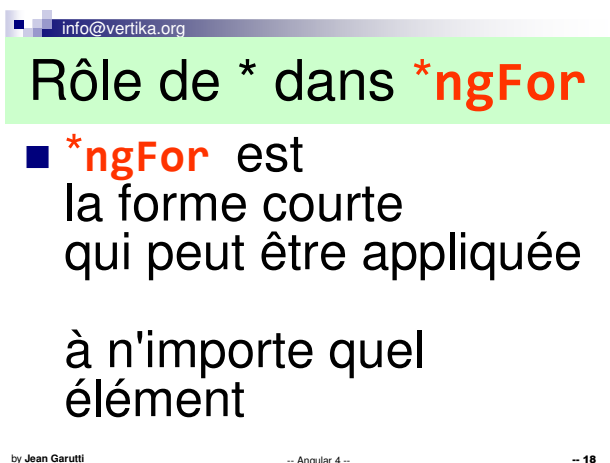


info@vertika.org

Rôle de * dans *ngFor

- **ngFor** ne peut être appliqué qu'à un `<template>`.
- l'élément du `<template>` est créé implicitement en arrière-plan.

by Jean Garutti -- Angular 4 -- 17



info@vertika.org

Rôle de * dans *ngFor

- ***ngFor** est la forme courte qui peut être appliquée à n'importe quel élément

by Jean Garutti -- Angular 4 -- 18

Rôle de * dans *ngFor

- Lorsque Angular voit l'astérisque (*) dans **ngFor**, il utilise son élément DOM

en tant que **template**

pour rendre la boucle.

Rôle de * dans *ngFor

1. `<div *ngFor="#personne of personnes">`
2. `{{ personne.name }}`
3. `</div>`

est équivalent à

1. `<template ngFor #personne [ngForOf]="personnes">`
2. `<div>`
3. `{{ personne.name }}`
4. `</div>`
5. `</template>`

ngFor

- C'est une **directive structurelle**, ce qui signifie qu'elle modifie la structure du DOM.

ngFor

- Il est préférable de répéter un modèle HTML donné
 - ☐ une fois pour chaque valeur dans un tableau,
 - ☐ chaque fois qu'il passe la valeur du tableau
 - ☐ comme contexte d'interpolation
 - ☐ ou de liaison de chaîne.

by Jean Garutti

-- Angular 4 --

-- 22

ngFor

- Pour être aussi efficace que possible,
 - ☐ **ngFor** tentera de ne pas supprimer puis recréer ces éléments DOM
 - ☐ mais plutôt de réutiliser et de recycler chaque fois que possible.

by Jean Garutti

-- Angular 4 --

-- 23

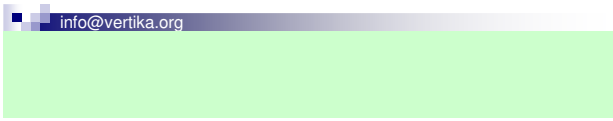
ngFor

- Cela contribue à la performance et empêche un ralentissement significatif dans votre DOM.

by Jean Garutti

-- Angular 4 --

-- 24



Merci !

Thanks !
