

# III-Directives



Dans l'architecture Angular, les directives sont finalement partout.

Les Components sont des directives à la seule différence qu'ils possèdent une fonctionnalité de templating. En TypeScript, une directive est une classe à laquelle on applique le décorateur @Directive.

Il existe deux sortes de directives :

- Les directives structurelles : Elles ont pour but de modifier le DOM en ajoutant, enlevant ou remplaçant un élément du DOM. (exemple: ngIf, ngFor)
- Les directives d'attribut: Elles ont pour but de modifier l'apparence ou le comportement d'un élément (exemple: ngStyle)

LES DIRECTIVES ( \*ngIf et \*ngFor )

Les directives sont des instructions intégrées dans le DOM que vous utiliserez presque systématiquement quand vous créerez des applications Angular.

Dans ce cours, nous parlerons des **directives structurelles**. Ce sont des directives qui, comme leur nom l'indique, modifient la structure du document.

Nous verrons deux d'entre elles (il en existe d'autres) :

**\*ngIf** : pour afficher des données de façon conditionnelle

**\*ngFor** : pour itérer des données dans un array, par exemple.

## \*ngIf

Un component auquel on ajoute la directive `*ngIf="condition"` ne s'affichera que si la condition est vraie (elle retourne la valeur `true` où la variable mentionnée est définie et non-nulle), comme un statement `if` classique.

Pour une démonstration simple, ajoutez une `<div>` rouge qui ne s'affichera que si la personne est présente :

```
mon-premier-component.component.html x TS mon-premier-component.component.ts
src > app > mon-premier-component > mon-premier-component.component.html > li.list-group-item > div
1 <li class="list-group-item">
2   <div style="width:20px;height:20px;background-color: green;" *ngIf="persoStatus === 'Présent'"></div>
3   <p>Etudiant : {{ persoName }} -- Statut : {{ getStatus() }}</p>
4   <input type="text" class="form-control" [(ngModel)]="persoName">
5 </li>
6
```

## Résultat:

## Liste



Etudiant : Emmanuelle -- Statut : Présent

Emmanuelle

Etudiant : Robin -- Statut : Absent

Robin



Etudiant : Dominique -- Statut : Présent

Dominique

☒ Utilisateur connecté

## \*ngFor

Lorsque l'on ajoute la directive `*ngFor="let obj of myArray"` à un component, Angular itérera l'array `myArray` et affichera un component par objet `obj`.

Créons un tableau directement dans `AppComponent` (voir image de droite)

Nous avons un array avec trois objets, chaque objet ayant une propriété `name` et une propriété `status`.

```
src > app > Ts app.component.ts > AppComponent > presenceTab
1  import { Component, Input } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'mon-premier-projet';
10   isAuthenticated = true;
11   onClick() {
12     alert('Cliqué !');
13   }
14
15   presenceTab = [
16     {
17       name: 'Emmanuelle',
18       status: 'Présent'
19     },
20     {
21       name: 'Robin',
22       status: 'Absent'
23     },
24     {
25       name: 'Dominique',
26       status: 'Présent'
27     }
28   ];
29
30 }
```

> app.component.html

```
src > app > app.component.html > div.container > div.row > div.col-xs-12 > ul.list-group > app-mon-premier-component
1  <div class="container">
2    <div class="row">
3      <div class="col-xs-12">
4        <h2>Liste</h2>
5        <ul class="list-group">
6          <app-mon-premier-component *ngFor="let element of presenceTab"
7                                     [persoName]="element.name"
8                                     [persoStatus]="element.status"></app-mon-premier-component >
9
10       </ul>
11       <input type="checkbox" [checked]="isAuth" (click)="onClique()">Utilisateur connecté
12     </div>
13   </div>
14 </div>
```

Le statement `let element of presenceTab`, comme dans une boucle `for` classique, itère pour chaque élément `element` (nom arbitraire) de l'array `presenceTab`. Après cette directive, vous pouvez maintenant utiliser l'objet `element`, dont vous connaissez la forme, à l'intérieur de cette balise HTML. Vous pouvez donc utiliser le `property binding`, et y passer les propriétés `name` et `status` de cet objet.

Remarque: N'oubliez pas l'astérisque devant ces directives, qui signifie à Angular de les traiter comme directives structurelles !

Résultat :

## Liste



Etudiant : Emmanuelle -- Statut : Présent

Emmanuelle

Etudiant : Robin -- Statut : Absent

Robin



Etudiant : Dominique -- Statut : Présent

Dominique

☒ Utilisateur connecté