

# Cours Angular : Directives et Control Flow

## 1. Introduction

Les **directives** permettent de modifier le comportement ou l'apparence du DOM. Le **control flow** sert à contrôler l'affichage du template (conditions, boucles, alternatives).

Dans Angular moderne, le control flow repose à la fois sur : - Les **directives structurelles classiques** (`*ngIf`, `*ngFor`, `*ngSwitch`) - Le **nouveau control flow Angular** (`@if`, `@for`, `@switch`) introduit dans les versions récentes.

Objectifs : - Comprendre les directives Angular - Maîtriser les directives structurelles et d'attribut - Comprendre et utiliser le control flow classique et moderne - Appliquer ces notions dans des cas pratiques

---

## 2. Les directives Angular

### 2.1 Définition

Une directive est une classe qui ajoute un comportement à un élément du DOM.

Angular distingue **3 types de directives** : 1. Directives de composants 2. Directives structurelles 3. Directives d'attribut

---

## 3. Directives structurelles

### 3.1 Principe

Les directives structurelles modifient la **structure du DOM** (ajout, suppression, répétition d'éléments).

👉 Elles sont reconnaissables par le symbole `*`.

---

### 3.2 `*ngIf`

Permet d'afficher ou masquer un élément selon une condition.

```
<p *ngIf="isLoggedIn">Bienvenue utilisateur</p>
```

Avec `else` :

```
<p *ngIf="isLoggedIn; else notConnected">Bienvenue</p>
```

```
<ng-template #notConnected>
  <p>Veuillez vous connecter</p>
</ng-template>
```

### 3.3 \*ngFor

Permet de parcourir une liste.

```
users = ['Ali', 'Fatou', 'Jean'];
```

```
<ul>
  <li *ngFor="let user of users; index as i">
    {{ i + 1 }} - {{ user }}
  </li>
</ul>
```

Variables utiles : - `index` - `first`, `last` - `even`, `odd`

### 3.4 \*ngSwitch

Alternative à plusieurs `ngIf`.

```
<div [ngSwitch]="role">
  <p *ngSwitchCase="'admin'">Administrateur</p>
  <p *ngSwitchCase="'user'">Utilisateur</p>
  <p *ngSwitchDefault>Invité</p>
</div>
```

## 4. Directives d'attribut

### 4.1 Principe

Elles modifient l'apparence ou le comportement d'un élément **sans changer sa structure**.

### 4.2 ngClass

Ajoute ou supprime des classes CSS dynamiquement.

```
<p [ngClass]="{ 'active': isActive, 'error': hasError }">  
  Texte dynamique  
</p>
```

### 4.3 ngStyle

Applique des styles dynamiques.

```
<p [ngStyle]="{ color: textColor, 'font-size.px': fontSize }">  
  Style dynamique  
</p>
```

## 5. Control Flow Angular (nouvelle syntaxe)

### 5.1 Pourquoi un nouveau control flow ?

Angular introduit une syntaxe plus : - Lisible - Performante - Proche de JavaScript

👉 Elle remplace progressivement `*ngIf`, `*ngFor`, `*ngSwitch`.

## 6. @if – Conditions

```
@if (isLoggedIn) {  
  <p>Bienvenue</p>  
} @else {  
  <p>Connexion requise</p>  
}
```

Avec `@else if` :

```
@if (role === 'admin') {  
  <p>Admin</p>  
} @else if (role === 'user') {  
  <p>User</p>  
} @else {  
  <p>Invité</p>  
}
```

## 7. @for – Boucles

```
products = ['PC', 'Téléphone', 'Tablette'];
```

```
@for (product of products; track product) {  
  <p>{{ product }}</p>  
}
```

Avec index :

```
@for (product of products; let i = $index) {  
  <p>{{ i + 1 }} - {{ product }}</p>  
}
```

## 8. @switch – Alternatives multiples

```
@switch (status) {  
  @case ('success') {  
    <p>Succès</p>  
  }  
  @case ('error') {  
    <p>Erreur</p>  
  }  
  @default {  
    <p>En attente</p>  
  }  
}
```

## 9. Comparaison ancienne / nouvelle syntaxe

Classique	Nouveau control flow
*ngIf	@if
*ngFor	@for
*ngSwitch	@switch

👉 Recommandé pour les nouveaux projets Angular.

## 10. Cas pratique complet

### Objectif

Afficher une liste d'utilisateurs uniquement si elle existe.

```
users = ['Ali', 'Fatou', 'Jean'];
```

```
@if (users.length > 0) {  
  @for (user of users; let i = $index) {  
    <p>{{ i + 1 }} - {{ user }}</p>  
  }  
} @else {  
  <p>Aucun utilisateur</p>  
}
```

---

## 11. Bonnes pratiques

- Préférer le nouveau control flow ( `@if` , `@for` )
- Toujours utiliser `track` dans `@for` pour la performance
- Éviter les conditions complexes dans le template
- Déplacer la logique métier dans le composant

---

## 12. Résumé

- Les directives contrôlent le DOM
- Directives structurelles : conditions, boucles, switch
- Directives d'attribut : styles et classes dynamiques
- Le nouveau control flow Angular est plus clair et performant

---

👉 Prochaine étape recommandée : **@Input / @Output, pipes et services**