

# Guide Angular avec Exemples

## 1. Étapes de création d'un composant dans Angular

Étape 1 : Générer un composant

Utilisez la commande Angular CLI pour générer un composant :

```
ng generate component nom-du-composant
```

Exemple :

```
ng generate component user-profile
```

Étape 2 : Définir la logique dans le fichier TypeScript :

```
import { Component } from '@angular/core';
```

```
@Component({  
  selector: 'app-user-profile',  
  templateUrl: './user-profile.component.html',  
  styleUrls: ['./user-profile.component.css']  
})
```

```
export class UserProfileComponent {  
  user = { name: 'John Doe', age: 25 };  
}
```

Étape 3 : Créer la vue dans le fichier HTML :

```
<div>  
  <h1>Profile de {{ user.name }}</h1>  
  <p>Âge : {{ user.age }}</p>
```

</div>

Étape 4 : Ajouter du style dans le fichier CSS :

```
h1 {{ color: blue; }}
```

Étape 5 : Utiliser le composant :

```
<app-user-profile></app-user-profile>
```

## 2. Utilisation d'un composant

Pour utiliser un composant, insérez simplement son sélecteur HTML dans un autre composant :

```
<app-user-profile></app-user-profile>
```

## 3. Comprendre le Data Binding

Interpolation :

```
<p>{{ user.name }}</p>
```

Property Binding :

```
<img [src]="user.avatarUrl" />
```

Event Binding :

```
<button (click)="greetUser()">Say Hello</button>
```

Two-way Binding :

```
<input [(ngModel)]="user.name" />
```

```
<p>{{ user.name }}</p>
```

## 4. Comprendre le Property Binding

Dans le fichier TypeScript :

```
userAvatar = 'https://example.com/avatar.jpg';
```

Dans le fichier HTML :

```
<img [src]="userAvatar" alt="User Avatar" />
```

## 5. Comprendre l'Event Binding

Dans le fichier TypeScript :

```
greetUser() {  
    alert('Hello, user!');  
}
```

Dans le fichier HTML :

```
<button (click)="greetUser()">Click Me</button>
```

## 6. Utiliser des modèles pour un code plus propre

Étape 1 : Créer un modèle :

```
export class User {  
    constructor(  
        public name: string,  
        public age: number,  
        public email: string  
    ) {}
```

```
}
```

Étape 2 : Utiliser le modèle dans le composant :

```
import { User } from './user.model';
```

```
@Component({
```

```
  selector: 'app-user-profile',
```

```
  templateUrl: './user-profile.component.html',
```

```
  styleUrls: ['./user-profile.component.css']
```

```
})
```

```
export class UserProfileComponent {
```

```
  user: User = new User('John Doe', 25, 'john.doe@example.com');
```

```
}
```