

Angular 24-25

Workshop 2

Tâche 1 : Configurer l'environnement Angular

1. **Objectif** : Installer les outils nécessaires pour commencer à travailler avec Angular.

2. **Instructions** :

- Installez **Node.js** (si ce n'est pas déjà fait) : [Node.js](#)
- Installez Angular CLI en exécutant cette commande :

```
npm install -g @angular/cli
```

- Créez un nouveau projet Angular en utilisant Angular CLI :

```
ng new my-angular-app
```

- Naviguez dans le répertoire de votre projet : `cd my-angular-app`
- Lancez le serveur de développement : `ng serve`
- Ouvrez `http://localhost:4200` dans un navigateur.

Tâche 2 : Créer votre premier composant

1. **Objectif** : Créer un nouveau composant pour afficher des informations sur un utilisateur.

2. **Instructions** :

- Générez un nouveau composant `user-profile` en utilisant Angular CLI :

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

- Ouvrez le fichier `user-profile.component.ts` et ajoutez un objet `user` avec les propriétés `name` et `age` :

```
export class UserProfileComponent {
```

```
  user = { name: 'John Doe', age: 30
```

```
};
```

```
}
```

- Mettez à jour le fichier `user-profile.component.html` pour afficher les informations de l'utilisateur :

```
<h1>Nom : {{ user.name }}</h1>
```

```
<p>Âge : {{ user.age }}</p>
```

Tâche 3 : Utiliser le composant dans un autre composant

1. **Objectif** : Afficher le composant `user-profile` dans le composant principal.
2. **Instructions** :
 - Ouvrez le fichier `app.component.html` (composant racine).
 - Utilisez le sélecteur de votre nouveau composant pour l'afficher :

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

Vérifiez que les informations de l'utilisateur s'affichent correctement dans le navigateur.

Tâche 4 : Comprendre le Data Binding

1. **Objectif** : Apprendre à utiliser les différents types de binding.
2. **Instructions** :
 - **Interpolation** : Mettez à jour `user-profile.component.html` pour afficher le nom de l'utilisateur en utilisant l'interpolation.

```
<h1>{{ user.name }}</h1>
```

- **Property Binding** : Ajoutez une image dans `user-profile.component.html` et liez la propriété `src` à une URL d'image :

```
<img [src]="\"https://example.com/avatar.png\" alt=\"Avatar\" />
```

- **Two-way Data Binding** : Ajoutez un champ de saisie qui modifie dynamiquement le nom de l'utilisateur en utilisant `ngModel` :

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

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

Tâche 5 : Event Binding

1. **Objectif** : Gérer les événements utilisateur tels que les clics de boutons.
2. **Instructions** :
 - Dans `user-profile.component.ts`, ajoutez une méthode `greetUser()` qui affiche un message dans la console :

```
greetUser() { console.log('Hello, ${this.user.name}!'); }
```

- Dans `user-profile.component.html`, ajoutez un bouton qui appelle cette méthode lorsque vous cliquez dessus :

```
<button (click)="greetUser()">Greet User</button>
```

- Ouvrez la console du navigateur et cliquez sur le bouton pour voir le message.

Tâche 6 : Utiliser des modèles pour un code plus propre

1. **Objectif** : Organiser les données avec des modèles (classes) pour améliorer la structure du code.
2. **Instructions** :
 - Créez un fichier `user.model.ts` dans le dossier `src/app` et définissez un modèle `User` :

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

- Mettez à jour le fichier `user-profile.component.ts` pour utiliser le modèle `User` :

```
import { User } from './user.model';  
  
export class UserProfileComponent {  
  
  user: User = new User('John Doe', 30, 'john.doe@example.com');  
  
}
```

- Dans `user-profile.component.html`, ajoutez un paragraphe pour afficher l'email de l'utilisateur :

```
<p>Email : {{ user.email }}</p>
```

Tâche 7 : Aller plus loin avec le Property Binding

1. **Objectif** : Travailler avec des propriétés plus complexes et les lier à des éléments HTML.
2. **Instructions** :

- Ajoutez un bouton désactivé par défaut et activez-le uniquement si l'utilisateur a plus de 18 ans :

```
<button [disabled]="user.age < 18">Submit</button>
```

Tâche 8 : Tester vos composants (Bonus)

1. **Objectif** : Comprendre les tests unitaires dans Angular.
2. **Instructions** :
 - Explorez le fichier `user-profile.component.spec.ts` généré automatiquement.
 - Ajoutez un test pour vérifier que le composant est créé correctement.