Exercice 02 — Typed forms & ControlValueAccessor

Objectif

Créer un `SharedSelectComponent` entièrement typé et compatible Angular Forms.

Etapes

1. **Créer le composant**

```
ng generate component shared-select --project=shared-ui --inline-style --
inline-template
```

2. **Définir l'API**

```
@Component({
  selector: "shared-select",
  standalone: true,
  imports: [NgFor, NgIf],
  templateUrl: "./shared-select.component.html",
  changeDetection: ChangeDetectionStrategy.OnPush,
  providers: [
    {
      provide: NG_VALUE_ACCESSOR,
      useExisting: forwardRef(() => SharedSelectComponent),
      multi: true,
    },
  ],
})
export class SharedSelectComponent<T> implements ControlValueAccessor {
  @Input({ required: true }) options: ReadonlyArray<{ label: string; value: T
}> = [];
  @Input() placeholder = "Sélectionner";
  // ...
}
```

3. **Implémenter le CVA**

   - `writeValue(value: T | null)` conserve la valeur.
   - `registerOnChange` et `onTouched` à stocker via `noop` initiaux.
   - Supporter `setDisabledState` (aria-disabled + classes).

4. **Typed forms**

   - Utiliser `FormBuilder.nonNullable` dans un composant host :

```
readonly form = this.fb.nonNullable.group({
  country: this.fb.control<string | null>(null, Validators.required),
```

```
});
```

- Lier `form.controls.country` à `shared-select`.

5. **Gestion des erreurs & a11y**
   - Ajoutez `aria-invalid`, `aria-describedby`.
   - Affichez les erreurs dans un `<p role="alert">`.

Résultats attendus

- Le sélecteur supporte n'importe quel type `T`.
- Les formulaires typés ne déclenchent aucun `any` / `unknown` implicite.
- Les validations s'affichent uniquement si le control est `touched`.

Pour aller plus loin

- Ajouter la navigation clavier (flèches, entrée) pour une meilleur UX.
- Supporter les `OptionGroup` (libellés de groupe) si besoin.

Consultez la solution dans `exercices/solutions/02-forms-typed.md` si nécessaire.