

Utilisation de ControlValueAccessor

Pourquoi ?

Si souhaitons créer un élément de formulaire personnalisé et souhaitons l'utiliser avec FormsModule ou ReactiveFormsModule

Par exemple:

```
<app-custom-field [(ngModel)]="val"></app-custom-field>
```

Créer le composant

```
@Component({
  selector: 'app-custom-field',
  template: '',
  providers: [
    {
      provide: NG_VALUE_ACCESSOR,
      useExisting: FieldComponent,
      multi: true
    }
  ]
})
export class FieldComponent implements ControlValueAccessor {
}
```

Le schéma de classe

```
export class FieldComponent implements ControlValueAccessor {

  private _value: any = ''
  @Input() disabled = false

  onChange = (val: any) => {};
  onTouch = () => {}

  // Appel de la méthode sur click d'un bouton (exemple)
  action(val: any) {
    if (!this.disabled) {
      this.writeValue(val)
    }
  }

  get value() {
    return this._value
  }
}
```

```
set value(val) {
    this._value = val
}

// Lors de l'écriture, quelles sont les autres instructions
writeValue(val: any): void {
    this.value = val
    this.onChange(val)
}

// On met un callback dans une propriété
registerOnChange(fn: (val: any) => void): void {
    this.onChange = fn;
}

// On met un callback dans une propriété
registerOnTouched(fn: () => void): void {
    this.onTouched = fn;
}

setDisabledState(isDisabled: boolean): void {
    this.disabled = isDisabled;
}
}
```