Fiche récap : Exploitez les Signals pour créer des interfaces réactives avec Angular

Développement

1. Découvrir les Signals pour créer des états réactifs simples.
2. Transférer des données avec les inputs entre components.
3. Faire remonter les interactions via les outputs et models.
4. Réagir aux changements grâce aux effects ciblés.

Code 💻

typescript

```typescript
// Création d'un Signal modifiable
header = signal<string>('Signalize');

// Modification avec .set et .update
setTimeout(() => this.header.set('Signalize!'), 2000);
setTimeout(() => this.header.update(h => h + '!'), 3000);

// Signal dérivé avec computed
tagline = computed(() => this._taglines[this.taglineIndex()]);

// Transmission de données avec input Signal
task = input.required<Task>();
completed = computed(() => this.task().completed);

// Réaction aux changements avec effect
constructor() {
  effect(() => console.log('Progression :', this.progressBarWidth()));
}
```

Bonnes pratiques 👍

- ✅ Créer un WritableSignal avec signal() pour gérer un état modifiable.
- ✅ Lire un Signal en l'appelant avec des parenthèses.
- ✅ Utiliser computed() pour dériver une valeur à partir d'un Signal.
- ✅ Employer input.required() pour rendre un InputSignal obligatoire.
- ✅ Utiliser output() pour faire remonter des événements enfants → parents.
- ✅ Transmettre des Signaux avec [(model)] pour un two-way binding.
- ✅ Placer les effect() dans un constructeur ou autre contexte d'injection.
- ✅ Employer @let pour simplifier l'accès aux propriétés dans les templates.

Erreurs classiques 👎

- ❌ Modifier un computed() directement — ce n'est pas un WritableSignal.
- ❌ Oublier les parenthèses en lisant un Signal dans le template.
- ❌ Utiliser effect() à tort pour modifier un Signal ou une variable.
- ❌ Ne pas rendre les input obligatoires quand nécessaires.
- ❌ Négliger de transmettre les output() jusqu'au parent.
- ❌ Utiliser model là où un simple input suffit.
- ❌ Omettre le track dans les boucles @for Angular.
- ❌ Ne pas vérifier les effets de bord potentiels dans les effect().

Définitions 🔎

Signal
Primitif réactif permettant de stocker et de lire une valeur, déclenchant la mise à jour du DOM.

WritableSignal
Signal modifiable via les méthodes .set() ou .update().

Computed
Signal calculé dynamiquement à partir d'autres Signaux.

Effect
Fonction qui réagit à toute modification des Signaux dont elle dépend.

Model
Signal partagé entre parent et enfant pour permettre une communication bidirectionnelle.