Updating

Phase de Updating

Le composant se re-rend quand ses props ou son state changent.

Avec les classes

```
componentDidUpdate(prevProps, prevState) {
  // Appelé après chaque re-render
  if (prevProps.userId !== this.props.userId) {
    // Les props ont changé → réagir
    this.fetchUser(this.props.userId);
  }

  if (prevState.count !== this.state.count) {
    // Le state a changé → réagir
    console.log("Count changé :", this.state.count);
  }
}
```

Avec les Hooks

```
// Réagir aux changements de props ou state
useEffect(() => {
  fetchUser(userId);
}, [userId]); // S'exécute quand userId change

useEffect(() => {
  console.log("Count changé :", count);
}, [count]);
```

Optimiser les re-renders

```
// Éviter les re-renders inutiles du composant enfant
const EnfantMemo = React.memo(function Enfant({ nom }) {
  return <p>{nom}</p>;
});

// shouldComponentUpdate (classe)
shouldComponentUpdate(nextProps, nextState) {
  return nextProps.value !== this.props.value;
}
```

Ordre d'exécution lors d'une mise à jour

1. Le state ou les props changent
2. React re-exécute la fonction du composant
3. React met à jour le DOM (diff)
4. Les effets dont les dépendances ont changé s'exécutent (cleanup de l'ancien effet d'abord)

Revision #1
Created 29 March 2026 12:57:33 by Hugo
Updated 29 March 2026 12:57:33 by Hugo