# React re-render rules for current component and child components

In React, understanding re-rendering rules for a component and its children is crucial for optimizing performance and writing predictable UI. Here's a breakdown of when a component re-renders and how it affects child components.

## ✅ When a Component Re-renders (Current Component)

A React component re-renders when:

* Its own state changes via useState, useReducer, or this.setState() (for class components).
* Its props change (coming from a parent component).
* Its context changes, if it consumes a context with useContext(MyContext) or <MyContext.Consumer>.
* Its parent re-renders, and the component is not memoized (React.memo for function components or PureComponent for class components).

## 🔁 How Re-renders Affect Child Components

By default:

* If a parent re-renders, all non-memoized children also re-render.
* Memoized children (with React.memo) will only re-render if their props change.

Key Rules for Children:

|  |  |
| --- | --- |
| **Child Type** | **Re-renders when...** |
| Normal component | Parent re-renders or props/context/state inside the child changes |
| React.memo | Only if **props change** or **context/state inside the child** changes |
| useMemo | Only re-evaluated if dependencies change, but doesn't control rendering directly |
| useCallback | Works with memoized children to prevent unnecessary re-renders due to new props |