Side effects

In React, a **side effect** refers to any operation that occurs outside of the scope of rendering component.

e.g of side effect

Fetching data from an API

Timers (setTimeout, setInterval)

Certain operations (like fetching data or subscribing to events) need to happen **after the render**, outside the normal data flow. Such operations are known as side effects. React provides ways to **safely** manage them without breaking the rendering process or causing unnecessary re-renders

React has a built-in mechanism to handle side effects, and the most common way of dealing with side effects in functional components is through the useEffect hook.

import { useState, useEffect } from 'react';

function App() {

  const [data, setData] = useState(null);

  useEffect(() => {

    // Side effect: Fetching data from an API

    fetch('https://api.example.com/data')

      .then(response => response.json())

      .then(data => setData(data));

  }, []);  // Empty array means this runs only once, after the first render

  return (

    <div>

      <h1>Data:</h1>

      <pre>{JSON.stringify(data, null, 2)}</pre>

    </div>

  );

}