|  |
| --- |
| import React, { useState, useEffect } from 'react'; |
| setTimeout(()=>{}, 1000)  useEffect(() => {  // Code to run after each render  }, [dependencies]);   * The function inside **useEffect** is the side effect code you want to run. * The second argument, which is an array **[dependencies]**, is optional. It specifies when the effect should run. If you pass an empty array **[]**, it runs only after the initial render. |
| import React, { useEffect, useState } from 'react';  function MyComponent() {  useEffect(() => {    console.log('Component has rendered');  }, [ ]); // Empty dependency array means this effect runs only once  return (  <div>  <h1>Hello, React!</h1>  </div>  );  } |
| import React, { useEffect, useState } from 'react';  function MyComponent() {  const [data, setData] = useState([ ]);  useEffect(() => {    fetch('https://api.example.com/data')  .then(response => response.json())  .then(data => setData(data));  }, [ ]); // Empty dependency array means this effect runs only once  return (  <div>  <h1>Data from API:</h1>  <ul>  {data.map(item => (  <li key={item.id}>{item.name}</li>  ))}  </ul>  </div>  );  } |
| import React, { useEffect, useState } from 'react';  function TimerComponent() {  const [count, setCount] = useState(0);  useEffect(() => {  const timer = setInterval(() => {  setCount(prevCount => prevCount + 1);  }, 1000);  // Cleanup function  return () => {  clearInterval(timer);  };  }, []);  return (  <div>  <p>Count: {count}</p>  </div>  );  } |