

Hooks - useEffect dependency array rules:

useEffect(() => {}, [dependencyArray])

1. **An empty dependency** array will execute the code inside the useEffect on the FIRST render of the component, so only once!
2. **No dependency array** will execute the code whenever ANY state is changed – this will likely cause problems and most likely lead to INFINITE LOOPS 🦴
🦴, at least in my experience this has happened
3. **A dependency array with a dependency inside** will fire useEffect ONLY AFTER THAT DEPENDENCY HAS BEEN CHANGED, this is the most common use case for **useEffect**. Eg, something has changed and we need to re-render the component to show that change ✓✓✓✓.

Hooks useState and useEffect simple example

```
import { useEffect, useState } from "react";

export default function App() {
  const [name, setName] = useState("Tom")
  useEffect(()=>{
    setName('David')
  },[name])
  return (
    <div>
      <h1>{name}</h1>
    </div>
  );
}
```