In **React functional components**, the persistence of variables depends on where and how they are declared:

**1️⃣ Variables Declared Inside the Functional Component**

These **reset on every re-render** because the function re-executes entirely.

**Example:**

function Counter() {

let count = 0; // Resets to 0 on every render

const increment = () => {

count++; // Modifies the variable, but doesn't persist

console.log("Count:", count);

};

return <button onClick={increment}>Increment</button>;

}

**Behavior:**

* Every time increment is clicked, count is reset to 0 before being incremented.
* The UI **never updates** because useState is not used.

➡ **Use useState instead** if you need persistence and re-renders.

**2️⃣ Variables Declared Outside the Functional Component**

These **persist across renders** but are **shared across all instances of the component** (because they are outside the function scope).

**Example:**

let count = 0; // Global variable (persists across renders and component instances)

function Counter() {

const increment = () => {

count++;

console.log("Count:", count);

};

return <button onClick={increment}>Increment</button>;

}

**Behavior:**

* count **persists across renders**.
* **But it is shared across all instances** of Counter (not component-specific).
* **Does not trigger re-renders**.

➡ **Use useRef for component-specific persistent values without triggering re-renders.**

**3️⃣ Using useRef for Persistent Variables (Without Re-Renders)**

To keep variables persistent **per component instance** without causing re-renders, use useRef.

function Counter() {

const countRef = React.useRef(0); // Persistent per component instance

const increment = () => {

countRef.current++;

console.log("Count:", countRef.current);

};

return <button onClick={increment}>Increment</button>;

}

**Behavior:**

* countRef.current **persists across renders**.
* **Component re-renders do not reset countRef.current**.
* **Does not trigger re-renders**.

**✅ When to Use What?**

| **Use Case** | **Solution** |
| --- | --- |
| Variable should persist **across renders** (per component) **without triggering re-renders** | useRef |
| Variable should persist **across renders and trigger re-renders** when changed | useState |
| Variable resets on re-renders (temporary computation) | Normal variable inside function |
| Variable shared across **all component instances** | Declare outside component |

Let me know if you need more examples! 🚀