

useRef

useRef

`useRef` retourne un objet `{ current: valeur }` qui **persiste entre les renders** sans déclencher de re-render quand il change. C'est sa différence fondamentale avec `useState`.

Syntaxe

```
const ref = useRef(valeurInitiale);  
// Accès : ref.current
```

Usage 1 : Accéder à un élément DOM

```
function InputAutofocus() {  
  const inputRef = useRef(null);  
  
  useEffect(() => {  
    inputRef.current.focus(); // Focus au montage  
  }, []);  
  
  return <input ref={inputRef} placeholder="Tapez ici..." />;  
}  
  
// Lire la valeur sans contrôler l'input (uncontrolled)  
function Form() {  
  const nameRef = useRef(null);  
  
  const handleSubmit = () => {  
    console.log(nameRef.current.value);  
  };  
  
  return (  
    <form onSubmit={handleSubmit}>
```

```
    <input ref={nameRef} />
    <button type="submit">Envoyer</button>
  </form>
);
}
```

Usage 2 : Stocker une valeur sans re-render

Idéal pour stocker des valeurs qui ne doivent pas provoquer de re-render : timers, valeurs précédentes, flags...

```
// Garder trace de l'ID d'un intervalle
function Timer() {
  const [count, setCount] = useState(0);
  const intervalRef = useRef(null);

  const start = () => {
    intervalRef.current = setInterval(() => {
      setCount(prev => prev + 1);
    }, 1000);
  };

  const stop = () => clearInterval(intervalRef.current);

  return (
    <div>
      <p>{count}</p>
      <button onClick={start}>Démarrer</button>
      <button onClick={stop}>Arrêter</button>
    </div>
  );
}
```

Usage 3 : Capturer la valeur précédente d'un state

```
function usePrevious(value) {
  const ref = useRef();
```

```
useEffect(() => {
  ref.current = value; // Met à jour APRÈS le render
});
return ref.current; // Retourne la valeur PRÉCÉDENTE
}

function Compteur() {
  const [count, setCount] = useState(0);
  const prevCount = usePrevious(count);

  return <p>Avant: {prevCount} | Maintenant: {count}</p>;
}
```

useRef vs useState

	useRef	useState
Persiste entre renders	☑	☐
Déclenche un re-render	☐	☑
Accès au DOM	☑	☐
Valeur mutable	☑ (ref.current)	☑ (setter requis)

Revision #1

Created 29 March 2026 12:57:35 by Hugo

Updated 29 March 2026 12:57:35 by Hugo