

SUMÁRIO

1. PERSISTINDO VALORES SEM RENDERIZAÇÃO.....	1
1.1. CASOS DE USO	1
2. USEREF PARA ALTERAR A DOM.....	1
2.1. TRANSMISSÃO DE REFS COM PROPS.....	2
3. FONTES	3

1. PERSISTINDO VALORES SEM RENDERIZAÇÃO

O Hook [useRef](#) é como o useState no sentido de que persiste valores, mas diferentes no que tange a sua renderização: seu valor não é renderizado.

No caso do useState, os valores persistidos, mesmo diante de mudanças, e que chamamos de estado, são renderizáveis, isto é, devem servir para fins visuais, ou visualizáveis imediatamente a cada mudança. Por um exemplo, um clássico contador reativo criado com useState, é um que, a cada alteração de valor, é renderizado novamente para que seja impresso com seu novo valor de estado.

Em contraste, no caso do useRef, o mesmo contador pode ser criado, mas o seu valor não é renderizado na tela a cada atualização. É sim, atualizado, o seu valor, mas não é impresso na tela.

E qual seria o propósito de utilizar isso? Um state que não é renderizado imediatamente? Existem alguns casos específicos de uso recomendados na documentação, mas alguns desenvolvedores o consideram uma alternativa para ganho de performance. Como o useRef não provoca uma nova renderização do componente em que é utilizado, não é necessário o processamento adicional de renderização que o useState demanda.

1.1. CASOS DE USO

Por ser mais performático, seria preferível em todos os casos em que a renderização não é necessária? Como já dito, não é recomendado essa forma de uso. Na própria **documentação** do React, na seção do useRef, **o uso de refs é aconselhado para apenas algumas situações**.

Clique [aqui](#) para ler sobre.

2. USEREF PARA ALTERAR A DOM

O useRef também pode ser utilizado para modificar elementos da DOM sem que sejam necessários os seletores convencionais do javascript. É possível referenciar um elemento HTML com useRef e utilizar o referencial para manipula-lo.

Para ligar o elemento HTML ao valor referencial, é preciso atribuir o Ref como valor do seu atributo “ref = {}”. Veja abaixo:

```
const valorRef = useRef();  
  
return(  
  <  
    <input type = "text" ref = {valorRef}  
  >  
>  
)
```

Considerando o exemplo acima, agora o elemento `<input>` pode ser modificado a partir da variável ref “valorRef”. Entende-se desta forma: o elemento input faz referência ao “valorRef”.

Para acessar o valor de um ref, quebrando expectativas, não basta utilizar a própria variável, que neste caso é “valorRef” mas sim, também, o atributo `.current`, desta forma: `valorRef.current`.

2.1. TRANSMISSÃO DE REFS COM PROPS

Sendo direto e breve, sim, é possível, como seria presumível, transmitir refs entre componentes, da mesma forma que também é com states. O funcionamento da transmissão de valores é o mesmo.

3. FONTES

<https://pt-br.reactjs.org/docs/hooks-reference.html#useref>

