**Para saber mais: Closures e o map**

No vídeo passado, precisamos criar um bloco para que pudéssemos utilizar o console.log, e utilizamos chaves para criar este bloco.

Tá Luiz, mas o que isto tem a ver com o título deste para saber mais? Vamos ver o código da função .map que temos:

{colaboradores.**map**((colaborador, indice) => <Colaborador key={indice} colaborador={colaborador} corDeFundo={time.corSecundaria} />)}

Primeiro destrincharemos esta linha de código e depois falaremos especificamente sobre o que ela tem a ver com o título.

1. Como estamos utilizando o .map, podemos deduzir que colaboradores é um **array**;
2. O **.map** aceita uma função como parâmetro, no nosso caso utilizamos uma arrow function para isto;
3. A arrow function não necessita de um nome, ou seja, ela é uma **função anônima**;
4. Como não temos chaves após a arrow(=>), sabemos que ela não tem um bloco de código, apenas o retorno;
5. Como não temos um bloco, sabemos que o retorno dela é o componente Colaborador.

Beleza, agora que conseguimos ter uma visão melhor, explica o que é closure!

Uma closure é um bloco dentro de uma função que nos permite colocar alguma informação dentro, ou seja, é o mesmo que um “bloco”, que é o que precisamos para colocar o console.log!

Veja estes 2 códigos:

<Componente onClick={() => executarFuncao()}

<Componente onClick={() => { executarFuncao() }}

Eles parecem muito parecidos, mas são diferentes! A diferença é que nesta arrow function, a primeira (sem as chaves) retorna a função e a segunda só a executa!

Na prática, isto não influencia em nada nesta parte do código, mas e aqui?

{colaboradores.**map**((colaborador, indice) => { <Colaborador key={indice} colaborador={colaborador} corDeFundo={time.corSecundaria} /> })}

O que acontece neste caso?

Neste caso, o código não funcionaria! O map necessita de um retorno, pois a diferença entre o map e o forEach é que o map muda o item de cada array, então se ele não recebe nenhum retorno, o map interpreta todos os itens como undefined, então, caso você queira utilizar um bloco (ou uma closure) dentro do map, você é **obrigado** a retornar o que você quer que ele mostre.

Beleza Luiz, mas no código não tá escrito return, como eu estou retornando algo que eu não digo que estou retornando?

Aí que está a magia da coisa, você só precisa dizer que está retornando algo se você precisar utilizar a closure! Caso você não queira, apenas utilizando parênteses ou simplesmente não colocando nada, você já diz que está retornando!

Veja este código:

pessoas.**map**(pessoa=> (

<Pessoa nome={pessoa.nome} />

))

Os parênteses neste caso são iguais a não colocar nada, mas ele permite que você tenha um **return** de mais de uma linha! Porém como não temos uma closure, podemos apenas colocar os parênteses, sem o return! Caso você queira fazer algum cálculo, pode fazer algo assim:

pessoas.**map**(pessoa => {

**const** maiorDeIdade = pessoa.idade > 18;

**return** (

<Pessoa nome={pessoa.nome} maiorDeIdade={maiorDeIdade} />

)

})

Neste caso, você está utilizando o closure (as chaves) e um return com mais de uma linha (com os parênteses) de uma vez só! Agora olhe o mesmo código:

* Sem closure e sem return de mais de uma linha:

pessoas.**map**(pessoa => <Pessoa nome={pessoa.nome} maiorDeIdade={pessoa.idade > 18} />)

* Sem closure e com return de mais de uma linha:

pessoas.**map**(pessoa => (

<Pessoa nome={pessoa.nome} maiorDeIdade={pessoa.idade > 18} />

))

Então as chaves, os parênteses ou nada têm significado neste caso, e você agora sabe de todos eles!

Caso tenha alguma dúvida sobre isso, sinta-se à vontade para perguntar no fórum!

Bons estudos!