**Para saber mais: renderização condicional**

No vídeo passado, precisamos criar um ternário para renderizar dois ícones de coração, um para favorito e um para não favorito, isto é chamado de conditional rendering, ou em português *renderização condicional*!

Isto é muito comum e muito importante dentro do React, pois com isto você pode renderizar qualquer coisa baseada em algum estado criado

Vamos ver alguns tipos de renderização condicional:

* Ternário

O ternário é bem comum e é utilizado se precisamos de uma renderização baseada em if/else, a primeira parte sempre é se a variável é true, e a segunda é se a variável é false.

{condicional ? ‘a condicional é **true**!’ : ‘a condicional é **false**!’}

* Operador lógico

Mas e se eu quiser renderizar apenas se for true? Para isto você pode utilizar o operador &&, que é o operador AND no Javascript, ou seja, só funcionará se a variável for true.

{condicao && outraCondicao && ‘todas **as** condições anteriores são **true**!’}

* if/else

O if/else não é aceito dentro do return, mas você pode utilizar fora do return!

**if**(!estaLogado) {

**return** (

<div>

você não está logado!

</div>

)

} **else** {

**return** (

<div>

você está logado!

</div>

)

}

**OBS:** neste caso você nem precisaria do else, pois o return de baixo só executaria se o if for false.

* Variável

No if/else, você não poderá executar o return de baixo caso o if seja true, para resolver este problema você pode colocar o valor dentro de uma variável, e colocar esta variável dentro do return!

**let** logado;

**if**(estaLogado) {

logado = ‘**Ol**á, você está logado!’

} **else** {

logado = ‘**Ol**á, você não está logado! :(’

}

**return** (

<div>

{logado}

</div>

)

* Função

Caso você queira fazer alguma comparação, poderá colocar esta condição dentro de uma função passando algum parâmetro!

**function** **verificaLogado**(logado) {

**if**(logado) {

**return** ‘**Ol**á, você está logado!’

}

**return** ‘**Ol**á, você não está logado! :(’

}

**return** (

<div>

{verificaLogado(estaLogado)}

</div>

)

**OBS:** na função, o que aparecerá na tela é o return, então não esqueça de sempre ter um return na função, dentro e fora das condições!