

React JS

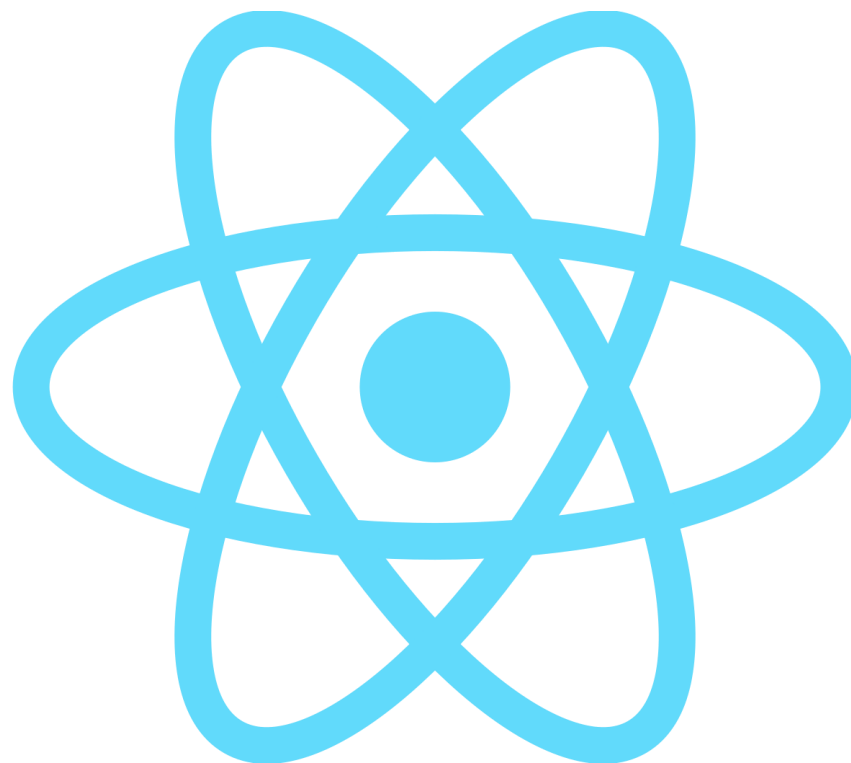
Aula 04

Instrutor: Rodolfo Moura

Intro

Revisão sobre
componentes

Realização da
atividade proposta 1



PROPS

O que são?

Valores passados para os componentes

Dinamicidade dos componentes

Mudança do valor para execução

Passado como atributo para componente

Necessário resgatar dentro de uma propriedade/argumento chamada props na função de definição do componente.

Props são somente de leitura

PROPS

Criação de componente e função

Adição de props dentro do parênteses da função

Adicionamos a propriedade {props.propriedade} dentro do código.

Exportamos a componente normalmente

```
function Item2(props) {
```

```
<div>  
  <h1>Componente teste {props.nome}</h1>  
</div>
```

PROPS

“Chamamos” a propriedade dentro da tag que chamamos a componente (no arquivo app.js), colocando em aspas o valor desejado.

Replicando o componente e modificando sua propriedade teremos algo diferente cada vez

```
<Item2 nome="Extra" />
```

Edit src/App.js and save to reload.

Componente teste Extra

```
<Item2 nome="Extra" />  
<Item2 nome="Repetido" />  
<Item2 nome="De novo" />
```

Edit src/App.js and save to reload.

Componente teste Extra
Componente teste Repetido
Componente teste De novo

PROPS

Podemos colocar também dados dinâmicos, e chamá-los da maneira como normalmente chamamos variáveis no React.

```
const nome = "Rodolfo"
```

```
<Item2 nome={nome} />
```

Também podemos trabalhar com componentes que possuem mais de uma propriedade.

PROPS

Podemos colocar também dados dinâmicos, e chamá-los da maneira como normalmente chamamos variáveis no React.

```
const nome = "Rodolfo"
```

```
<Item2 nome={nome} />
```

Também podemos trabalhar com componentes que possuem mais de uma propriedade.

```
<Pessoa nome="Rodolfo" idade="30" />
```

PROPS

Em caso de um objeto com muitas propriedades, podemos também chamar suas propriedades diretamente no () da função, sem mencionar “props” posteriormente no corpo do código.

Tratamos como um objeto, inclusive no uso de {}

```
function Pessoa ({nome, foto, idade, funcao})  
|   return (
```


PROPS

Além de atributos, nas nossas props nós também poderemos adicionar métodos, da mesma maneira como trabalhamos com objetos anteriormente.

Exemplo, temos a função no nosso código (em App.js):

```
function logando () {  
  | console.log("Teste")  
}
```

PROPS

Fazemos algo no nosso componente que realize a chamada de uma função, nesse caso, em um botão:

```
<button size="small" onClick={props.funcao}>Botão</button>
```

Em seguida chamamos a função da mesma maneira que chamamos nossas props, ao realizar a chamada do componente:

```
funcao={logando}/>
```