Componentes funcionais

FUTURE4

Sumário

Sumário

- O que é um componente
- Componentes em React
- Props

Componentes

FUTURE4

O que é um componente?

- **Pedaço/bloco** de interface
- Blocos **independentes**
- Permitem desenvolvimento **isolado** de cada parte da interface
- Conceitualmente, são iguais a **funções**

Lembrando funções...

Entrada
(Input)

Saída
(Output)

FUTURE4

No caso do componente...

Input de dados

Output de UI

React

FUTURE4

Dividindo em componentes

FUTURE4

Componentes no React

FUTURE4

Tipos de componente

- Componentes funcionais
  - Escritos por meio de uma **função**
  - Simples - possuem pouca lógica (por enquanto)

- Componentes de classe
  - Escritos por meio de uma classe
  - Possuem estado e ciclos de vida, permitindo lógicas complexas

Criando um componente funcional

```
function MeuComponente() {
  return (
    <div>
      <p>Este é o meu componente!</p>
    </div>
  )
}
```

FUTURE4

Criando um componente funcional - regras

- Nome SEMPRE começando com letra maiúscula
- Deve sempre retornar um JSX
- JSX deve ser englobado por uma única tag

Usando um componente

- Chamar como se fosse uma tag HTML no JSX
- Nome deve ser o mesmo, sempre com letra maiúscula
- Permite sintaxe *self-closing*
  - <Componente />

Usando um componente

```
function App() {
  return (
    <div>
      <MeuComponente />
    </div>
  );
}
```

FUTURE4

Pausa

Props

O que são e por que props?

- Componente é uma função, e como qualquer função, pode receber **parâmetros**
- Props é como chamamos os parâmetros de um componente
- Permite a criação de componentes **reutilizáveis** - dados não são estáticos

Como funciona?

- Função do componente recebe como primeiro argumento um objeto que chamamos de **props**

- Parâmetros são passados como se fossem atributos HTML

- Props são *read-only*, ou seja, não podemos mudar esse objeto, nem nenhuma de suas propriedades

Como funciona?

- Função do componente recebe como primeiro argumento um objeto que chamamos de **props**
- Parâmetros são passados como se fossem atributos HTML

- Props são ***READ-ONLY*****,** ou seja, **NÃO PODEMOS MUDAR ESSE OBJETO**, nem nenhuma de suas propriedades

Usando props

```
function App() {
  return (
    <div>
      <MeuComponente texto={'Este é o meu componente'}/>
    </div>
  );
}

function MeuComponente(props) {
  return (
    <div>
      <p>{props.texto}</p>
    </div>
  )
}
```

FUTURE4

Usando props

```
function App() {
  return (
    <div>
      <MeuComponente texto={'Este é o meu componente'}/>
    </div>
  );
}

function MeuComponente(props) {
  const textoComExclamacoes = props.texto + '!!!!'
  return (
    <div>
      <p>{textoComExclamacoes}</p>
    </div>
  )
}
```

FUTURE4

Usando props

FUTURE4

Children

- Componentes React, assim como elementos HTML, podem ter filhos
- Porém, precisamos pegar e renderizar esses filhos
- Filhos são passados em uma props especial, chamada de *children*

Children

```
function App() {
  return (
    <div>
      <MeuComponente>
        <p>Este é o meu componente</p>
      </MeuComponente>
    </div>
  );
}

function MeuComponente(props) {
  return (
    <div>
      {props.children}
    </div>
  )
}
```

FUTURE4

PropTypes

FUTURE4

Tipagem de props

- Como já vimos, variáveis JavaScript não possuem tipo definido
- Muitas vezes vamos usar componentes prontos, que outras pessoas fizeram
- Quando um app cresce muito, fica difícil saber o que cada componente espera

PropTypes

- Para resolver esse problema, foi criado o PropTypes
- Biblioteca que serve para definir e checar tipos das props
- Cada componente que recebe props deve ter sua definição de PropTypes

Instalando

```
$ npm install prop-types
```

 Lembrar de instalar toda vez que começar um projeto novo

Usando

```
import PropTypes from 'prop-types';

function MeuComponente(props) {
  const textoComExclamacoes = props.texto + '!!!!'
  return (
    <div>
      <p>{textoComExclamacoes}</p>
    </div>
  )
}

MeuComponente.propTypes = {
  texto: PropTypes.string.isRequired,
}
```

FUTURE4

Pausa

Coding Together

FUTURE4

FutureTube

Busca

- Inicio
- Em alta
- Inscrições
- Originais

- Biblioteca
- Histórico

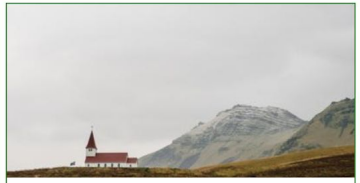
Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem



Titulo da imagem

Titulo da imagem

Titulo da imagem

Oi, eu moro no footer!

FUTURE4

FutureTube

Busca

- Inicio
- Em alta
- Inscrições
- Originais

- Biblioteca
- Histórico

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Titulo da imagem

Oi, eu moro no footer!

FUTURE4

Dúvidas?

FUTURE4

Obrigado!

FUTURE4