

transforme ■ se



Bem-vindos ao curso de Java



Sumário

- Node.js
 - O que é Node.js
 - Gerenciamento de pacotes
 - Links
- React
 - O que é React
 - Criando um projecto de React
 - Componentes
 - Docs

Node.js

O que é Node.js

Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web.

Node.js

Gerenciamento de pacotes

O gerenciamento dos pacotes é feito através do **node package manager (npm)**, e **(node package extractor) (npx)**

O primeiro tem o propósito de instalar para uso posterior código armazenado num package de nodejs, instalando o software globalmente ou localmente

Já o segundo tem o propósito de executar a nível local o código instalado globalmente. Um exemplo disso é o **npx create-react-app** que tem como propósito instalar em nível local um template vazio de um site de react, pronto a ser usado, através de uma fórmula instalada com npm.

Node.js

Links

- <https://nodejs.org/en>

React

O que é React

O **React** (também denominado React.js ou ReactJS) é uma biblioteca front-end JavaScript de código aberto com foco em criar interfaces de usuário em páginas web.

Criado em 2011 pelo Facebook (Meta), com a criação de views declarativas e baseando-se em componentes, possuía o intuito de otimizar a atualização e a sincronização de atividades simultâneas no feed de notícias da rede social e melhorar a manutenção de código.

React

Criando um projecto de React

- Abra um terminal e execute (Onde “my-app” é o nome do seu projeto):
 - *npx create-react-app my-app*
- Navegue para dentro do Projeto com:
 - `cd my-app`
- E o inicialize com:
 - `npm start`

React

Componentes

Aplicações React são feitas de componentes. Um componente é um pedaço da UI (Interface do Usuário), que tem sua própria lógica e aparência. Um componente pode ser pequeno como um botão, ou grande como uma página completa

```
function MyButton() {  
  return (  
    <button>I'm a button</button>  
  );  
}
```

Componentes em React são funções em Javascript que retornam um código HTML

React

Componentes

Com o componente criado, você pode anexá-lo em outros componentes;

```
export default function MyApp() {  
  return (  
    <div>  
      <h1>Welcome to my app</h1>  
      <MyButton />  
    </div>  
  );  
}
```

Componentes em React utilizam PascalCase para ser identificados

React

Adicionando Estilo

Em React, você especifica uma classe CSS através do atributo **className**, ela funciona da mesma maneira que o atributo **class** das TAGs HTML

```
<img className="avatar" />
```

Para importar um arquivo CSS, basta utilizar a sintaxe:

```
import './App.css';
```

React

Adicionando Lógica dentro das TAGs

Em React, se utiliza chaves `{}` para escapar do HTML e retornar ao Javascript, assim sendo possível utilizar variáveis e funções dentro do seu código HTML

```
return (  
  <>  
    <h1> {user.name} </h1>  
    <img className="avatar" src={user.imageUrl} />  
  </>  
) ;
```

React

Condicionais

Em React, não existe uma sintaxe especial para escrever uma condicional, você utiliza as mesmas sintaxes do que vc escreveria num código JavaScript.

Mas caso queira utilizar um condicional dentro do seu código HTML, você pode utilizar o operador condicional ?

```
<div>
  {isLoggedIn ? (
    <AdminPanel />
  ) : (
    <LoginForm />
  ) }
</div>
```

React

Renderizando Listas

Em React, é utilizado o loop **for** e a função de array **map()** para renderizar listas de componentes

```
const list = ["a", "b", "c",];
```

Dentro do componente se usa a função **map** para criar um array de itens de lista ``

```
const listItems = list.map(item =>  
  <li key={item}>  
    {item}  
  </li>  
);
```

React

Eventos

É possível responder a eventos declarando *event handlers* dentro dos seus componentes

```
function MyButton() {  
  function handleClick() {  
    alert('You clicked me!');  
  }  
}
```

```
  return (  
    <button onClick={handleClick}>  
      Click me  
    </button>  
  );  
}
```

React

Estados

Componentes também são capazes de “lembrar” informações e apresentá-las, como por exemplo, contar o total de vezes que um botão é clicado, para isso é adicionado um **estado** ao seu componente.

```
import { useState } from 'react';
```

```
function MyButton() {  
  const [count, setCount] = useState(0);  
  // ...  
}
```


React

Docs

- <https://react.dev/>

MUITO OBRIGADO
PELA ATENÇÃO

Até a próxima aula!

transforme ■ se

O conhecimento é o poder
de transformar o seu futuro.