

AULA 3/6

Props

Props

- Props são basicamente argumentos que podemos passar para os nossos componentes React.
- Por exemplo, você quer customizar a página de entrada da sua aplicação dando as boas vindas para cada usuário.
- Para isso, é preciso saber o nome do usuário. O nome do usuário pode ser passado por argumentos no momento de invocação do componente React.
 - Todo parâmetro que for passado na chamada do componente pode ser acessado como um argumento da função no componente

Props

App.jsx

BoasVindas.jsx ✕

src > BoasVindas.jsx > BoasVindas

```
1 export default function BoasVindas(props){
2   return (
3     <h1>Seja bem vindo: {props.nome}</h1>
4   );
5 }
```

App.jsx ✕

BoasVindas.jsx

src > App.jsx > ...

Seja bem vindo: Sidartha

```
1 import './App.css'
2 import BoasVindas from './BoasVindas'
3
4 function App() {
5   return (
6     <>
7       <BoasVindas nome="Sidartha" />
8     </>
9   )
10 }
11
12 export default App
13
```

Props

- Então, podemos customizar os componentes reutilizando o código base e alterando somente os parâmetros.

Seja bem vindo: Sidartha

Seja bem vindo: João

Seja bem vindo: Maria

```
function App() {  
  return (  
    <>  
      <BoasVindas nome="Sidartha" />  
      <BoasVindas nome="João" />  
      <BoasVindas nome="Maria" />  
    </>  
  )  
}  
  
export default App
```

Props

- É possível quebrar os diversos parâmetros recebidos no componente React usando {}
- Ao invés de acessar com props.nome, podemos fazer:

```
export default function BoasVindas({nome}){  
  return (  
    <h1>Seja bem vindo: {nome}</h1>  
  );  
}
```

Props

- Se precisar de mais parâmetros, só adicionar

```
export default function BoasVindas({primeiroNome, ultimoNome}){  
  return (  
    <h1>Seja bem vindo: {primeiroNome} {ultimoNome}</h1>  
  );  
}
```

```
function App() {  
  return (  
    <>  
    <BoasVindas primeiroNome="Sidartha" ultimoNome="Carvalho" />  
    <BoasVindas primeiroNome="João" ultimoNome="da Silva" />  
    <BoasVindas primeiroNome="Maria" ultimoNome="Ferraz" />  
    </>  
  )  
}
```

Props

- Para valores numéricos, adicione { } ao invés de “ ”

```
function App() {  
  return (  
    <>  
    <BoasVindas primeiroNome="Sidartha" ultimoNome="Carvalho" idade={32} />  
  )  
}
```


Props

- Arrays e Objects
 - Tudo que estiver dentro de { } será considerado JS

```
<Componente  
  arrayValores={ [1, 23, 44, 667] }  
  
  dicioValores={ {valor1: 23, valor2: 434} } />
```

Estilização Dinâmica

CSS Dinâmico

- É possível alterar o CSS de forma dinâmica dos componentes React
- Por exemplo, se o meu usuário for Admin, quero que o nome dele seja impresso em vermelho.
 - O default é false, ou seja, se não for informado parâmetro, considerar que não é Admin.

CSS Dinâmico

```
export default function BoasVindas({primeiroNome, ultimoNome, idade, isAdmin = false}){  
  let adminStyle = {color: isAdmin ? "red" : "green"};  
  
  return (  
    <h1 style={adminStyle}>Seja bem vindo: {primeiroNome} {ultimoNome} -- {idade} </h1>  
  );  
}
```

```
function App() {  
  return (  
    <>  
    <BoasVindas primeiroNome="Sidarta" ultimoNome="Carvalho" idade={33} isAdmin={true} />  
    <BoasVindas primeiroNome="João" ultimoNome="Silva" idade={66} />  
    </>  
  )  
}
```

Map to HTML

- É possível transformar um Array / Map em elementos HTML dentro do React

App.jsx

ColorList.jsx

BoasVindas.jsx

src > ColorList.jsx > ColorList

```
1 export default function ColorList({mapColors}){
2
3     return (
4         <div>
5             <h1>Color List</h1>
6             <ul>
7                 { mapColors.map((color) => (<li>{color}</li>)) }
8             </ul>
9         </div>
10    );
11 }
```



App □ `<ColorList mapColors={ ["blue", "red", "yellow", "black"] } />`

Map to HTML

- Além disso, é possível usar o valor para definir o CSS também

```
export default function ColorList({mapColors}){  
  
  return (  
    <div>  
      <h1>Color List</h1>  
      <ul>  
        { mapColors.map((color) =>  
          (<li style={{ color: color }}> Minha cor é: {color} </li>)) }  
        </ul>  
      </div>  
    );  
  }  
}
```

Color List

-
-
-
-

Minha cor é: blue

Minha cor é: red

Minha cor é: yellow

Minha cor é: black

SlotMachine

- Componente React



Você perdeu! Tente Novamente!



Você Ganhou!!!

SlotMachine

o.jsx

CassinoSlot.jsx ●

ColorList.jsx

BoasVindas.jsx

CassinoSlot.jsx > CassinoSlot

```
export default function CassinoSlot({firstColumn, secondColumn, thirdColumn}){  
  
  const isWinner = (firstColumn === secondColumn) && (firstColumn === thirdColumn);  
  
  return (  
    <div>  
      <h1>{firstColumn} - {secondColumn} - {thirdColumn}</h1>  
      <h2> {isWinner ? "Você Ganhou!!!" : "Você perdeu! Tente Novamente!"}</h2>  
    </div>  
  );  
}
```

```
<CassinoSlot firstColumn="🍒" secondColumn="🍌" thirdColumn="🍌" />
```

```
<CassinoSlot firstColumn="🍒" secondColumn="🍒" thirdColumn="🍒" />
```


SlotMachine

- Você também pode alterar o CSS para a cor do texto quando vencer ou perder, mas isso pode tornar o código menos legível!

```
<div>
  <h2 style={{color: isWinner ? "green" : "red"}}>
    {isWinner ? "Você Ganhou!!!" : "Você perdeu! Tente Novamente!"}
  </h2>
</div>
```

React Developer Tools

React Developer Tools

- Para auxiliar na construção de aplicativos React, podemos fazer uso de um plugin no Chrome
 - React Developer Tools

<https://chromewebstore.google.com/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?pli=1>



React Developer Tools

Add to Chrome

Featured 4.0 ★ (1.5K ratings)

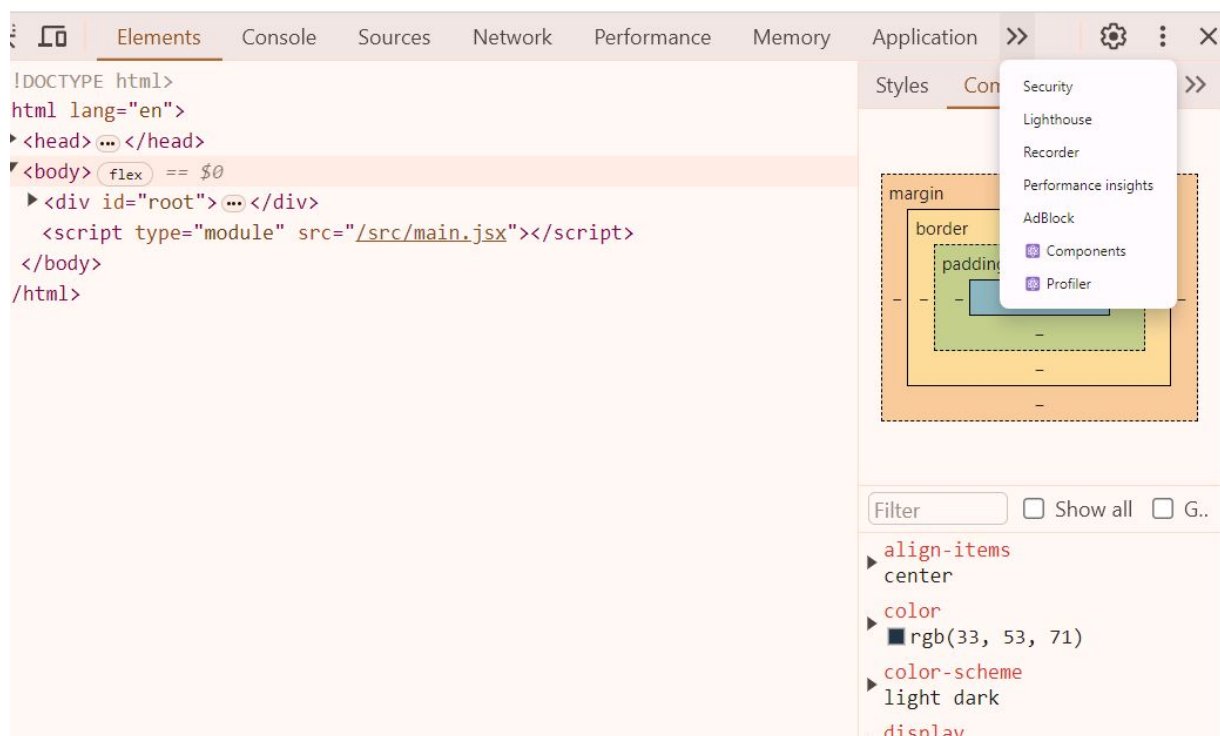
Extension

Developer Tools

4,000,000 users

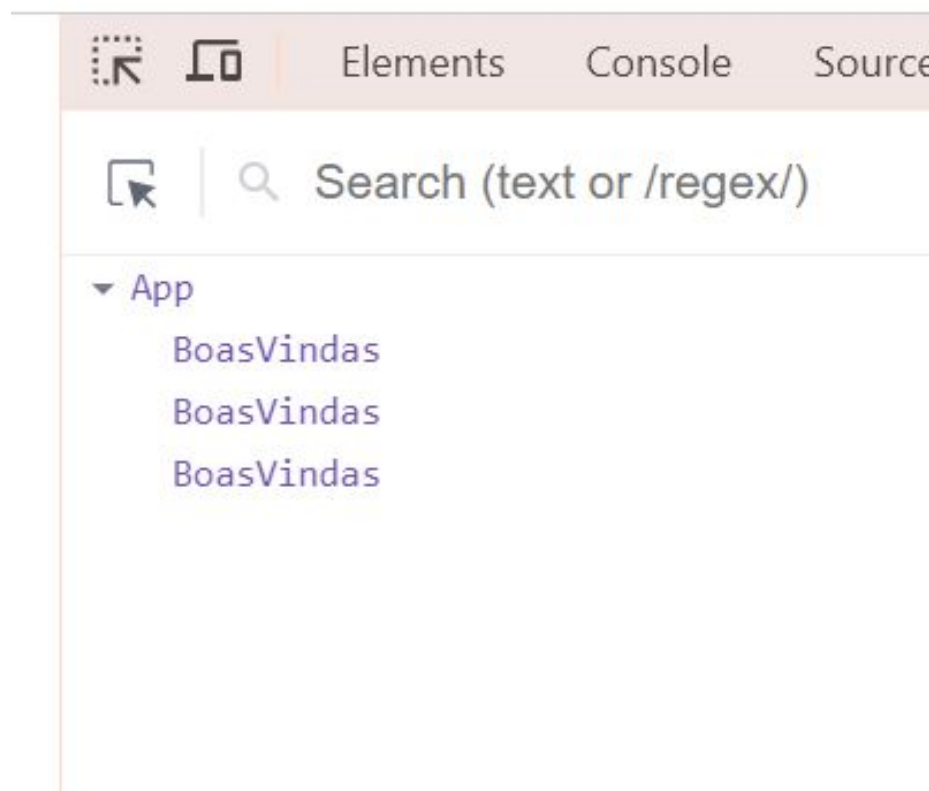
React Developer Tools

- Ao abrir o Developer Mode no Chrome, há duas novas opções:
 - Componentes e Profiler



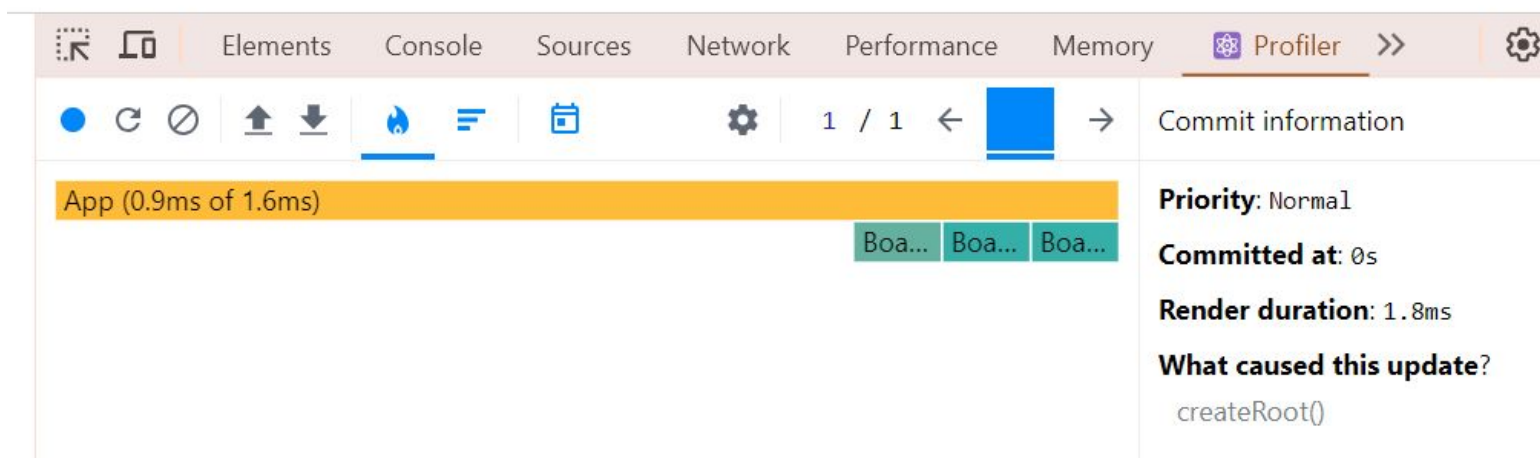
React Developer Tools

- Na aba **Components** é possível visualizar os componentes React e suas **Props** que foram criados na página e a hierarquia entre eles



React Developer Tools

- Na aba **Profile** é possível identificar problemas de desempenho no seu aplicativo.
- É possível ver quanto tempo cada componente demorou para ser construído e exibido ao usuário, dentre outras coisas.



React Developer Tools

- Na página do Airbnb é possível ver centenas de componentes anônimos.
- Provavelmente isso aconteceu no momento de criação do aplicativo com a compactação dos nomes para reduzir espaço e tornar o site menos “copiável”.

