

AULA 2/6

Fragments React

- Vimos nos exemplos anteriores que, no React, só é possível retornar um elemento por função / componente.
- Em muitos casos não queremos ter que adicionar uma `<div>` ou `<form>` no nosso componente React.
- Se não quiser incluir elementos HTML para englobar os elementos criados, é possível fazer o uso de Fragments no React.

Fragments React

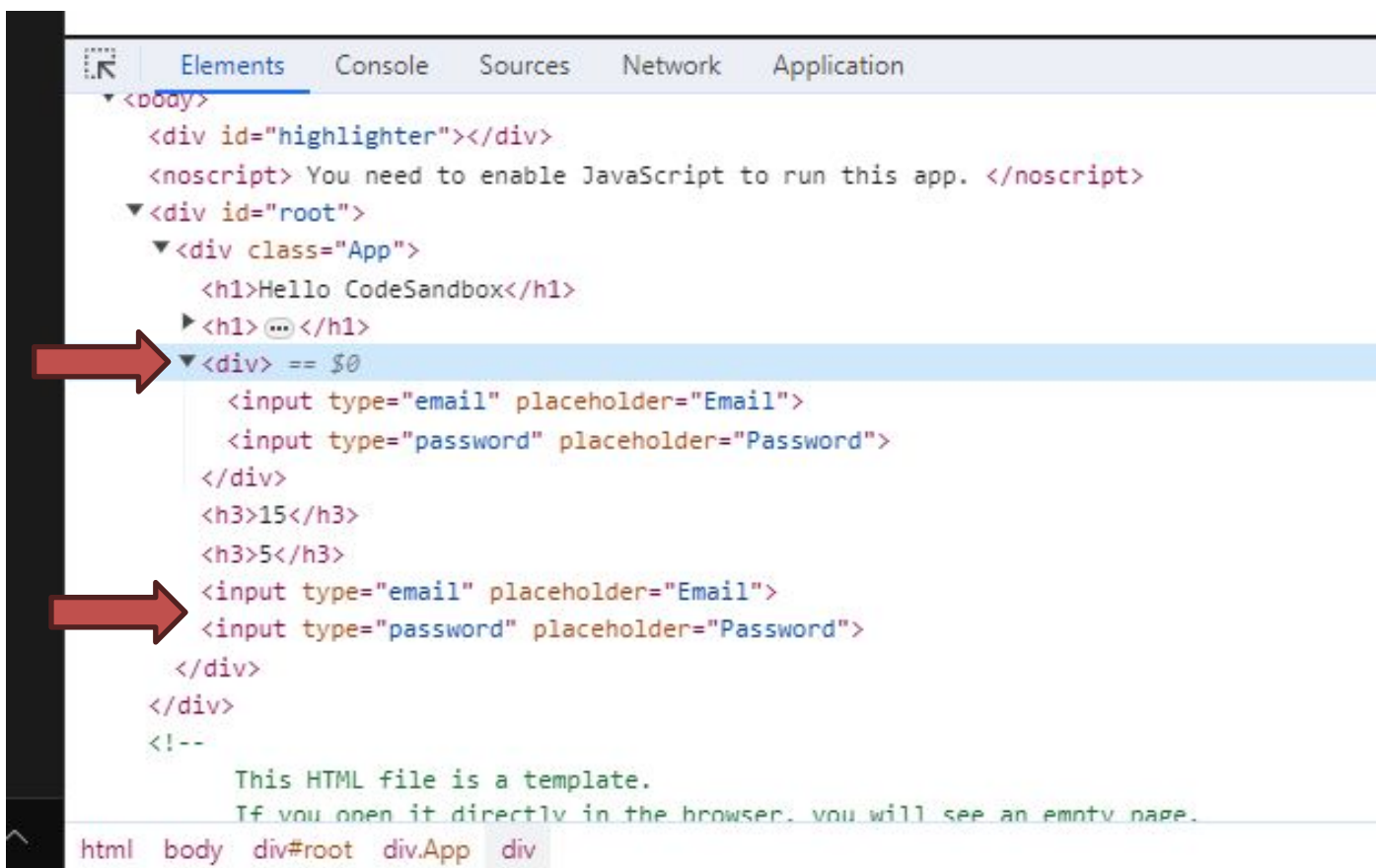
- Para incluir um Fragment React é só incluir `<>` no lugar do `<div>` ou `<form>`

```
FragmentReact.js > ...
1 export default function FragmentReact() {
2   return (
3     <>
4       <input type="email" placeholder="Email" />
5       <input type="password" placeholder="Password" />
6     </>
7   );
8 }
9
```

```
src > App.js > ...
1 import './styles.css';
2 import MeuComponente from './MeuComponente.js';
3 import MeuDuploComponente from './MeuDuploComponente.js';
4 import { add, sub } from './Calculadora';
5 import FragmentReact from './FragmentReact.js';
6
7 let soma = add(5, 10);
8
9 export default function App() {
10   return (
11     <div className="App">
12       <h1>Hello CodeSandbox</h1>
13       <MeuComponente />
14       <MeuDuploComponente />
15       <h3>{soma}</h3>
16       <h3>{sub(10, 5)}</h3>
17       <FragmentReact />
18     </div>
19   );
20 }
21
22
```

Fragments React

- HTML gerado com e sem o Fragment React

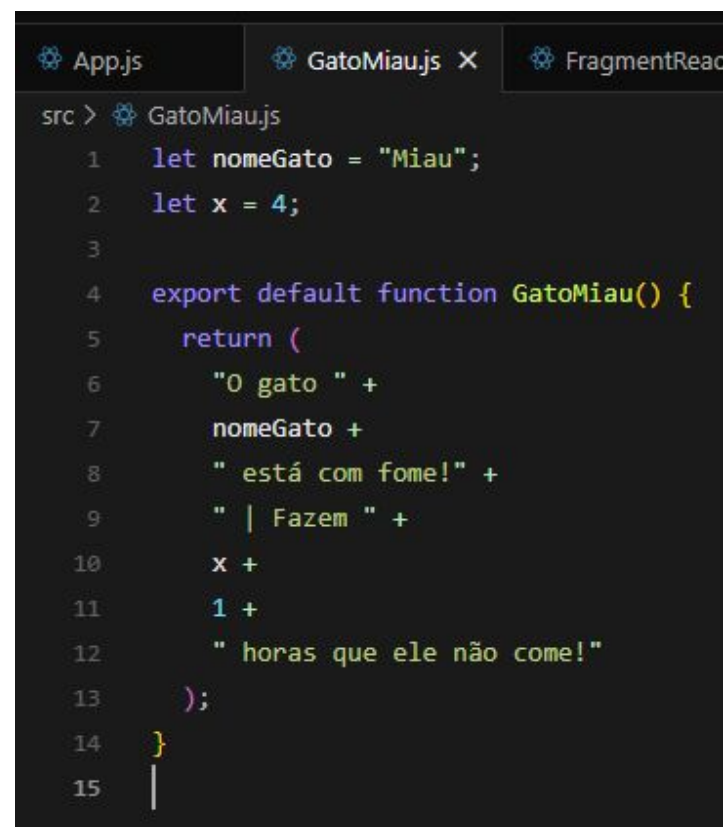


```
<body>
  <div id="highlighter"></div>
  <noscript> You need to enable JavaScript to run this app. </noscript>
  <div id="root">
    <div class="App">
      <h1>Hello CodeSandbox</h1>
      <h1>...</h1>
      <div> == $0
        <input type="email" placeholder="Email">
        <input type="password" placeholder="Password">
      </div>
      <h3>15</h3>
      <h3>5</h3>
      <input type="email" placeholder="Email">
      <input type="password" placeholder="Password">
    </div>
  </div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.
  -->
```

html body div#root div.App div

Código JS no React

- É possível executar código puramente JS dentro do React, para isso, deve-se utilizar o código JS dentro de { } ou dentro do componente diretamente.



```
src > GatoMiau.js
1  let nomeGato = "Miau";
2  let x = 4;
3
4  export default function GatoMiau() {
5    return (
6      "O gato " +
7      nomeGato +
8      " está com fome!" +
9      " | Fazem " +
10     x +
11     1 +
12     " horas que ele não come!"
13   );
14 }
15 |
```

Componente de Componentes

- Como vimos, o React é voltado para componentização e reuso de código.
- Com isso, é recomendado a criação de pequenos componentes React e depois a estruturação de novos componentes reutilizando os componentes criados.

Componente de Componentes

- Por exemplo, vamos criar um Casino online.
- Nosso **primeiro jogo** é o rolamento de dados
 - Neste jogo, o dado é rolado e retorna um número aleatório entre 1 a 6 (Dado.js)
- O **segundo jogo** é a mesa de dados, onde são rolados 3 dados, cada um retornando um valor aleatório de 1 a 6 (MaoDeDados.js)
- Iremos utilizar o primeiro componente para criar o segundo.

Componente de Componentes

```
App.js  MaoDeDados.js  Dado.js  X
src > Cassino > Dado.js
1  export default function Dado() {
2    let valorDado = Math.floor(Math.random() * 6);
3    return "O dado rolou com: " + valorDado;
4  }
5
```

O dado rolou com: 3

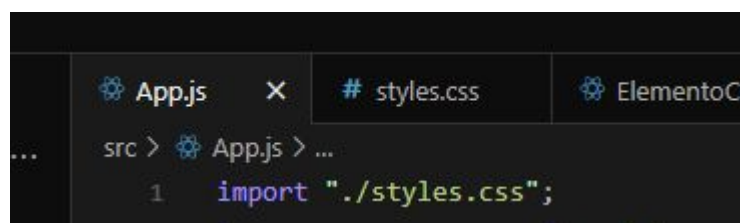
Mão de Dados

O dado rolou com: 1
O dado rolou com: 3
O dado rolou com: 0

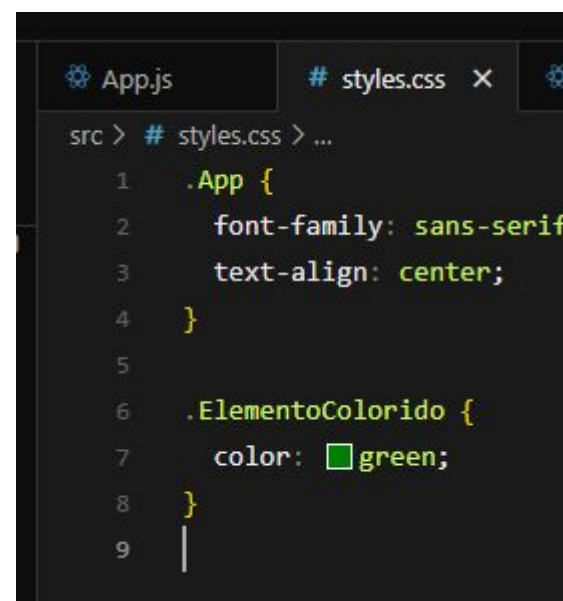
```
App.js  MaoDeDados.js  X  Dado.js
src > Cassino > MaoDeDados.js > ...
1  import Dado from "../Dado.js";
2
3  export default function MaoDeDados() {
4    return (
5      <>
6        <h3>Mão de Dados</h3>
7        <Dado /> <br />
8        <Dado /> <br />
9        <Dado /> <br />
10     </>
11   );
12 }
13 |
```


Estilizando Componentes React

- Pode-se utilizar o elemento *classname*, similar ao *class* do CSS clássico
- Desta forma, todas as definições de CSS podem estar em um único arquivo, o *styles.css* que é importado no *App.js*.



```
src > App.js > ...  
1 import "../styles.css";
```



```
src > # styles.css > ...  
1 .App {  
2   font-family: sans-serif  
3   text-align: center;  
4 }  
5  
6 .ElementoColorido {  
7   color: green;  
8 }  
9 |
```

Estilizando Componentes React

```
pp.js  # styles.css  ElementoColorido.js X
> ElementoColorido.js > ...
1  export default function ElementoColorido() {
2    return <h2 className="ElementoColorido">Elemento colorido!</h2>;
3  }
4  |
```

ato Miau está com fome! | Fazem 41 horas que ele não come!

O dado rolou com: 1

Mão de Dados

O dado rolou com: 2

O dado rolou com: 5

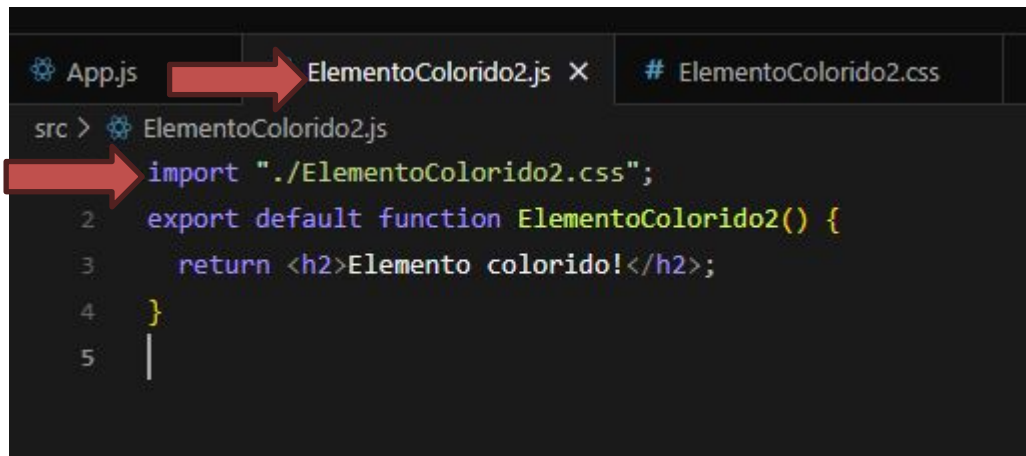
O dado rolou com: 1

Elemento colorido!

Estilizando Componentes React

- Porém... é recomendado que cada componente possua seu próprio arquivo de CSS, permitindo que outros desenvolvedores encontrem os estilos mais rapidamente dentro do seu projeto React.
- É recomendado que o arquivo .css possua o mesmo nome do componente.js
- Por exemplo
 - ElementoColorido.js □ ElementoColorido.css.

Estilizando Componentes React



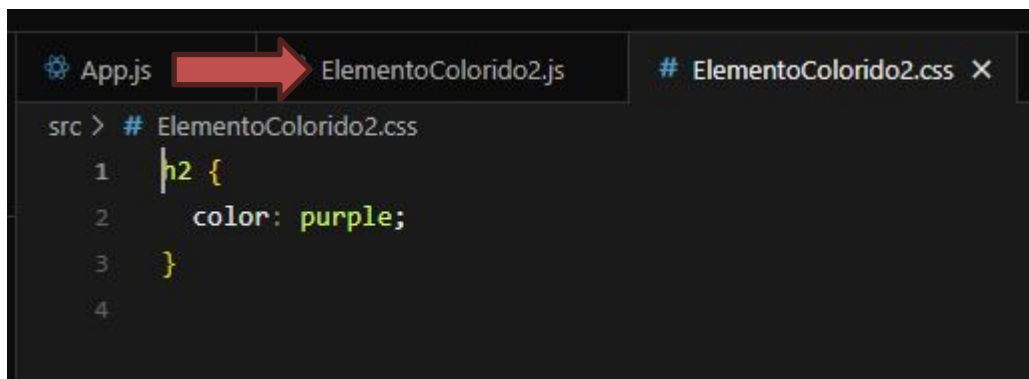
```
App.js ElementoColorido2.js X # ElementoColorido2.css
src > ElementoColorido2.js
import './ElementoColorido2.css';
2 export default function ElementoColorido2() {
3   return <h2>Elemento colorido!</h2>;
4 }
5 |
```

Esta abordagem permite que o CSS seja encapsulado dentro do componente e facilmente encontrado!

O dado rolou com: 0
O dado rolou com: 2

Elemento colorido!

Elemento colorido!



```
App.js ElementoColorido2.js X # ElementoColorido2.css X
src > # ElementoColorido2.css
1 h2 {
2   color: purple;
3 }
4
```

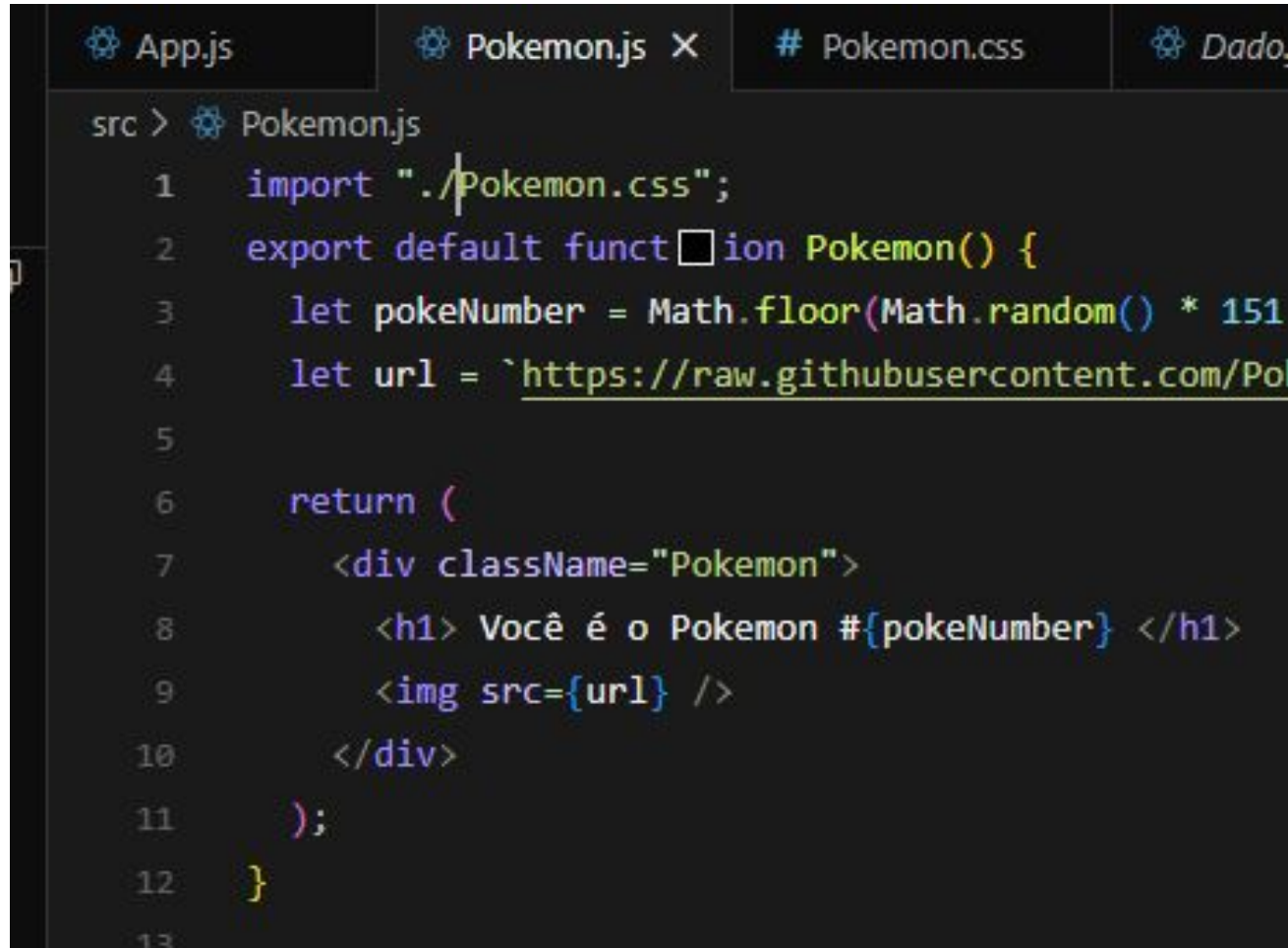
Exemplo Prático

Adivinhador Pokemon

- Qual Pokemon você é?



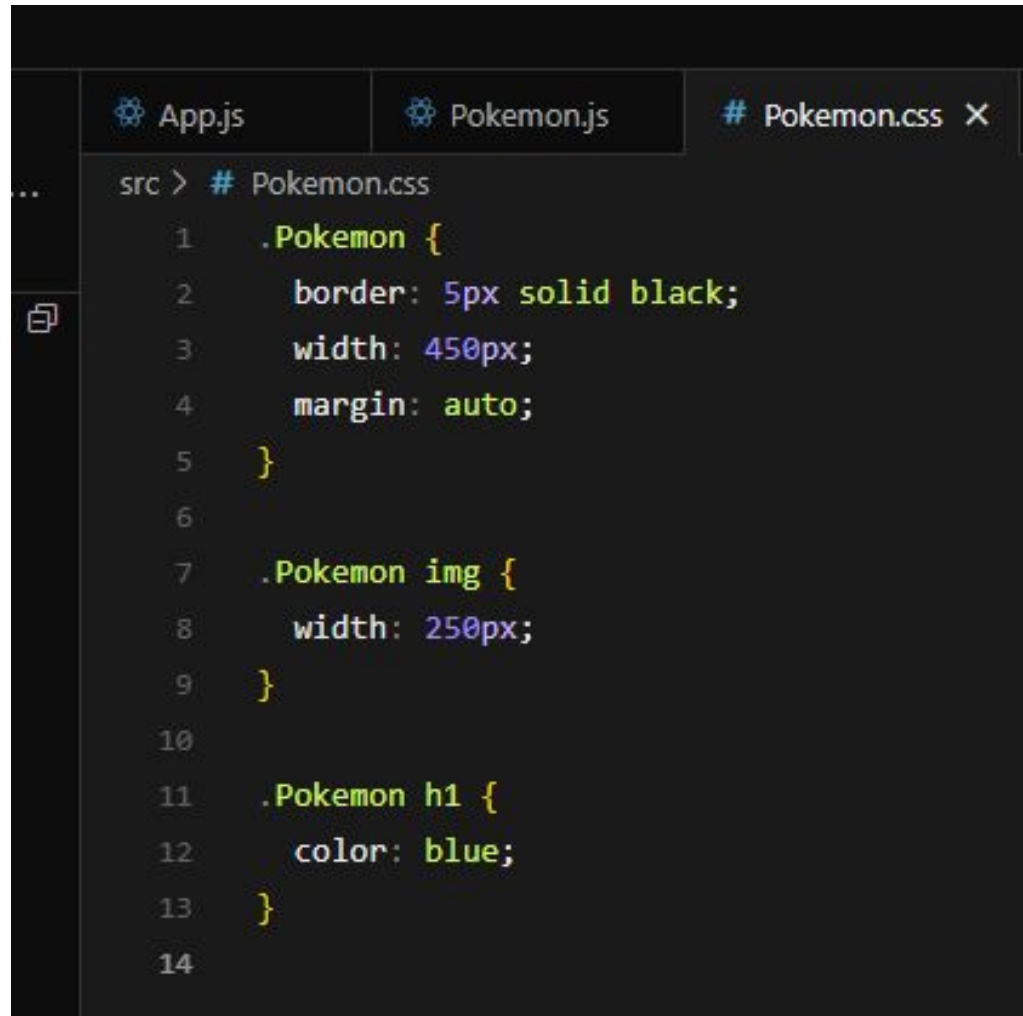
Adivinhador Pokemon

A screenshot of a code editor with four tabs: App.js, Pokemon.js (active), Pokemon.css, and Dado.js. The Pokemon.js file contains the following code:

```
src > Pokemon.js
1  import "../pokemon.css";
2  export default function Pokemon() {
3    let pokeNumber = Math.floor(Math.random() * 151);
4    let url = `https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/${pokeNumber}.png`;
5
6    return (
7      <div className="Pokemon">
8        <h1> Você é o Pokemon #{pokeNumber} </h1>
9        <img src={url} />
10      </div>
11    );
12  }
```

let url = `https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/\${pokeNumber}.png`;

Adivinhador Pokemon



The image shows a code editor with three tabs: 'App.js', 'Pokemon.js', and '# Pokemon.css'. The active tab is '# Pokemon.css', which contains the following CSS code:

```
src > # Pokemon.css
1  .Pokemon {
2    border: 5px solid black;
3    width: 450px;
4    margin: auto;
5  }
6
7  .Pokemon img {
8    width: 250px;
9  }
10
11 .Pokemon h1 {
12   color: blue;
13 }
14
```


Componentes Prontos

Componentes Prontos

- Componentes prontos para uso
 - Material by Google

<https://mui.com/material-ui/all-components/>

- Comunidade

<https://github.com/brillout/awesome-react-components>

Material Components

- <https://mui.com/material-ui/react-rating/>

Basic rating

Controlled



Read only



Disabled



No rating given

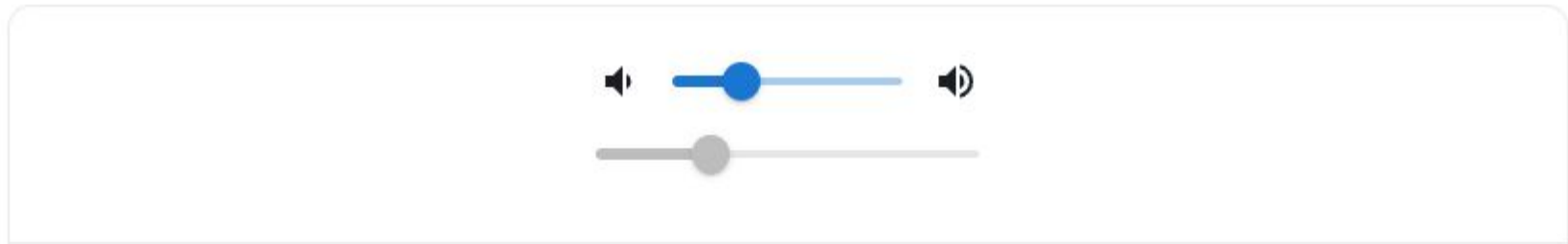


Material Components

- <https://mui.com/material-ui/react-slider/>

Continuous sliders

Continuous sliders allow users to select a value along a subjective range.

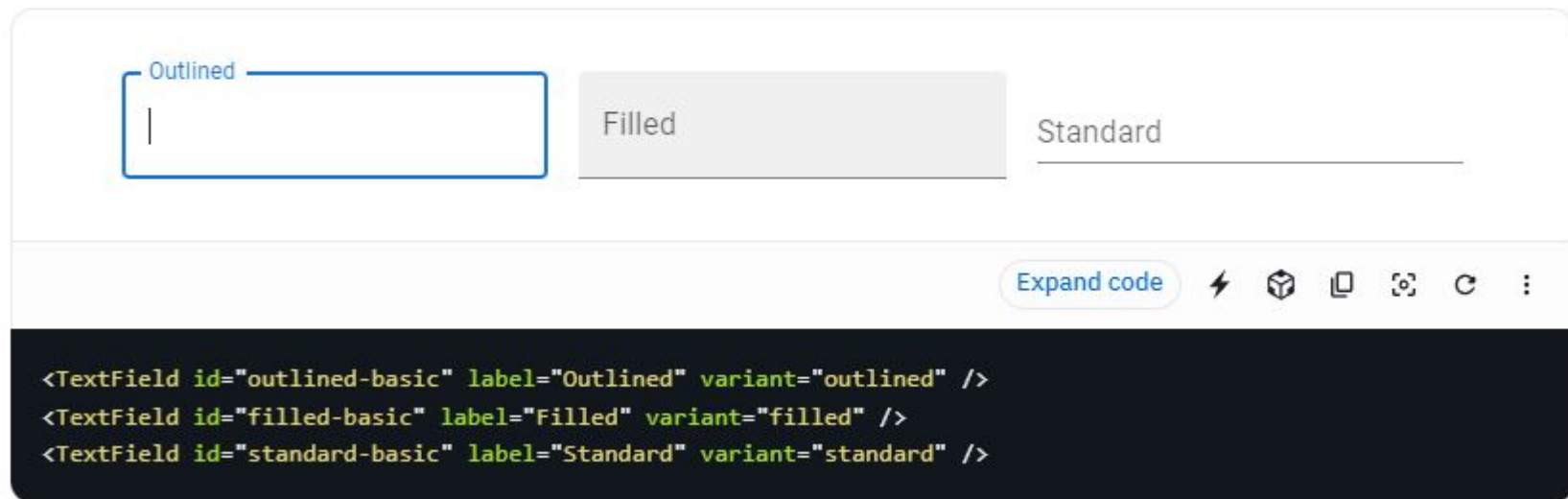


Material Components

- <https://mui.com/material-ui/react-text-field/>

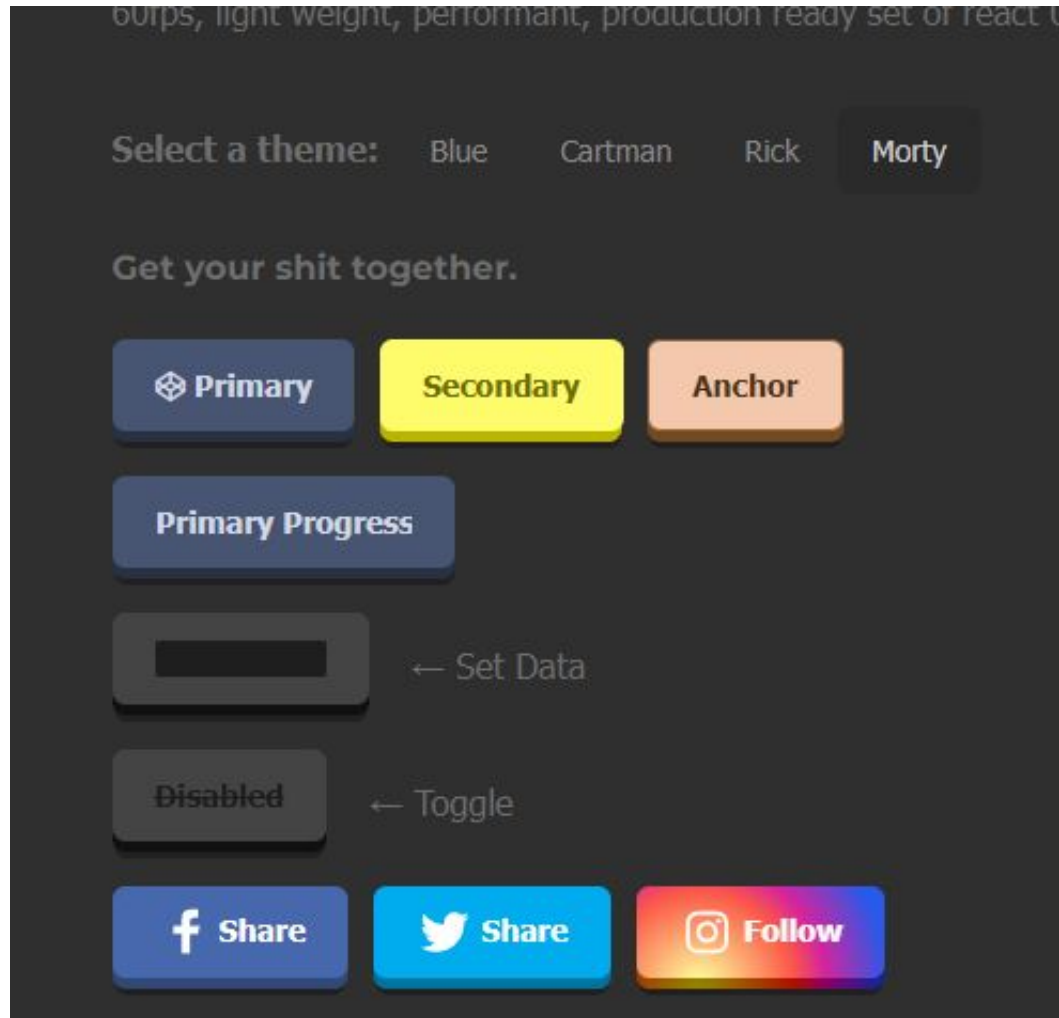
Basic TextField

The `TextField` wrapper component is a complete form control including a label, input, and help text. It comes with three variants: outlined (default), filled, and standard.



Comunidade

- <https://caferati.me/demo/react-awesome-button>



Comunidade

- <https://komarovalexander.github.io/ka-table/#/overview>

Basics			
Column 1	Column 2 ↑	Column 3	Column 4
column:1 row:0	column:2 row:0	column:3 row:0	column:4 row:0
column:1 row:1	column:2 row:1	column:3 row:1	column:4 row:1
column:1 row:2	column:2 row:2	column:3 row:2	column:4 row:2
column:1 row:3	column:2 row:3	column:3 row:3	column:4 row:3
column:1 row:4	column:2 row:4	column:3 row:4	column:4 row:4
column:1 row:5	column:2 row:5	column:3 row:5	column:4 row:5
column:1 row:6	column:2 row:6	column:3 row:6	column:4 row:6
column:1 row:7	column:2 row:7	column:3 row:7	column:4 row:7
column:1 row:8	column:2 row:8	column:3 row:8	column:4 row:8
column:1 row:9	column:2 row:9	column:3 row:9	column:4 row:9

Exemplos MIU

<https://playcode.io/mui>

```
import React from 'react'
import Button from '@mui/material/Button'
import ShoppingCartIcon from '@mui/icons-material/ShoppingCart';
```

```
function ReusableButton({ variant = 'contained', label, onClick,
icon }) {
  return (
    <Button
      variant={variant}
      startIcon={icon || <ShoppingCartIcon />}
      onClick={onClick}
      sx={{ margin: 1 }}
    >
      {label || 'Botão Padrão'}
    </Button>
  );
}
```

```
export function App(props) {
  return (
    <div className='App'>

    <ReusableButton
      variant="outlined"
      label="Adicionar ao Carrinho"
      onClick={() => console.log('Clicado!')}
    />

    </div>
  )
}
```



```
import React from 'react'
import { Card, CardMedia, Typography, CardContent, Button } from '@mui/material';
```

```
export function App(props) {
  return (
    <div className='App'>

      <ProductCard product={productData} />

    </div>
  )
}

function ProductCard({ product }) {
  return (
    <Card sx={{ maxWidth: 345 }}>
      <CardMedia
        component="img"
        height="140"
        image={product.imageUrl || '/placeholder.jpg'}
        alt={product.name}
      />
      <CardContent>
        <Typography gutterBottom variant="h5">
          {product.name}
        </Typography>
        <Typography color="text.secondary">
          Preço: R$ {product.price.toFixed(2)}
        </Typography>
      </CardContent>
      <Button variant="contained" fullWidth>
        Comprar
      </Button>
    </Card>
  );
}
```

```
// Dados de exemplo:
const productData = {
  name: 'Nokia Tijolão',
  price: 1999.99,
  imageUrl:
'https://s2-techtudo.glbimg.com/fyC0QV72cee07ckqHTs3f_ZFWi0=/0x0:695x401/984x0/smart/filters:strip_icc()/i.s3.glbimg.com/v1/AUTH_08fbf48bc0524877943fe86e43087e7a/internal_photos/bs/2017/9/4/j6fITqTW2nce3A8ZA3IQ/2-nokia-3310.jpg'
};
```

