

JSX



Elementos são os menores blocos de construção de aplicativos React.

Podemos chamar as *tags* para o DOM de **objetos JSX**, ou **elementos**, ou nós.

As *tags* são sempre escritas em minúsculas.



A tag sempre precisa ser fechada

```
<p>Olá Serratec!</p>
```

Mesmo que com auto fechamento

```
<hr />
```



Os elementos podem ser atribuídos em variáveis.

```
const saudacao = <h1>Bora Serratec</h1>;
```

Cada instrução retorna somente um único elemento. Mas elementos podem conter outros elementos. Para retornar mais elementos, estes devem ser envolvidos em um elemento pai, seguindo a estrutura parecida com HTML normal.

O elemento pai será considerado o *único elemento*.

```
var bloco = <div><h2>Artista: Linkin Park</h2></div>;
```

ELEMENTO REACT



Elementos em múltiplas linhas devem ser colocados entre parênteses.

```
var bloco = (  
  <div>  
    <h2>Artista: Linkin Park</h2>  
    <hr />  
  </div>  
);
```

ATRIBUTOS



Um elemento pode conter atributos. Se o atributo é *string*, pode ser passado entre aspas.

```
const painel = <div id="painel">Lista de músicas</div>;
```

ATRIBUTOS



Para atributos de outros tipos de dados, devem ser utilizadas expressões React.

Os atributos em React devem ser escritos seguindo a notação camelCase.

```
<input type="text" readOnly />
```



class e **for** são palavras reservadas em JS. Portanto, esses atributos devem ser substituídos por **className** e **htmlFor**

EXPRESSÃO REACT



Uma **expressão React** é uma expressão JavaScript que permite também a utilização de elementos React. As **expressões React** são envolvidas entre chaves.

```
<textarea rows={3} />
```



Um inteiro deve ser passado para um atributo como *expressão*.

Podemos fazer associação com as expressões utilizadas em *template strings*.

EXPRESSÕES COM ELEMENTOS



Expressões podem ser incluídas em qualquer parte da instrução.

```
let nome = "Maria"
```

```
let oi = <p>Oi {nome + "!"}</p>;
```

E expressões podem renderizar outros elementos.

```
let oi = <p>Oi Maria!</p>;
```

```
let texto = <div>{oi}</div>;
```

ARRAYS DE ELEMENTOS



Uma vez que variáveis podem receber elementos React, então, arrays também podem. Uma expressão React pode renderizar um array inteiro de uma vez.

```
const frutas = [<em>🍓</em>, <strong>🍌</strong>, <span>🍉</span>];  
let feira = <p>{frutas}</p>;
```

ELEMENTOS REPETIDOS



Ao trabalhar com elementos que se repetem, é necessário a inclusão do atributo **key**.

Elementos repetidos precisam ter uma chave de identificação única.

```
<ul>  
  <li key={musica.id}>{musica.nome}</li>  
</ul>
```



Manipular eventos em elementos React lembra muito a manipulação de eventos no DOM.

Diferenças:

- Assim como atributos, eventos em React são nomeados usando camelCase.
- A função de manipulação deve ser referenciada em uma expressão.



Em React, estilos **não são *strings***, e sim, **objetos**. Você já reparou que **nada em React é string**. Devem ser passados ao elemento em uma expressão.

```
const estilo = {  
  background: "tomato",  
};  
let quadrado = <div style={estilo}>🍅</div>;
```

COMENTÁRIOS



Não é possível fazer um comentário usando tags JSX. Contudo, expressões React permitem comentários multilinha, e dessa forma podemos comentar trechos do código.

```
{/* <!-- tags podem ser comentadas? dentro de expressões, sim! -->  
  Papel não compila :)  
*/}
```


