Front End III

Regras para escrever JSX

Finalidade de JSX

A finalidade do JSX é que ele seja usado por transpiladores que convertam os tokens JSX para JavaScript padrão.

Regras para escrita de JSX

- Os componentes definidos pelo usuário devem ter iniciais maiúsculas. De fato, o React recomenda nomear os componentes com uma letra em maiúscula. Caso exista um componente que começa com minúscula, uma variável em maiúsculas deve ser atribuída antes dele ser usado com sintaxe JSX. Por exemplo:

```
/* Não faça isso! Este é um componente, portanto a primeira letra
       deve ser maiúscula */
       function meuComponente() {
        /* O uso de div em minúscula é correto sim
           porque div é uma tag válida de HTML */
        return <div>Eu sou um componente React</div>;
    }

    function minhaFuncao() {
        /* Não faça isso! O React acha que <meuComponente /> é uma tag
HTML
```

```
      porque ele não tem inicial maiúscula */

        return <meuComponente />;

    }
```

- Para os elementos HTML que possuem tags de fechamento automático <img>, <hr>, <input> e
, a barra diagonal antes do colchete angular de fechamento é obrigatória em JSX (embora seja opcional em HTML). Por exemplo, isto irá funcionar:

```
<input/>
```

  Mas isto não:

```
<input>
```

- As expressões que figuram entre chaves são avaliadas como JavaScript, por isso, devem ser expressões válidas de JavaScript. Por exemplo, a linha a seguir produz um paragraph com o conteúdo `2 + 3 = 5` porque a parte entre chaves {2 + 3} está avaliando o 5:

```
   <p> 2+3 = {2+3} </p>;
```

- JSX aceita aninhamento, porém, a expressão deve ser apenas um elemento externo. Por exemplo, isto funciona:

```
const headings = (

    <div id = "outermost-element">

        <h1>Eu sou um cabeçalho h1</h1>

        <h2>Eu sou um cabeçalho h2</h2>

    </div>

);
```

  Isto não:

```
  const headings = (
```

```
    <h1>Eu sou um cabeçalho h1</h1>
    <h2>Eu sou um cabeçalho h2</h2>
);
```

- As propriedades (props) em JSX utilizam a convenção de nomeação de camelCase, ao invés da utilizada para nomes de atributos em HTML, uma vez que são baseadas na API DOM, e não nas especificações da linguagem HTML. Por exemplo, deve-se usar *className* e *tabIndex* em vez de classname e *tabindex* respectivamente, como quando usamos JavaScript para manipular o DOM.

```
const ola = <h1 className="welcome"> Ola Mundo </h1>
```

Isto não irá funcionar:

```
    const ola = <h1 classname="welcome"> Ola Mundo </h1>
```

Isso é assim, também, para os gerenciadores de eventos em JSX. Por exemplo, isto funciona:

```
<button onClick ={handleClick}>Clique em mim</button>
```

Já isso não:

```
<button onclick ={handleClick}>Clique em mim</button>
```

Por enquanto, é isso. Até a próxima!