Certified Tech Developer
The Ultimate Degree

Front End III

Reglas para escribir JSX

Intención de JSX

La intención con JSX es que sea utilizado por transpiladores que conviertan los tokens JSX en JavaScript estándar.

Reglas para escribir JSX

- Los componentes definidos por el usuario deben empezar con mayúscula. De hecho, React recomienda nombrar los componentes con una letra mayúscula. Si se tiene un componente que comienza con una letra minúscula, debe asignarse a una variable en mayúscula antes de usarlo con sintaxis JSX. Por ejemplo:

```
/* ¡No hacer! Este es un componente, por lo que la primera letra
   debe ser mayúscula */
function miComponente() {
    /* El uso de div en minúscula sí es correcto
       porque div es una etiqueta válida de HTML */
    return <div>Soy un componente React</div>;
}
```

```
function miFuncion() {
    /* ¡No hacer! React piensa que <miComponente /> es una etiqueta HTML
        porque no empieza con mayúscula */
    return <miComponente />;
}
```

- Para los elementos HTML que tienen etiquetas de cierre automático como <img>, <hr>, <input> y
, la barra diagonal antes del corchete angular de cierre es obligatoria en JSX (aunque es opcional en HTML). Por ejemplo, esto funcionará:

```
<input/>
```

Esto no funcionará:

```
<input>
```

- Las expresiones entre llaves se evalúan como JavaScript, por lo que deben ser expresiones válidas de JavaScript. Por ejemplo, la siguiente línea produce un paragraph con el contenido `2 + 3 = 5` porque la porción entre llaves {2 + 3} evalúa a 5:

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

- JSX acepta anidación, pero la expresión debe tener solo un elemento externo.
Por ejemplo, esto funciona:

```
const headings = (
   <div id = "outermost-element">
       <h1>Soy un encabezado h1</h1>
       <h2>Soy un encabezado h2</h2>
   </div>
);
```

Esto no funcionará:

```
const headings = (
    <h1>Soy un encabezado h1</h1>
    <h2>Soy un encabezado h2</h2>
);
```

- Las propiedades (*props*) en JSX utilizan la convención de nomenclatura de camelCase en lugar de la usada para nombres de atributos en HTML, porque se basan en la API DOM, no en las especificaciones del lenguaje HTML. Por ejemplo, se debe usar *className* y *tabIndex* en lugar de *classname* y *tabindex* respectivamente, igual que cuando usamos JavaScript para manipular el DOM. Por ejemplo, esto funciona:

```
const hola = <h1 className="welcome"> Hola Mundo </h1>
```

Esto no funcionará:

```
const hola = <h1 classname="welcome"> Hola Mundo </h1>
```

Esto también es cierto para los manejadores de eventos en JSX. Por ejemplo, esto funciona:

```
<button onClick = {handleClick}>Hazme click</button>
```

Esto no funcionará:

```
<button onclick = {handleClick}>Hazme click</button>
```

Eso es todo por ahora. ¡Hasta la próxima!