😌

JSX Syntax

Date	@February 28, 2023 → March 6, 2023
Assign	
Status	In progress
↗ ✅ Tasks	

Que es JSX?

Es la forma en que le decimos a React lo que queremos mostrar en la pantalla, no es Javascript ni HTML, es similar a un templating language con los beneficios de Javascript, este despues es transformado en javascript y enviado al cliente.

Uno de los beneficios de JSX es que nos da la oportunidad de tener encapsulado en un solo archivo la logica y el UI de nuestros componentes, evitandonos asi tener multiples archivos para un mismo elemento, permitiendonos tener componentes ordenados y faciles de entender, mantener y escribir.

Expresiones en JSX

Cuando queremos agregar expresiones de Javascript dentro de JSX debemos envolverlos en llaves `{ expresion JS }`, esto sera evaluado y se mostrara el resultado en el html final.

Por ejemplo, si tenemos esto

```
const user = {
  name: "Pepito",
  lastName: "Perez",
  email: "pperez@email.com",
  isAdminUser: false
}
```

```
const element = <h1>El usuario {user.name} {user.LastName}
                {user.isAdminUser ? "si es" : "no es"} un administrador.</h1>;
```

obtendremos como resultado `<h1>El usuario Pepito Perez no es un administrador.</h1>` , Ademas podemos hacer operaciones matematicas y/o llamados a funciones que tengamos definidas previamente

```
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}

const user = {
  firstName: 'Pepito',
  lastName: 'Perez'
};

const element = <h1>Hello, {formatName(user)}!  </h1>;
const element2 = <h1>{23 + 3}</h1>;
```

y obtendremos como resultado `<h1>Hello, Pepito Perez<h1/>` y `<h1>26</h1>`, podemos especificar atributos usando llaves

```
const element = <img src={user.avatarUrl}></img>;
```

Renderizando componentes en React con JSX

Para renderizar componentes en React con JSX, simplemente necesitamos importar el componente y usarlo como si fuera una etiqueta HTML. Por ejemplo, si tenemos un componente `MiComponente` y queremos usarlo en otro componente, simplemente hacemos lo siguiente:

```
import MiComponente from './MiComponente';

function App() {
  return (
    <div>
      <h1>Mi aplicación</h1>
      <MiComponente />
    </div>
```

```
  );
}
```

De esta manera, el componente `MiComponente` se renderizará dentro del componente `App`. También podemos pasar props a los componentes, de la misma manera que lo hacemos con las etiquetas HTML.

Ventajas de JSX

- Permite tener componentes más legibles y fáciles de mantener.
- Facilita la escritura de código al permitir el uso de operaciones y llamados a funciones de JavaScript para generar el contenido de los componentes.
- Al ser transformado en JavaScript, permite utilizar todo el poder de este lenguaje para manipular el DOM y generar contenido dinámicamente.

Desventajas de JSX

- Puede resultar confuso para aquellos que no estén familiarizados con la sintaxis de JSX.
- Al ser una extensión de JavaScript, puede requerir herramientas adicionales para su compilación y uso en navegadores antiguos.
- Puede llevar a la creación de componentes sobrecargados y difíciles de seguir si no se toman medidas para mantenerlos organizados.

Consideraciones al escribir JSX

Al escribir JSX es importante tener en cuenta que la sintaxis es similar a HTML, pero no es lo mismo. Además, es recomendable separar la lógica del componente de su presentación, manteniendo así un código más limpio y organizado. También es importante utilizar nombres descriptivos para los componentes y evitar componentes sobrecargados con demasiadas responsabilidades.