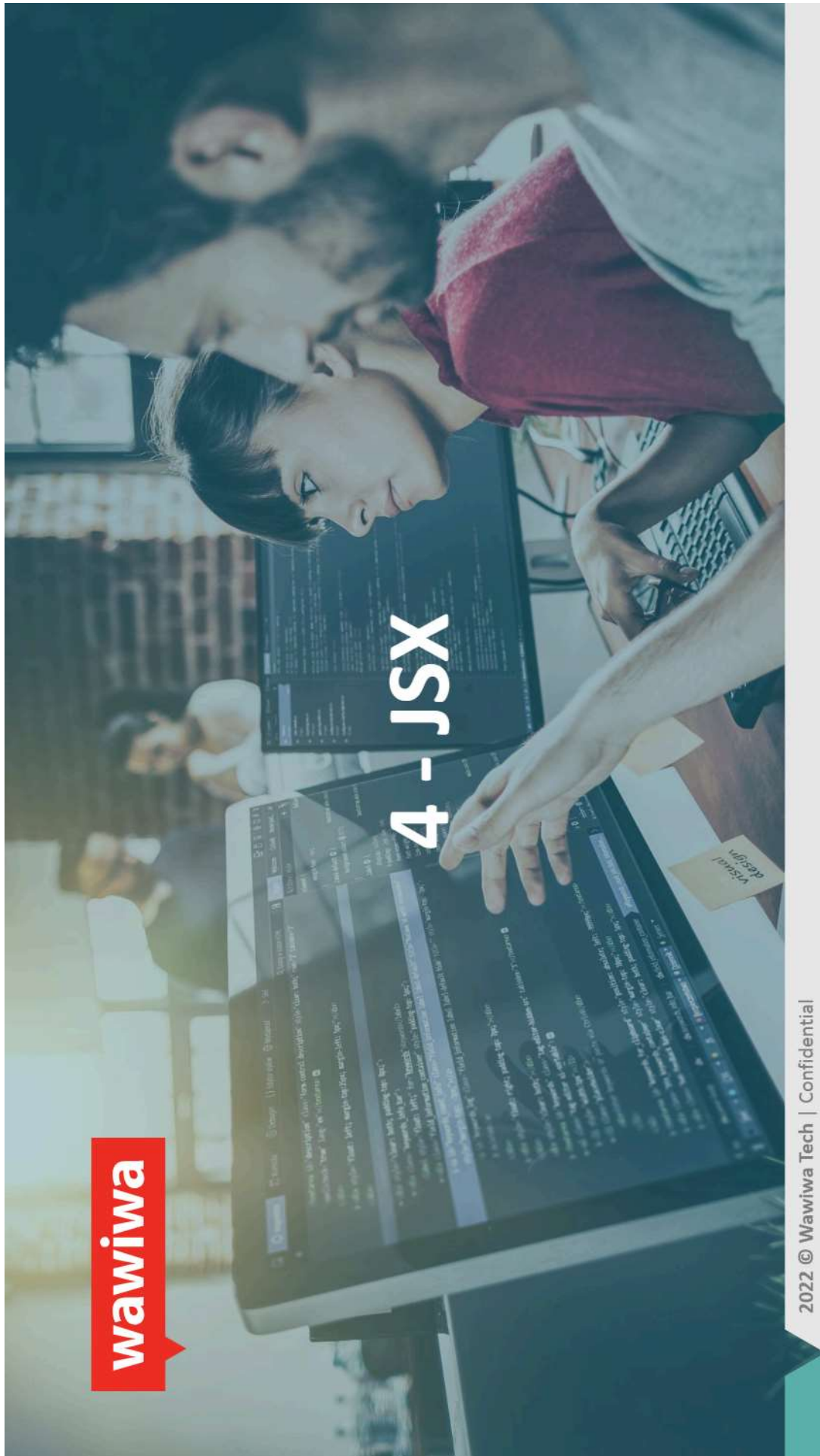




wawiwa

React

2022 © Wawiwa Tech | Confidential



2022 © Wawiwa Tech | Confidential



Introducción

En el capítulo previo usamos un código similar a este código:

```
const element = <div>Hello world</div>;
```

Este es un simple código JSX en React. Es JSX, no HTML dentro de Javascript.

Sin embargo, el navegador no entiende este JSX porque no es código JavaScript válido.

Esto se debe a que le estamos asignando un tag HTML a una variable que no es un string, es código HTML.



¿Qué es JSX?

JSX significa **JavaScript XML**, es una extensión de sintaxis de JavaScript.

JSX nos permite **escribir HTML** en React.

```
const element = <div>Hello world</div>;
```

Aunque JSX no es requerido para React, la mayoría de los usuarios lo encuentran útil como una asistencia visual. También permite la visualización de mensajes de error y advertencia más útiles.



¿Cómo funciona JSX?

Como el navegador no puede leer JSX, es **traducido a version JavaScript regular** a la hora de ejecutarse. Así es como funciona:

El código JSX se pasa a Babel (un compilador de JavaScript) que lo convierte el código Javascript que todos los navegadores pueden entender. Este compilador también convierte cualquier JavaScript ES6 funcionalidad en cualquier cosa que los navegadores viejos puedan reconocer. Por ejemplo convierte la clave "const" en "var".

*Un compilador - es un programa especial que traduce el código fuente en código de máquina, bytecode o cualquier otro lenguaje de programación.

*BABEL*

Ejemplo de cómo funciona JSX

Uso de JSX así:

```
<h1 id="jsx">This is JSX</h1>
```



React convierte este JSX a este código:

```
React.createElement("h1", { id: "jsx" }, "This is JSX");
```

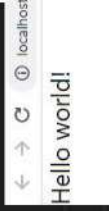


Incrustar (embedding) expresiones en JSX

Usar una variable:

Podemos declarar una variable llamada "name" y luego usarla dentro de JSX envolviendolo con llaves:

```
1 function App() {  
2   const message = "Hello world!";  
3   return (  
4     <div>{message}</div>  
5   );  
6 }  
7  
8 export default App;
```



```
1 function App() {  
2   const name = "Joe";  
3   return (  
4     <div>Hello {name}</div>  
5   );  
6 }  
7  
8 export default App;
```



Puedes poner cualquier expresión válida de Javascript dentro de llaves en JSX. Por ejemplo, $2 + 2$, `user.firstName`, o `functionName(parameter)` son todas expresiones válidas de Javascript.



Incrustar expresiones en JSX

Ejemplo del uso de una función:

```
function App() {  
  function getFullName(user) {  
    return user.firstName + ' ' + user.lastName;  
  }  
  const user = {  
    firstName: 'John',  
    lastName: 'Cena'  
  }  
  return (  
    <div>Hello, {getFullName(user)}!</div>  
  );  
}  
export default App;
```





Ejercicio 1

En App.js crea una función llamada `NumberRangeChecker`. La función debe aceptar un número.

La función debe retornar el rango de números basado en estas condiciones:

- Debajo de 50
- 50-100
- Sobre 100

Dentro de los componentes de JSX, usa expresiones incrustadas (embedded) para mostrar el número y el rango del número determinado.