

JSX

- Extensión de la sintaxis del lenguaje JavaScript que proporciona una forma de estructurar la representación de componentes utilizando una sintaxis familiar para muchos desarrolladores
- Apariencia similar a HTML y JavaScript.
- Sintaxis jugar entre HTML y JS.
- React recomienda este lenguaje para poder escribir los componentes que contienen el maquetado y la logica juntos, ya que se enfoca en las “piezas” que conforman al sitio.
- Esta extension nos permite crear desde variables que contengan por ejemplo un titulo, hasta un componente que nos muestre la estructura del componente y los metodos que se involucran en el mismo.
- Nos permite utilizar todas las herramientas de JS como bucles,condicionales,etc; hasta etiquetas de maquetacion,anclas y llamado a otros componentes utilizando escritura de etiquetas.
- En el caso que necesitemos colocar expresiones de JavaScript en atributos o contenido de etiquetas, debemos utilizar llaves que envuelvan dicho bloque de JS.
- Utiliza escritura camelCase y tiene algunas modificaciones de nombre en algunos atributos como “class” que ahora pasaria a ser “className” para que no se confunda con la palabra class recervada para crear los componentes de clase.
- Si una etiqueta tiene cierre en si misma, puedes cerrarla inmediatamente con “ /> “ .
- Esta extension es compilada por Babel para que el navegador pueda interpretarlo.

```
let persona={
  nombre:"Ana",
  edad:20,
  localidad:"Vicente Lopez"
}
let titulo = <h1>Hola {persona.nombre},te damos la bienvenida! </h1>
```

TIPOS DE COMPONENTES

COMPONENTES DE CLASES

- Forma de declarar componentes utilizando el identificador class y arma al componente utilizando un constructor como en la POO.
- Tambien utilizan un metodo llamado render() para poder devolvernos algo,usualmente el componente.
- Posee algunas características particulares al momento de escribirlo com el acceso al objeto de informacion que se le pasan a esos componentes.
- Necesita de que el nombre de dicho componente inicie con la primer letra mayuscula.
- Anteriormente se utilizaba este metodo de escritura para los componentes pero en la actualidad es mas utilizado los componentes funcionales.

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

imagen obtenid ade <https://es.reactjs.org/>

COMPONENTES FUNCIONALES

- Forma de declarar componenetes en forma de una funcion (puede ser utilizando la definicion de funciones tradicionales como arrow function).
- Necesita de que el nombre de dicho componente inicie con la primer letra mayuscula.
- Utiliza la palabra recervada return para devolver algo, en este caso el contenido del componente.

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

imagen obtenid ade <https://es.reactjs.org/>