

# React JS

**DEV.F**

dev

# ¿Qué es React?

**Creador:**

Jordan Walke - 2013

**Soportado:**

Facebook

**Ultima Version:**

React 18.2.0



React

es una biblioteca JavaScript de código abierto

Diseñada con el objetivo de crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones

En una sola página (SPA - Single Page Application).

Es mantenido por Facebook y la comunidad de software libre,



# ¿Qué necesito para aprender ReactJS?

Hay muchas comunidades de código para el aprendizaje sobre React

Bases: **Html** , **Css**, **JavaScript**

Recomendado: **Node.js**, **ES6,7**

Extras: Git & Terminal o línea de comandos.



# Ruta del desarrollador Frontend

Ya vimos html y css, trabajamos un poco de maquetación, estructuración y hemos visto la lógica con javascript.

Ahora con react vamos a tener un nuevo superpoder en programación.





# ¿Por qué aprender ReactJS?

React es una librería o biblioteca que se destaca por ser simple y por tener una api muy amigable que la podemos entender como si fuera JavaScript

Empresas y Startups como, Airbnb, Facebook, Instagram, Uber utilizan React.

# FRAMEWORK



# LIBRERÍA

Conjunto de herramientas que trabajan en un proyecto completo bajo ciertas reglas.



Herramienta con una sola utilidad específica.



Tiene funcionalidades integradas para que no necesites librerías externas.



La compatibilidad de sus funcionalidades está asegurada.



El framework define la forma en que debes desarrollar el proyecto.



Eres libre de usar las librerías que desees en la estructura que quieras.



Debes controlar la compatibilidad de cada librería con las demás.

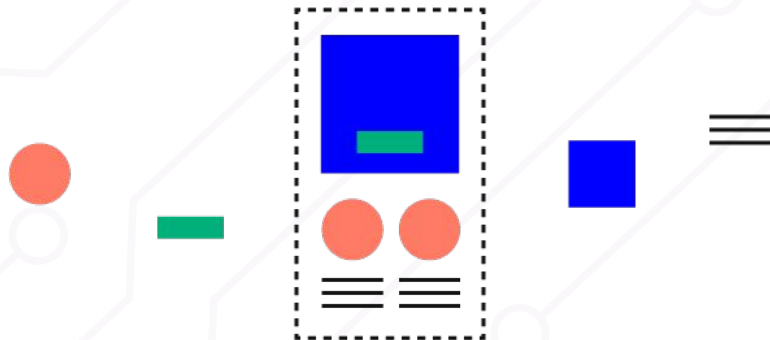


Puedes usar varias librerías según tus necesidades.



# Virtual Dom

El virtual Dom es una copia en memoria del dom del navegador, es decir del html, cuando hacemos una aplicación web hacemos un CRUD utilizando partes de HTML



Un ejemplo, es que si nosotros hacemos un cambio en el código para reflejar en pantalla React solo va a renderizar el cambio que acabamos de hacer

# **Cómo crear aplicaciones React**

**Tres formas diferentes de crear una aplicación React**

- 1. Poner React en un archivo HTML, con scripts externos.**
- 2. Utilizar un entorno React en el navegador como CodeSandbox.**
- 3. Crear una aplicación React en el ordenador usando una herramienta como Create React App / Vite.**



# JSX

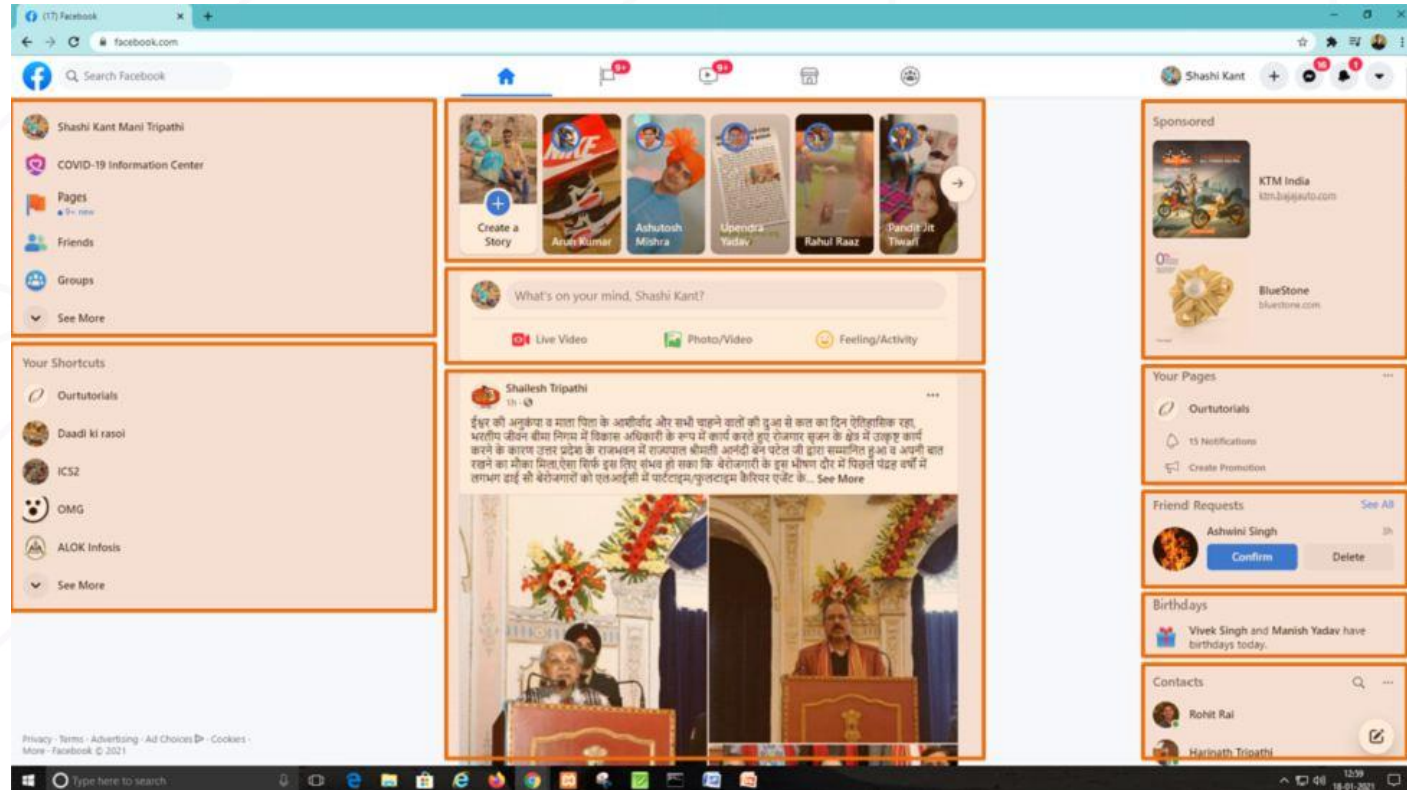
```
1  import * as React from 'react';
2  import * as ReactDOM from 'react-dom';
3
4  ReactDOM.render(
5    <h1 style={{color: 'red'}}>
6      Hello Lemoncode with TSX
7    </h1>
8    , document.getElementById('root'));
```

**JSX ES LA FORMA EN LA COMBINACIÓN NUESTRO HTML Y JAVASCRIPT, ES DECIR VAMOS A TRABAJAR TEMPLATES O BLOQUES DE CÓDIGO, DENTRO DE NUESTRA LÓGICA DE JAVASCRIPT**

# Componentes funcionales / clase



# Componentes



# Props en React

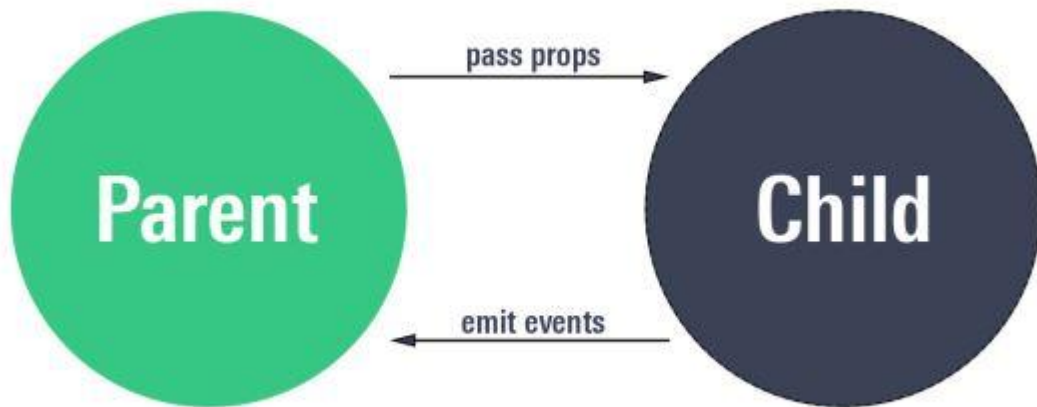
Los componentes y variables, estados pueden ser pasados por valores usando props

Los datos que se pasan a los componentes en JavaScript se llaman **props**.

```
ReactDOM.render(  
  <Greeting usuario="John!" />,  
  document.getElementById("root")  
);  
  
function Saludos(props) {  
  return <h1>Hola {props.usuario}</h1>;  
}
```

Los props están disponibles en los parámetros del componente al que se le pasan. Los props siempre se incluyen como propiedades de un objeto.

# Props





# Estados y ciclos de vida de React

El **estado (state)** es un concepto que se refiere a cómo cambian los datos de nuestra aplicación a lo largo del tiempo.

Para cambiar nuestra aplicación de elementos HTML estáticos a una dinámica con **la que el usuario pueda interactuar**, necesitamos estados.

# State



Data in the State control what you see in the View

```
const data = [  
  {  
    "name": "AFC Bournemouth",  
    "logo": "",  
    "manager": "Eddie Howe",  
    "stadium": "Dean Court",  
    "capacity": 11360  
  }  
]
```

EPL Teams	
1. AFC Bournemouth	
2. Arsenal	
3. Brighton & Hove Albion	
4. Burnley	
5. Chelsea	
6. Crystal Palace	
7. Everton	

# Props vs State



## Props vs State



- ✓ props are read-only
- ✓ props can not be modified
- ✓ state changes can be asynchronous
- ✓ state can be modified using `this.setState`

# ¿Qué son los eventos en React?

Los eventos son formas de obtener datos sobre una determinada acción que un usuario ha realizado en nuestra app.

Los props más comunes utilizados para manejar eventos son `onClick` (para eventos de click), `onChange` (cuando un usuario escribe en una entrada) y `onSubmit` (cuando se envía un formulario).

```
function ActionLink() {  
  function handleClick(e) {  
    e.preventDefault();  
    console.log('The link was clicked.');  }  
  
  return (  
    <a href="#" onClick={handleClick}>  
      Click me  
    </a>  
  );  
}
```