**React**

**Objetos literales.**

Documentación: <https://es.reactjs.org/>

<https://es.reactjs.org/docs/getting-started.html>

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working_with_Objects>

Los objetos literales también son conocidos como diccionarios o mapas en otros, en los lenguajes de programación. Un objeto literal se declara con las llaves **“{}”** y muchas personas confunden los objetos literales con un set, pero no es lo mismo.

Ejemplo de un objeto literal:

const persona = {};

Algo muy importante que debemos de comprender de los objetos literales en JS, es que **los objetos son pasados por referencias y no por valores**, esto quiere decir que cualquier modificación que sufra el objeto inicial, las demás referencias que se hicieron al objeto inicial se verán afectadas de igual manera, ya que todas las referencias están apuntando al mismo lugar. Ejemplo:

const persona = {

    nombre: "Benjamín",

    apellido: "Ayala",

    edad: 30,

    domicilio:{

        pais: "México",

        estado: "Colima",

        municipio: "Manzanillo",

        calle: "San pedrito, sector 7 #570",

        cp: 28219,

    }

};

const persona2 = persona;

persona2.nombre = "Pedro"; // Este cambio en nombre de persona2, afectará también el nombre en persona.

console.log(persona);

console.log(persona2);

**Nota importante.**

**Para poder romper esa referencia de un objeto inicial y hacer que no se modifiquen los cambios en el objeto inicial, una manera muy sencilla es utilizar algo conocido como el operador SPREAD, este operador se especifica con los “…” a la izquierda del objeto que se hará la referencia, esto esparcirá cada una de las propiedades del objeto al que se indica, sin apuntar a su referencia. Mejor dicho este operador crea un nuevo objeto y rompe la referencia del otro objeto. Esto es una sintaxis bastante común hoy en día en JS.**

[**https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Spread\_syntax**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Spread_syntax)

**Ejemplo:**

const persona = {

    nombre: "Benjamín",

    apellido: "Ayala",

    edad: 30,

    domicilio:{

        pais: "México",

        estado: "Colima",

        municipio: "Manzanillo",

        calle: "San pedrito, sector 7 #570",

        cp: 28219,

    }

};

const persona2 = persona;

persona2.nombre = "Pedro"; // Este cambio en nombre de persona2, afectará también el nombre en persona.

const persona3 = { ... persona }

persona3.nombre = "Juan"; // Este cambio en nombre de persona3, no afectará el nombre en persona.

console.log(persona);

console.log(persona2);

console.log(persona3);

**Existen otro operador que se llamado REST y la diferencia está dependiendo de donde se utilice con el objeto.**

[**https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions/rest\_parameters**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions/rest_parameters)