

Objetos en JavaScript II

Fundamentals

Existencia de propiedades (in)

El operador *in*, nos permite comprobar si una propiedad existe dentro de un objeto. Su sintaxis es la siguiente:

`'key' in obj`

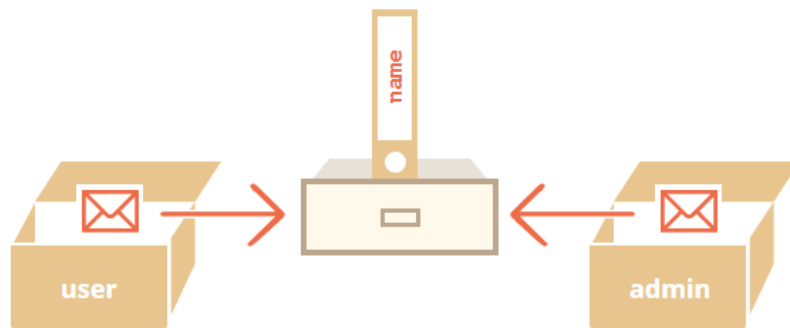
El resultado es un valor booleano, true si existe o false en caso contrario.

```
JS app.js > ...
1  let persona = {nombre: 'John', apellido: 'Doe', edad: 35};
2
3  if ('apellido' in persona) {
4      console.log('La propiedad existe en el objeto');
5  } else {
6      console.log('No existe a propiedad en el objeto');
7  }
```

Referencias y valores en Objetos

A diferencia de las variables ordinarias, los objetos no copian su contenido, si no una referencia al objeto.

```
1 let user = { nombre: 'John'};  
2 let admin = user; // copia la referencia al objeto
```



Se crea una copia de la referencia al objeto.

Referencias y valores en Objetos

Dos objetos son iguales si y solo si ellos son el mismo objeto

```
let objA = {}  
let objB = objA;  
  
console.log(objA == objB); // True  
console.log(objA === objB); // True
```

Ambos hacen referencia al mismo objeto

```
let objA = {}  
let objB = {};  
  
console.log(objA == objB); // False  
console.log(objA === objB); // False
```

Los objetos son diferentes entre si.

Clonado y fusionado de Objetos

El proceso de copia de objetos. Crea una referencia al mismo objeto, pero si yo quisiera una copia de la estructura del objeto, para efectos prácticos?

?

Clonado y fusión de Objetos

```
let persona = { nombre: 'John', apellido: 'Doe', edad: 30};
let copia = {};

// Recorreremos el objeto persona y copiamos su estructura en
// el objeto copia
for (let key in persona) {
  copia[key] = persona[key];
}

console.log(persona);
console.log(copia);

console.log(persona == copia); // false
console.log(persona === copia); // false
```

Clonado y fusionado de Objetos

Pero también podemos utilizar el método `Object.assign` de JS, cuya sintaxis es la siguiente:

`Object.assign(destino, fuente-1, fuente-2, ..., fuente-n);`

```
let persona = { nombre: 'John', apellido: 'Doe', edad: 30};  
  
let copia = Object.assign({}, persona);  
  
console.log(persona);  
console.log(copia);  
  
console.log(persona == copia);  
console.log(persona === copia);
```

Clonado y fusión de Objetos

Este método copia todas las propiedades de uno o varios objetos a un objeto destino.

```
let destino = { a: 1, b: 2 };  
let fuente = {b: 4, c: 5}  
  
let resultado = Object.assign(destino, fuente);  
|  
console.log(destino); // { a: 1, b: 4, c: 5};  
console.log(resultado); // { a: 1, b: 4, c: 5};
```

Si la propiedad existe se sobrescribe

Clonado y fusiónado de Objetos

También podemos copiar varias propiedades de otros objetos.

```
let user = { nombre: 'George', age: 30};  
let permisoVer = { puedeVer: true};  
let permisoEditar = {puedeEditar: true};  
  
console.log(user);  
|  
Object.assign(user, permisoVer, permisoEditar);  
  
console.log(user);
```

Fuente: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/assign

Claves, valores y entradas de Objetos

En JavaScript, contamos con tres métodos importantes para conocer las keys o propiedades de un objeto, así como sus valores y sus pares *clave:valor*, estos métodos son:

```
Object.keys(Obj); // retorna un array con las propiedades del objeto  
Object.values(Obj); // retorna un array con los valores del objeto  
Object.entries(Obj); // retorna un array de pares clave:Valor
```

Claves, valores y entradas de Objetos

```
let persona = { nombre: 'George', apellido: 'Rush', age: 30};  
  
console.log(Object.keys(persona));  
console.log(Object.values(persona));  
console.log(Object.entries(persona));
```

Hands On

Reto 1

Se desea almacenar los datos de un producto en un registro. Cada producto debe contener información concerniente a: *código, nombre del producto, precio*.

Diseñar un programa que permita cargar la información de n productos en una lista.

Reto 2

Realice las modificaciones necesarias, para que al programa creado en el reto anterior, nos permita mostrar una tabla con la siguiente información

Nombre	Precio
Producto1	precio1
Producto2	Precio2
Producto-n	Precio-n
Total	TotalProductos

Reto 3

La empresa 'ACME LTDA' tiene un registro de empleados, por cada empleado se guarda: DNI, nombre, apellido, zona (norte, sur, centro), sueldo base, comisión y total.

Desarrolle un programa que calcule el total del salario por empleado, utilizando la formula:

$\text{salario} = \text{sueldo base} + \text{comisión}$

Se debe imprimir en pantalla, el nombre y el salario de cada empleado. El salario debe ser una propiedad de cada objeto creado.

Reto 4 – RECORDAR ES VIVIR...!

Dado un array de N elementos enteros entre 0 y 100, crear un programa que invierta el contenido de dicho array.

Nota

No esta permitido el uso de la función reverse.

Reto 5 – RECORDAR ES VIVIR...!

Dado el string 'JavaScript is super fun!', crear un programa que imprima el string al revés: '!nuf repus si tpircsavaj'.

Nota

No esta permitido el uso de la función reverse.

Reto 6

Dado un objeto persona con propiedades: *nombre*, *apellido*, *edad*, *ciudad*, crear una copia del objeto en una variable copiaPersona, añadir los atributos puedeLeer, puedeEditar, puedeBorrar.

Nota

No se debe afectar el objeto original.

Reto 7

Del reto anterior eliminar la propiedad ciudad, en el objeto persona si existe.

Nota

No se debe afectar el objeto copia.