

Inmutabilidad de datos primitivos

Como ya vimos anteriormente los tipos de datos string, number, boolean, undefined y null son primitivos, esto, entre otras cosas, significa que son INMUTABLES, es decir que su valor nunca cambiará salvo que hagamos una reasignación.

```
const string = 'hola';  
  
string.toUpperCase();  
  
console.log(string);
```

```
let string = 'hola';  
  
string = string.toUpperCase();  
  
console.log(string);
```



Mutabilidad de los datos referencia



Por otro lado los arrays y los objetos son conocidos como datos de tipo referencia, eso significa que cuando creamos un array o un objeto su valor no se guarda directamente, si no que guardamos una REFERENCIA a su valor, eso significa que al modificarlos, cambiamos el valor de la referencia y se quedan modificados, por lo que destruimos el array inicial.

```
const numbers = [1, 2, 3, 4, 5];  
  
numbers.pop();  
  
console.log(numbers);
```

El spread operator es un operador que nos va a permitir expandir los elementos iterables (arrays) y los objetos, esto simplifica la recogida de valores en una estructura de datos Su sintaxis es “...” más simple imposible.

```
const numbers = [1, 2, 3, 4, 5];  
  
console.log(...numbers); // 1 2 3 4 5
```

Esto nos sirve, entre otras cosas, para copiar arrays de forma segura creando una nueva referencia en lugar de hacer una copia de la misma referencia.

```
const numbers = [1, 2, 3, 4, 5];  
  
const numbersCopy = [...numbers];  
  
console.log(numbersCopy); // [1, 2, 3, 4, 5]
```

Este operador también lo podemos usar en objetos

```
const person = {  
  name: 'Dorian',  
  age: 25  
};  
  
const address = {  
  street: 'Calle Falsa 123',  
  postalCode: '00001'  
};  
  
const personWithAddress = { ...person, ...address };
```

```
{  
  name: 'Dorian',  
  age: 25,  
  street: 'Calle Falsa 123',  
  postalCode: '00001'  
}
```