



Certified Tech
Developer

The Ultimate Degree

Front End III

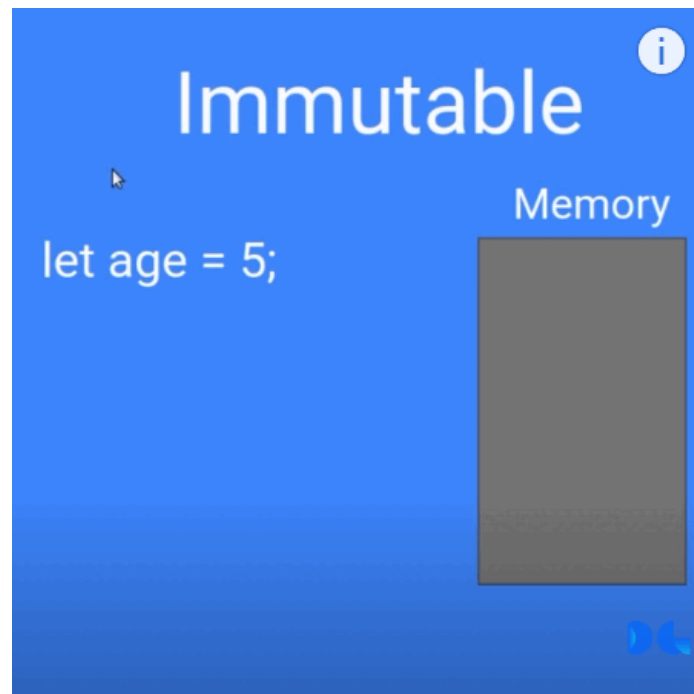
Concepto de Inmutabilidad

¡Ya lo dijimos! Aunque no lo creamos, ya venimos trabajando con el concepto de inmutabilidad. Ahora, ¡vamos a ponerle palabras!

Lo cierto es que en JS los tipos de datos primitivos como string, boolean, number, undefined y null son inmutables. Veamos un ejemplo:

```
let name = "Digital"
let copyName = name
copyName = "House"
//name = "Digital" y copyName = "House"
```

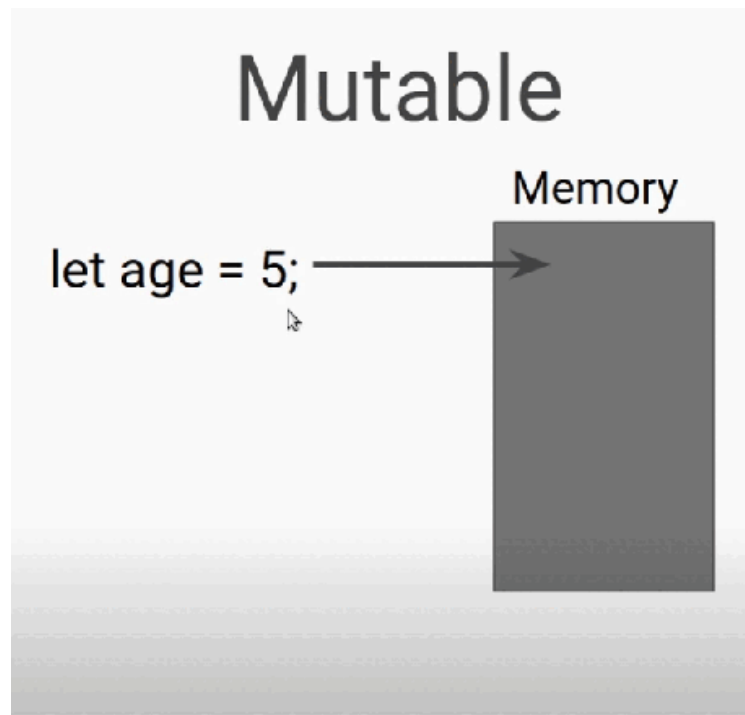
Esto parece un comportamiento muy normal del lenguaje, pero lo cierto es que esto sucede porque el tipo de dato string es **immutable**. Cada variable es una posición en memoria distinta y entonces al modificar una no afecta a la otra.



En cambio, algunos ejemplos de mutabilidad en JS son los objetos y arreglos. Podemos verlo en el siguiente ejemplo:

```
let user = {  
  name: "Digital",  
  surname: "House"  
}  
  
let copyUser = user  
  
copyUser.surname = "School"  
  
//user.surname = "School" y copyUser.surname = "School"
```

En este caso, ambos objetos (datos **mutables** en JS) a pesar de ser variables distintas, pertenecen a la misma porción de memoria. Por lo que si se modifica una, por consiguiente se modificará la otra.



Ahora bien, **el concepto de inmutabilidad se aplica y se utiliza con mucha frecuencia en el mundo de React**. Las valiosas props de los componentes de React son objetos **read-only**, es decir, inmutables. A pesar de que JS nos permite mutar objetos, aplicado al contexto de la librería, esto sería un error.

Más adelante veremos qué aspectos de un componente podemos **mutar** y cómo hacerlo, pero por el momento quedémonos con que **las props son inmutables**.

Algunas ventajas que trae esta implementación son:

- **Side effects free:** con esta implementación nos aseguramos que nuestros objetos y componentes no se modifiquen en lugares inesperados, afectando con ello la ejecución de nuestro programa.
- Código mucho más **limpio y predecible**.
- **Cambios centralizados:** sabemos que los objetos inmutables son asignados solamente una vez, por lo que sería fácil trackear su valor.
- Mayor facilidad a la hora de **testear y debuggear**.