

JavaScript desde cero

Módulo 5

Declaración de variables

Declaración de variables

En versiones anteriores de ECMAScript, teníamos una sola palabra para declarar variables: la palabra **var**.

Con esta palabra reservada podíamos crear variables con dos posibilidades: **reasignación** y **redefinición**.

¿Qué significa esto?



Un poco de contexto

Recordemos que una variable representa un espacio de memoria reservado que puede contener un único dato a la vez. ¿Un qué?

Veamos...

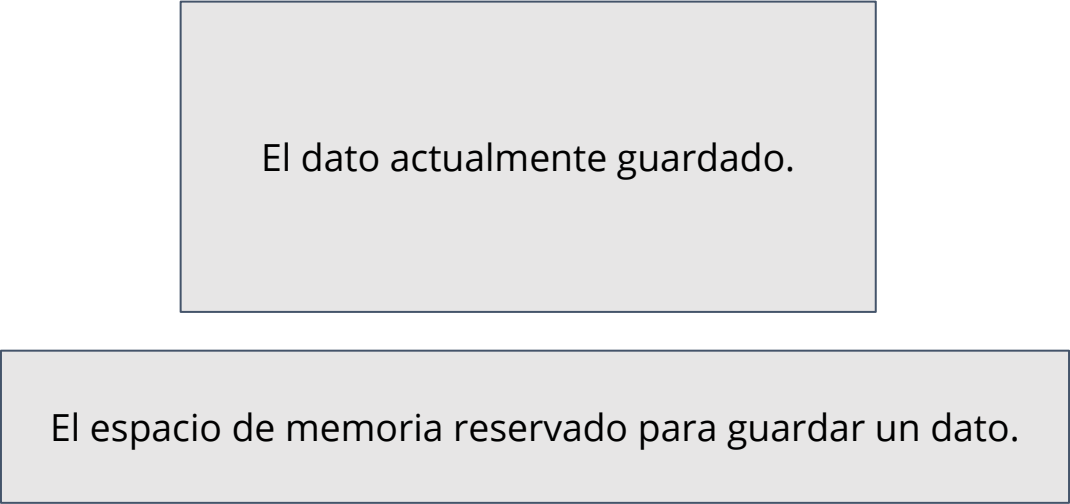
Las computadoras fueron diseñadas con, a grandes rasgos, dos lugares en donde guardar datos para su posterior uso:

- Un lugar para guardar datos a largo plazo, como es el disco duro / estado sólido.
- Un lugar para guardar datos de corto plazo, como es la memoria RAM.

Las variables permiten hacer uso de la memoria RAM para guardar información de corta duración. Para almacenar esta información, se reserva un espacio en esta memoria. Ese espacio no podrá ser ocupado por otra variable hasta que se “libere” cuando el programa termine.

Un poco de contexto

Entonces, una variable tiene estas dos partes:



El dato actualmente guardado.

El espacio de memoria reservado para guardar un dato.

Un poco de contexto

¿Y que tiene que ver todo esto con la palabra reservada **var**?
Pues, con **var** podemos...

Reasignar una variable, es decir,
cambiar el dato actual.

Redefinir una variable, es decir, cambiar su ubicación
en memoria.

Declaración de variables en ES6

El estándar ECMAScript 6 introduce dos palabras nuevas para la creación de variables:

- Mediante la palabra reservada **const** creamos una variable que no puede redefinirse ni reasignarse (no puede cambiar su posición ni su valor).
- Mediante la palabra reservada **let** creamos una variable que no puede redefinirse, pero si reasignarse (solo puede cambiar su valor).



Declaración de variables en ES6

¿Cuál es su utilidad? Promueven la **integridad de los datos**. ¿Qué es eso?

Supongamos que te encuentras en la oficina de tu trabajo. En un momento Juan, un empleado, llama a secretaría para pedirle un reporte de ventas y le dice "Hola, necesito el reporte".

En otro lado de la oficina, se encuentra Santiago, que llama a la misma oficina para pedirle el mismo reporte, pero de una forma diferente: "Hola, necesito el reporte de ventas que abarca desde el 01-01-2021 hasta el 02-03-2021 en mi escritorio, a más tardar a las 18hs de este día"

En el ejemplo anterior, Juan dio una orden a Secretaría muy poco específica, por lo que el personal deberá deducir muchas cuestiones. Si hay varias personas que se comunican como Juan, sucederá que en esa deducción, Secretaría puede cometer errores.

Sin embargo, en el caso de Santiago, la orden fue muy precisa. Si todas las personas se comunican como lo hizo Santiago, entonces Secretaría podrá organizarse con mayor facilidad y así cometer menos errores a la hora de repartir esos reportes.

Declaración de variables en ES6

¿Qué tiene que ver esto con integridad de los datos? Pues, Santiago proveyó mayor integridad de datos a Secretaría que Juan. En esta analogía, podemos comparar a Santiago y a Juan con los programas de computadora, y a Secretaría con el Sistema Operativo.

Cuando un programa, a través de un lenguaje de programación, le da instrucciones precisas de donde guardar la información, el Sistema Operativo comete menos errores ya que debe “deducir” menos cosas.

Si declaro una variable con `var`, estoy obligando al Sistema Operativo a deducir donde colocar esa variable, y si cambia su posición, a hacer los cambios necesarios. En cambio, con `const` y `let`, al definir su posición como fija, se evita que el SO deba hacer dichas deducciones y, por lo tanto, el dato se vuelve más predecible.

Declaración de variables en ES6

Tabla comparativa de las palabras reservadas para declarar variables

	Reasignar (cambiar valor)	Redefinir (cambiar posición)
const	NO	NO
let	SI	NO
var	SI	SI

Se recomienda intentar siempre utilizar **const**. Si se necesita *reasignar* (es decir, cambiar el valor) de la variable, se puede utilizar **let**. La palabra **var** la vamos a dejar como última opción.

¡Muchas gracias!

¡Sigamos trabajando!