



Las variables

DigitalHouse>



**Certified Tech
Developer**

The Ultimate Degree

“

Las **variables** son espacios de memoria en la computadora donde podemos almacenar distintos tipos de datos.



”

Tipos de variables

En JavaScript existen estos tipos de variables:

- let
- const

Para declarar una variable escribimos el tipo y el nombre que le queremos dar a la variable:

```
{ }  
let contador;  
const url;
```

Veamos cada parte con más detalle...

Declaración de una variable

```
let nombreSignificativo;
```



let

La palabra reservada **let** le indica a JavaScript que vamos a **declarar una variable de tipo let**.



Nombre

- Solo puede estar formado por letras, números y los símbolos \$ (pesos) y _ (guion bajo).
- No pueden empezar con un número.
- No deberían contener ñ o caracteres con acentos.



Es una buena práctica que los nombres de las variables usen el formato Camel case, como `variableEjemplo` en vez de `variableejemplo` o `variable_ejemplo`.

Declaración de una variable

```
let miVariable;
```

No es lo mismo que:

```
let MiVariable;
```



JavaScript es un lenguaje que **hace diferencia entre MAYÚSCULAS y minúsculas**. Por eso, es bueno seguir un estándar a la hora de escribir nombres.

“

Las **buenas prácticas**, si bien no son obligatorias para que nuestro código funcione, van a permitir que este sea **más fácil de leer y de mantener.**

”



Asignación de un valor

```
let miApodo = 'Hackerman';
```



Nombre

El nombre que nos va a servir para identificar nuestra variable cuando necesitemos usarla.

Asignación

Le indica a JavaScript que queremos guardar el valor de la derecha en la variable de la izquierda.

Valor

Lo que vamos a guardar en nuestra variable. En este caso, un texto.

Asignación de un valor

La **primera vez** que declaramos una variable es necesaria la palabra reservada **let**.

```
{ } let miApodo = 'Hackerman';
```

Una vez que la variable ya fue declarada, le asignamos valores **sin let**.

```
{ } miApodo = 'El Barto';
```


Declaración con const

Las variables **const** se declaran con la palabra reservada **const**.

```
{} const email = "mi.email@hotmail.com";
```

Las variables declaradas con **const** funcionan igual que las variables **let**, estarán disponibles solo en el bloque de código en el que se hayan declarado.

Al contrario de **let**, una vez que les asignemos un valor, no podremos cambiarlo.

```
email = "mi.otro.email@hotmail.com";  
{}  
// Error de asignación, no se puede cambiar  
// el valor de un const
```

Declaración con let o const

Como dijimos antes, tanto **let** como **const** son accesibles dentro del bloque donde son declaradas.

Por esta razón solo podemos declararlas una vez. Si volvemos a declararlas, JavaScript nos devolverá un error.

```
{  
  let contador = 0;  
  let contador = 1;  
  // Error de re-declaración de la variable  
  
  const email = "mi.email@hotmail.com";  
  const email = "mi.nuevo.email@hotmail.com";  
  // Error de re-declaración de la variable  
}
```

Declaración con var

Existe otra forma de declarar variables que **ya no se utiliza y no lo veremos en la materia**, pero que puede ser que la encuentres en códigos que tengan algunos años.

Estas variables se declaran de una manera similar, con la diferencia que utilizamos la palabra reservada **var**.

```
{ } var contador = 1;
```

La principal diferencia entre **var** y **let** es que **var** será accesible de manera global por todo nuestro código y no estará limitado el acceso a solo el bloque de código donde fue declarado como es el caso de **let**.

Por convención y buenas prácticas es recomendado el uso de **let**.

Los bloques de código son normalmente determinados por las llaves **{ }**.

Veamos un ejemplo:

var

```
if (true) {  
  var nombre = "Juan";  
}  
  
console.log(nombre);  
// Ok, muestra "Juan"
```

Cuando usamos **var**, JavaScript ignora los bloques de código y convierte nuestra variable en global.

Eso quiere decir que si hay otra variable **nombre** en nuestro código, seguramente estemos pisando su valor.

let

```
if (true) {  
  let nombre = "Juan";  
}  
  
console.log(nombre)  
// Error: nombre no existe
```

Cuando usamos **let**, JavaScript respeta los bloques de código. Eso quiere decir que nombre no podrá ser accedida fuera del **if**.

También quiere decir que podemos tener variables con el mismo nombre en diferentes bloques de nuestro código.

“

Las **palabras reservadas** como **var**, **let** y **const** solo pueden utilizarse para el propósito que fueron creadas.

No pueden ser utilizadas como:
nombre de variables, funciones,
métodos o identificadores de objetos.

”



DigitalHouse>