



Variables

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

TIPOS DE VARIABLE

En Javascript existen tres tipos de variables:

- **var** (⚠ es la forma más antigua de declarar variables. Se desaconseja su uso).
- **let**
- **const**

Para declarar una variable escribimos el **tipo** (var, let o const) y el **nombre** que le queremos dar a la variable:

```
{  
var nombre;  
let contador;  
const url;
```

Veamos cada parte más en 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

Sólo puede estar formado por letras, números y los símbolos \$ y _ (guión bajo).
No pueden empezar con un número.
No deberían contener ñ o caracteres con tildes.



Es una **recomendable** que los nombres de las variables usen el formato **camelCase**, 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 **recomendaciones** (o buenas prácticas) permiten que el código sea **más fácil de leer, compartir y de mantener.**

ASIGNACIÓN DE VALORES A UN VARIABLE

Asignamos valores a una variable usando el **operador de asignación**.



```
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.

MODIFICANDO EL CONTENIDO DE UNA VARIABLE

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

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

⚠ Si la variable ya fue declarada le asignamos valores ***sin la palabra*** reservada **let**.

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



Nuestra variable **guardará siempre el último valor** asignado, eso quiere decir que ***si volvemos a asignarle un valor pisamos el anterior.***

DECLARACIÓN DE VARIABLES CON **LET**

Se declaran utilizando la **palabra reservada let**.

```
{ let contador = 0;
```

La variable declarada con **let** únicamente será accesible en el bloque de código en el que fue declarada.

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

Las variables de tipo **let** permiten cambiar su valor luego de asignado.

DECLARACIÓN CON **CONST**

Se declaran con la palabra reservada **const**.

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

Las variables declaradas con **const** también estarán disponibles sólo en el bloque de código en el que se hayan declarado.

Las variables de tipo **const NO permiten** cambiar su valor luego de asignado.

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

REDECLARACIÓN DE UNA VARIABLE

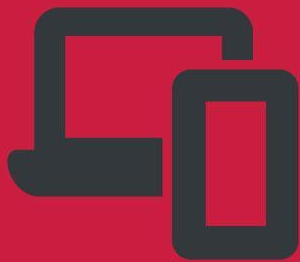
Tanto las variables de tipo **let** como las de tipo **const** son accesibles dentro del bloque donde son declaradas.

Por esta razón sólo podemos **declararlas una vez dentro de dicho bloque**. 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
```

{}



Tipos de datos

Llamamos **tipos de datos** a los diferentes valores que puede recibir una variable.

Los **tipos de datos permiten conocer** las **características y funcionalidades** que estarán disponibles para un variable.

TIPOS DE DATOS SIMPLES (o primitivos)

NUMÉRICOS (number)

```
{}
```

```
let edad = 35; // número entero  
let precio = 150.65; // con decimales
```



Como JavaScript está escrito en inglés
usaremos un punto para separar los
decimales.

CADENAS DE CARACTERES (string)

```
{}
```

```
let nombre = 'Mamá Luchetti'; // comillas simples  
let ocupacion = "Master of the sopas"; // comillas dobles  
tienen el mismo resultado
```

LÓGICOS O BOOLEANOS (boolean)

```
{}
```

```
let laCharlaEstaReCopada = true;  
let hayAsadoAlFinal = false;
```

TIPOS DE DATOS ESPECIALES

NaN (NOT A NUMBER)

```
{}
```

```
let malaDivision = "hola"/2; // NaN no es un número
```

NULL (VALOR NULO)

Lo asignamos nosotros para indicar un valor vacío o desconocido.

```
{}
```

```
let temperatura = null; // No llegó un dato, algo falló
```

UNDEFINED (valor sin definir)

Las variables tienen un valor indefinido hasta que les asignamos un valor.

```
{}
```

```
let otraVariable; // undefined, no tiene valor  
otraVariable = "¡Hola!"; // Ahora si tiene un valor
```

COMBINANDO TEXTO Y VARIABLES

Template strings

Son cadenas de texto que permiten intercalar variables.

- Se utilizan las **comillas invertidas** (```) para delimitar las cadenas.
- La sintaxis `${ }` permite interpolar las variables.

```
let nombre = "Marcelo";
```

```
let apellido = "Gallardo";
```

ES6

```
console.log(`Hola ${nombre} ${apellido}`);
```

Template strings

También podemos hacer operaciones dentro de la interpolación.

```
let nombre = "Yuri";
```

```
let edad = 18;
```

ES6

```
console.log(`Soy nombre y el próximo año cumpliré edad + 1`);
```

COMENTANDO EL CÓDIGO

Los comentarios son partes de nuestro código pero **no se ejecutan**.

Siempre comienzan con dos barras inclinadas //

Los usamos para explicar o **dejar información útil** para nuestro equipo de trabajo, otros programadores o simplemente para nuestro yo del futuro.



```
// Math.round() retorna el valor redondeado al entero  
más cercano.
```

```
let redondeado = Math.round(20.49);
```

DigitalHouse >
Coding School



¡NOS VEMOS LA PRÓXIMA!