



Tecnología

JAVASCRIPT 01 | Módulo: Repaso general,
Objetos Literales y Estructuras de control

<TEMA DE LA CLASE>

OBJETIVOS |

LAS VARIABLES

VARIABLES |

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

VARIABLES |

TIPOS DE VARIABLE

En Javascript existen tres tipos de variables:

- var
- let
- const

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

```
{ }  
var nombre;  
let contador;  
const URL;
```

VARIABLES |

DECLARACIÓN DE UNA VARIABLE

`var nombreSignificativo;`



var

La palabra reservada **var** le indica a Javascript que vamos a **declarar una variable de tipo var**.



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 acentos.



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

VARIABLES |

DECLARACIÓN DE UNA VARIABLE

```
var miVariable;
```

¡...no es lo mismo que...!

```
var 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.

VARIABLES |

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

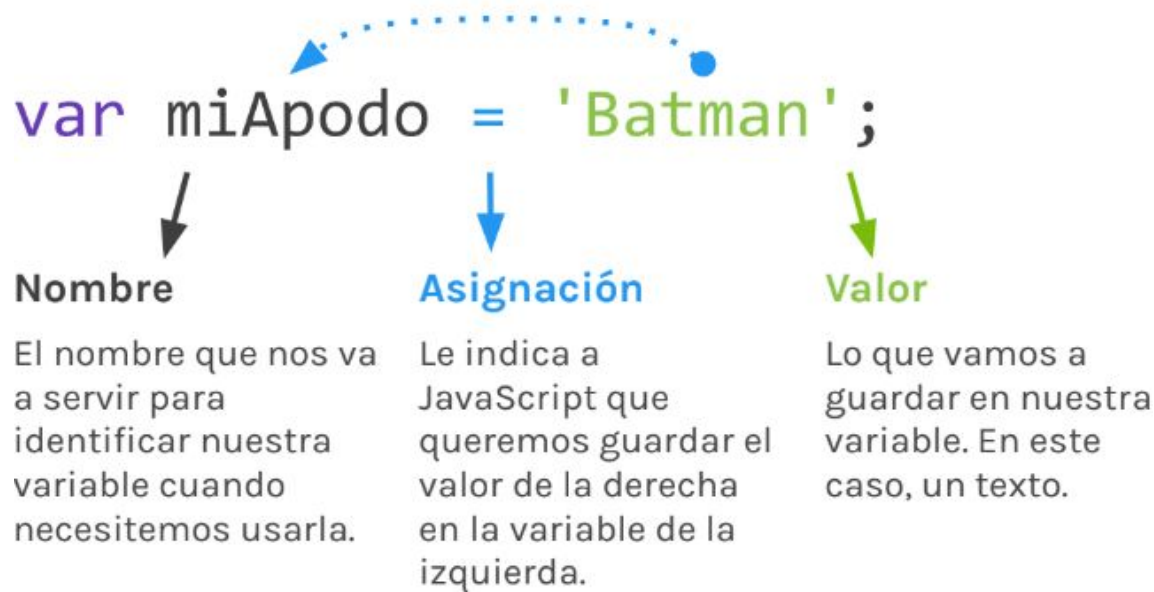
VARIABLES |

ASIGNACIÓN DE UN VALOR

Cuando declaramos una variable, también podemos al mismo tiempo asignarle un valor. Eso lo hacemos con el operador de asignación.

VARIABLES |

ASIGNACIÓN DE UN VALOR



VARIABLES |

ASIGNACIÓN DE UN VALOR

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

```
{ } var miApodo = 'Batman';
```

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

```
{ } miApodo = 'Acuaman';
```



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

VARIABLES |

DECLARACIÓN CON LET

Estas variables se declaran de una manera similar con la diferencia que utilizamos la palabra reservada `let`.

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

La principal diferencia entre `var` y `let` es que `let` sólo será accesible en el bloque de código en el que fue declarada.

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

VARIABLES |

DECLARACIÓN CON LET

VAR

```
if (true) {  
  var nombre = "Julieta";  
}
```

```
console.log(nombre);  
// Ok, muestra "Julieta"
```

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 = "Julieta";  
}
```

```
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.

VARIABLES |

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 sólo en el bloque de código en el que se hayan declarado. Al contrario que **let**, una vez que les asignemos un valor, no podremos cambiarlo.

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

VARIABLES |

DECLARACIÓN CON **CONST**

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

Las variables
las variables
código



Como habrás visto, el nombre de las constantes (variables declaradas con la palabra reservada **const**) suele declararse en MAYÚSCULAS, ¡esto es una buena práctica!

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

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

VARIABLES |

DECLARACIÓN CON LET O CONST

Como dijimos antes tanto `let` como `const` son accesibles dentro del bloque donde son declaradas. Por esta razón sólo 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
```


VARIABLES |

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

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