

# JavaScript - JS

- Es un lenguaje de programación interpretado
- Sirve para programar el comportamiento de las páginas
- Funciona junto con HTML y CSS

## Consola

La consola es un panel que muestra la información de la web. Tiene una línea para ejecutar las instrucciones que le daremos a la computadora con nuestro código.

## console.log ()

El método para imprimir mensajes en la consola. Imprime lo que esté entre ( )

```
console.log('Hi there!');  
// Prints: Hi there!
```

## Comentarios

Sirven para explicar un bloque de código, ordenarlo o hacer aclaraciones al respecto sin afectar el código.

De una línea se crean con dos barras diagonales consecutivas //

```
// This line will denote a comment
```

De párrafo /\* para abrir

y para cerrar \*/

```
/*  
The below configuration must be  
changed before deployment.  
*/
```

## Tipos de Datos

1. **number**: cualquier número, incluidos los números con decimales.

```
let amount = 6;  
let price = 4.99;
```

2. **String**: cadenas, son agrupación de caracteres. Pueden ser letras, números, espacios, símbolos, etc. Se usan con comillas simples: '...'

```
let single = 'Wheres my bandit hat?';  
let double = "Wheres my bandit hat?";
```

3. **boolean**: solo tiene dos valores posibles: TRUE (verdadero) o FALSE (falso).

```
let lateToWork = true;
```

4. **null**: ausencia intencional de un valor.

```
let x = null;
```

5. **undefined**: representa la ausencia de un valor, no se ha definido un valor. Las variables no inicializadas tendrán este valor.

```
var a;  
  
console.log(a);  
// Prints: undefined
```

6. **simbol**: símbolos, son identificadores únicos, útiles en codificaciones más complejas.
7. **object**: colecciones de datos relacionados.

## Variables

Las variables se utilizan para almacenar valores de datos.

Utilizamos la palabra clave **var** para declarar variables. El signo = asigna el valor a la variable.

En el ejemplo se define x como la variable. Luego a la variable x se le asigna el valor 6.

```
var x;  
  
x = 6;
```

Tener en cuenta que JavaScript es Case Sensitive, VAR o Var no es lo mismo que la palabra clave es **var**.

Los nombres de las variables se llaman identificadores.

Un identificador tiene que empezar por una letra, un guion o un signo de peso, los caracteres siguientes si pueden ser números. Tener en cuenta la distinción de mayúsculas y minúsculas.

## Declarar variables:

**var** declara una variable, su scope es local. Se le puede asignar un valor y puede cambiar su valor.

**let** declara una variable local en un bloque de código. Se le puede asignar un valor y puede cambiar su valor.

**const** declara un variable de solo lectura en un bloque de código. No es posible cambiar su valor mediante asignación.

### Operadores de Asignación

asignan valores: (+ , - , \* , / , % , \*\*)

```
// Addition
5 + 5
// Subtraction
10 - 5
// Multiplication
5 * 10
// Division
10 / 5
// Modulo
10 % 5
```

**%** es Modulo, devuelve el resto de una división, el número de la derecha se divide en el número de la izquierda tantas veces como sea posible.

```
// calculates # of weeks in a year, rounds down to
nearest integer
const weeksInYear = Math.floor(365/7);

// calculates the number of days left over after 365 is
divded by 7
const daysLeftOver = 367 % 7 ;

console.log("A year has " + weeksInYear + " weeks and "
+ daysLeftOver + " days");
```

**\*\*** es Exponencial

**++** es incremento, **i++** es igual que **i+1**

**--** es decremento, **i--** es igual que **i-1**

Un operador de asignación asigna un valor a su operando izquierdo basado en el valor de su operando derecho.

(+= , -= , \*= , /= , %= , \*\*=)

Ej. **x += y** es lo mismo que **x = x + y**

```
let number = 100;

// Both statements will add 10
number = number + 10;
number += 10;

console.log(number);
// Prints: 120
```

## Operadores de Relacionales / de Comparación

igualdad ( **==** )

desigualdad ( **!=** )

estrictamente iguales ( === ) (ambos operandos iguales y del mismo tipo)

estrictamente desiguales ( !== )

mayor que ( > )

mayor igual que ( >= )

menor que ( < )

menor igual que ( <= )

### Operadores lógicos

no ( ! )

and ( && )

or ( || )

### Conversion de Tipos de Datos

**typeof** nos dice que tipo de dato es la variable

JS es un lenguaje dinamico, la misma variable se puede utilizar para contener diferentes tipos de datos.

```
var x;           // Now x is undefined
x = 5;           // Now x is a Number
x = "John";      // Now x is a String
```

```
var x = 16 + 4 + "Volvo";
```

Resultado: "20Volvo"

La lectura es de izquierda a derecha por lo tanto primero hace la suma entre 16 (number) + 4 (number) y al resultado le asigna Volvo convirtiendo el resultado en un String

```
var x = "Volvo" + 16 + 4;
```

Resultado: Volvo164

Como el primer dato es de tipo String para procesar el pedido transforma el 16 (number) a tipo String, lo concatena y hace lo mismo con el 4 siguiente.

**parseInt()** es para cambiar una variable de tipo String a number

**\*\*resumen en progreso CONTINUARÁ\*\***