

INTRO A JAVASCRIPT

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Objetivo de la sesión

- Hablaremos un poco de **JavaScript**
- Veremos que es el estándar de **ecmascript**
- Hablaremos de la declaración de variables en las **versiones de ecma 6+**

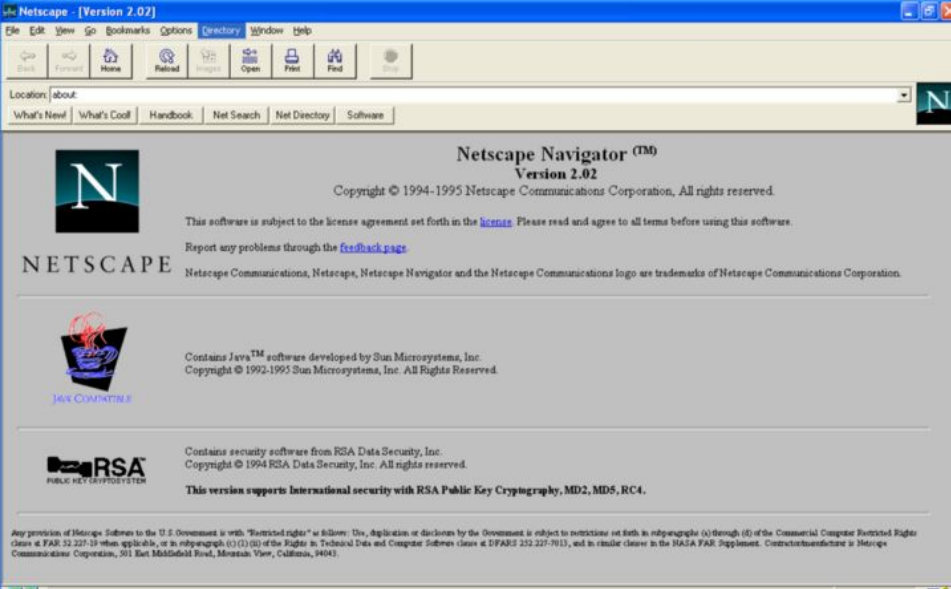


Javascript

JS

DEV.F
DESARROLLAMOS(PERSONAS);

dev

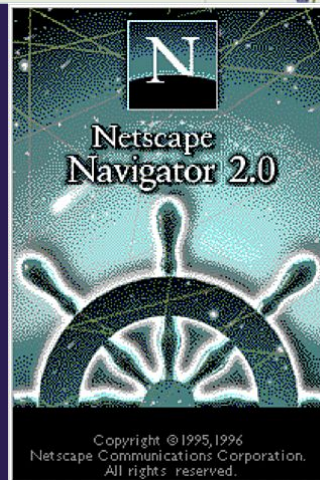
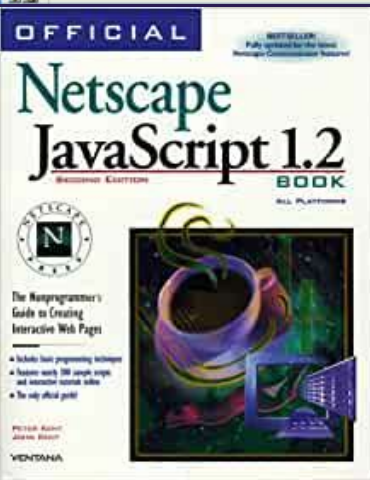


Un poco de historia...

Javascript es un lenguaje creado por Netscape. Fue introducido en su navegador Netscape 2.0 en 1995.

Se dice que fue creado en una semana por Brendan Eich.

Originalmente fue llamado Mocha y previo a su lanzamiento LiveScript, pero gracias a un acuerdo con Sun Microsystem (antigua propietaria de Java), pasó a llamarse finalmente JavaScript.



Hablemos de JavaScript

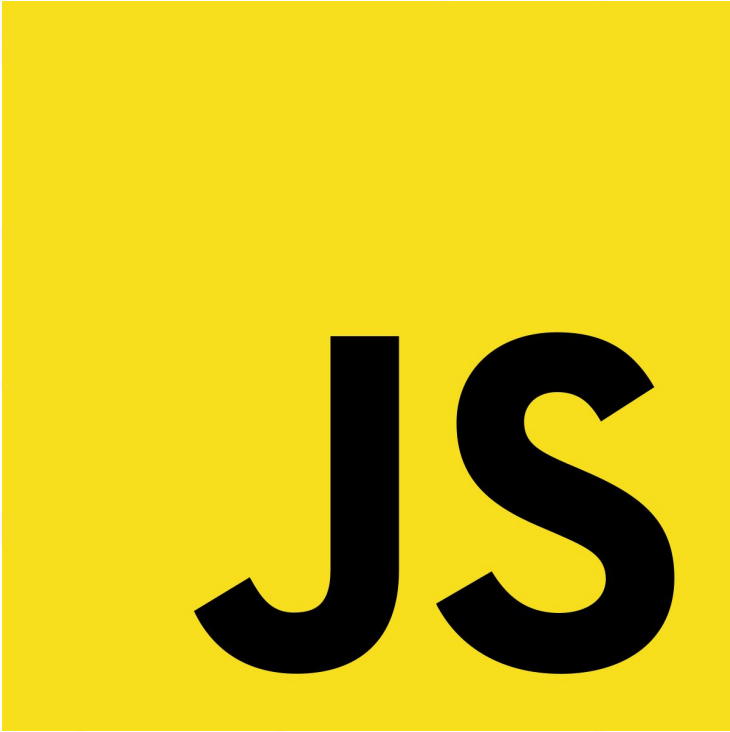
JavaScript es uno de los lenguajes de programación más populares en todo el globo terráqueo, es el lenguaje de programación que tiene mayor dominio de la web.



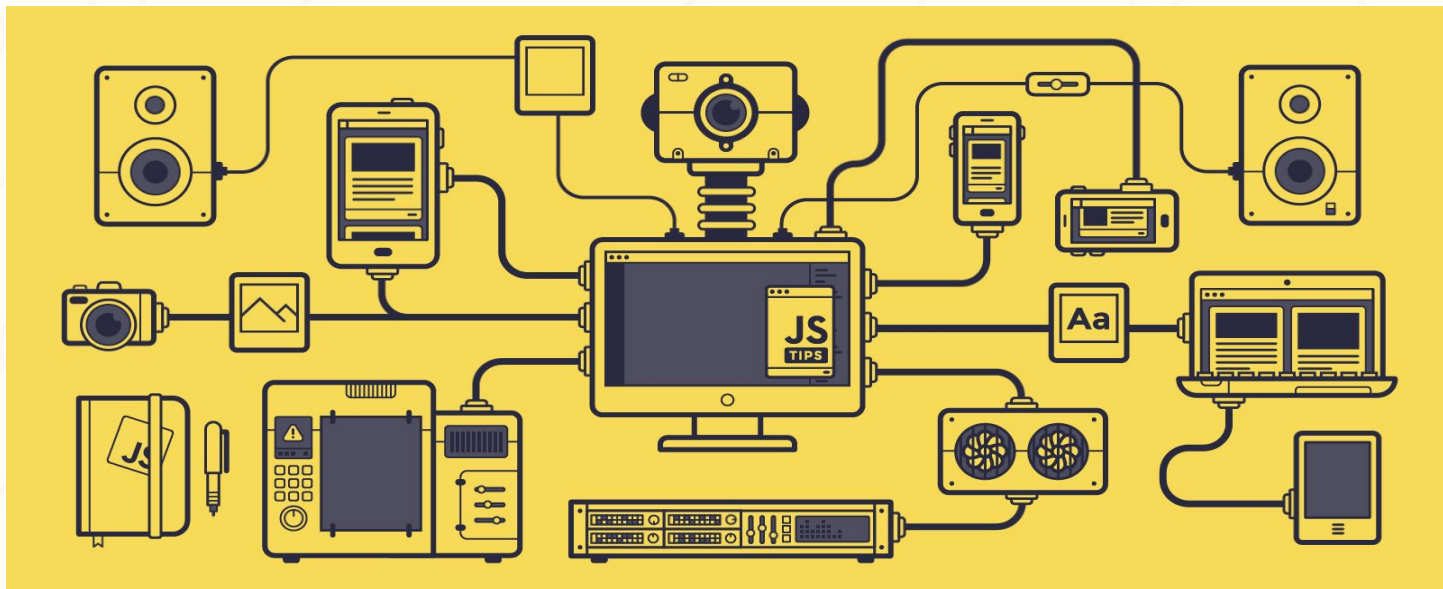
JavaScript

Hablemos de JavaScript

Es tan popular, que un intérprete tan poderoso como NODEJS puede interpretarlo de manera nativa en cualquier navegador web.

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing the JavaScript logo.

Hablemos de JavaScript

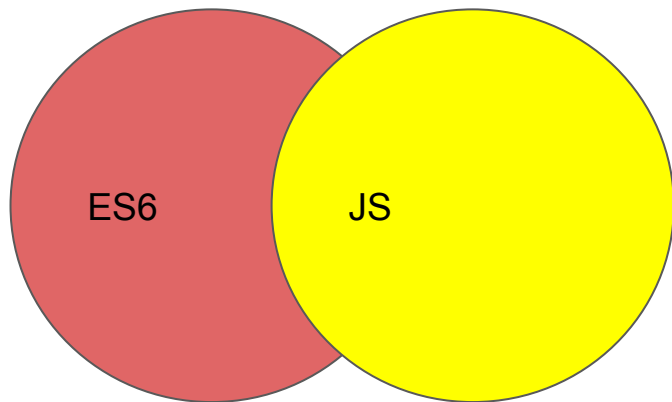


Ahora nuestros sitios web pasarán de ser estáticos a dinámicos, permitiendo una interacción entre el sitio web y los usuarios finales.

Un poco de historia ...

- A raíz de la competencia que se imponía con Microsoft y las mejoras que había presentado, el equipo de Netscape hizo alianzas con la empresa **Ecma International** (European Computer Manufacturer Association) una organización sin fines de lucro que se encarga de regular el funcionamiento de muchos estándares de la industria mundial, no solo en Europa si no en otros continentes.
 - Desde entonces en el año 1997 es diseñado el estándar para JavaScript DOM (Document Object Model) a fin de evitar la incompatibilidad entre navegadores, siendo la primera versión de esta ECMA-262 bajo la ISO/IEC 16262 saliendo oficialmente el 1 de junio de este año (1997), con el transcurso del tiempo las versiones han evolucionado y los estándares de JavaScript se rigen por ECMAScript a la actualidad.





Cosas que faltan
por implementar.

Cosas que los engines y
navegadores agregan.

Los Engines **son herramientas que los programadores han creado para hacer del desarrollo de videojuegos algo más ágil y sencillo**, sin quitarle el alto nivel de especialización necesario.

¿Qué es ECMAScript?

ECMAScript es el estándar y no el lenguaje.

Puede llegar a tomar mucho tiempo su implementación en un navegador.

¿Qué es ECMAScript?

ECMAScript específicamente es el estándar que a partir del año 2015 a la actualidad se encarga de regir como debe ser interpretado y funcionar el lenguaje JavaScript, siendo este (JS – JavaScript) interpretado y procesado por multitud de plataformas

A large yellow rectangular area containing the text "ES6" in a bold, dark grey sans-serif font. The text is centered within the yellow area.

ES6



¿Qué es un lenguaje de programación?

Es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes.

Que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

VARIABLES

(var, let, const)

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Variables

Una variable es un **símbolo que permite identificar a un elemento.**

X

=

1

VARIABLE

VALOR

Variables

En programación debemos asignar la palabra **var a nuestra variable para indicar que en efecto es una variable.**

var X = 1

VARIABLE

VALOR

Variables

Normalmente se utilizan para guardar/almacenar información para su uso posterior.



```
/* Guardo el string Pedro en la variable nombre */  
var nombre = 'Pedro';
```

Variables

En programación debemos asignar la palabra **var a nuestra variable para indicar que en efecto es una variable.**

```
> var x = 1  
var y = 2  
var texto = "Aquí va un texto"  
var nombre = "LUJAN"
```


Tips: Nombrando Variables

Es importante dar un nombre descriptivo a nuestra variable. Por ejemplo, es mejor: **var temperatura;** que **var t;**

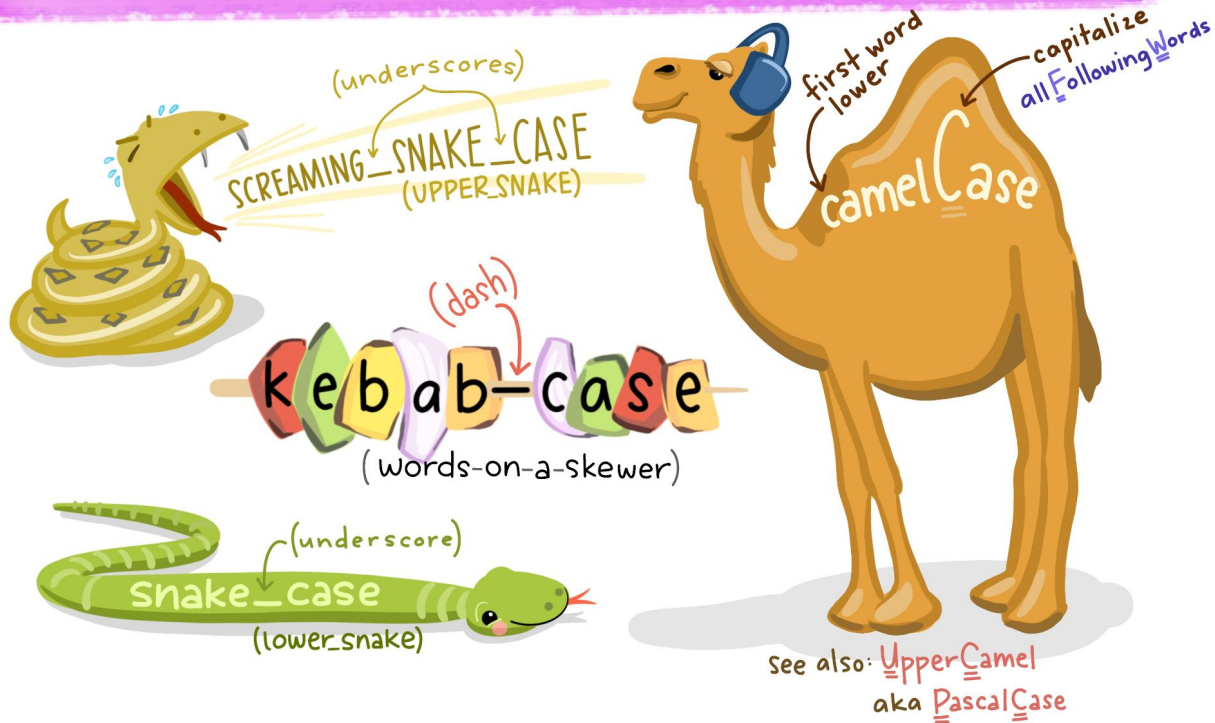
En JavaScript escribimos los nombres de variables usando **camelCase.**

Es decir, primera letra minúscula, y siguientes palabras comienzan con mayúscula, sin usar espacios.

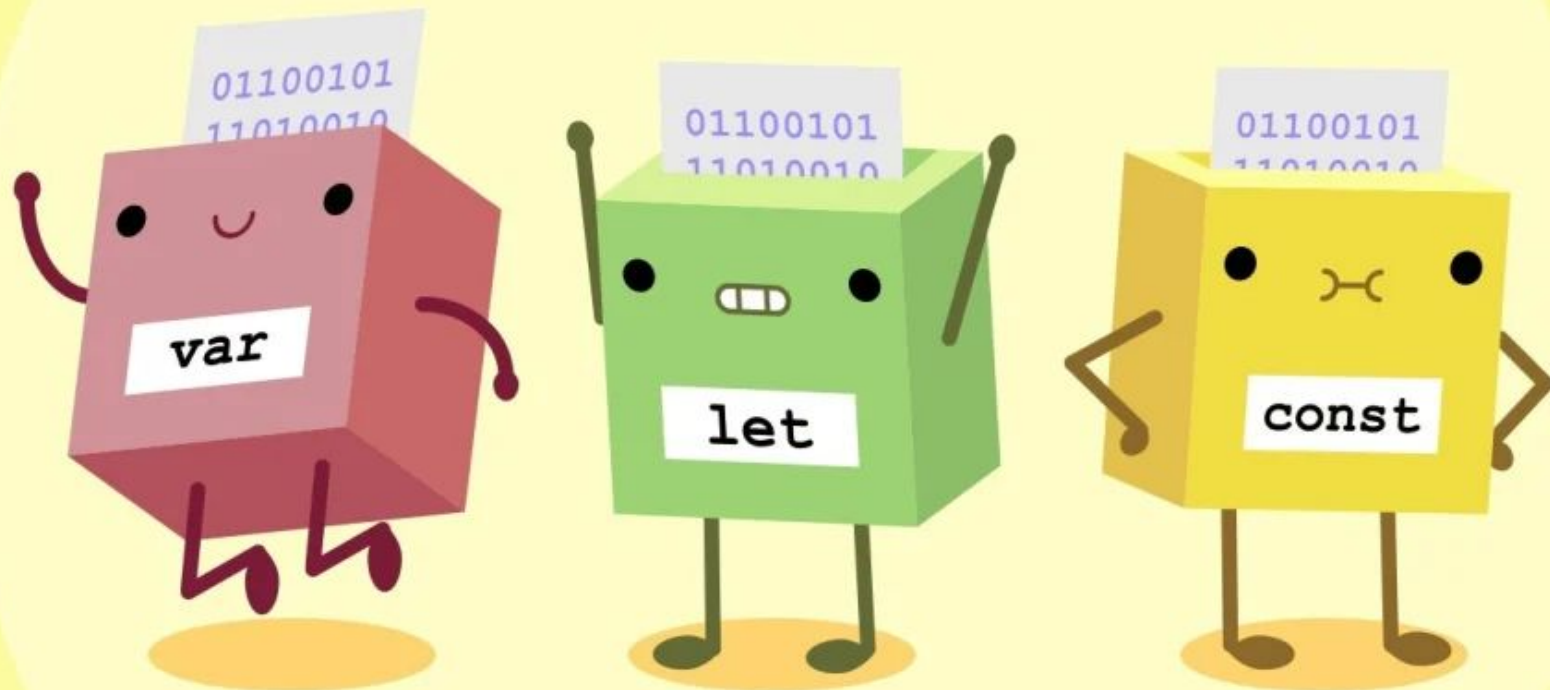


Dato Cultural: ¿Qué otras formas existen?

in that case...



A QUÉ BLOQUES DE CÓDIGO VA A PODER ACCEDER UNA VARIABLE



var, let, const

SON LAS MANERAS DE PODER **INICIAR UNA VARIABLE**

¿Qué es una variable?

Es una pequeña porción de memoria donde guardamos cierta información.

```
var ataque = 100;  
let pokemonType= 'electrico';  
const pokemonName = 'pikachu';
```

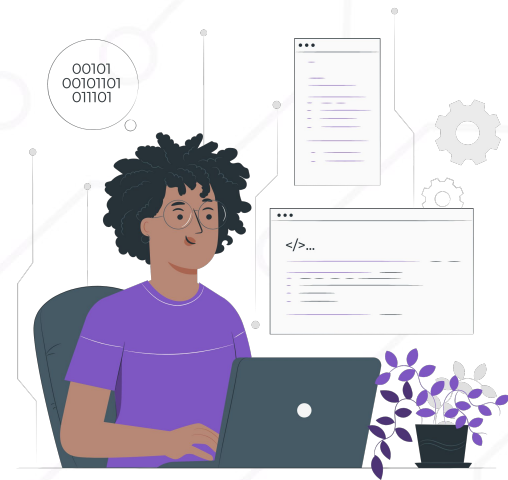


VAR EVERYWHERE!

```
> var age = 31  
  var name = 'JESSICA AGUILAR'  
  var profession = 'SOFTWARE ENGINEER'  
  var isAlive = true  
  var description = 'TECH LEADER AS A FRONTEND  
  DEVELOPER'
```

SCOPE EN LAS VARIABLES

Antes de empezar a analizar las nuevas maneras de declarar variables **(let y const)** es necesario comprender el ámbito de las mismas en JavaScript.



SCOPE EN LAS VARIABLES

Scope

```
let year = '2020';
```

Global Scope

```
function theYear() {  
  let text = "The year is"  
  return text + " " + year;  
}
```

Function Scope

```
if(10 < 20) {  
  let greeting = "hi";  
  return greeting  
}
```

Block Scope

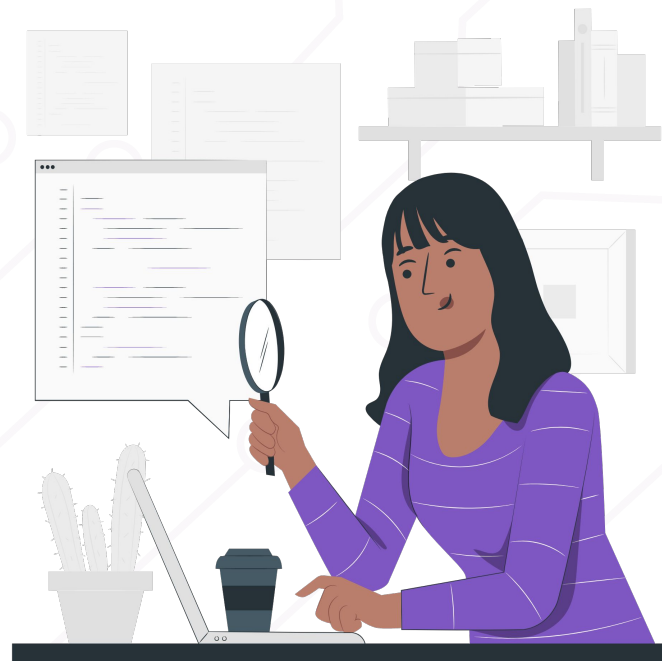
El alcance en JavaScript se refiere al contexto actual del código, que determina la accesibilidad de las variables a JavaScript.

Los dos tipos de alcance son locales y globales: Las variables globales son aquellas declaradas fuera de un bloque. Las variables locales son aquellas declaradas dentro de un bloque.

¿Qué sucede si declaramos una variable sin var, let o const?

Si declaramos una variable dentro o fuera de un bloque de código sin usar ninguna palabra reservada esta automáticamente se transforma en una variable global y su tipo de variable será por defecto var

```
> lo_queSea = "el texto que sea"  
< 'el texto que sea'  
  
> lo_queSea  
< 'el texto que sea'
```



¿Qué es let? 🤔

let y const existen desde ECMAS6 y surgieron por el año 2015/16

let es una nueva forma de declarar variables, permite al programador proteger el valor de una variable **dentro del bloque donde se la asigna.**



¿Qué es const?

const permite declarar constantes, ósea, espacios en memoria donde sus valores no cambian en el tiempo.

```
> const PI = Math.PI  
console.log(PI)  
3.141592653589793
```

Por convenciones se acostumbra declarar las constantes con mayúsculas

Una **constante** no permite que su valor se pueda alterar una vez declarada, el siguiente ejemplo devolvería un error:

```
> const fruta = "Manzana"  
let fruta = "Manzana"  
console.log(fruta)
```

✖ Uncaught SyntaxError: [VM3511:2](#)
Identifier 'fruta' has already
been declared

Conclusiones

- Las variables locales deben ser declaradas con las palabras reservadas **var**, **let** o **const**, de lo contrario el intérprete de JavaScript las convierte automáticamente en un variable global.
- **let** encapsula una variable dentro de un bloque, porque no podrá ser usada fuera del mismo.
- **const** evita reescribir el valor de una variable, cosa que **var** no hace.
- **var** y **let** comparten el mismo scope.
- Usa **let** o **const** de forma correcta para tener un mejor código y no ocupar mayor espacio de memoria, **var** es una forma anticuada de declarar variables pero aún se siguen utilizando.

Tipos de Datos

DEV.F
DESARROLLAMOS(PERSONAS);

dev

“Strings”

El tipo String de JavaScript se utiliza para representar datos textuales.

Normalmente se representan encerrando el valor textual entre comillas simples o dobles.



```
"Soy un String";  
'También soy un String';
```

Numbers

Number es un objeto primitivo envolvente que permite representar y manipular valores numéricos como 37 o -9.25. El constructor Number contiene constantes y métodos para trabajar con números. Valores de otro tipo pueden ser convertidos a números usando la función Number()..

```
123      // one-hundred twenty-three  
123.0    // same  
123 === 123.0  // true
```

También pueden contener valores **NaN**
("Not a Number" o No es un número).

Booleans

En términos de informática, un **boolean** es un dato lógico que solo puede tener los valores **true (verdadero)** o **false (falso)**.



```
/* Tipo Boolean */  
true; // Verdadero  
false; // Falso
```

Juegan un papel fundamental en la lógica de programación, ya que podemos establecer acciones dependiendo si se cumple o no cierta condición.

Null

El tipo Null tiene el valor: **nulo**.

```
var miVariable = null  
  
console.log(miVariable) // null
```

Un valor **nulo** significa que es un valor desconocido, indefinido o no inicializado

Undefined

Una variable a la cual no se le haya asignado valor tiene el valor **indefinido**.

```
var saludo;  
console.log(typeof saludo); // "undefined"
```

Symbol

Symbol es un tipo de datos cuyos valores son **únicos** e **inmutables**. Dichos valores pueden ser utilizados como identificadores (claves) de las propiedades de los objetos. Cada valor del tipo **Symbol** tiene asociado un valor del tipo String o Undefined que sirve únicamente como descripción del símbolo.

Object

Un Objeto es un valor en memoria al que podemos acceder por un identificador.

```
var persona = {  
  nombre: "Marcel",  
  edad: "38",  
  ciudad: "Alaior",  
};
```

En JavaScript los objetos pueden ser vistos como una colección de propiedades.

Operadores Matemáticos

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Operadores Matemáticos

Suma (+)

Resta (-)

Multiplicación (*)

División (/)

Residuo de una división (%)

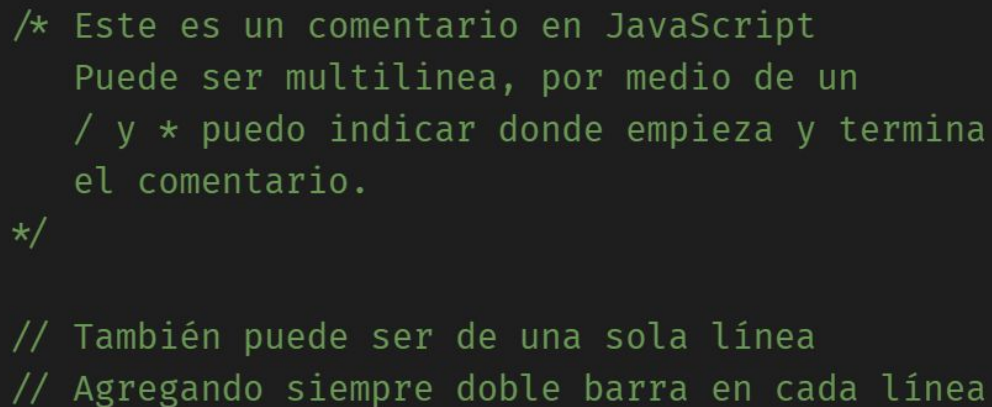
Exponencial (**)

Comentarios

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Formas de Comentar Código en JavaScript



```
/* Este es un comentario en JavaScript
   Puede ser multilínea, por medio de un
   / y * puedo indicar donde empieza y termina
   el comentario.
*/

// También puede ser de una sola línea
// Agregando siempre doble barra en cada línea
```