**2UD: DESARROLLADOR WEB FRONT END**

**UNIDAD 1**

**Sesión 1: Conceptos básicos de JavaScript**

**1. Introducción a JavaScript**

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS).

JavaScript se diseñó con una sintaxis similar a C++ y Java,​ aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor.

Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX.

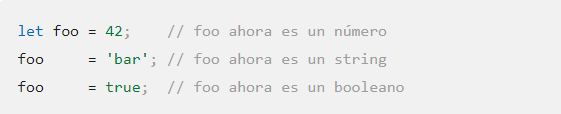
A partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor Node.js es uno de los notables ejemplos de JavaScript en el lado del servidor, siendo usado en proyectos importantes.

Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje.

* **HTML** es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página.
* **CSS** es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.
* **JavaScript** es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.
  1. ***¿Para qué se utiliza JavaScript?***
* Actualización dinámica de contenidos.
* Desarrollo de aplicaciones web y móviles.
* Desarrollo de aplicaciones del lado del servidor.
* Desarrollo de videojuegos

**2. Tipos de datos y estructuras en JavaScript**

Todos los lenguajes de programación tienen estructuras de datos integradas, pero estas a menudo difieren de un lenguaje a otro.

Las variables en JavaScript no están asociadas directamente con ningún tipo de valor en particular, y a cualquier variable se le puede asignar (y reasignar) valores de todos los tipos:

El último estándar ECMAScript define nueve tipos.

Seis tipos de datos primitivos, controlados por el operador typeof:

* Undefined: typeof instance === "undefined“. Una variable a la que no se le ha asignado un valor tiene el valor undefined.
* Boolean: typeof instance === "boolean“. Representa una entidad lógica y puede tener dos valores: true y false.
* Number: typeof instance === "number“. Es un valor en formato binario de 64 bits de doble precisión IEEE 754.
* String: typeof instance === "string“. Se utiliza para representar datos textuales. Es un conjunto de "elementos" de valores enteros sin signo de 16 bits.
* BigInt: typeof instance === "bigint“. Almacena y opera de forma segura números enteros grandes incluso más allá del límite seguro de enteros para Numbers.
* Symbol: typeof instance === "symbol“. Un símbolo es un valor primitivo único e inmutable y se puede utilizar como clave de una propiedad de objeto.
* Null: typeof instance === "object". Tipo primitivo especial que tiene un uso adicional para su valor: si el objeto no se hereda, se muestra null;
* Object: typeof instance === "object". Tipo estructural especial que no es de datos pero para cualquier instancia de objeto construido que también se utiliza como estructuras de datos: new Object, new Array, new Map, new Set, new WeakMap, new WeakSet, new Date y casi todo lo hecho con la palabra clave new.
* Function: typeof instance === "function". Una estructura sin datos, aunque también responde al operador typeof. Esta simplemente es una forma abreviada para funciones, aunque cada constructor de funciones se deriva del constructor Object.

**3. Declaración de variables en JavaScript.**

JavaScript tiene tres tipos de declaraciones de variables.

* var: Declara una variable, opcionalmente la inicia a un valor.
* let: Declara una variable local con ámbito de bloque, opcionalmente la inicia a un valor.
* const: Declara un nombre de constante de solo lectura y ámbito de bloque.

Utiliza variables como nombres simbólicos para valores en tu aplicación. Los nombres de las variables, llamados identificadores, se ajustan a ciertas reglas.

Un identificador de JavaScript debe comenzar con una letra, un guión bajo (\_) o un signo de dólar ($). Los siguientes caracteres también pueden ser dígitos (0-9).

Dado que JavaScript distingue entre mayúsculas y minúsculas, las letras incluyen los caracteres "A" a "Z" (mayúsculas), así como "a" a "z" (minúsculas).

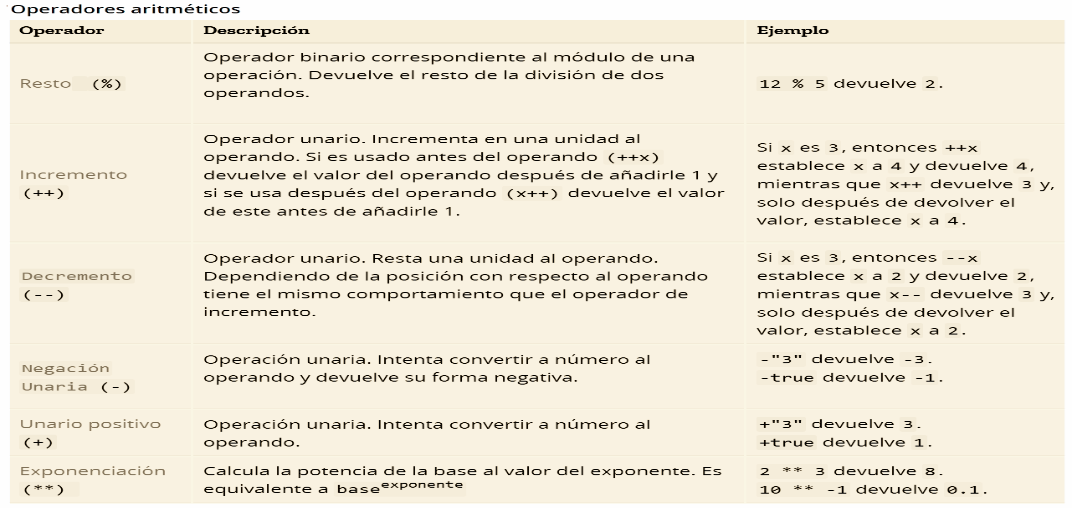
Puedes declarar una variable de dos formas:

* Con la palabra clave var. Por ejemplo, var x = 42. Esta sintaxis se puede utilizar para declarar variables locales y globales, dependiendo del contexto de ejecución.
* Con la palabra clave const o let. Por ejemplo, let y = 13. Esta sintaxis se puede utilizar para declarar una variable local con ámbito de bloque.

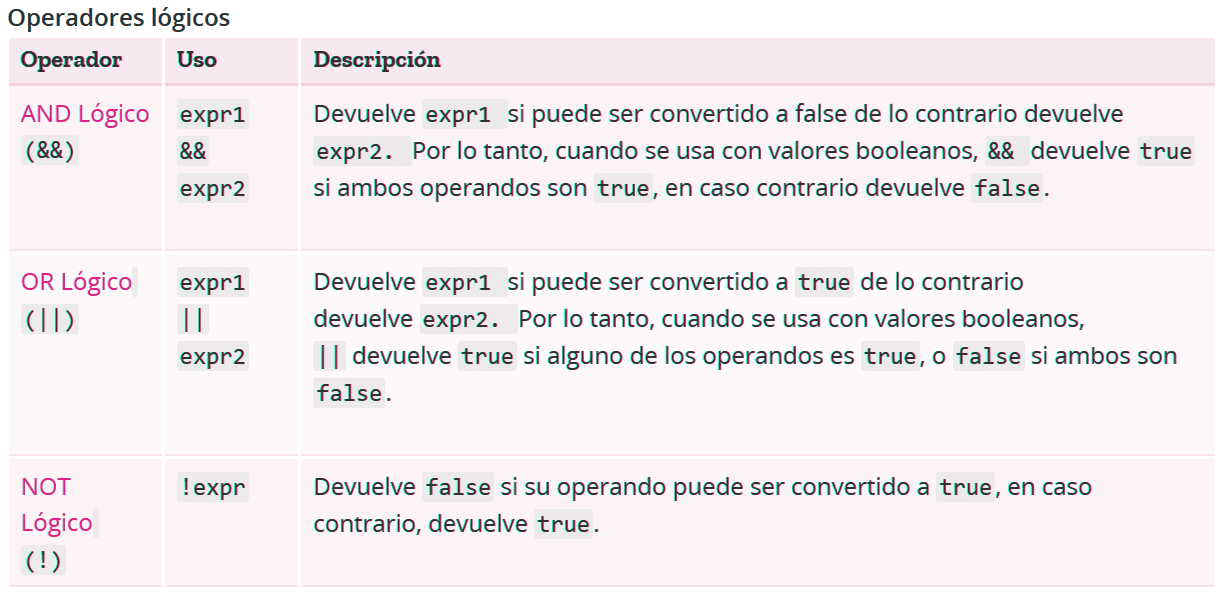
**4. Operadores en JavaScript.**

JavaScript tiene los siguientes tipos de operadores. Esta sección describe los operadores y contiene información sobre la precedencia de los mismos.

* Aritméticos
* Lógicos
* Asignación
* Comparación
* ***Operadores Aritméticos***



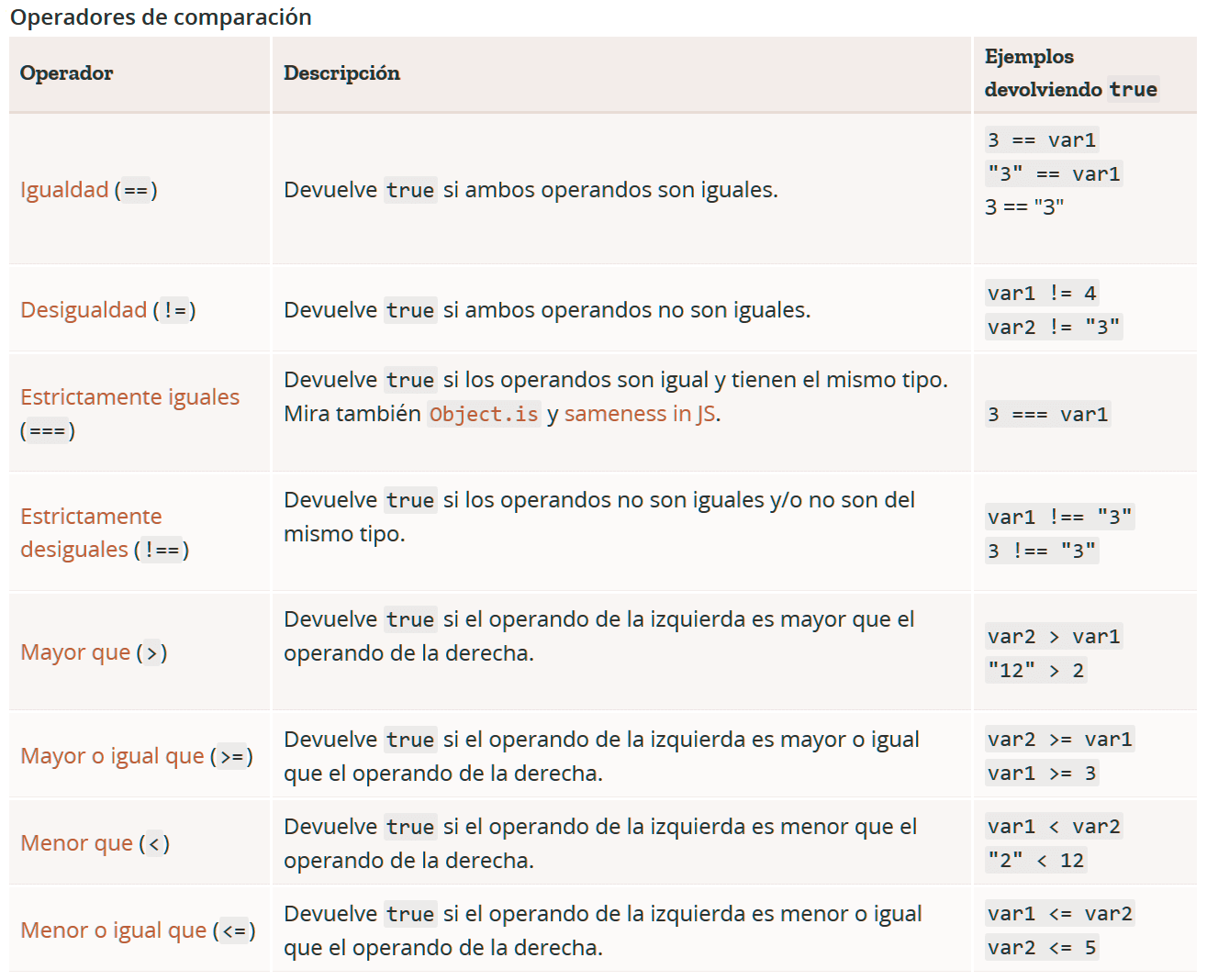
* ***Operadores Lógicos***



* ***Operadores Asignación***



* ***Operadores Comparación***



**5. Funciones predefinidas en JavaScript.**

* eval(string): Esta función recibe una cadena de caracteres y la ejecuta como si fuera una sentencia de Javascript.
* parseInt(cadena, base): Recibe una cadena y una base. Devuelve un valor numérico resultante de convertir la cadena en un número en la base indicada.
* parseFloat(cadena): Convierte la cadena en un número y lo devuelve.
* isNaN (número): Devuelve un booleano dependiendo de lo que recibe por parámetro. Si no es un número devuelve un true, si es un número devuelve false.
* escape(string): Crea una nueva cadena de caracteres en los que ciertos caracteres han sido sustituidos por una secuencia hexadecimal de escape.