

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- Qué es JavaScript.
- Variables, condicionales y funciones.
- Consola de JavaScript.

JavaScript, abreviado comúnmente como JS es un lenguaje de programación ligero, interpretado o compilado justo a tiempo, es decir, se compila durante el tiempo de ejecución de un programa (compilar es la traducción de un código fuente a código de máquina). Fue desarrollado en 1995 por **Netscape** y, al igual que otros lenguajes, permite definir y almacenar variables, realizar operaciones y trabajar con paradigmas de programación como la Programación Orientada a Objetos (**POO**).

Este lenguaje es el más utilizado en **FrontEnd** y conocido como un lenguaje para páginas web, sin embargo, puede ser utilizado en otros entornos de trabajo distintos al navegador, como **Node.js** para desarrollo de **BackEnd**.

Como el resto de las tecnologías en programación, **JavaScript** se rige por un estándar, el cual es **ECMAScript**.

QUÉ ES ECMAScript

ECMAScript es el lenguaje de *scripting* (escritura de comandos) que forma la base del lenguaje **JavaScript**. Se encuentra estandarizado por la organización [ECMA Internacional](#).

Desde el año 2012 todos los navegadores modernos soportan **ECMAScript 5.1**. En 2015 **ECMA Internacional** publicó oficialmente **ECMAScript 2015**, conocido como **ECMAScript 6** o **ES6**.

Los estándares de **ECMAScript** se lanzan anualmente, siendo la última versión preliminar la de [ECMAScript 2020](#).

IMPORTANCIA DE APRENDER JAVASCRIPT

JavaScript es el lenguaje de programación más popular del mundo, utilizado en la famosa triada para desarrollo **FrontEnd HTML, CSS y JS**.

Estas tres tecnologías se utilizan en conjunto por los desarrolladores web para:

- **HTML**: definir el contenido de las páginas web.
- **CSS**: especificar el diseño de las páginas web.
- **JavaScript**: programar el comportamiento de las páginas web.



Ilustración 1 Triada Front End

COMO INSTALAR JAVASCRIPT

JavaScript no requiere ser instalado o descargado, ya se encuentra en nuestros navegadores, es de uso gratuito y solo debemos indicar la extensión del archivo y utilizar la sintaxis correcta para poder comenzar a escribir en este lenguaje de programación.

QUE PODEMOS HACER CON JAVASCRIPT

Como previamente decíamos, **JavaScript** es un lenguaje de programación que se utiliza comúnmente en conjunto con **HTML** y **CSS** que hasta este punto ya hemos conocido.

Utilizando **JavaScript** podemos modificar el contenido **HTML**, modificar los valores de los atributos de **HTML**, cambiar estilos de **CSS**, ocultar o mostrar elementos **HTML**, consumir **API's**, entre otras cosas.

CARACTERÍSTICAS GENERALES DE JAVASCRIPT

- Es muy liviano, por lo que se ejecuta correctamente sin requerir de grandes características de hardware.
- Es multiplataforma, ya que depende del navegador más que del sistema operativo. Una aplicación hecha en **JavaScript** se ejecutará de igual manera en **Windows, Linux o Mac**.
- Es multiparadigma, es decir, cumple con múltiples formas de programar: es orientado a objetos, orientado a eventos, de carácter imperativo, entre otros.
- Es interpretado, por lo que no es necesario compilar nuestro código para ejecutarlo. Basta solo con actualizar el archivo y veremos inmediatamente el resultado en el navegador.

VENTAJAS

Es el lenguaje de programación del lado del cliente más popular y utilizado por los desarrolladores de **FrontEnd** de todo el mundo, ya que es un lenguaje sólido, liviano, simple y muy funcional.

JavaScript permite manipular el **DOM** (*Document Object Model*), obtener datos de sus elementos, modificarlos, eliminarlos, animarlos y guardarlos, además de permitir interactuar con el servidor, enviando o pidiendo datos.

Existen numerosos **Frameworks**, o librerías de código que nos permiten trabajar de mejor manera, tales como:

- [AngularJS](#)
- [React](#)

- [Node.js](#)
- [Vue.js](#)

COMENZAR A PROGRAMAR

A diferencia de **HTML** y **CSS**, al utilizar **JavaScript** ya estaremos programando y programar es mucho más que escribir código. Cuando comenzamos a programar, escribimos una serie de instrucciones que una computadora deberá seguir para conseguir alcanzar la solución a una problemática planteada.

Es por eso por lo que hablamos de algoritmo al momento de hablar de programación.

QUE ES UN ALGORITMO

Según la **RAE** un algoritmo es *“Un conjunto ordenado y finito de operaciones que permite hallar la solución a un problema”*.

En otras palabras, es detallar los pasos a seguir de manera lógica y ordenada para solucionar un problema. Destacamos que un algoritmo siempre será finito, es decir, tendrá un paso de inicio y, tras seguir una secuencia lógica de distintos pasos, tendrá siempre un paso final.

Los algoritmos no son algo exclusivo de la programación y un ejemplo clásico son las instrucciones que aparecen en un manual para armar algo o una receta de cocina que indica los pasos para lograr preparar alguna deliciosa comida.



Ilustración 2 Algoritmo en lenguaje natural

Existen 4 formas de representar un algoritmo, siendo la primera forma la que utilizamos comúnmente en nuestra vida conocida como "lenguaje natural", con la cual damos las instrucciones para conseguir algo, como, según el ejemplo previo, indicar los pasos para usar un extintor.

Existen también los **diagramas de flujo**, que son una forma visual o esquemática de representar la secuencia de pasos de un algoritmo. Esto permite que sea entendible para personas ajenas a la programación y cuenta con simbologías bien definidas para facilitar su comprensión.

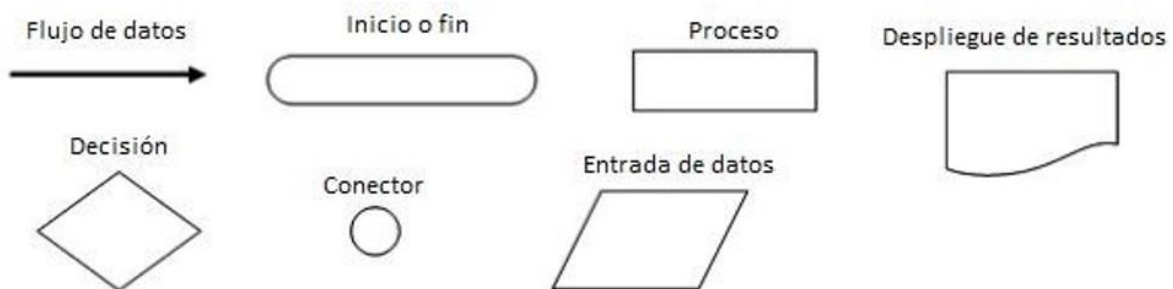


Ilustración 3 Algoritmo en diagrama de flujo

Otra forma de representar un algoritmo es a través de un **pseudocódigo** que nos permite detallar los pasos de nuestro algoritmo, siendo un punto intermedio entre el lenguaje natural y el lenguaje de programación.

```
Proceso Suma

    Definir A,B,C como Reales;

    Escribir "Ingrese el primer numero:";
    Leer A;

    Escribir "Ingrese el segundo numero:";
    Leer B;

    C <- A+B;

    Escribir "El resultado es: ",C;

FinProceso
```

Ilustración 4 Algoritmo en pseudocódigo

Una buena técnica para poder comprender mejor la programación es practicar con un pseudocódigo, utilizando herramientas como PSeint.

Y finalmente, tenemos los lenguajes de programación con los que daremos instrucciones a la máquina, pudiendo ser este lenguaje JavaScript, Pascal, Java, PHP, Python, C#, Cobol, Go, entre muchos otros.



Ilustración 5 Lenguajes de programación

EL PENSAMIENTO LÓGICO

El pensamiento lógico es indispensable a la hora de programar, este nos permite reconocer y analizar los problemas y posibles soluciones. Nos permitirá anticiparnos a los diferentes comportamientos e interacciones que se puedan presentar y programar un código que tenga en cuenta todas estas situaciones. Mientras más desarrollado se encuentre el pensamiento lógico, más a menudo se logrará que el programa haga lo que esperamos que haga y más rápido se obtendrán soluciones a los problemas diarios que se presenten.

RETOMANDO JAVASCRIPT

Como decíamos previamente, **JavaScript** es un lenguaje multiplataforma y orientado a objetos que se utiliza para dar interactividad a las páginas web. Contiene una biblioteca estándar de objetos, como **Array**, **Date** y **Math** y un conjunto básico de elementos del lenguaje como operadores, estructuras de control y declaraciones.

JAVASCRIPT Y JAVA

Java y **JavaScript** muchas veces llevan a la confusión de ser el mismo lenguaje por lo parecido de su nombre (siendo, en reiteradas ocasiones, confundido pensando que **Java** es la abreviación de **JavaScript**). Tienen algunas similitudes y diferencias, como el fuerte tipado de **Java** en comparación con la flexibilidad de **JavaScript**.

JavaScript	Java
Orientado a objetos. No hay distinción entre tipos de objetos. La herencia se realiza a través del mecanismo de prototipo, y las propiedades y métodos se pueden agregar a cualquier objeto de forma dinámica.	Basado en clases. Los objetos se dividen en clases e instancias con toda la herencia a través de la jerarquía de clases. Las clases y las instancias no pueden tener propiedades o métodos agregados dinámicamente.
Los tipos de datos de las variables no se declaran (tipado dinámico, tipado flexible).	Los tipos de datos de las variables se deben declarar (tipado estático, fuertemente tipado).
No se puede escribir automáticamente en el disco duro.	Puede escribir automáticamente en el disco duro.

Ilustración 6 Tabla de comparación de Java y JavaScript

SINTAXIS DE JAVASCRIPT

A pesar de no tener ninguna relación con **Java**, su sintaxis es muy parecida. También es muy similar a **C**. La sintaxis de **JavaScript** sigue las siguientes normas básicas:

- Espacios en blanco y nuevas líneas no se toman en cuenta. El intérprete de **JavaScript** ignorará cualquier espacio en blanco extra, lo que permite un mejor ordenamiento del código, pudiendo usarse tabulación y saltos de línea.
- Es *case sensitive* o sensible a mayúsculas y minúsculas, por lo que dos variables con diferencias en las mayúsculas (como letra y Letra) serán consideradas como distintas.
- No es necesario indicar el tipo de variable que se está definiendo. Además, una misma variable puede almacenar distintos tipos de datos durante la ejecución de un script.
- Se recomienda terminar una sentencia con punto y coma (;), aunque no es obligatorio.
- Se pueden incluir comentarios utilizando //.

TIPOS DE DATOS

Existen 5 tipos de datos en **JavaScript**, conocidos como datos primitivos. Un dato primitivo es un tipo de dato que representa un solo valor que no es modificable.

- **Boolean**: representa una entidad lógica y puede tomar dos valores: true/false.
- **Number**: representa un valor numérico, entero o racional.
- **String**: representa una cadena de texto o caracteres.
- **Undefined**: representa una variable a la que no se le ha asignado un valor, por lo que su valor es undefined.
- **Null**: representa una variable a la que se asigna un valor nulo o vacío.

Si no estamos seguros del tipo de dato al que pertenece una variable, podemos utilizar el operador ***typeof***, que devolverá una cadena de texto con el tipo de dato.

VARIABLES Y CONSTANTES

Una variable, se utiliza en programación para almacenar datos. En **JavaScript**, una variable se declara con la palabra **`var`**, palabra que se encarga de decirle al intérprete que lo que viene a su derecha es

una variable. Luego de la palabra **var**, se escribe el nombre de la variable y, después del nombre y un signo igual (=), se escribe el dato que se quiere almacenar.

```
1 var n = 5;
```

En adición a **var**, existe también la instrucción **let**, la cual declara una variable de alcance local con ámbito de bloque, es decir, que su alcance solo llega al bloque en el que está declarada.

```
1 if (a < b) {  
2     let c = a * b;  
3 }
```

Como indica su nombre, el valor de una variable puede variar. En cambio, el valor de una constante es inmutable. Una constante se define en **JavaScript** utilizando la palabra **const**, seguida del nombre de la constante, un signo igual y su valor.

```
1 const g = 9.8;
```

PALABRAS RESERVADAS

Las declaraciones de **JavaScript** comienzan a menudo con una palabra clave para identificar la acción que se debe realizar. Acá veremos una lista con estas “palabras claves”, más conocidas como palabras reservadas para declaraciones.

Palabra clave	Descripción
var	Declara una variable
let	Declara una variable de bloque
const	Declara una constante de bloque
if	Marca un bloque de declaraciones que se ejecutarán con una condición.
switch	Marca un bloque de sentencias que se ejecutarán en diferentes casos.
for	Marca un bloque de declaraciones que se ejecutarán en un bucle.
function	Declara una función
return	Sale de una función
try	Implementa el manejo de errores en un bloque de declaraciones.

Ilustración 7 Palabras reservadas de JavaScript

Para conocer más sobre **JavaScript** podemos ingresar a la [introducción a JavaScript](#) de **Mozilla**.