

00 | INTRODUCCIÓN

DEBATAMOS: ¿QUÉ ES LA PROGRAMACIÓN?



Definición: Programación

Es enseñarle o mostrarle a la computadora una forma de resolver una tarea o problema que tengamos diciéndole paso a paso lo que debe hacer. Una vez que se lo enseñemos entonces la computadora va a poder resolverlo con su gran capacidad de cómputo y cálculo todas las veces que lo necesitemos.

Para comunicarnos con la computadora y explicarle qué pasos debe seguir existen los famosos **lenguajes de programación**. Cada uno tiene distintas ventajas y desventajas teniendo que elegir cuál queremos usar.

Los más conocidos son: Java, Python, JavaScript, Ruby, C#, C++, C.
A lo largo de este curso van a aprender **JavaScript**

Definición: Algoritmo

Guía ordenada paso a paso de como resolver un problema que tengamos.

Se puede asociar a la idea de tener una receta en cocina.

Patatas fritas

Receta de patatas fritas caseras.

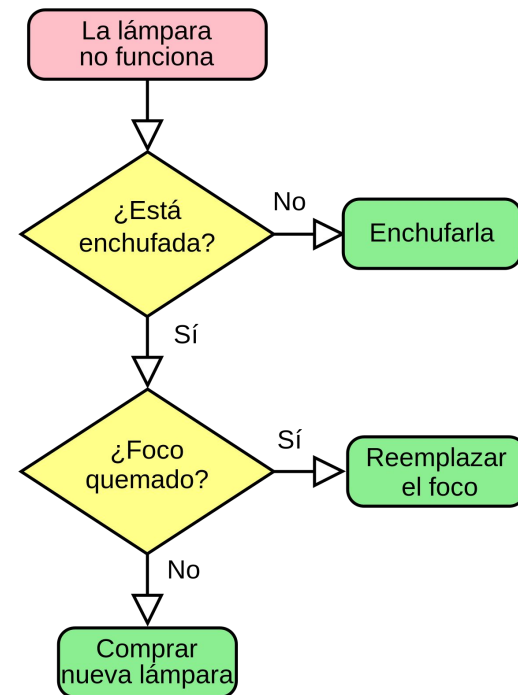


Ingredientes

- 3 ó 4 patatas (300g.)
- 4 dientes de ajo
- Aceite de oliva
- Sal

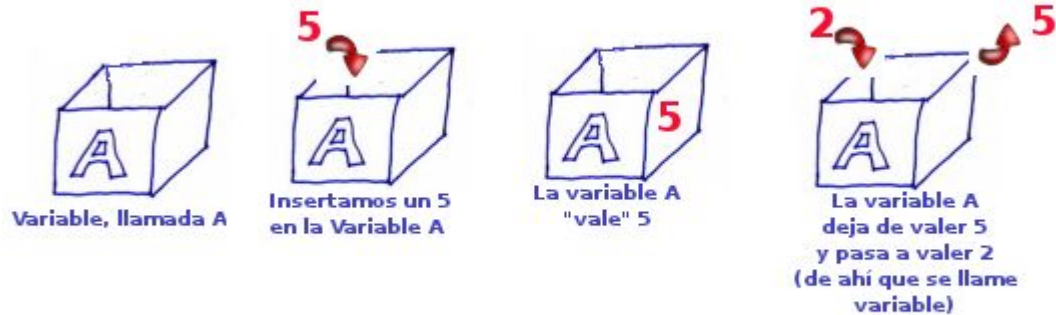
Elaboración (Pasos)

1. Calentar aceite en una sartén.
2. Añadir las patatas cortadas, la sal y los ajos.
3. Freír al gusto.
4. Servir en plato.



Definición: Variable

Es un elemento dentro del mundo de la programación que me permite guardar valores para usarlos más adelante en el algoritmo.



Definición: Función

Es un elemento el cual tiene una entrada y una salida, generándose en el medio una transformación.

$$f(x) = 3x + 2$$

En el colegio ustedes vieron que las funciones tenían como entrada números y como salida obtenían números distintos en un principio


En programación esto avanza más y las funciones pueden tener varios tipos distintos de entrada y varios tipos distintos de salida.

Por ejemplo una función en donde entre un número y salga como resultado un texto

01 | JAVASCRIPT

JavaScript - Definición

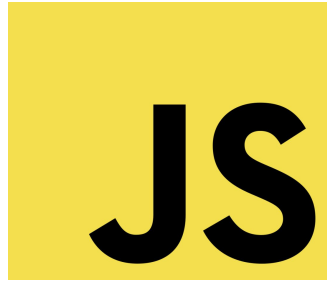
JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web.

A large, bold, black 'JS' logo is centered on a solid yellow rectangular background. The letters are thick and stylized, with the 'J' having a curved bottom and the 'S' being a simple, rounded shape.

JavaScript – 1 de las 3 tecnologías “core” de desarrollo



**Estructura
(Contenido)**



**Comportamiento
(Lenguaje de
programación)**



Diseño

JavaScript – Sustantivo, adjetivo, verbo



CONTENIDO

Sustantivo

`<p></p>`

Significa
párrafo



DISEÑO

Adjetivo

`P { color: red; }`

Significa “el párrafo
es rojo”



PROGRAMACIÓN

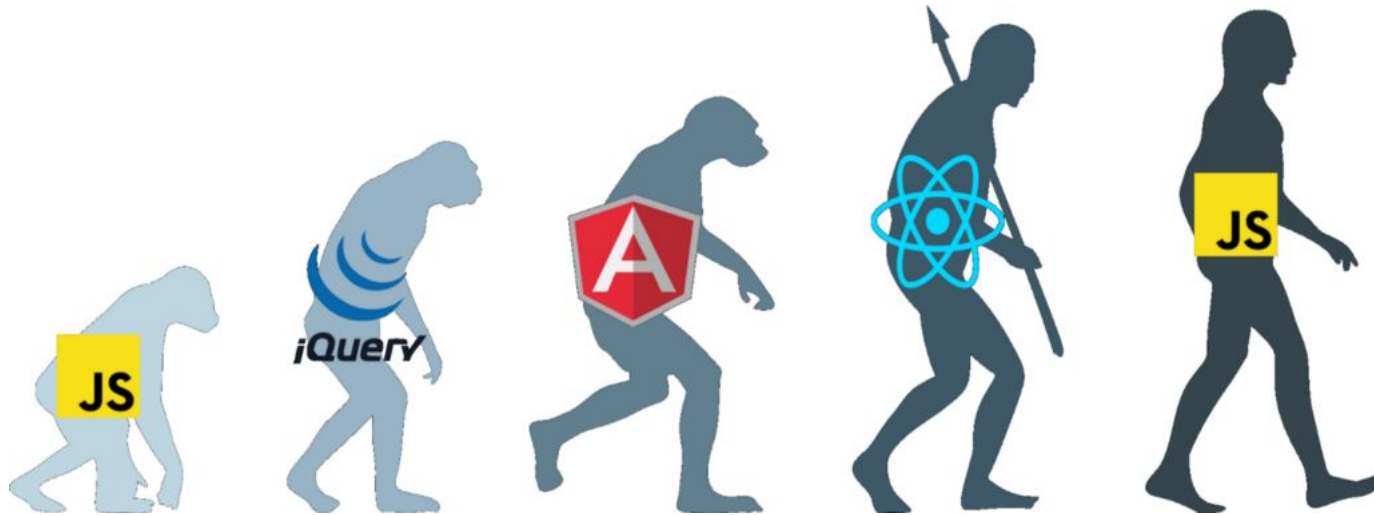
Verbo

`p.hide();`

Significa “oculta el
párrafo”

JavaScript - Aplicación

JavaScript se utiliza actualmente del lado del cliente (navegador), como del servidor. De momento, nos estaremos focalizando en lo que sucede del lado de nuestro navegador.



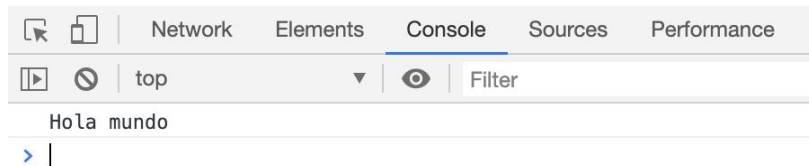
JS – Formas de insertar código

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Primer ejemplo de JS</title>
  </head>
  <body>
    <h1>Primer ejemplo en JS</h1>
    <script>
      console.log("Hola mundo");
    </script>
  </body>
</html>
```

Documento HTML

En el siguiente ejemplo, se muestra cómo insertar código de JS dentro del mismo documento HTML.

Luego, en la consola de desarrolladores de Chrome, en la solapa “Console”, vemos la salida del texto.



Consola de desarrolladores

JS – Formas de insertar código

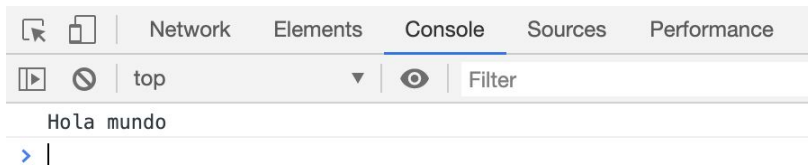
```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>HTML con hoja externa de JS</title>
  </head>
  <body>
    <h1>HTML con hoja externa de JS</h1>
    <script src="vinculado.js"></script>
  </body>
</html>
```

Documento HTML

En el siguiente ejemplo, se muestra cómo vincular una hoja de JS externa al documento HTML.

```
console.log("Hola mundo");
```

Hoja de JavaScript



Consola de desarrolladores

JS - Variables

Esta es la instrucción para crear una variable y asignarle un valor en JS.
Como toda instrucción termina en ; como limitador.



JS – Tipos de dato

Veamos a continuación los 5 tipos de datos primitivos que existen en JavaScript:

1. **Números**: Con punto flotante, decimales y enteros.
2. **String**: Cadena de caracteres.
3. **Booleano**: Tipo de dato lógico, cuyos posibles valores son “true” o “false”.
4. **Undefined**: Tipo de dato de una variable a la cual aún no se le asignó valor.
5. **Null**: Ausencia de valor.

Cada tipo de dato tiene ciertas operaciones asociadas, por ejemplo:

a los **números** los puedo sumar, restar, multiplicar, dividir,

a los **booleanos** los puedo operar/comparar usando los operadores/comparadores lógicos

JS – Tipado Dinámico

JavaScript cuenta con lo que se conoce como “dynamic typing”, el cual nos permite a nosotros los programadores declarar variables sin indicar el tipo de dato que almacenará.

Al ejecutarse, JS resolverá automáticamente qué tipo de dato debe asignarle a cada variable, dependiendo de su valor.

Algunas reglas que sigue javascript son...

```
var numero = 5; // Escribo un número entero
var otro_numero = 10.1; // Escribo un número con punto
var texto = "Hola, soy un texto"; // Escribo texto entre ""
var bool = true; // Escribo true ó false
```


JS – Reglas de nombre

No podemos declarar una variable/función que comience con

- Números
- Caracteres especiales
- Palabras reservadas de JS (como var ó function)

Debemos hacerlo con:

- Guion bajo
- Signo dólar
- Cualquier palabra que no rompa con las dos primeras reglas

```
> var 3test = 1;
```

✖ Uncaught SyntaxError: Invalid or unexpected token

```
> var @miVariable = "Juan";
```

✖ Uncaught SyntaxError: Invalid or unexpected token

JS – Coerción de tipos

Como dijimos antes cierto tipo de dato tiene ciertos operadores válidos asociados... entonces qué pasa si nosotros hacemos algo como esto?

```
var nombre = "Juan";  
var edad = 40;  
  
console.log(nombre + edad);
```

Bueno lo que pasa es que JavaScript convierte las variables a un tipo de dato en que se puedan operar. En este caso convierte el 40 (número) a "40" (string) y concatena el texto quedando como resultado "Juan40"

JS – Operadores aritméticos

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma dos números
-	Substracción	7 - 9	Resta dos números
*	Multiplicación	6 * 3	Multiplifica dos números
/	División	4 / 8	Divide dos números
%	Módulo: el resto después de la división	7 % 2	Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1
++	Incremento.	a++	Suma 1 al contenido de una variable.
--	Decremento.	a--	Resta 1 al contenido de una variable.
-	Invierte el signo de un operando.	-a	Invierte el signo de un operando.

JS – Operadores lógicos

Operador	Significado
Operador &&	Operador lógico and
Operador	Operador lógico or
Operador !	Operador lógico not

Tabla de "y, and, &&"		
valor1	valor2	valor1 && valor 2
true	true	true
true	false	false
false	true	false
false	false	false

Tabla de "o, or, "		
valor1	valor2	valor1 valor 2
true	true	true
true	false	true
false	true	true
false	false	false

Tabla de "no, not, !"	
valor1	!valor1
true	false
false	true

JS – Comparadores lógicos

Operador	Significado
<code>a === b</code>	Igual a
<code>a !== b</code>	Distinto
<code>a > b</code>	Mayor que
<code>a >= b</code>	Mayor o igual que
<code>a < b</code>	Menor
<code>a <= b</code>	Menor o igual que

JS – Precedencia de operadores

La precedencia de operadores determina el orden en el cual los operadores son evaluados. Los operadores con mayor precedencia son evaluados primero.

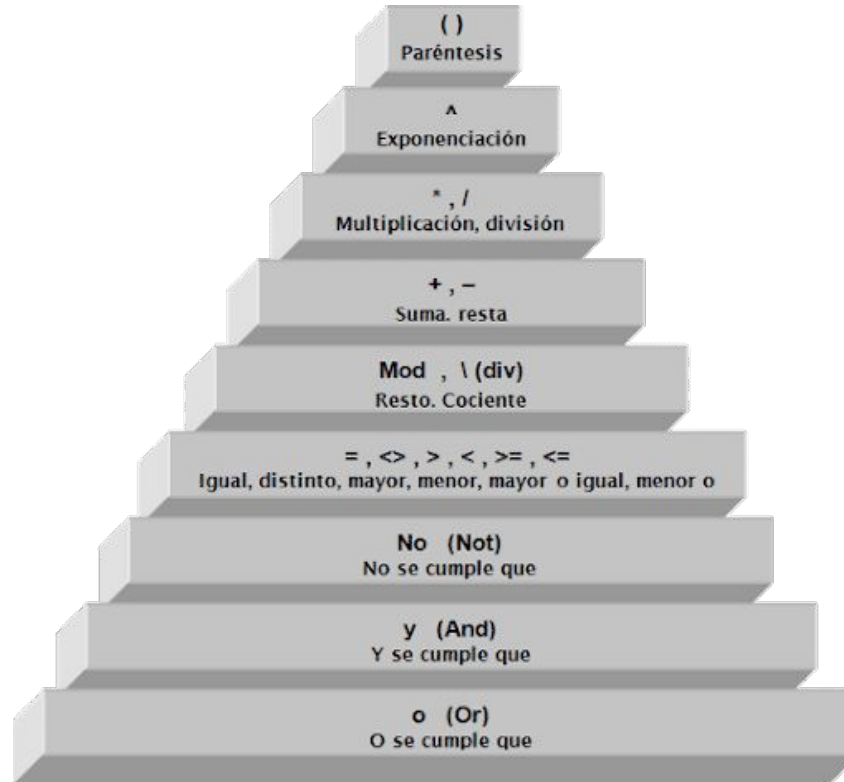
Analicemos el siguiente ejemplo:

Ejemplo:

```
1 | 3 + 4 * 5 // retorna 23
```

El operador de multiplicación ("*") tiene una precedencia mas alta que el operador de suma ("+") y por eso sera evaluado primero.

JS – Precedencia de operadores



DUDAS

