



JAVASCRIPT

BRASCO



JAVASCRIPT – JS

Es un lenguaje de programación **interpretado** e **imperativo**.



Orientado a objetos, **débilmente tipado** y **dinámico**, basado en **prototipos**.

Se utiliza principalmente del **lado del cliente**, pero también del **lado de servidor**.

LADO CLIENTE

Librerías y frameworks: jQuery,
AngularJS, EmberJS, VueJS.

LADO SERVIDOR

Modelo de eventos NodeJS.
(Basic, Express)

POPULARIDAD

Gran cantidad de recursos y
documentación.

MULTIPLATAFORMA

Navegadores: Chrome, Safari, Edge
Sistemas: Windows, Mac, Linux.

VERSATILIDAD

Compatible con otros lenguajes:
PHP, Pearl, Java.

CARACTERÍSTICAS LENGUAJE INTERPRETADO

EJECUCIÓN

El navegador interpreta y ejecuta ese código directamente en su entorno de ejecución interno.

LÍNEA POR LÍNEA

El motor analiza y ejecuta cada línea de código a medida que se encuentra durante el proceso de carga de la página.

FLEXIBILIDAD

No es necesario compilar el código antes de probarlo, lo que permite un desarrollo rápido y una iteración ágil.



USOS PRINCIPALES

1

**Desarrollo de sitios web del lado del cliente
(front end, en el navegador).**

2

**Desarrollo de aplicaciones para dispositivos
móviles, híbridas o que compilan a nativo.**

3

**Desarrollo de aplicaciones de escritorio para
sistemas Windows, Linux y Mac.**

4

**Construcción de servidores web y aplicaciones
de servidor.**



FUNCIONES EN JAVASCRIPT

- Un conjunto de instrucciones que realiza una tarea o calcula un valor.

Debe tomar alguna entrada y devolver una salida. ●

- Debe definirse en algún lugar del ámbito desde el que deseas llamarla.

```
function square(number) {  
    return number * number;  
}
```

```
const square = function (number) {  
    return number * number;  
};  
  
var x = square(4); // x obtiene el valor 16
```

VARIABLES JAVASCRIPT

VAR

Declara una variable de scope global o local para la función sin importar el ámbito de bloque.

```
var i = 20
```

LET

Declara una variable de scope global, local para la función o de bloque. Es reassignable.

```
let i = 0;  
if(true) {  
  let i = 1;  
}
```

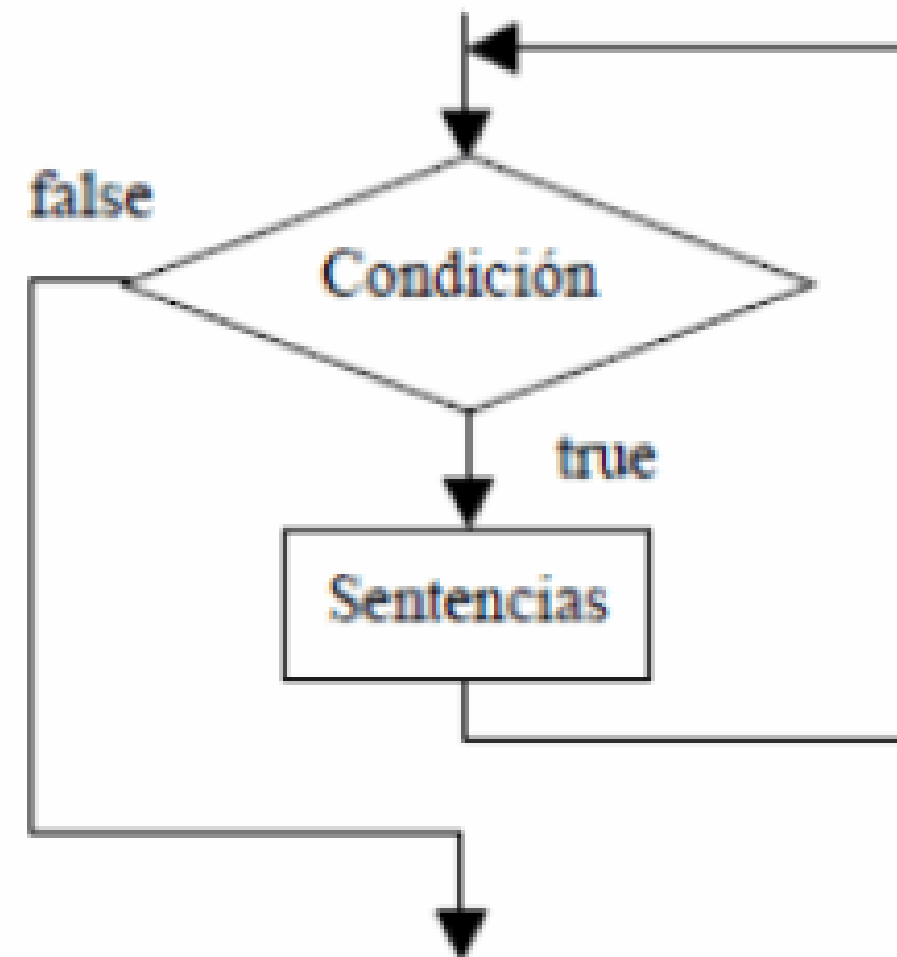
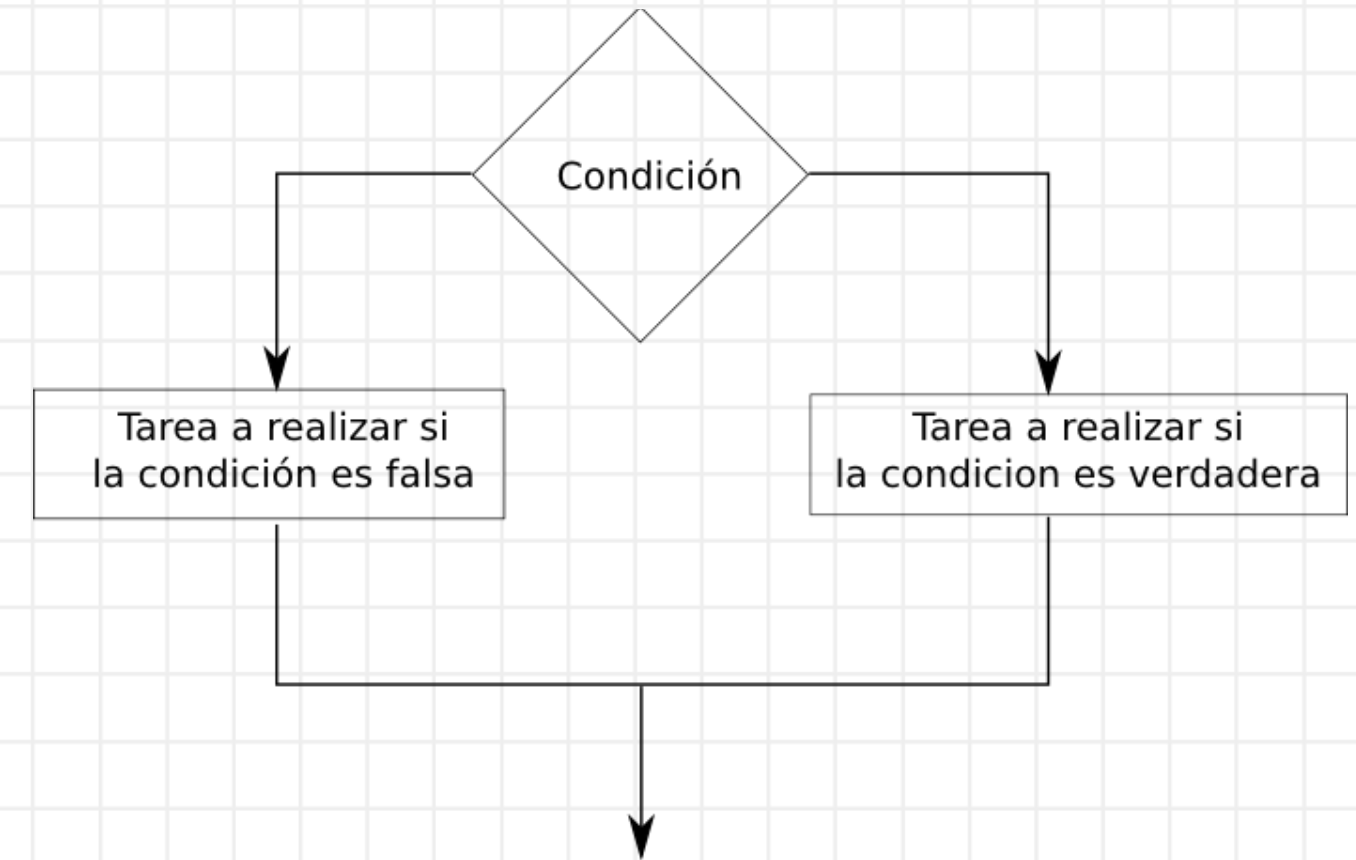
CONST

Declara variables que son constantes. No se puede reasignar su valor.

```
const x = 5
```


Tipos de estructuras de control en javascript

Estructuras condicionales:
Realizar una bifurcación del flujo de instrucciones.
Condiciones.



Estructuras de repetición:
Se utilizan para realizar de forma repetida varias acciones. **Bucles.**

ESTRUCTURAS CONDICIONALES

IF ELSE

```
var edad = prompt(«Dime tu
edad»);
if (num > 17) {
alert('Eres mayor de edad, puedes
acceder');
}else {
alert('Eres menor de edad; NO
puedes acceder');
}
```

SWITCH

```
var hoy = new Date()
var nombresmes
var mes=hoy.getMonth()
switch (mes) {
case 0 : nombresmes=» Enero»
break
case 1 : nombresmes=» Febrero»
break
case 6 : nombresmes=» Julio»
break
case 9 : nombresmes=» Setiembre»
break
default : nombresmes= «... no se el mes» }
alert(«Estamos en el mes llamado» + nombresmes)
```

ESTRUCTURAS DE REPETICIÓN

WHILE

```
var i = 0;
while (i<9) {
    alert (i);
    i++;
}
alert('»Salimos» del
while porque i vale: `
+i)
```

DO WHILE

```
var resultado = 1;
var numero = 5;

do {
    resultado = resultado *
    numero;
    numero--;
} while(numero > 0);

alert(resultado);
```

FOR

```
for (i=0; i<10; i++)
{
    alert (i);
}
```

Estructuras de control de errores.

- Permiten controlar los errores que el usuario final comete de forma fortuita o intencionada y poder seguir trabajando de forma normal.

TRY CATCH

- ```
try {
 document.write(10/variable1) ;
 //Variable insegura del usuario (puede introducir un 0)
}catch(e){
 alert(e.message);
 // Mensaje en caso de error }
```
-

```
<html>
<TITLE>Ejemplo06.htm</TITLE>

<head>
<script>
 // Ejemplo que visualiza un reloj digital.
 function Ver_Hora()
 {
 var mihora = new Date();
 var horas = mihora.getHours().toString();
 var minutos = mihora.getMinutes().toString();
 if (minutos.length == 1) minutos = "0" + minutos;
 var segundos = mihora.getSeconds().toString();
 if (segundos.length == 1) segundos = "0" + segundos;

 document.forms[0].mireloj.value = horas + " : " + minutos + " : "
+ segundos;
 }

</script>
</head>
<body>
<form>
<p align="center">
<input type="text" size="10" name="mireloj">
</p>
</form>
<script>
var r = setInterval ("Ver_Hora()",500);
</script>
</body>

</html>
```



# CONCLUSIÓN

Javascript es un lenguaje enfocado al desarrollo web.

Potente y amplio en el lado del cliente, pero funcional desde servidor.

Su estructura y funcionamiento es similar a POO base, pero mas simple.

De uso popular, estandarizado y de fácil acceso.





# INTEGRANTES DE BRASCO



No Brandon  
(Erick Siqueiros)



Brandon 2  
(David Vargas)



Brandon 1  
(Isaac Leyva)



<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>

<https://www.manualweb.net/javascript/caracteristicas-javascript/>

<https://cursohacker.es/estructuras-de-control-en-javascript/>

<https://cybmeta.com/var-let-y-const-en-javascript>

<https://blog.hubspot.es/website/que-es-javascript>

*Referencias.*

