

# Java Script - Conceptos Básicos

...

Resumen en base a  
<https://developer.mozilla.org/es/docs/Web/JavaScript>

# ¿Qué es JavaScript?

JavaScript® (a menudo abreviado como JS) es un lenguaje ligero, interpretado y orientado a objetos con funciones de primera clase (Esto significa que el lenguaje admite pasar funciones como argumentos a otras funciones, devolverlas como valores de otras funciones y asignarlas a variables o almacenarlas en estructuras de datos), y mejor conocido como el lenguaje de programación para las páginas Web, pero también se utiliza en muchos entornos que no son de navegador. Es un lenguaje de scripts que es dinámico, multiparadigma, basado en prototipos y admite estilos de programación orientados a objetos, imperativos y funcionales.

# Ejemplo, dentro de un archivo html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hola mundo");
  </script>
</body>
</html>
```

# Declaración de variables

Utilizando la palabra reservada `var` (no recomendable desde ES6).

Uno de los mayores problemas con la declaración de variables utilizando la palabra clave `var` es que tú puedes fácilmente sobrescribir declaraciones de variables:

```
<script>  
  var j = 10;  
  var j = "carlos";  
  console.log(j);  
</script>
```

# Declaración de variables utilizando let

Una palabra clave llamada let fue introducida en ES6, una actualización importante para JavaScript, para resolver este problema potencial con la palabra clave var.

Este ejemplo da error:

```
<script>  
  let j = 10;  
  let j = "carlos";  
  console.log(j);  
</script>
```

# Declarando constantes

Para declarar constantes, se utiliza la palabra reservada `const`, es práctica común que el nombre de la constante sea en mayúscula, separada por guiones bajos (snake case).

Ejemplo:

```
<script>  
  const NUMERO_PI = 3.14;  
  console.log(NUMERO_PI);  
</script>
```

# Inmutabilidad de cadenas

Las cadenas en javaScript son inmutables, es decir no se pueden cambiar los elementos que la componen, ejemplo:

```
<script>  
  let palabra = "Carlos";  
  palabra[0] = "c";  
  console.log(palabra);  
</script>
```

Para realizar un cambio de un elemento, hay que reemplazar toda la cadena.

# Arreglos

En javaScript, los arreglos pueden contener distintos tipos de datos, ejemplo:

```
<script>  
  const miLista = ['Juan',34,true];  
  console.log(miLista);  
  console.log(miLista[0]);  
  console.log(miLista[1]);  
  console.log(miLista[2]);  
</script>
```

En este caso la palabra reservada `const`, evita que se cambie el array a otro tipo de variable.



# Ejercicio

Dada esta lista, [3,-2,3,10,12,-14,-8], según los temas vistos , contar positivos y negativos, acumular positivos y negativos, contar los números mayores a 3.

Mostrar los resultados por consola del navegador.

# Métodos de los arreglos

Para agregar elementos a una lista (por ejemplo a [3,-2,3,10,12,-14,-8])

```
<script>  
  const miLista = [3,-2,3,10,12,-14,-8];  
  miLista.push(30);  
  console.log(miLista);  
  console.log('Tamaño de la lista ' + miLista.length)  
</script>
```

Para quitar un elemento del final de la lista `miLista.pop()`

# Funciones

La estructura es la siguiente: palabra reservada function, nombre de la función, parametros (optativos) y la palabra reservada return (optativo), ejemplo:

```
function calcularScoring(tiempo, ventas) {  
    if(tiempo>10) tiempo = 10;  
    if(ventas>100000) ventas = 100000 ;  
    return tiempo/10 * 100 + ventas/100000 * 100;  
}  
console.log(calcularScoring(10,100000));
```

# Ambito y alcance de las variables

Las variables declaradas dentro de una función, solo tienen alcance dentro de la misma.

```
let mivariable = 100;  
function calcular() {  
    let mivariable = 200;  
    console.log(mivariable);  
}  
calcular();  
console.log(mivariable);
```

# Función que no devuelve un valor

Una función puede no devolver un valor, en caso de llamar a una función que no devuelve un valor, se utiliza la palabra reservada undefined.

```
function calcular(){  
    let mivariable = 200;  
    mivariable++;  
}  
console.log(calcular());
```

# Operación de estricta igualdad

El operador `==` realiza una conversión de tipos de datos para funcionar.

El operador de estricta igualdad `===` compara valor y tipo, ejemplo:

`(5===5)` devuelve `true`

`(5=== '5')` devuelve `false`

# Tipo de variable

En javascript , para saber el tipo de variable, se utiliza el operador `typeof`

Ejemplo:

`typeof 5`, devuelve la cadena `number`

`typeof "5"` devuelve la cadena `string`.

# Ejercicio

Dada una lista de nombres de repuestos de autos ( bomba de agua, correa de distribución, aceite, junta sello eje, bujías ) y otra lista aparte, con los precios ( 15000, 3000, 3500, 5000, 2000 ), para determinar el precio del repuesto, se accede a la misma posición entre la lista de repuestos y la lista de precios, ejemplo: posición 1, correa de distribución, corresponde el precio de la posición 1 de la lista de precios: 3000.

Crear dos listas, con los nombres de los repuestos, para precios mayores a 3000 y otra con precios menores o iguales a 3000. Mostrar las dos listas por la consola del navegador.