

Scope (alcance) de variables

En js las variables tienen 3 diferentes alcances.

- Global: Son aquellas que son accesibles desde cualquier parte del código. Si se declara en la raíz del código esta puede ser utilizada en todas las subfunciones. Estas pueden saltar incluso de un script a otro en etiquetas html.

```
let nombre = "Luis"
function saludar(){
  console.log("Hola " + nombre);
}
saludar();
```

- Alcance Local: Son variables definidas dentro de funciones o pequeños fragmentos de código. Por lo que solo podrá ser llamada dentro de ese fragmento de lo contrario no funcionará.

```
function saludar(){
  let nombre = "Luis";
  console.log("Hola ", nombre);
}

console.log(nombre); //Error nombre no esta definido.
saludar(); // Imprime Hola Luis
```

- Bloque: las variables con let o const dentro de un bloque({}) solo son accesibles dentro de ese bloque. Algunos de ejemplos de bloque son: if o for.

```
if(true){
  let nombre = "Juan";
  console.log("Hola ", nombre); // Hola Juan
}
console.log("Hola ", nombre); // Error nombre no esta definido
```

Hoisting

En js cuando definimos una variable con var estas se mueven al inicio del script. Lo que puede provocar cosas inesperadas. Al usar **let** o **const** estas variables no se elevarán al inicio del script lo que ayuda a evitar errores.

Operadores de comparación

Comparación del lado derecho con el izquierdo para obtener un valor booleano.

- Igual (==): Compara si dos valores son iguales.

- Estrictamente igual (===): Compara si los dos valores son iguales y del mismo tipo.
- Distinto (!=): Compara si dos valores son distintos.
- Estrictamente distinto (!===)
- Mayor que (>) : Compara si un valor es mayor que otro
- Menor que (<): Compara si un valor es menor que otro
- Mayor o igual que(>=): Compara si un valor es mayor o igual que otro
- Menor o igual que(<=): Compara si un valor es menor o igual que otro.

```
let numero = 10;
let esPar = if(numero%2 == 0) ; //true
let esImpar = if(numero%2 == 1 ); //false

let num1 = "5";
let num2 = 5;
let sonIguales = if(num1 == num2)//true
    sonIguales = if(num1 === num2)// false
```

Truthy/Falsy

Las variables en JS pueden tener un valor de truthy o falsy. Un valor truthy es aquel que se evalúa como verdadero en un contexto de booleanos mientras que el falsy se evalúa como falso.

Podemos usar esta propiedad para la toma de decisiones en nuestro código.

Cualquier número positivo o negativo devuelve un valor truthy y el 0 devuelve un falsy. Un arreglo vacío y un objeto vacío me devuelven un truthy.

Valores falsy:

- undefined
- null
- NaN
- 0
- "" (Cadena vacía)
- false

```
let edad = 18;
if(edad >= 18){
    // se ejecuta si el valor es truthy
    console.log("Eres mayor de edad");
}else{
    // se ejecuta si el valor es falsy
    console.log("Eres menor de edad");
}

let numero = 16;
if(numero%2 == 0){
    console.log("El número ", numero , "es un número par");
}
```

```
}else{
    console.log("El número ", numero ,"es un número impar");
}

//16%2 =0 y 0 es falsy
if(numero%2){
    console.log("El número ", numero ,"es un número impar");
}else{
    console.log("El número ", numero ,"es un número par");
}

if(false){
    // esto no se ejecuta
}else{
    // esto si se ejecuta
}
```

Reducción de condicionales quitando el else

Podemos simplificar nuestro código condicional quitando el else. Esto es útil cuando no necesitamos un bloque false o cuando tenemos múltiples condiciones.

```
let user= "cei";
let password ="abc123";
let mensaje = "Clave correcta";

if(password.length <= 8){
    mensaje = "La pass debe de tener al menos 8 caracteres";
}
if(password.includes(" ")){
    mensaje = "La pass no debe de tener espacios";
}
if(password.includes(user)){
    mensaje = "La pass no puede incluir tu nombre de usuario";
}
if(password.toLowerCase() === password){
    mensaje = "La pass debe de contener una letra mayúscula ";
}
console.log(mensaje);
```

Manipulación de textos

1. Concatenación

```
let nombre ="Luis"
let apellido = "Estrada"
let nombreCompleto = nombre + " " + apellido;
```

```
// Concatenar utilizando +=  
let numero = 5;  
numero += 3;  
  
let saludo = "Hola, ";  
saludo += "Como estas?";
```

2. Interpolación: Con las plantillas literales (template literals), pueden insertar variables o expresiones dentro de una cadena usando ```.

```
let nombre = "Luis"  
let apellido = "Estrada"  
let nombreCompleto = `${nombre} ${apellido}`;  
  
let total = 50;  
let iva = 0.21;  
console.log(`El total con impuestos es: ${total + (total * iva)}`)
```

3. Métodos de Strings Js nos ofrece una variedad de métodos para manipular cadenas de textos. Algunos ejemplos comunes son:

- `toUpperCase()` y `toLowerCase()` Colocar las letras a mayúsculas o todas minúsculas
- `split()` divide una cadena en una lista/array basada en un separador. Devolviendo una lista.
- `slice()` Extrae una porcion de la cadena basada en indices de inicio y fin.Sin colocar el indice final.
- `replace()` reemplaza un parte de la cadena por otra (la primer ocurrencia).
- `trim()` Elimina los espacios en blanco al inicio o fin de la cadena.
- `includes()` Verifica si una cadena contiene una subcadena específica.
- `startsWith()` y `endsWith()` Verifica si una cadena inicia o termina con una cadena especifica.
- `repeat()` Repite una cadena un número específico de veces.

```
let text = "Me encanta JavaScript";  
  
// UppeCase LowerCase  
console.log(text.toUpperCase()); // ME ENCANTA JAVASCRIPT  
console.log(text.toLowerCase()); // me encanta javascript  
  
//Split  
let palabras = text.split(" "); // ['Me','encanta','javaScript']  
  
//slice  
let frase = "Aprender javascript es divertido"
```

```
console.log("Parte de frase: ", frase.slice([9,19]));// "javascript"
console.log("Desde el inicio 9 al final: ", frase.slice(9));// "javascript es divertido"

// replace
let frase2 = "Hola mundo, hola universo";
console.log("Reemplazar 'hola' por 'Hola' : ", frase2.replace('hola','Hola'));//
Reemplazar 'hola' por 'Hola' : Hola mundo,Hola universo
console.log("Reemplazar todos los 'hola' por 'Hola': ",
frase.replace(/hola/g,"Hola"));

//trim
let frase3 = "      Hola mi      nombre es      Luis      ";
console.log(frase3.trim());// Hola mi      nombre es      Luis

//include
let frase4 = "El sol brilla en el cielo";
console.log("Contiene sol?: " , frase4.include("sol"));// Contiene sol?: true
console.log("Contiene luna?: ", frase4.include("luna"));// Contiene luna?: false

//startsWith endsWith
let frase5 = "JavaScript es genial";
console.log("Empieza con Java?: " , frase5.startsWith("Java"));// Empieza con
Java?:true
console.log("Termina con genial?: ", frase5.endsWith('genial'))//Termina con
genial?: true

// repeat
let risa = "Ja";
console.log("Me causa mucha gracia: ", risa.repeat(3));// Me causa mucha gracia
JaJaJa
```

Podemos combinar los métodos como en el ejemplo siguiente.

```
let frase = "    No me gusta Javascript en diseño web    ";
let resultado = frase.trim()// elimina espacios
                .toUpperCase()// convierte el texto a mayúsculas
                .replace('NO ME GUSTA', ' ME ENCANTA');// reemplaza el texto
en mayúsculas porque el método anterior le hizo toUpperCase().
console.log(resultado)// ME ENCANTA JAVASCRIPT EN DISEÑO WEB
```