

1.- JavaScript : SCOPE y STRICT MODE

Recordar que cuando se carga código como un MODULO este se carga de forma estricta (STRICT) es decir que todas las variables deben definirse antes de usarlas, no podemos hacer `a=5` sin antes haberla definido con `var`, `let` o `const`. En cambio si no estamos en modo estricto si podemos hacerlo.

Var no respeta el SCOPE ya que se puede modificar su valor dentro del bloque debido a lo cual **NO DEBE USARSE** porque es muy fácil perder el control, modificando el valor de la variable dentro de una función sin querer...

```
var a = 5;
{
  var a = 6;
  console.log("A dentro del bloque", a);
}
console.log("A fuera del bloque", a);
```

Mucho mejor solo usar **let** y **const** que si respetan el scope/ámbito definido por `{ }`

```
let b = 5;
{
  let b = 6;
  console.log("B dentro del bloque", b);
}
console.log("B fuera del bloque", b);
```

Ojo con **let**, si no ponemos dentro del bloque la palabra `let` la variable cambia de valor

```
let c = 5;
{
  c = 6;
  console.log("C dentro del bloque", c);
}
console.log("C fuera del bloque", c);
```

1.- JavaScript : SCOPE y STRICT MODE

CONST no permite reasignar valores es decir el valor es constante en este caso parece que reasignamos valores pero no es así porque d fuera del bloque es una variable diferente a dentro del bloque porque la hemos vuelto a definir con const pero si a la linea const d = 6; le quitamos el const fallaría porque estaríamos reasignando el valor

```
const d = 5;
{
  const d = 6;
  console.log("D dentro del bloque", d);
}
console.log("D fuera del bloque", d);
```

Bibliografía

Let y const: https://exploringjs.com/impatient-js/ch_variables-assignment.html#let

Strict mode: https://exploringjs.com/impatient-js/ch_syntax.html#strict-mode

Ejercicios

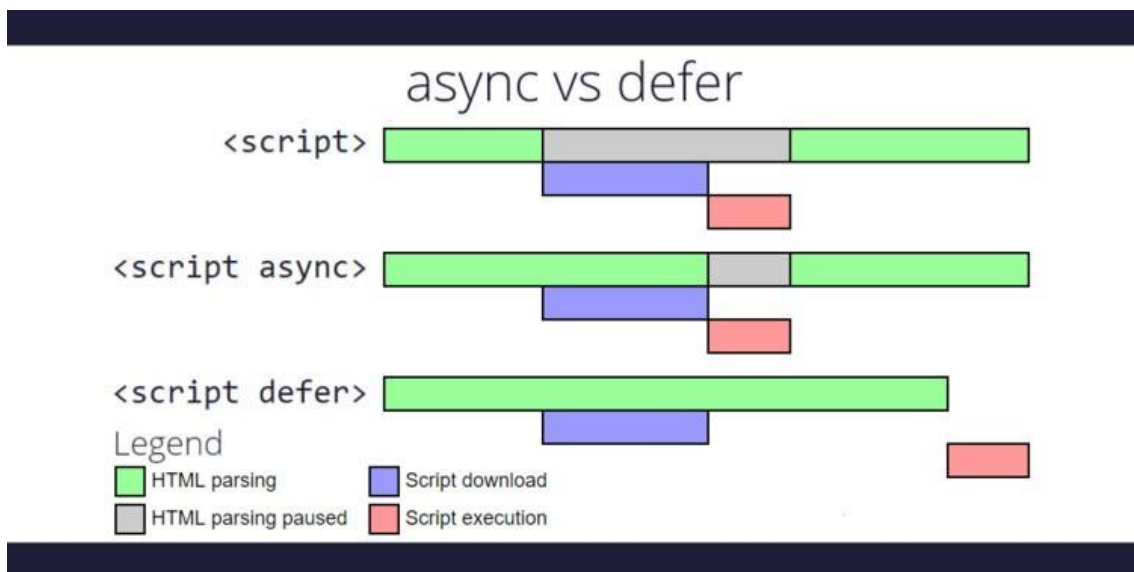
1. Prueba los trozos de código y anota los resultados de las variables
2. Crea algún ejemplo más donde se demuestre si respetan el scope tanto para VAR, LET y CONST
3. Amplia la documentación con información buscada en internet

AMPLIACIÓN

Atributos async y defer en la Etiqueta <script>

A partir de HTML5, la etiqueta <script> admite dos atributos, async y defer, que permiten que los scripts comiencen su descarga inmediatamente sin interrumpir el proceso de parseo del documento HTML.

1.- JavaScript : SCOPE y STRICT MODE



Async

El atributo `async` permite que el script se ejecute de manera asíncrona tan pronto como se haya descargado. Esto significa que:

- El script se descargará en paralelo con el parseo del documento.
- Se ejecutará tan pronto como la descarga haya terminado, sin esperar a que el documento se haya parseado por completo.
- Los scripts marcados con `async` pueden no ejecutarse en el orden en el que aparecen en el documento HTML, ya que la ejecución depende de cuál script se descargue primero.

Ejemplo:

```
<script src="scripts.js" async></script>
```

defer

El atributo `defer` asegura que los scripts se ejecutarán en el orden en el que aparecen en el documento HTML. Las características de `defer` incluyen:

- El script se descarga en paralelo con el parseo del documento, similar a `async`.
- La ejecución del script se pospone hasta que el documento se haya parseado completamente.
- La ejecución ocurre justo antes del evento `DOMContentLoaded`, pero después de que todo el HTML haya sido procesado.

1.- JavaScript : SCOPE y STRICT MODE

Ejemplo:

```
<script src="scripts.js" defer></script>
```

Sin async ni defer

Si un script no tiene ninguno de estos atributos, el comportamiento por defecto es que el script se descarga y se ejecuta de manera síncrona. Esto significa que:

- El parseo del documento se pausa hasta que el script haya sido descargado y ejecutado completamente.
- Los scripts se ejecutan en el orden en que aparecen en el documento.

Ejemplo:

```
<script src="scripts.js"></script>
```

OTROS

En este caso investiga o busca recursos (autoaprendizaje) sobre:

- ¿Cómo ofuscar código JS? Hay páginas webs que lo hacen. Escribe un programa en JS y ofúscalos y prueba que sigue funcionando
- Busca un editor “bueno” ONLINE y prueba que funciona.
 - Codesandbox
 - project idx (de google)
- Páginas de Retos de programación
 - CodeSignal (Intenta hacer los primeros 8 ejercicios del modo INTRO)

1.- JavaScript : SCOPE y STRICT MODE

RELACIÓN DE EJERCICIOS DE REPASO

1. **Imprimir "Hola, Mundo"**
Escribe un programa que muestre "Hola, Mundo" en la consola.
2. **Suma de dos números**
Crea una función que acepte dos números como argumentos y devuelva su suma.
3. **Par o impar**
Escribe una función que determine si un número dado es par o impar.
4. **Números del 1 al 10 en un bucle**
Usa un bucle `for` para imprimir los números del 1 al 10.
5. **Suma de números del 1 al 100**
Escribe un programa que calcule la suma de los números del 1 al 100.
6. **Factorial de un número**
Crea una función que calcule el factorial de un número dado.
7. **Número mayor entre tres**
Escribe una función que acepte tres números y devuelva el mayor.
8. **Números primos del 1 al 100**
Escribe un programa que imprima todos los números primos entre 1 y 100.
9. **Invertir una cadena de texto**
Crea una función que acepte una cadena de texto y devuelva la cadena invertida.
10. **Contar vocales en una cadena**
Escribe una función que cuente el número de vocales en una cadena.
11. **Multiplicación de dos números sin usar el operador ***
Escribe una función que multiplique dos números sin utilizar el operador de multiplicación.
12. **Validar una dirección de correo electrónico**
Crea una función que valide si una cadena es un correo electrónico válido (simplemente revisa si contiene "@" y ".").
13. **Generar números aleatorios**
Escribe un programa que genere un número aleatorio entre 1 y 100.
14. **Array de números pares**
Crea un array que contenga los números pares entre 1 y 50.
15. **Eliminar duplicados de un array**
Escribe una función que acepte un array y elimine los valores duplicados.
16. **Calcular promedio de un array**
Crea una función que acepte un array de números y devuelva el promedio de los mismos.
17. **Palíndromo**
Escribe una función que determine si una palabra o frase es un palíndromo.
18. **Contar palabras en una oración**
Crea una función que cuente cuántas palabras hay en una cadena de texto.
19. **FizzBuzz**
Escribe un programa que imprima los números del 1 al 100, pero para múltiplos de 3 imprime "Fizz", para múltiplos de 5 "Buzz", y para múltiplos de ambos "FizzBuzz".
20. **Encontrar el número más pequeño en un array**
Crea una función que encuentre el número más pequeño en un array de números.

1.- JavaScript : SCOPE y STRICT MODE

ESTO ES LO QUE ME DICE CHATGPT pero hay muchas cosas que no sabemos de las soluciones aportadas por esta IA, intenta hacerlos tú usando lo que si sabes (if, bucles, cadenas...)

En rojo he puesto lo que sospecho que no sabes, puedes buscarlo en internet y hacerte una idea de porqué funciona, pero seguro que con lo que sabes ahora y pensando “un poco” puedes hacerlos, así es como se aprende a programar (lógica de programación)

1. Imprimir "Hola, Mundo"

```
console.log("Hola, Mundo");
```

2. Suma de dos números

```
function sumar(a, b) {  
    return a + b;  
}  
console.log(sumar(5, 3)); // 8
```

3. Par o impar

```
function esPar(numero) {  
    return numero % 2 === 0 ? "Par" : "Impar";  
}  
console.log(esPar(4)); // Par  
console.log(esPar(7)); // Impar
```

4. Números del 1 al 10 en un bucle

```
for (let i = 1; i <= 10; i++) {  
    console.log(i);  
}
```

5. Suma de números del 1 al 100

```
let suma = 0;  
for (let i = 1; i <= 100; i++) {  
    suma += i;  
}  
console.log(suma); // 5050
```

6. Factorial de un número

```
function factorial(n) {  
    let resultado = 1;  
    for (let i = 1; i <= n; i++) {  
        resultado *= i;  
    }  
    return resultado;  
}  
console.log(factorial(5)); // 120
```

7. Número mayor entre tres

```
function mayorDeTres(a, b, c) {  
    return Math.max(a, b, c);  
}  
console.log(mayorDeTres(5, 9, 3)); // 9
```

8. Números primos del 1 al 100

```
function esPrimo(num) {  
    if (num <= 1) return false;  
    for (let i = 2; i < num; i++) {
```

1.- JavaScript : SCOPE y STRICT MODE

```
        if (num % i === 0) return false;
    }
    return true;
}

for (let i = 1; i <= 100; i++) {
    if (esPrimo(i)) {
        console.log(i);
    }
}
```

9. Invertir una cadena de texto

```
function invertirCadena(cadena) {
    return cadena.split("").reverse().join("");
}
console.log(invertirCadena("javascript")); // tpircsavaj
```

10. Contar vocales en una cadena

```
function contarVocales(cadena) {
    let contador = 0;
    let vocales = "aeiouAEIOU";
    for (let letra of cadena) {
        if (vocales.includes(letra)) {
            contador++;
        }
    }
    return contador;
}
console.log(contarVocales("Hola Mundo")); // 4
```

11. Multiplicación de dos números sin usar el operador *

```
function multiplicar(a, b) {
    let resultado = 0;
    for (let i = 0; i < Math.abs(b); i++) {
        resultado += a;
    }
    return b < 0 ? -resultado : resultado;
}
console.log(multiplicar(5, 3)); // 15
console.log(multiplicar(5, -3)); // -15
```

12. Validar una dirección de correo electrónico

```
function validarCorreo(correo) {
    return correo.includes("@") && correo.includes(".");
}
console.log(validarCorreo("test@example.com")); // true
console.log(validarCorreo("testexample.com")); // false
```

13. Generar números aleatorios

```
let numeroAleatorio = Math.floor(Math.random() * 100) + 1;
console.log(numeroAleatorio);
```

14. Array de números pares

```
let numerosPares = [];
for (let i = 1; i <= 50; i++) {
    if (i % 2 === 0) {
        numerosPares.push(i);
    }
}
```

1.- JavaScript : SCOPE y STRICT MODE

```
}  
console.log( numerosPares );
```

15. Eliminar duplicados de un array

```
function eliminarDuplicados(array) {  
    return [...new Set(array)];  
}  
console.log(eliminarDuplicados([1, 2, 2, 3, 4, 4, 5])); // [1, 2, 3, 4, 5]
```

16. Calcular promedio de un array

```
function calcularPromedio(array) {  
    let suma = array.reduce((a, b) => a + b, 0);  
    return suma / array.length;  
}  
console.log(calcularPromedio([5, 10, 15])); // 10
```

17. Palíndromo

```
function esPalindromo(cadena) {  
    let cadenaLimpia = cadena.toLowerCase().replace(/[^\w ]/g, "");  
    let invertida = cadenaLimpia.split("").reverse().join("");  
    return cadenaLimpia === invertida;  
}  
console.log(esPalindromo("Anita lava la tina")); // true  
console.log(esPalindromo("Hola Mundo")); // false
```

18. Contar palabras en una oración

```
function contarPalabras(oracion) {  
    return oracion.split(" ").length;  
}  
console.log(contarPalabras("Hola, ¿cómo estás?")); // 3
```

19. FizzBuzz

```
for (let i = 1; i <= 100; i++) {  
    if (i % 3 === 0 && i % 5 === 0) {  
        console.log("FizzBuzz");  
    } else if (i % 3 === 0) {  
        console.log("Fizz");  
    } else if (i % 5 === 0) {  
        console.log("Buzz");  
    } else {  
        console.log(i);  
    }  
}
```

20. Encontrar el número más pequeño en un array

```
function encontrarMinimo(array) {  
    return Math.min(...array);  
}  
console.log(encontrarMinimo([3, 1, 4, 1, 5, 9])); // 1
```