

Bootcamp de Full Stack

Bienvenidos a la clase N°32

- Modo estricto
- Operadores avanzados:
 - Ternario
 - Optional chaining
 - Nullish coalescing
 - Logical OR
- IIFE + Patrón Módulo
- Proyecto Integrador F1 (i5)

JavaScript

Modo Estricto

El modo estricto en ECMAScript 5 (introducido en 2009) es una forma de ejecutar JavaScript en una variante más restringida y segura del lenguaje. Se activa explícitamente utilizando la directiva "use strict" al comienzo de un archivo o función. Su propósito es ayudar a los desarrolladores a escribir código más robusto, predecible y compatible con futuras versiones del lenguaje. Esta directiva no afecta a todo el entorno por defecto, lo que permite migrar el código gradualmente al modo estricto.

El modo estricto introduce varios cambios respecto al modo no estricto (modo "sloppy"):

- Convierte errores silenciosos en errores explícitos, facilitando la depuración.
- Desactiva características problemáticas del lenguaje, como el uso implícito de variables globales.
- Prohíbe sintaxis que podría introducir conflictos con futuras versiones del estándar ECMAScript.

Nota: La versión actual del estándar ECMAScript es ECMAScript 2021 (ES12).

JavaScript

Operadores Avanzados

Ternario: Es un operador condicional que evalúa una expresión booleana y devuelve uno de dos valores según si la condición es verdadera o falsa.

```
let edad = 21;  
const message = edad >= 18 ? "Mayor de edad" : "Menor de edad";
```

Optional Chaining: Este operador permite acceder a propiedades o métodos de forma segura, evitando errores si alguna referencia intermedia es null o undefined.

```
let person = {  
  name: "Juan",  
  address: {  
    street: "Av. siempreviva 100"  
  }  
};  
  
const ciudad = person?.address?.country;
```

JavaScript

Operadores Avanzados

Nullish Coalescing: Es un operador lógico que devuelve el operando del lado derecho solo si el operando del lado izquierdo es `null` o `undefined`. En caso contrario, devuelve el operando del lado izquierdo. Los valores considerados son `null` y `undefined`.

```
let greet = null;  
const message = greet ?? "Hola Mundo";
```

Logical OR: Este operador devuelve el operando del lado derecho si el del lado izquierdo es un valor “falsy”. Caso contrario, devuelve el operando izquierdo. Los valores considerados como falsos (falsy) son: `false`, `0`, `-0`, `""`, `null`, `undefined`, `NaN`. En otras palabras, son los valores no “truthy”.

```
let greet = '';  
const message = greet || "Hola Mundo";
```

JavaScript

IIFE + Patrón Módulo

Un **IIFE** (Immediately Invoked Function Expression) es una función que se define y se ejecuta automáticamente en el momento en que se declara. Se usa para crear un ámbito privado y evitar contaminar el ámbito global.

```
((() => {  
  console.log("Soy un IIFE");  
}))();
```

El **Patrón Módulo** es una técnica de diseño que organiza el código en partes independientes, encapsulando datos y exponiendo solo lo necesario. Este patrón suele usar IIFE para crear un ámbito privado y retornar un objeto con funciones y/o propiedades accesibles desde afuera.

```
const counter = (() => {  
  let value = 0;  
  return {  
    increment: function () { return value += 1; },  
  };  
})();  
console.log(counter.increment()); // 1  
console.log(counter.increment()); // 2
```

BREAK

Descansemos 10 minutos



Proyecto Integrador F1

Iteración N°5



Para llevar a cabo el proyecto integrador F1, accede al documento de la consigna a través del siguiente enlace:

<https://docs.google.com/document/d/181methzV3GUL-YCBNp-qE94qE8-eEiA9Q169gyzUvSs/edit?usp=sharing>

CIERRE DE CLASE

Continuaremos en la próxima clase

