**Modo estricto (“use strict”):** Basicamente esto se usa para que tengamos buenas prácticas, de lo contrario nos corregirá mediante la aparición de errores.

-Convierte errores de javascript en excepciones.

-Mejorando la optimización de los errores y consigue mejores tiempos de ejecución.

-Evita sintaxis usadas en posteriores a ES6.

-No permite que el programador realice una “mala sintaxis”.

**Como usar modo estricto:**

-**Variables declaradas:** Básicamente nos hace definir una variable estrictamente.

"use strict";

let nombre = "juan";

console.log(nombre)

-**Modificar propiedades (defineProperty() y writeable):** Básicamente si aplicamos esto las propiedades ya definidas no se podrán sobreescribir.

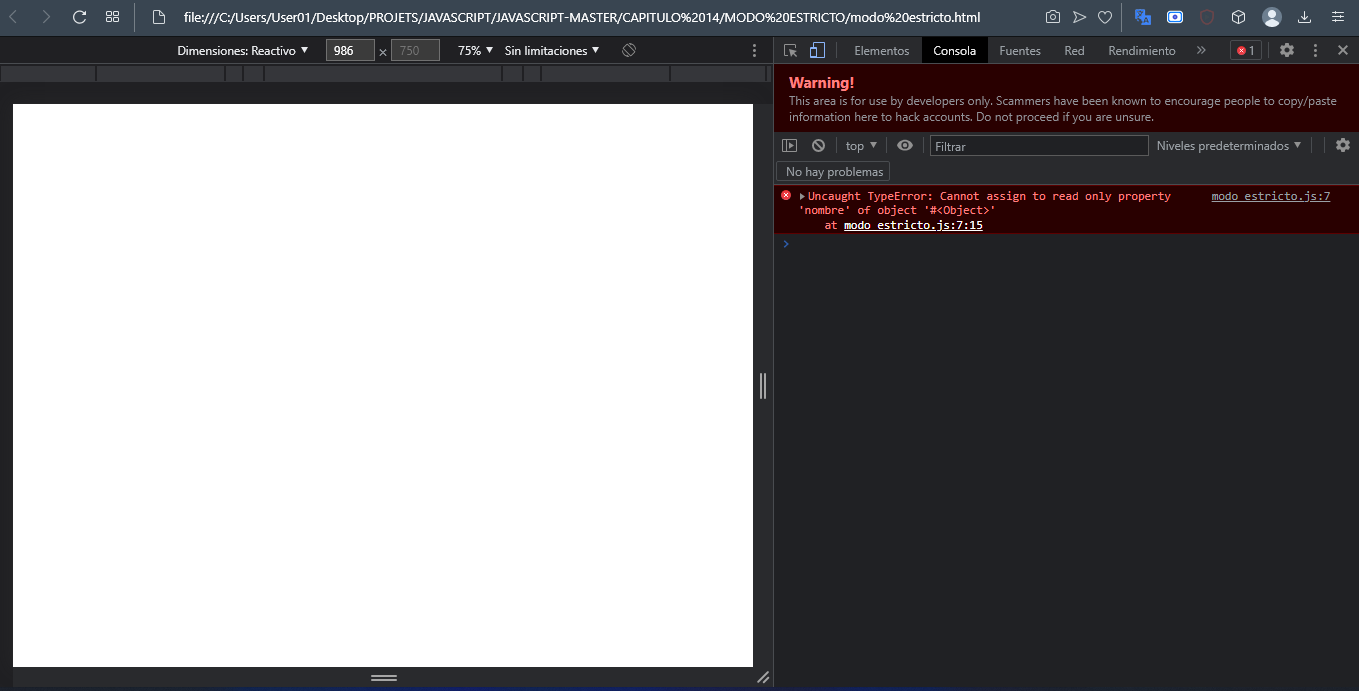
"use strict";

const objeto ={}; //creamos un objeto

Object.defineProperty(objeto,'nombre',{value: "pedro",writeable: false});//creamos lo que va a ir dentro del objeto.

objeto.nombre = "robert lewandowsky" //esto es para sobreescribir y que nos aparezca este nuevo nombre en consola, pero como aplicamos "writeable: false", saldrá un error ya que le indicamos que no se puede sobreescribir el nombre.

console.log(objeto) //mostramos el objeto en consola



-**Agregar propiedades:** aplicando preventExtensions al OBJETO no se podrá agregarle más contenido dentro de él

"use strict";

const objeto ={nombre:"juan"}; //creamos un objeto

Object.preventExtensions(objeto);

objeto.apellido = "perez"//este contenido ya no se podrá agregar al objto

console.log(objeto.apellido)

-**No se puede agregar propiedades a un string.**

"use strict";

const str = "juan perez";

str.canal = "jota pe" //debido a esta linea saltará un error usando "use stric" ya que no se puede agregar propiedades(o más contenido) a un string.

console.log(str)

-**No existen las multiples variables en una función.**

//"use strict";

//ESTO NO EXISTE. NO PUEDE HABER 2 PARAMETROS IGUALES (texto,texto)

function hablar(texto,texto){ //1:texto=undefined 2: texto=es el que va a tomar para mostrar en consola

    console.log(texto)

}

hablar("pedro","pedro") //Mostrará el 2do en consola, esto está mal. En modo estricto esto no sucede

"use strict";

//ESTO SI EXISTE

function hablar(texto){ //1:texto=undefined 2: texto=es el que va a tomar para mostrar en consola

    console.log(texto)

}

-**Delete en objeto o variables(SIGNIFICA QUE NO SE PUEDEN ELIMINAR VARIABLES y OBJETOS EN MODO ESTRICTO, PERO SI LAS PROPIEDADES QUE CONTIENE UN OBJETO).**

"use strict";

//ESTO NO EXISTE (NO SE PUEDEN ELIMINAR VARIABLES, PERO SI LO QUE HAY DENTRO DE UN OBJETO)

let variable = "juan";

delete variable;

console.log(variable)

"use strict";

//ESTO SI EXISTE

const objeto ={

    nombre: "juan"

}

delete objeto.nombre

console.log(objeto.nombre) //nos dirá que ya esta undefined porque lo eliminamos.

-**Las palabras reservadas(como delete,package, return,etc) no pueden ser usadas como variables.Si las usamos nos saltará un error.**

"use strict";

let delete = "pedro";

console.log(delete)

"use strict";

let package = "pedro";

console.log(package)

**-Cambia el This.**

**THIS USADO EN FUNCIONES EN MODO ESTRICTO**

"use strict";

//ESTO NO EXISTE EN MODO ESTRICTO. PERO SI, SI NO USAMOS MODO ESTRICTO

function saludar(){

    this.nombre = "pedro";

    this.saludar = function(){

        console.log(`Hola ${this.nombre}`)

    }

}

obj = new saludar(); //DEFINIMOS LA VARIABLE

obj.saludar()//LLAMAMOS A LA FUNCIÓN

//RESULTADO EN CONSOLA: Hola pedro

**THIS USADO DE FORMA CLASICA EN UN CONSTRUCTOR**

"use strict";

class pedro{

    constructor(nombre){

        this.nombre = nombre

    }

}

**-Numeros octales con una “0” adelante y tampoco existe with.**

//"use strict";

//ESTO NO EXISTE

console.log(0025)//esto está en base 8

**-Arguments y Eval NO pueden ser variables.**

"use strict";

//ESTO NO EXISTE

let arguments ="hola"

console.log(arguments)

"use strict";

//ESTO NO EXISTE

let eval ="hola"

console.log(eval)