Ejercicio Java Script

Declaracion de Variables:

- var miVar = 1234;
- var miCadena = 'Hola, mundo';

Constantes:

- const a = 7;
- const COLOR_RED = "# F00";

Operadores:

- Operadores de asignación
  - x = y
  - x += y
- Operadores de comparación
  - `3 == var1`
  - `var1 != 4`
- Operadores aritméticos
  - `12 % 5` devuelve `2`.
  - `2 ** 3` devuelve `8`.
- Operadores bit a bit

Expresión	Resultado	Descripción binaria
15 & 9	9	1111 & 1001 = 1001
15 | 9	15	1111 | 1001 = 1111

- Operadores lógicos
  - var n1 = !true; // !t devuelve false
  - var n2 = !false; // !f devuelve true
- Operadores de cadena de caracteres
  - console.log("mi " + "string"); // lanza el String "mi string" en la consola.
  - var mistring = "alfa";
    mistring += "beto"; // devuelve "alfabeto" y asigna este valor a "mistring".
- Operador condicional (ternario)
  - condición ? valor1 : valor2
  - var estado = (edad >= 18) ? "adulto" : "menor";
- Operador coma
  - for (var i = 0, j = 9; i <= j; i++, j--)
    console.log("a[" + i + "][" + j + "]= " + a[i][j]);

Expresiones:

1. Expresiones primarias
   a. **This**
      i. this["nombreDePropiedad"]
      ii. this.nombreDePropieda

iii.
```
function validate(obj, lowval, hival){

    if ((obj.value < lowval) || (obj.value > hival))

        alert("¡Valor inválido!");
    }
```

iv.
```
<b>Ingrese un número entre 18 y 99:</b>
<input type="text" name="age" size=3 onChange="validate(this, 18, 99);">
```

v.
```
<form name="miFormulario">

    Nombre del formulario:

    <input type="text" name="text1" value="Beluga"/>

    <input type="button" name="button1" value="Mostrar nombre del formulario"

        onClick="this.form.text1.value = this.form.name;"/>

</form>
```

b. **Operador de agrupación**

i.
```
var a = 1;
var b = 2;
var c = 3;
```

ii.
```
// se evalúa por defecto como
a + (b * c)   // 7
```

2. Expresiones al lado izquierdo

a. New

i.
```
var nombreDeObjeto = new tipoDeObjeto([param1, param2, ..., paramN]);
```

b. super

i.
```
super([argumentos]); // llama al constructor padre.
```

ii.
```
super.funcionEnPadre([argumentos]);
```

c. **Operador de propagación**

i.
```
var partes = ["hombros", "rodillas"];
```

ii.
```
var letra = ["cabeza", ...partes, "y", "dedos"];
```

iii.
```
function f(x, y, z) { }
var args = [0, 1, 2];
f(...args);
```

Sentencias Selectivas:

1. if():

a.
```
var x = 33;
 if( x == 33 )
{

   x = 55;

}
```

2. sentencia condicional doble
    a. var x = 80;

```
if( x >= 100 )
{
    alert("Mayor o igual de 100");
}
else
{
    alert("Menor de 100");
}
```

3. Sentencia condicional Multiple

```
var x = -80;
if( x >= 100 )
{
    alert("El valor es igual o mayor de 100");
}
else if( x >= 0 )
{
    alert("El valor está entre 0 y 99");
}
else
{
    alert("El valor es negativo");
}
```

4. **switch()**
    - var dia = 3;

```
switch( dia )
{
    case 1:
      alert("Lunes");
       break;
```

```
    case 2:
        alert("Martes");
        break;
    case 3:
        alert("Miércoles");
        break;
    case 4:
        alert("Jueves");
        break;
    case 5:
        alert("Viernes");
        break;
    case 6:
        alert("Sábado");
        break;
    case 7:
        alert("Domingo");
        break;
     default:
        alert("Valor no válido");
}
```

- ```
  var dia = "MIERCOLES";
  switch( dia )
  {
      case "LUNES":
          alert(1);
           break;
      case "MARTES":
          alert(2);
          break;
      case "MIERCOLES":
  ```
  ```

```
        alert(3);
        break;
    case "JUEVES":
        alert(4);
        break;
    case "VIERNES":
        alert(5);
        break;
    case "SABADO":
        alert(6);
        break;
    case "DOMINGO":
        alert(7);
        break;
     default:
        alert("Día no válido");
}
```

Estructuras iterativas: while, do.. While, For:

1. **while().**

```
var contador = 1;
 while( contador <= 5 )
{
    alert( "Ahora contador tiene valor [" + contador + "]" );
     contador = contador + 1;
}
```

2. **do ... while()**

```
var contador = 1;
 do
{
    alert( "Ahora contador tiene valor [" + contador + "]" );
     contador = contador + 1;
```

```
} while( contador <= 5 )
```

3. **for()**

```
var contador;
 for(contador=1; contador <= 5; contador++ )
{
    alert( "Ahora contador tiene valor [" + contador + "]" );
}
```

Try, catch, throw:

1. **try ... catch**

```
var x = 33;
 try
{
    x = x + y;
}
 catch( e )
{
    alert( "error: " + e.message );
}
```

2. **Throw**
   - throw "Error2"; // genera una excepción con un valor cadena
   - throw 42; // genera una excepción con un valor 42
   - throw true; // genera una excepción con un valor true
   - function ExceptionUsuario(mensaje) {

```
  this.mensaje = mensaje;
  this.nombre = "ExceptionUsuario";
}

function getNombreMes(mes) {
  mes = mes - 1; // Ajustar el número de mes al índice del arreglo (1 = Ene, 12 = Dic)
```

```
   var meses = new Array("Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul",
      "Ago", "Sep", "Oct", "Nov", "Dic");
   if (meses[mes] != null) {
      return meses[mes];
   } else {
      miExcepcionUsuario = new ExceptionUsuario("NumeroMesNoValido");
      throw miExcepcionUsuario;
   }
}


try {
   // sentencias para try
   nombreMes = getNombreMes(miMes);
} catch (excepcion) {
   nombreMes = "desconocido";
   registrarMisErrores(excepcion.mensaje, excepcion.nombre); // pasa el objeto exception al
manejador de errores
}
```

Objeto Error: