Desarrollo Entorno Cliente

# 2.A.- Estructura del lenguaje JavaScript.

## 1.- Fundamentos de JavaScript

El lenguaje de programación objeto de este módulo es JavaScript.

JScript es la implementación particular de Microsoft, problemas de incompatibilidades con múltiples navegadores. W3C diseño el DOM para evitar esto. JScript se ha integrado en la plataforma de desarrollo .NET de Microsoft.

A partir de 1997 se rigen por un estándar ECMA. En el documento **ECMA**-**262** es dónde se detallan dichas especificaciones. JavaScript y JScript son compatibles con el estándar ECMA-262.

JavaScript sintaxis similar al lenguaje C. No tiene relación con Java ya que tienen semánticas y propósitos diferentes.

JavaScript fue desarrollado por **Brenda Eich**, lo llamo Mocha >> LiveScript >> JavaScript.

JS es uno de los lenguajes más extendidos y usados en la Web tanto en el desarrollo cliente como servidor que permiten desarrollar aplicaciones altamente acopladas con el cliente lo que quiere a éstas un altísimo grado de interactividad.

## 2.- Comentarios en el código.

A la hora de programar en cualquier lenguaje es muy importante incluir comentarios en el código para describir aspectos y facilitar el trabajo.

Los comentarios son sentencias que el intérprete de JavaScript ignora. Permite a los desarrolladores dejar notas cómo funcionan las cosas en sus programas.

Muy recomendable el que se documente el código siempre con límites.

JavaScript permite dos estilos de comentarios.

**//** este es un comentario de una línea

Para comentarios más largos /\* \*/

Funcionalidad que también tienen los comentarios:

* Desactivar el código para que deje de ejecutarlo sin tener que borrar nada.
* Dejar información de contacto para que otros programadores contacten contigo.

3.- Constantes y variables

En JavaScript a estos símbolos se les puede asignar un valor una única vez (constantes) o más de una (variables).

No es necesario declarar las variables antes de asignarles un valor.

Palabra reservada **const**.

const cajas = 3;

Dos palabras reservadas para declarar variables **var** y **let**:

**Var**, permite declarar la variable en su contexto de ejecución.

**Let**, permite declarar la variable limitado a su declaración al bloque, declaración o expresión en el se va a utilizar.

Las variables se pueden declarar y luego asignar o hacer la declaración y asignación junta.

Var edad = 38;

Se puede omitir la palabra **var** sin embargo se recomiendo su uso.

Var altura, peso, edad;

Una variable de JavaScript podrá almacenar diferentes tipos de valores. No tienen tipo.

Dar nombres a las variables:

* no podremos usar palabras reservadas.
* ni símbolos de puntuación en el medio de la variable
* ni la variable podrá contener espacios en blanco.
* Se han de construir con caracteres alfanuméricos y el carácter subrayado (\_).
* No podremos utilizar caracteres raros como el signo +, un espacio, %, $, etc.
* En los nombres de variable y estos nombres no podrán comenzar con un número.
* Dos palabras se separaran con el símbolo “\_” o mayúsculas.

## 4.- Tipos de datos.

Tres tipos de datos:

* Booleano
* Numérico
* Cadena de caracteres

Palabra reservada **null** permite asignar un valor a una variable cuando todavía se desconoce su tipo o tiene un valor inválido.

El valor **undefined** se puede asignar a una variable cuando todavía no conocemos su valor.

## 4.1.- Conversiones de tipos de datos.

Aunque los tipos de datos en JS son muy sencillos, a veces las operaciones no se realizan correctamente y esto es debido a la conversión de tipos. JS lo realiza de la mejor forma y automática.

La expresión se evalúa de izquierda a derecha.

Para convertir cadenas a números dispones de las funciones: **parseInt() y parseFloat().**

La conversión de números a cadenas, ya que simplemente se concatenará una cadena vacía al principio.

Los **paréntesis** fuerzan la conversión del número a una cadena.

JavaScript tiene una propiedad asociada con ella que es la longitud **length**.

## 5.- Operadores.

JS es un lenguaje rico en operadores: símbolos y palabras que realizan operaciones sobre uno o varios valores, para obtener un nuevo valor.

Cualquier valor se denomina operando. Y una expresión puede contener un operando y un operador o bien dos operandos separados por un operador.

Categorías de operadores en JavaScript:

* Comparación: compara los valores de 2 operandos, devolviendo un resultado de true o false. **== != === !== > >= < <=**
* Aritméticos: unen dos operandos para producir un único valor. **+ - \* / % ++ — +valor –valor**
* Asignación: asigna el valor a la derecha de la expresión a la variable que está a la izquierda. **= += -= \*= /= %= <<= >= >>= >>>= &= |= ^= []**
* Booelan: operaciones booleanas aritméticas sobre uno o dos operandos. **&& || !**
* Bit a bit: realizan operaciones aritméticas o de desplazamiento de columna. **& | ^ ~ << >> >>>**
* Objeto: ayudan a los scripts a evaluar la herencia y capacidades de un objeto. **[] . () delete in instanceOf new this**
* Misceláneos: operadores que tienen un comportamiento especial. **, ?: typeof void**