



Instituto de Enseñanza Secundaria
Santiago Hernández

DESARROLLO WEB EN ENTORNO CLIENTE

**UD2. INTRODUCCIÓN A JAVASCRIPT.
PARTE 1
SINTAXIS BÁSICA**

CARACTERÍSTICAS DE JAVASCRIPT

Lenguaje:

- Interpretado (el intérprete traduce a lenguaje máquina, no se compila)
- Dialecto del estándar ECMAScript con características adicionales (ECMA International: ONG dedicada a estandarizar el uso de las TIC y los dispositivos electrónicos)
- Orientado a objetos (cada objeto ofrece una funcionalidad para manipular los datos de entrada y obtener datos de salida específicos)
- Basado en prototipos (sin clases, los objetos no se instancian)
- Imperativo (ejecuta una instrucción detrás de otra)
- Débilmente tipado (las variables no se definen con tipos)
- Dinámico (en tiempo de ejecución se puede cambiar el tipo de una variable o los métodos de un objeto)

CARACTERÍSTICAS DE JAVASCRIPT

Peculiaridades del lenguaje:

- Case sensitive (distingue entre mayúsculas y minúsculas)
- Obvia los espacios en blanco de más y las nuevas líneas: permite tabular y ordenar como queramos
- Es recomendable terminar cada sentencia con “;”, pero no obligatorio
- Comentarios:
 - // todo lo que sigue a la doble barra en una línea será un comentario
 - /* para hacer comentarios de más de una línea */
- Aunque la sintaxis básica es similar a Java o C, no olvidar que no hay ninguna relación entre estos lenguajes.

JAVASCRIPT Y HTML

Como incluir JS en HTML:

- **Embebido:** usando la etiqueta `<script>` dentro de `<body>` o `<head>`:
 - HTML5 utiliza por defecto JavaScript
 - Recomendación: en head (funciones) o al final de body (imperativo)
- Usando la etiqueta `<script>` para referenciar un **archivo externo** de extensión .js
 - `<script src="script.js"></script>`
 - Es la opción más profesional.
- **En línea:** incrustando el código JS dentro de un elemento HTML:
 - `<p onclick="alert('Me han clicado')">TEXTO PARA HACER CLICK</p>`

JAVASCRIPT EMBEBIDO

La importancia del orden

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  <meta charset="utf-8">
5  <title>Este texto es el título del documento</title>
6  <script>
7  function mostraralerta(){
8  alert('hizo clic!');
9  }
10 function hacerclic(){
11 document.getElementsByTagName('p')[0].onclick=mostraralerta;
12 }
13 window.onload=hacerclic;
14 </script>
15 </head>
16 <body>
17 <div id="principal">
18 <p>Hacer Clic</p>
19 <p>No puede hacer Clic</p>
20 </div>
```

SINTAXIS BÁSICA

Palabras reservadas

- No pueden usarse como variables, funciones, métodos o identificadores de objetos.
 - https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Palabras_Reservadas

Variables (ES6)

- Tres formas de definir variables
 - **var**: variable global
 - **let**: variable local al bloque donde se define. Es necesario inicializarla
 - **const**: variable constante, su valor no puede ser alterado en todo el código.
 - Se recomienda nombrarlas con letras mayúsculas para mayor claridad del código.
 - Aplicado a un objeto, el objeto será constante pero no así sus propiedades.
- No se define el tipo, que puede cambiar a lo largo de la ejecución.
- Se recomienda el uso de let salvo que su ámbito restringido nos haga usar var y cuando debamos ajustarnos al estándar ES5 o anteriores.

SINTAXIS BÁSICA

Tipos de datos primitivos:

- **Booleanos:** true, false
- **Números:** enteros o en coma flotante (la coma se escribe con punto). Sin comillas. Todos los números son coma flotante de doble precisión (64 bits).
- **Cadenas de caracteres.** Entre cualquier tipo de comillas.
- **Symbol** (ES6).
- **Undefined:** cuando se ha declarado una variable pero todavía no se ha asignado un valor ni un tipo de dato.
- **Null:** cuando no se ha definido, no existe ningún objeto.

Tipos de datos no primitivos:

- **Objeto:** colección de parejas (nombre, valor)
 - **Array:** objeto de alto nivel

OPERADORES DE JAVASCRIPT

Asignación

- **"="** Asigna un valor a una variable
- Puede utilizarse al declarar la variable o en cualquier otro momento.

Operadores aritméticos

- | | |
|-----------------------------|----------------------------|
| ➤ "+" Suma | "-" Resta |
| ➤ "++" Incrementar 1 | "--" Disminuir 1 |
| ➤ "*" Multiplicación | "/" División |
| ➤ "%" Resto | "**" Exponenciación |
- Objeto Math: valor absoluto, funciones senoidales, logaritmos, raíces... y lo que lo queramos extender.

OPERADORES DE JAVASCRIPT

Operadores lógicos

- “!” Negación && And || Or
 - “>” Mayor que “<” Menor que
 - “>=” Mayor o igual que “<=” Menor o igual que
 - “==” Igual a “!=” Distinto a
 - “===” Igualdad estricta (mismo tipo primitivo)
-
- Para números y para cadenas.

ESTRUCTURAS DE CONTROL Y REPETICIÓN

Estructuras de decisión

```
if (condición 1) {  
    instrucciones para cuando se cumpla la condición 1;  
}else if (condición 2){  
    instrucciones opcionales para cuando no se cumpla la condición 1 y sí la 2;  
}else {  
    instrucciones opcionales para cuando no se cumpla ninguna condición;  
}
```

ESTRUCTURAS DE CONTROL Y REPETICIÓN

Estructuras de decisión

```
switch (condición) {  
    case valor1:  
        instrucciones para cuando condición === valor;  
        break; //sale del switch sin evaluar el resto de posibles valores  
    case valor2:  
    case valor3:  
        instrucciones para cuando condición === valor2 || condición === valor3;  
        break;  
    default:  
        instrucciones opcionales para cuando condición no coincida con ningún valor;  
        break;  
}
```

Nota: sólo permite evaluar valores concretos, no rangos.

ESTRUCTURAS DE CONTROL Y REPETICIÓN

Estructuras de repetición

```
while (condición) {  
    instrucciones a ejecutar mientras condición === true;  
}
```

```
do {  
    instrucciones a ejecutar mientras condición === true;  
}while (condición)
```

ESTRUCTURAS DE CONTROL Y REPETICIÓN

Estructuras de repetición

```
for (inicialización;condición;incremento) {  
    instrucciones a ejecutar en cada iteración;  
}
```

var i=0; //Variación inicializando fuera:

```
for (;condición;incremento) {  
    instrucciones a ejecutar en cada iteración;  
}
```

```
for (inicialización;;) {  
    if (condición) break; //condición dentro  
    instrucciones a ejecutar en cada iteración;  
    i++; //incremento dentro  
}
```

ESTRUCTURAS DE CONTROL Y REPETICIÓN

Estructuras de repetición variantes de for

for (contador **in** objeto) {

instrucciones a ejecutar mientras se recorre con i las claves de un objeto;

}

for (**var** contador **of** lista) {

instrucciones a ejecutar mientras se recorre con i el contenido de una lista;

}

TIPO STRING

Definición

- **var** cadena = “esto es una cadena”;
- **var** cadena2 = **String**(“esto también es una cadena”);

Propiedades

- cadena.**length** : devuelve el número de caracteres de la cadena

Métodos

- **.charAt()**: carácter en la posición especificada.
- **.indexOf()**: ubicación del primer elemento que coincide. Si no existe devuelve -1.
- **.match()**: empareja una expresión regular con una cadena. Devuelve un array con las coincidencias.
- **.split()**: devuelve un array con los fragmentos creados al partir el string con la expresión pasada
- https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/prototype

TIPO NUMBER

Definición

- Sistema decimal: **var** nDec = 2.34;
 - Notación científica: **var** nCient = 1.43**e**7; ($1,43 \cdot 10^7$)
- Sistema octal: **var** nOctal = **0**347; (el 0 a la izquierda lo identifica)
 - Si algún dígito es mayor que 7 lo interpretará como un decimal.
- Sistema binario: **var** nBin = **0b**1100; (también puede usarse **0B**)
 - Si algún dígito no es 0 ó 1 dará error.
- Sistema hexadecimal: **var** nHex = **0x**1100; (también puede usarse **0X**)
 - Si se usa algún símbolo que no corresponde al sistema hexadecimal dará error.

Valores especiales

- Infinity -Infinity -NaN (not a number)

TIPO NUMBER

Objeto Math:

- Objeto incorporado con propiedades y métodos para constantes y funciones matemáticas.
- Constantes:
 - Math.E (constante de Euler)
 - Math.PI
- Métodos:
 - Math.abs(): devuelve el valor absoluto
 - Math.cos(): devuelve el coseno
 - Math.random(): devuelve un número pseudo-aleatorio del intervalo [0, 1).
 - https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math

TIPO NUMBER

Precisión:

- 17 dígitos decimales.
- Fijar nosotros la precisión:
 - `Math.toPrecision()`: permite indicar con cuantos dígitos representamos.
 - `Math.trunc()` / `Math.round()`: devuelven la parte entera.
 - `Math.toFixed()`: devuelve el número con los decimales que fijamos redondeado.

Conversiones:

- `parseInt(string, base)`; utiliza `isNaN()` para devolver NaN caso necesario.
- `parseFloat()`;
- `number.toString()`;
- `Object.valueOf()`;

TIPO ARRAY

Definición:

- **var** lista = ['esto', 'es', 1, 'array'];
- Es un objeto de alto nivel.
- Puede contener cualquier tipo en cualquier elemento.
- lista[0]; accedemos al valor de cada elemento con su índice.
- Propagación: permite separar elementos dentro de un elemento
 - **var** listaLista = ['esto', 'es', 1, ...otraLista, 'en la lista'];

Métodos:

- .sort(): ordena los elementos.
- .join(): devuelve un string con los elementos separados por el parámetro que le pasemos.
- .pop():saca el último elemento . push(): añade elemento(s) y devuelve la longitud.