



JS

UT2. INTRODUCCIÓN A JS

Desarrollo Web en Entorno
Cliente
2ºDAW

1. SINTÁXIS GENERAL

ECMAScript tiene una sintaxis similar a C++ o Java.

Comentarios de una línea `// comment`

Comentarios multilínea `/* comment */`

El punto y coma al final de línea es opcional pero recomendable

No usar palabras reservadas para nombrar variables

El tipado de datos es débil. No hace falta declararlo.

Gramática léxica de JS:

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Lexical_grammar

2. LA CONSOLA DE DEPURACIÓN

Podemos ver la consola en el navegador en la opción “inspeccionar”

Podemos interactuar con el objeto del sistema `console` invocando sus métodos

```
console.log('Hola mundo')           // mensaje general
console.info('Esto es un mensaje'); // mensaje de información
console.warn('Cuidadito')           // mensaje de advertencia
console.error('Error fatal');        //mensaje de error
```

Con la orden `debugger` podemos parar la ejecución del script para depurar

2.1 ALERT, PROMPT Y CONFIRM

Sirven para mostrar alertas del navegador, solicitar un dato y confirmar algo con un dialogo de Aceptar/Cancelar.

Intentaremos evitarlos siempre que podamos. Los usaremos solo ahora al principio, ya que aún no conocemos otra forma de comunicación dinámica con el usuario.

- `alert('Mensaje')`
- `confirm('Mensaje')` // --> devuelve un booleano
- `prompt('mensaje',[default])` // --> devuelve una cadena

3. DECLARACIÓN DE VARIABLES

Podemos declarar de tres formas nuestras variables

- `var` → forma tradicional. Se puede acceder desde cualquier punto del script.
- `let`. → variable que solo es accesible desde el “scope” donde se declara (función o bloque de código)
- `const` → constante que solo es accesible desde el “scope” donde se declara.

```
var num = 5; //variable tradicional. Accesible en todo el script
let cadena = "hola"; //variable accesible solo en su "scope"
const booleano = true; //constante accesible en su "scope"
```

Se recomienda usar `let` o `const` dependiendo de si queremos que el valor varíe o no

4. TIPOS DE DATOS

Tipos de datos primitivos

- Number: Números
- String: Cadena de texto
- Boolean: true / false
- Undefined: no definido
- (*)Null: valor nulo

4.1 NUMBERS

Cualquier positivo o negativo

- Ej. 4, 5, 34, 0, -4, -30

Cualquier número con decimales o sin ellos

- Ej. 34.4 , 0,5 , -9,45 , 150000000

Operaciones aritméticas

- + - * /

Operaciones de comparación (Explicación igualdades)

- < > >= <= == === !=

4.1 OPERACIONES CON NUMBERS

typeof

NaN

isNaN()

toString()

.toFixed()

4.2 STRINGS

Una cadena de texto entre comillas

- “Hello world”
- ‘Hello world’

Operación de concatenar

- cadena = ‘Hola’ + ‘mundo’

Operaciones de comparación de cadenas

- == , === , !=

4.2 OPERACIONES CON STRINGS

Concatenar texto

Template literals

typeof

.length

.includes()

.slice(start,end)

.replace('este texto','por este otro')

.trim()

.split(',')

4.2 EJERCICIO CON STRINGS

Te han pedido que incluyas en el texto de la próxima newsletter unas variables dinámicas que vienen del usuario.

Nombre: Amparo

Interés: Breath of the wild

Descuento: 20% en tu próxima compra

Texto: Hola USUARIO,

Como regalo de bienvenida al Club de Gamers

queremos ofrecerte un descuento del DESCUENTO en el próximo lanzamiento de tu interés JUEGO.

Esperamos que lo disfrutes,

UN saludo,

4.3 BOOLEAN

Solo admite dos valores: `true` o `false`

- Es útil para comprobar estados de la aplicación

Se aconseja que su nombre defina el estado positivo

- Ej: `userIsLogged`

4.3 BOOLEAN

Boolean(valor); nos devuelve el valor booleano de una condición o variable

True (Verdadero)

1

"Whatever"

3.14

100 > 5

1 < 100

'1' == 1

False (Falso)

0, -0

""

NaN

null

undefined

'1' === 1

4.3 BOOLEAN. EJERCICIO PRÁCTICO

Mostrar una información al usuario dependiendo de si está conectado o no.

Variable `USER_LOGGED`

- * Mensaje si no está conectado: Debes registrarte para leer este artículo.
- * Mensaje si está conectado: Haz click aquí para ver el contenido.

El mensaje se ha de mostrar dentro de un párrafo en la página.

4.4 UNDEFINED

Es un tipo de datos que indica **ausencia de valor**

Ej:

```
var a
```

```
console.log(a)
```

```
console.log(typeof a) // → undefined
```

4.5 NULL

Sí que tiene valor pero es null (un objeto que apunta a vacío)

Ej:

```
Var a = null
```

```
console.log (a)
```

```
Console.log (typeof a) // → objetc
```


5. SENTENCIAS DE CONTROL

Sentencias condicionales:

If... [else[if...]]

switch

Operador ternario condicional

5. SENTENCIAS DE CONTROL

Bucles o repeticiones:

for (tradicional bucle que itera con un contador)

for..in (recorre propiedades de un objeto)

for..of (EC6) (recorre una colección de objetos)

While (tradicional bucle mientras);

Nota: Diferencias entre for..in y for..of