
UD3 Objetos predefinidos

2º CFGS DAW

Desarrollo Web en Entorno Cliente

2019-20

1 – Introducción

Como en la mayoría de lenguajes de programación, en JavaScript se incorporan librerías de uso común para facilitar labores a los programadores.

Estas librerías constan de funciones y objetos predefinidos.

Mediante las funciones y los objetos predefinidos los programadores disponen de funcionalidades ya programadas pudiéndose centrar en la lógica de negocio de la aplicación.

2 – Funciones predefinidas

Tratamiento numérico

En la unidad anterior ya se utilizaron dos funciones predefinidas:

- parseInt: convierte una cadena de texto a número entero.
- parseFloat: convierte una cadena de texto a número decimal.

En esta unidad se añade la función:

- isNaN: comprueba si una cadena se puede considerar como un número, devuelve un booleano. NaN = Not a Number.

2 – Funciones predefinidas

Tratamiento numérico

Script que pide un número asegurándose que será un número:

```
do {  
    var numero = prompt("Esto se repetirá hasta que metas un numero");  
} while(isNaN(numero));
```

2 – Funciones predefinidas

Función eval

La función **eval** evalúa una cadena de texto como si fuera código JavaScript.

También puede evaluar expresiones numéricas devolviendo el resultado.

Con esta función se puede construir código de manera dinámica.

Solo se aconseja su uso en situaciones muy concretas ya que la cadena a evaluar puede contener código malicioso.

La función **eval** es muy útil para procesar **JSON** como se verá más adelante.

2 – Funciones predefinidas

Función eval

```
var operacion = prompt("Introduce una operación matemática");  
// el usuario introduce por ejemplo: 7*13  
alert(eval(operacion));
```

```
var x = 3;  
var y = 2;  
var a = eval("2+3");  
var b=eval("x*y");  
eval("alert('a vale '+ a +', b vale '+b) ");
```

Actividades

Actividad 1

Usando la función **eval**.

3 – Objetos predefinidos

JavaScript permite la programación orientada a objetos.

Además de poder crear objetos propios como se verá en la unidad 7, JavaScript ofrece unos objetos predefinidos.

En esta unidad se verán los objetos:

- String
- Date
- Array
- Math
- Window

4 – String

Toda cadena en JavaScript se convierte a objeto **String** de manera implícita en su creación.

De esta manera, automáticamente se dispone de las propiedades y funciones propias del objeto String.

Los métodos de **String** en **JavaScript** no modifican el objeto original, lo que hacen es devolver el resultado obtenido.

```
cadena = cadena.metodoFicticio();    // así se modificaría el String
```

4 – String

Documentación: https://www.w3schools.com/js/js_string_methods.asp

El objeto **String** tiene una propiedad **length** que devuelve los caracteres que tiene la cadena.

Además tiene una serie de métodos que se pueden consultar en la documentación.

4 – String

Método	Descripción
toLowerCase()	Devuelve la cadena en minúsculas.
toUpperCase()	Devuelve la cadena en mayúsculas.
concat(cadena)	Devuelve el objeto concatenando la cadena pasada por parámetro.
charAt(posición)	Devuelve el carácter que se encuentre en la posición indicada.
indexOf(texto [, índice])	Devuelve la primera posición de texto . Con índice se indica que empiece a buscar desde esa posición, por defecto es cero .
lastIndexOf(texto [, índice])	Devuelve la última posición de texto . Con índice se indica que empiece a buscar desde esa posición hacia delante, por defecto es cero .
replace(texto1, texto2)	Devuelve el objeto sustituyendo texto1 por texto2 .
split(carácter [, trozos])	Separa la cadena cada vez que encuentra carácter , devolviendo un array con los trozos separados. El parámetro trozos indica el máximo de elementos que tendrá el array.
substring(inicio [, fin])	Devuelve la subcadena comprendida entre las posiciones inicio y fin . Si no se indica el parámetro fin se toma hasta el final de la cadena.

4 – String

```
var cad = "Álex:Torres:654654654:alex@mail.com";  
var telefono;
```

```
cad = cad.toUpperCase();  
alert(cad);
```

```
splitTodosCampos = cad.split(":");  
split1Campo = cad.split(":")  
alert(splitTodosCampos);  
alert(split1Campo);
```

```
telefono = splitTodosCampos[2];  
// Se cambia en el teléfono los números 5 por 0  
telefono = telefono.replace("5", "0");  
alert(telefono);
```

```
// Se muestra el quinto número del teléfono  
alert(telefono.charAt(4));
```

5 – Date

Documentación: https://www.w3schools.com/js/js_dates.asp

El objeto **Date** permite trabajar con fechas.

El constructor para crear una fecha admite muchos formatos.

Nota: los meses se numeran de 0 (enero) a 11 (diciembre).

5 – Date

```
// Crea una fecha con la fecha y hora del sistema  
var fecha = new Date();
```

```
// Crea una fecha basándose en la cadena de fecha especificada  
fecha = new Date("October 13, 2014 11:13:00");
```

```
// Indicando año, mes, día, horas, minutos, segundos milisegundos  
fecha = new Date(99,5,24,11,33,30,0);
```

```
// Fecha indicando año, mes, día.  
fecha = new Date(99,5,24);
```

5 – Date

Método	Descripción
Setters y getters: getMinutes() getMonth() setDate() setHours()	Permiten consultar y cambiar alguna propiedad de la fecha: https://www.w3schools.com/js/js_date_methods.asp Obtiene los minutos. Obtiene el mes como número 0-11. https://www.w3schools.com/js/js_date_methods_set.asp Cambia el número del día de la fecha 1-31. Cambia la hora de la fecha 0-23.
toDateString()	Convierte la fecha en una cadena.
toGMTString()	Convierte la fecha en una cadena en formato GMT.
toUTCString()	Convierte la fecha en una cadena en formatoUTC.

5 – Date

```
var d = new Date();
```

```
alert(d.getDay());
```

```
d.setDate(1);
```

```
alert(d.getDay());
```


Actividades

Actividad 2

Trabajando con cadenas.

Actividad 3

Usando el objeto **Date**.

6 – Array

Documentación: https://www.w3schools.com/js/js_arrays.asp

Un **array** es una colección de elementos.

Los elementos que se almacenan en un **array** pueden ser **de cualquier tipo**.

Todo lo que se vio sobre arrays en la unidad anterior es aplicable a los objetos tipo **array**.

Por regla general los métodos de array **sí modifican** el objeto original.

6 – Array

```
var x = [1,2,3];
```

```
// Se añade un elemento al vector
```

```
x[3] = 4;
```

```
var coches = ["Seat", "Ford", "Tesla", "Audi"];
```

```
// usando la palabra reservada new
```

```
var coches = new Array("Seat", "Ford", "Tesla", "Audi");
```

```
coches[4] = "Peugeot";
```

6 – Array

Método	Descripción
join([separador])	Devuelve en una cadena todos los elementos del array concatenados con el texto separador .
push(elemento1 [, elemento2 [, ...]])	Añade al final del array los parámetros que se introduzcan en la función.
pop()	Devuelve y elimina el último elemento del array.
reverse()	Invierte el orden de los elementos del array.
sort()	Ordena alfabéticamente los elementos del array.
slice(inicio [,final])	Devuelve los elementos del array comprendidos entre los índices inicio y final . Si no se indica final se devolverá hasta el último elemento del array.

6 – Array

```
var x = [1,2,3];  
var y = [2,3,10,2];  
// Se invierte el vector  
x.reverse();  
alert(x);  
// Se ordena (alfabéticamente)  
y.sort()  
alert(y);  
// Se elimina el último elemento  
var sacado = y.pop();  
alert(sacado);  
alert(y);
```

7 – Math

Documentación: https://www.w3schools.com/js/js_math.asp

El objeto **Math** dispone de métodos que ayudan en las tareas matemáticas.

Dispone de constantes y de métodos.

7 – Math

Constantes:

Constante	Descripción
E	Número de Euler.
PI	Número Pi.
LN2	Logaritmo neperiano de 2.
LN10	Logaritmo neperiano de 10.
SQRT2	Raíz cuadrada de 2.
SQRT1_2	Raíz cuadrada de ½.

7 – Math

Métodos:

Método	Descripción
<code>floor(número)</code>	Redondea hacia el siguiente entero menor.
<code>ceil(número)</code>	Redondea hacia el siguiente entero mayor.
<code>round(número)</code>	Redondea hacia el siguiente entero más cercano.
<code>abs(número)</code>	Devuelve el número en valor absoluto (sin signo).
<code>max(h, i [, j [, k [, ...]]])</code>	Devuelve el mayor número de entre todos los parámetros.
<code>min(h, i [, j [, k [, ...]]])</code>	Devuelve el menor número de entre todos los parámetros.
<code>pow(x, y)</code>	Devuelve la operación x^y .
<code>random()</code>	Devuelve un número aleatorio decimal entre 0 y 1.
<code>sqrt(número)</code>	Devuelve la raíz cuadrada de número .

7 – Math

```
// Se obtiene un entero entre 1 y 11  
var x = parseInt( (Math.random()*10)+1 )
```

```
// Se redondea hacia abajo  
var y = Math.floor(11.5);
```

```
// Uso de la constante PI  
alert("Perímetro de círculo de radio 5 = "+ 2*Math.PI*5);
```

Actividades

Actividad 4
Arrays.

Actividad 5
Calculando.

8 – BOM (Browser Object Model)

El BOM permite que JavaScript se pueda “comunicar” con el navegador.

No existe un estándar oficial para el BOM aunque en las versiones modernas de los navegadores los desarrolladores han implementado los mismos métodos y propiedades.

Los elementos del BOM son: Window, Screen, Location y History.

En la unidad anterior ya se vieron métodos del objeto Window:
alert, confirm y prompt.

8 – BOM (Browser Object Model)

Window:

Métodos	Descripción
setTimeout(función([parámetros]), tiempo)	<p>Realiza la llamada a función pasados los milisegundos indicados en tiempo.</p> <p>Devuelve el identificador del Timeout para poder cancelarlo.</p> <p>Se ejecuta función solo una vez.</p>
setInterval(función([parámetros]), tiempo)	<p>Realiza una llamada a función cada vez que pasan los milisegundos indicados en tiempo.</p> <p>Devuelve el identificador del Interval para poder cancelarlo.</p> <p>Se ejecuta repetidamente función hasta que se cancele</p>
clearTimeout(id)	Anula el Timeout correspondiente a id .
clearInterval(id)	Anula el Interval correspondiente a id .

8 – BOM (Browser Object Model)

```
// Intervalo cada 15 segundos  
var idIntervalo15s = setInterval("alert('hola');", 15000);  
  
// Timeout para dentro de 3 segundos  
var idIntervalo3s = setTimeout("alert('adios'); ", 3000);  
  
// Timeout para dentro de 5 segundos  
var idIntervalo5s = setTimeout("alert('OLA K ASE'); ", 5000);  
  
// Se cancela el último timeout  
clearTimeout(idIntervalo5s);
```

Actividades

Actividad 6
Cuenta atrás.