

Desarrollo Web en Entorno Cliente

UD 04. Objetos predefinidos

Parte 2

Actualizado Octubre 2023

Licencia



Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE DE CONTENIDO

1. Objeto predefinido Math	3
2. Objeto predefinido Window	3

1. OBJETO PREDEFINIDO MATH

El objeto Math es un objeto que contiene una colección métodos que nos ayudarán a trabajar realizar operaciones aritméticas, redondeos, etc.

Algunas de las constantes matemáticas predefinidas más importantes son:

- **E**: almacena el número de Euler.
- **PI**: almacena el número Pi.
- **LN2 / LN10**: logaritmo neperiano de 2 / 10.
- **LOG2E / LOG10E**: logaritmo en base 2 / 10 del número de Euler.

Algunos de los métodos más importantes que tiene disponibles son:

- **Redondeo**:
 - **floor(numero)**: redondea hacia abajo un número.
 - **ceil(numero)**: redondea hacia arriba un número.
 - **round(numero)**: redondea al entero mas cercano.
- **Operaciones matemáticas**:
 - **abs(numero)**: devuelve el número en valor absoluto.
 - **max / min (x,y)**: devuelve el mayor / menor de dos números x e y.
 - **pow(x,y)**: devuelve x elevado a y.
 - **random()**: devuelve un número aleatorio con decimales entre 0 y 1.
 - **sqrt(numero)**: devuelve la raíz cuadrada del número indicado.

Ejemplo:

```
// Obtenemos un entero entre 1 y 11
let x=parseInt( (Math.random()*10)+1 )
// Redondeamos y hacia abajo
let y=Math.floor(11.5);
alert(Math.PI);
```

2. OBJETO PREDEFINIDO WINDOW

El objeto Window es un objeto que solo está presente al trabajar con Javascript en navegadores, no estando presente en entornos como NodeJS.

El motivo de que este objeto sólo exista en navegadores es que fuera de ellos carece de sentido, ya que posee propiedades y controla elementos relacionado con lo que ocurre en la “ventana” del navegador.

Los métodos que estudiamos en el tema anterior cómo alert, prompt, etc. forman parte del objeto Window. Para hacer llamada a estos métodos no hace falta nombrar explícitamente Window (el navegador ya se encarga de ello).

Algunos de los métodos más importantes no estudiados previamente son:

- **setTimeout(cadenaFuncion, tiempo):** este método ejecuta la llamada a la función proporcionada por la cadena (se puede construir una cadena que lleve parámetros) y la ejecuta pasados los milisegundos que hay en la variable tiempo. Devuelve un identificador del "setTimeout" que nos servirá para referenciar lo si deseamos cancelarlo. SetTimeout solo ejecuta la orden una vez.
- **setInterval(cadenaFunción, tiempo):** exactamente igual que setTimeout, con la salvedad de que no se ejecuta una vez, sino que se repite cíclicamente cada vez que pasa el tiempo proporcionado.
- **clearTimeout / clearInterval (id):** se le pasa el identificador del timeout/interval y lo anula.

Ejemplo:

```
// Creamos un intervalo que cada 15 segundos muestra mensaje hola
let idA=setInterval("alert('hola');",15000);
// Creamos un timeout que cuando pasen 3 segundos muestra mensaje adios
let idB=setTimeout("alert('adios');",3000);
// Creamos un timeout que cuando pasen 5 segundos muestra mensaje
estonoseve
let idC=setTimeout("alert('estonoseve');",5000);
// Cancelamos el último timeout
clearTimeout(idC);
```