Resumen JavaScript Tema 2

MÉTODOS

|  |  |
| --- | --- |
| SIGNATURA MÉTODO | DESCRIPCIÓN – UTILIDAD |
| 1. prompt( ); 2. parseInt(prompt( )); | 1. Pide al usuario que introduzca algo por teclado. 2. Obliga al usuario a introducir un número, si no recibe un número da error |
| typeof (no es un método) | Devuelve el tipo de dato de la variable que se encuentra a continuación |

TIPOS DE DATOS

NUMBER

|  |  |
| --- | --- |
| SIGNATURA MÉTODO | DESCRIPCIÓN – UTILIDAD |
| isNaN(…) | Devuelve true, si el dato pasado por parámetro NO es un número, |
| typeof (no es un método) | Devuelve el tipo de dato de la variable que se encuentra a continuación |
| 1. toString(16) 2. toString(8) 3. toString(2) | 1. Convierte el número a base 16 y lo devuelve 2. Convierte el número a base 8 y lo devuelve 3. Convierte el número a base 2 y lo devuelve |
| toExponential(…) | Devuelve el número en notación exponencial redondeado a la cifra pasada como parámetro. |
| toFixed(…) | Devuelve el número redondeando las cifras decimales al número pasado como parámetro  X = 9.687  x.toFixed(0) = 10  x.tofixed(2) = 9.69 |

STRING

|  |  |
| --- | --- |
| SIGNATURA MÉTODO | DESCRIPCIÓN – UTILIDAD |
| length | Longitud de una cadena |
| charAt(…) | Devuelve el carácter que se encuentra en la posición indicada como parámetro |
| concat(…) | Se utiliza para combinar dos o más cadenas. Este método no cambia las cadenas existentes, sino que retorna una nueva cadena. |
| * indexOf(…) * lastIndexOf(…) | * Devuelve la posición de la primera coincidencia del valor pasado como parámetro * Como el anterior, pero devuelve la posición de la última coincidencia. |
| match(…) | Se usa para obtener todas las ocurrencias de una expresión regular dentro de una cadena. |
| toUpperCase() | Devuelve el valor convertido en mayúsculas de la cadena que realiza la llamada. |
| toLowerCase() | Devuelve el valor convertido en minúsculas de la cadena que realiza la llamada. |
| cadena.split([separador][,limite]) | Divide un objeto de tipo String en un array de cadenas mediante la separación de la cadena en subcadenas. |
| cadena.slice(inicioTrozo[, finTrozo]) | Extrae una sección de una cadena y devuelve una cadena nueva. |

**En ocasiones podemos convertir una variable de un tipo a otro (typecasting)**

* **toString()**: permite convertir variables de cualquier tipo a variables de cadena de texto.

Ejemplo:

var variable1 = true;

variable1.toString(); // devuelve "true" como cadena.

* **String():** convierte el valor pasado en string
* **Boolean():** convierte el valor pasado en un booleano
* **parseInt()** : convierte la variable que se le indica en un número entero
* **parseFloat(),** que convierte la variable que se le indica en un número decimal

OPERADORES

* **Operadores condicionales (?:)**

+ Permite indicar al navegador que ejecute una acción en concreto después de evaluar una expresión.

+ Si la expresión antes del operador es verdadera, se utiliza el primer valor a la derecha. En caso contrario se utiliza el segundo valor a la derecha.

(divisor != 0) ? alert(dividendo/divisor) : alert("No es posible la división por cero");