



Carmelo José Jaén Díaz



Objetos nativos. Boolean

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Nov
 - iPasen - cjaedia071@g.educaand.es

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Aprender cómo se puede manejar el tiempo en JavaScript.
- Saber cómo se programa la ejecución de funciones de forma puntual y periódica en el tiempo.
- Registrar en el front-end información de navegación, usuario, etc., mediante cookies o mediante el almacenamiento local que ofrece el nuevo estándar HTML5.
- Profundizar en el concepto de objeto.
- Conocer y manejar funciones relativas al lenguaje sobre arrays, strings, números.

GLOSARIO



Backtracking. Estrategia utilizada en algoritmos que resuelven problemas que tienen ciertas restricciones. Este término fue creado por primera vez por el matemático D. H. Lehmer en la década de los cincuenta.

BOM (Browser Object Model). Convención específica implementada por los navegadores para que JavaScript pudiese hacer uso de sus métodos y propiedades de forma uniforme.

Expresión regular. Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

NaN. Propiedad que indica Not a Number (valor no numérico).

Objeto window. Aquel que soportan todos los navegadores y que representa la ventana del navegador. Se estudiará en profundidad en capítulos posteriores.

URI (Uniform Resource Identifier). Cadena de caracteres que identifica un recurso en una red de forma unívoca. Una URI puede ser una URL, una URN o ambas.

GLOSARIO



URN. Localizador de recursos en la web que funciona de forma parecida a una URL, pero su principal diferencia es que no indica exactamente dónde se encuentra dicho objeto.

INTRODUCCIÓN



Los **datos primitivos de tipo boolean** permiten almacenar un valor verdadero (**true**) o falso (**false**). También existen los **objetos nativos Boolean**, que a diferencia de los anteriores, y como dice la palabra, son objetos.

A continuación, veremos en qué consiste la **función Boolean** y qué podemos hacer con ella.

Además, distinguiremos qué **valores son considerados ciertos** en este tipo de objetos y cuáles, por el contrario, son **falsos**. ¡Ojo, que hay más de los que crees!

INTRODUCCIÓN



¡Recuerda! El valor por defecto de un objeto nativo Boolean siempre es false. Y undefined, NaN, O... 🤔

FUNCIÓN *Boolean*



El objeto *Boolean* tiene una función asociada que al pasarle como argumento una variable o expresión, nos devuelve si esta es cierta o falsa.

EJEMPLO DE IMPLEMENTACIÓN

```
alert (Boolean(10>9)); //Devuelve true
```

En JavaScript existe una particularidad del método *alert* y es que al pasarle una expresión como argumento la va a evaluar mediante la función *Boolean*, por lo tanto, la expresión siguiente es análoga a la anterior.

```
alert (10>9); //Devuelve true
```


FUNCIÓN *Boolean*



La función *Boolean* nos permite saber si unos valores son ciertos o falsos.

Hay que tener en cuenta que será cierto todo valor que es real.

EJEMPLO DE IMPLEMENTACIÓN

```
alert("100 es "+Boolean(100));  
alert("3.4 es "+Boolean(3.4));  
alert("Hola es "+Boolean("Hola"));  
alert("false "+Boolean("false")); //false entre comillas se trata como una cadena  
alert("-15 es "+Boolean(-15));  
alert("5<6 es "+Boolean(5<6));
```

FUNCIÓN *Boolean*



La función *Boolean* nos permite saber si unos valores son ciertos o falsos.

Por contra, todos los valores que no sean reales serán falsos.

EJEMPLO DE IMPLEMENTACIÓN

```
alert("0 es "+Boolean(0));  
alert("-0 es "+Boolean(-0));  
alert("undefined es "+Boolean(undefined));  
alert("null es "+Boolean(null));  
alert("false es "+Boolean(false));  
alert("NaN es "+Boolean(NaN));
```

EJEMPLO DE USO



De lo anterior, se deduce que al evaluar esa expresión en un *if*.

```
let g;  
  if (!g) // Es equivalente a g==undefined  
  {  
    alert ("La g no está definida");  
  }
```

MÉTODOS DE *Boolean*



Podemos encontrar todos los métodos de *Boolean* en [W3Schools.com>JavaScript>JS References](#).

Como se puede comprobar solo tiene dos métodos:

- `toString()`: convierte un valor booleano en una cadena.
- `valueOf()`: devuelve el valor primitivo de un objeto booleano.

Aunque su uso resulta de vital importancia en las comparaciones.