



# Introducción a los objetos en Javascript

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

Carmelo José Jaén Díaz

# 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

---



En esta lección vamos a hacer una pequeña introducción a todos los tipos de datos que nos podemos encontrar en este lenguaje de programación. Y por ello, es importante diferenciar los datos primitivos de los objetos en Javascript.

Los datos primitivos hasta la versión ES5 eran cinco:

- Cadena.
- Número.
- Booleano.
- Nulo o null.
- No definido o undefined.

En la versión ES6 se incorpora un nuevo dato primitivo: symbol, el cual trataremos más adelante.

# INTRODUCCIÓN

---



Pero lo que nos incumbe en esta lección son los objetos, el otro tipo de datos que debemos sumar a los seis anteriores.

En programación, un objeto es un valor en memoria al que podemos referirnos mediante un identificador.

Podemos hablar de dos tipos de objetos en Javascript:

- Los objetos nativos, que no dependen del navegador.
- Los objetos de alto nivel, que dependen del navegador.

Por último, se tratarán los objetos definidos por el usuario.

# DATOS PRIMITIVOS

---



Como se ha introducido anteriormente los datos primitivos son los números, cadenas de caracteres, booleanos, valores *null* y *undefined*, con los que hemos trabajado en el RA2.

Por ejemplo:

`let cadena = "Carmelo";` *//Dato primitivo de tipo cadena de caracteres. En ningún momento se determina que es de tipo cadena, al asignarle mediante el operador = un dato entre comillas, la variable se define como tipo cadena.*

Esto se aplica a los cinco tipos de datos primitivos comentados anteriormente.



# OBJETOS NATIVOS

---



Son aquellos objetos que **NO DEPENDEN** del navegador. En JS son los siguientes:

- String
- Number
- Boolean
- Data
- Math
- RegExp (Expresiones regulares)
- Array
- Function
- Object

**NOTA:** Los objetos *String*, *Number* y *Boolean* nos permiten crear cadenas, números o booleanos pero no es recomendable su uso ya que su implementación mediante datos primitivos es mucho más rápida y eficaz

# OBJETOS DE ALTO NIVEL

---



Son aquellos objetos que **DEPENDEN** del navegador. En JS son los siguientes:

- Window
- Screen
- Navigator
- Location
- History
- Document

Este último es un objeto especial, puesto que se corresponde con el **modelo de objetos del documento (DOM)**.

Nos permite trabajar con los elementos HTML que tenemos dentro de nuestro documento.

El resto corresponden al **modelo de objetos del navegador (BOM)**. Nos permiten trabajar con el historial, de qué tipo es el navegador, el tamaño de las ventanas, etc.

# OBJETOS DEFINIDOS POR EL USUARIO

---



Un objeto, si no se hace uso de la palabra reservada `object`, se define como:

```
let docente = {nombre: "Carmelo", apellido: "Jaen", inicio: 2020};
```

Dicho objeto tiene:

- **Propiedades:** se definen en modo *clave:valor*. En el ejemplo serían: *nombre*, *apellido* e *inicio*.
- **Métodos:** acciones que se pueden ejecutar sobre un objeto. En el ejemplo no se ha creado ningún método pero se podría implementar el método *nombreCompleto* que devolvería la concatenación de *nombre* y *apellido*, es decir, “*Carmelo Jaen*”.

## RESUMEN

Las variables sólo almacenan un único dato.

Los objetos pueden almacenar uno o más datos con el formato *clave:valor*. Estos pueden ser datos primitivos, otros objetos, o incluso, ser funciones (métodos del objeto).