



Carmelo José Jaén Díaz



Objetos. Clases

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

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- 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



Las clases en Javascript se introdujeron en esta versión (ECMAScript2015 o ES6) y supusieron una mejora significativa en la herencia basada en prototipos de Javascript a la que estábamos acostumbrados. Estas clases nos dan una sintaxis más clara y simple para crear objetos y trabajar con herencia.

A diferencia de la versión anterior de Javascript en la que utilizábamos la palabra reservada **function** para definir una “clase” (entre comillas), en esta versión se utiliza **class**. Esto evita confusiones, ya que realmente los objetos nos son funciones.

Además, las propiedades se deben indicar dentro de un método constructor.

INTRODUCCIÓN



La sintaxis sería la siguiente:

```
class NombreClase {  
    constructor(parametro1 [, parametro2...]) {  
        this.propiedad1 = parametro1;  
        [this.propiedad2 = parametro2;]  
    }  
}
```

Para definir un objeto de la clase, utilizamos la palabra reservada **new**.

```
let|const nombreObjeto = new NombreClase (argumentos);
```

CLASES



Al implementar la versión ES6, el concepto de `clase`, JavaScript se asemeja a un lenguaje orientado a objetos (POO).

Decimos que se asemeja ya que internamente, Javascript traduce estas clases al sistema basado en prototipos que usa en realidad, sin embargo, los desarrolladores no necesitarán saber cómo funcionan los prototipos, sino que les bastará con utilizar las clases a modo de POO.

SINTAXIS DE CREACIÓN DE UNA CLASE



Recuerda: Es una buena práctica utilizar la primera letra en mayúscula para nombrar a las clases.

Para definir una clase en JavaScript utilizamos la siguiente sintaxis. Recuerda que siempre tenemos que definir su *constructor* (si no, se creará uno por defecto).

```
class Docente {  
  constructor (nom, ape) {  
    this.nombre = nom;  
    this.apellido = ape;  
  }  
}
```

SINTAXIS DE CREACIÓN DE UNA CLASE



Para *instanciar* un objeto de la clase **Docente**, es decir, crear un objeto de tipo **Docente**:

```
let docenteSAI = new Docente ("Carmelo", "Jaén");
```

Para acceder y visualizar sus propiedades por consola:

```
console.log(docenteSAI.nombre+" "+docenteSAI.apellido);
```