



4.A. Estructuras definidas por el usuario en JavaScript

3.- Creación de objetos definidos por el usuario.

En unidades anteriores has visto, como toda la información que proviene del navegador o del documento está organizada en un modelo de objetos, con propiedades y métodos. Pues bien, JavaScript también te da la oportunidad de crear tus propios objetos en memoria, objetos con propiedades y métodos que tú puedes definir a tu antojo. Estos objetos no serán elementos de la página de interfaz de usuario, pero sí que serán objetos que podrán contener **datos (propiedades) y funciones (métodos)**, cuyos resultados sí que se podrán mostrar en el navegador. El definir tus propios objetos, te permitirá enlazar a cualquier número de propiedades o métodos que tú hayas creado para ese objeto. Es decir, tú controlarás la estructura del objeto, sus datos y su comportamiento.

También hay que dejar claro que, **JavaScript no es un lenguaje orientado a objetos de verdad en sentido estricto**. Se considera que, JavaScript **es un lenguaje basado en objetos (u orientado a prototipos)**. La diferencia entre orientado a objetos y basado en objetos es significativa, y tiene que ver sobre todo en cómo los objetos se pueden extender.

Un objeto en JavaScript es realmente una colección de propiedades almacenadas en un array asociativo. Las propiedades pueden tener forma de datos, tipos, funciones (métodos) o incluso otros objetos. De hecho sería más fácil de entender un objeto como un array de valores, cada uno de los cuales está asociado a una propiedad (un tipo de datos, método u objeto). Un momento: ¿Un método puede ser una propiedad de un objeto? Pues en JavaScript parece que sí.

Una función contenida en un objeto se conoce como un **método**. Los métodos no son diferentes de las funciones que has visto anteriormente, excepto que han sido diseñados para ser utilizados en el contexto de un objeto, y por lo tanto, tendrán acceso a las propiedades de ese objeto. Esta conexión entre propiedades y métodos es uno de los ejes centrales de la orientación a objetos.

Los objetos se crean empleando una función especial denominada **constructor**, determinada por el nombre del objeto. Ejemplo de una función constructor:

```
function Coche( ) {  
  
    // propiedades y métodos  
  
}
```

Aunque esta función no contiene código, es sin embargo la base para crear objetos de tipo Coche. Puedes pensar en un constructor como un anteproyecto o plantilla, que será utilizada para crear objetos. Por convención, los nombres de los constructores se ponen generalmente con las iniciales de cada palabra en mayúscula, y cuando creamos un objeto con ese constructor (**instancia de ese objeto**), lo haremos empleando minúsculas al principio. Por ejemplo: `var unCoche = new Coche();`

La palabra reservada **new** se emplea para crear objetos en JavaScript. Al crear la variable `unCoche`, técnicamente podríamos decir que hemos creado una instancia de la clase Coche, o que hemos instanciado el objeto Coche, o que hemos creado un objeto Coche, etc. Es decir hay varias formas de expresarlo, pero todas quieren decir lo mismo.

A partir del estándar ECMAScript 2015 aparece la sintaxis de **clases** que permiten definir objetos utilizando expresiones más apropiadas de un lenguaje orientado a objetos (basado en clases).

Para saber más

[Trabajando con objetos](#)

[Clases en JavaScript](#)

Créditos de la imagen.

Autoría: Joost J. Bakker Ijmuiden.

Licencia: CC BY 2.0.

