



M4.250 - Introducción a la programación en JavaScript I aula 1

Objetos

Inicio:

13/05/20

Fin:

09/06/20

Introducción

Los objetos son una estructura de datos común en muchos lenguajes de programación. Nos permiten programar tratando la información de una manera similar a cómo la tratamos en el mundo real.

Así, un coche tiene un conjunto de características (color, marca, modelo, potencia, precio) y podemos crear un objeto que reúna esas características.

Vamos a ver un ejemplo que vamos a ir construyendo sobre la marcha explicando todas las posibilidades. Esta guía está basada en el material de Mozilla sobre objetos (https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Trabajando_con_objetos)

Sin embargo está simplificada y añade ejemplos. Se recomienda usar esta guía para entender cómo funcionan los objetos y pasar al material de Mozilla como ampliación cuando se tengan los conceptos claros.

Nuestro ejemplo se va a basar en el caso de un pequeño concesionario de automóviles. Tan pequeño que solo dispone de 10 coches que vende por internet. Son coches de segunda mano y, por tanto, de ellos nos interesarán los siguientes datos:

- Marca
- Modelo
- Potencia
- Antigüedad
- Precio

A estos datos que guardaremos de cada coche. Cuando hablemos de objetos les llamaremos propiedades .

Creando un objeto

Vamos a empezar creando un primer objeto.

```
var miAuto = new Object();
```

Con esta instrucción ya hemos creado el objeto. Ahora le añadiremos las propiedades:

```
miAuto.marca = "Seat";  
miAuto.modelo = "Ibiza";  
miAuto.potencia = 90;  
miAuto.antiguedad = 2010;  
miAuto.precio = 5000;
```

Probad este código aquí: <https://codepen.io/ccasadam/pen/mYVLZb?editors=0012>

Si por casualidad preguntamos por una propiedad que no hemos definido, la respuesta será `undefined` .

Consultando las propiedades de un objeto

Para acceder al contenido de las propiedades lo podemos hacer de dos maneras:

con la sintaxis de punto

```
console.log(miAuto.precio);
```

O con la notación de corchetes

```
console.log(miAuto["precio"]);
```

La ventaja de la notación de corchetes es que puedes acceder a una propiedad mediante el valor guardado en una variable:

```
var propiedad = "precio";  
console.log(miAuto[propiedad]);  
// Escribirá en la consola el valor de la  
// propiedad precio.
```

Otra ventaja del uso de la notación de corchetes es la posibilidad de usar un `for` para recorrer todos los elementos del objeto. Mira este ejemplo:

```
for (var i in miAuto) { // i irá tomando los  
                        // nombres de todas las  
                        // propiedades que tenga  
                        // miAuto  
    console.log(miAuto[i]);  
    // Escribirá en la consola los valores de todas las propiedades  
}
```

Puedes ver este ejemplo funcionando aquí: <https://codepen.io/ccasadam/pen/zQraKJ?editors=0012>

Inicializadores de objeto

En vez de crear un objeto e ir incorporando después sus propiedades, tenemos la opción de crear un objeto y sus propiedades de una sola vez:

```
var miAuto = { marca : "Seat",  
modelo : "Ibiza",  
potencia : 90,  
antiguedad : 2010,  
precio : 5000 };
```

Función constructora

También podemos usar una función constructora. En este caso definimos primero la función constructora:

```
function Auto(marca, modelo, potencia, antiguedad, precio) {  
    this.marca = marca;  
    this.modelo = modelo;  
    this.potencia = potencia;  
    this.antiguedad = antiguedad;  
    this.precio = precio;  
}
```

El uso de `this` es para indicar que nos referimos a `Auto` (`Auto.marca`). En este caso los nombres de los parámetros coinciden con los nombres de las variables. Se suele hacer así, aunque no es obligatorio.

Para tener nuestro objeto nos queda un paso:

```
var miAuto = new Auto("Seat", "Ibiza", 90, 2010, 5000);
```

Este es el sistema que deberíamos usar habitualmente para crear objetos. Y tiene una explicación.

Este sistema es el que más se asemeja al que usan otros lenguajes de programación cuando quieren crear objetos. Fijaros que a la variable constructora le hemos puesto el nombre con la primera letra en mayúscula (a diferencia de lo que solemos hacer con los nombres de variable que siempre los iniciamos con letra minúscula). Es una convención que se acostumbra a respetar.

En otros lenguajes de programación en vez de crear una función, hubiésemos creado una clase.

Las clases se comportan como un nuevo tipo de datos, así, podemos crear variables de números, de cadenas de caracteres o de Autos...

Ahora puedo crear otra variable de tipo `Auto`:

```
var otroAuto = new Auto("Opel", "Astra", 92, 1995, 500);
```

Desgraciadamente no vamos a poder comparar (con el signo `=`) dos objetos como si fuesen dos variables normales, pero siempre podremos crear una función que compare todas sus propiedades.

Podéis ver un ejemplo que hace eso (crea una función para comparar dos objetos del mismo tipo) aquí:

<https://codepen.io/ccasadam/pen/EzPLrL?editors=0012>

Definiendo los métodos

Los métodos son funciones que están asociadas a una clase (y por tanto a un conjunto de objetos).

Aunque podemos definir los métodos de una clase fuera de ella, nos resultará más fácil de entender si los definimos dentro:

```
function Auto(marca, modelo, potencia, antiguedad, precio) {  
  this.marca = marca;  
  this.modelo = modelo;  
  this.potencia = potencia;  
  this.antiguedad = antiguedad;  
  this.precio = precio;  
  this.edadAprox = function (anio) {  
    return (anio - this.antiguedad);  
  }  
}
```

Aquí hemos definido un método llamado `edadAprox` que nos calcula los años de vida de un vehículo a partir de un año que le pasamos como parámetro. De nuevo, con un ejemplo se ve mejor:

```
var otroAuto = new Auto("Opel", "Astra", 92, 1995, 500);  
console.log(otroAuto.edadAprox(2019));
```

Se puede ver este código funcionando aquí: <https://codepen.io/ccasadam/pen/RmryQd?editors=0012>

Hasta aquí la parte más importante del uso de la orientación a objetos en JavaScript.

Tenéis dos vídeos que pueden ayudaros a entender mejor el funcionamiento aquí:

- https://www.youtube.com/watch?v=JtvE_kpfgDE&list=PLiXW1QK5nFVNgarf4nYUuw4x88dEValf5
- <https://www.youtube.com/watch?v=QowgJdfpJP0&list=PLiXW1QK5nFVNgarf4nYUuw4x88dEValf5&index=3>

Son vídeos hechos para otra asignatura, pero os serán útiles igualmente.

Recursos de aprendizaje

Fuentes de información
