



APUNTES 3.5. OBJETOS EN JAVASCRIPT

“En JavaScript”, prácticamente todo es un objeto”.



3.5. Objetos en JavaScript.

Salvo valores y tipos primitivos como boolean, string, number, bigint, undefined, null, symbol, los demás componentes del lenguaje son objetos de JavaScript.

Aunque se desaconseja, se pueden declarar tipos primitivos como objetos:

```
var a = new String();  
var b = new Number();  
var c = new Boolean();
```



3.5. Objetos en JavaScript.

Por lo tanto, en el siguiente código:

```
var cadena = "Hola";  
var cadena2 = new String("Hola");
```

Tanto “cadena” como “cadena2” contendrán lo mismo y, si se hace una comparación el resultado será “true”, pero, en una comparación estricta el resultado será “false”. Esto es así porque “cadena” es de tipo string, mientras que “cadena2” es de tipo object.

```
alert("'cadena'es de tipo = "+typeof(cadena));  
alert("'cadena2'es de tipo = "+typeof(cadena2));
```



3.5. Objetos en JavaScript

En JavaScript, pueden crearse objetos de dos formas:

```
var coche = {  
  modelo: "Mercedes C320",  
  color: "azul",  
  kms: 15000,  
  combustible: "diesel"  
};
```

```
var coche = new Object();  
coche.modelo = "Mercedes C320";  
coche.color = "azul";  
coche.kms = 15000;  
coche.combustible = "diesel";
```



3.5.1. Recorrer la información de un objeto.

En ocasiones, es necesario recorrer los campos de un objeto para poder procesarlos.

```
var usuario={nombre: "Felipe", apellido:"Ranas", edad:30, esAdmin:true};  
for(campo in usuario){  
    alert(campo);  
    alert(usuario[campo]);  
}
```

El navegador mostrará mensajes emergentes con el nombre y valor de cada uno de los atributos del objeto (nombre, Felipe, apellido, Ranas, etc.)



3.5.1. Recorrer la información de un objeto.

Teniendo el siguiente objeto:

```
var usuario={nombre:{first:'Pedro',last:'Salas'},edad:30,esAdmin:true};  
    alert(usuario.nombre.first);  
    alert(usuario['nombre']['first']);
```

Las dos líneas de “*alert*” mostrarían la misma información. La primera forma de acceso a los miembros de un objeto es parecida a la utilizada en Java.



3.5.2. Constructores de JavaScript.

Los métodos en los objetos se llaman constructores y se declaran de la siguiente forma:

```
function coche(modelo, color, kms, combustible){  
    this.modelo = modelo;  
    this.color = color;  
    this.kms = kms;  
    this.combustible = combustible;  
}  
  
var elmio = new coche("Mercedes E330", "negro", 120000, "diésel");  
var eltuyo = new coche("BMV 318", "blanco", 210000, "gasolina");
```



3.5.2. Constructores de JavaScript.

Si escribimos lo siguiente:

```
var miBMW = eltuyo;
```

Aquí JavaScript no creará una copia del objeto, las variables miBMW y eltuyo apuntarán al mismo objeto. Cualquier modificación de un atributo en una variable repercutirá en la otra.



3.5.2. Constructores de JavaScript.

JavaScript puede añadir nuevos atributos, a pesar de que el objeto haya sido creado previamente. La siguiente línea de código funcionará sin problema alguno:

```
elmio.matricula = "4321 JPH"
```



3.5.2. Constructores de JavaScript.

Un objeto se caracteriza por tener un estado (atributos) y un comportamiento (métodos). Estos son algunos métodos (un setter y un getter) para la clase coche creada previamente:

```
function coche(modelo, color, kms, combustible){  
    this.modelo = modelo;  
    this.color = color;  
    this.kms = kms;  
    this.combustible = combustible;  
    this.setModelo = function (nuevomodelo){  
        this.modelo = nuevomodelo;  
    }  
    this.getModelo =function (){  
        return this.modelo;  
    }  
}
```