

Objetos definidos por el usuario

Objetos definidos por el usuario

En Javascript, un objeto es una colección de propiedades que a su vez pueden ser datos o métodos.

Constructor: es una función especial para crear un objeto. Empieza en mayúscula:

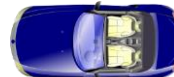
```
function Coche() {  
    //Propiedades y métodos  
}
```

Creación de un objeto:

```
var unCoche = new Coche();
```



Coche()



```
var cocheazul = new Coche();
```



```
var cocherojo = new Coche();
```



```
var cocheverde = new Coche();
```



Objetos definidos por el usuario

Definir propiedades de un objeto:

Se crean en el constructor precedidas por la palabra **this**.

“Constructor” sin parámetros:

```
function Coche() {  
    this.marca = ""; // Vacío  
    this.modelo = ""; // Vacío  
    this.combustible = "Diesel"; // Inicializado  
    this.cantidad = 0; //Inicializado  
}
```

“Constructor” con parámetros:

```
function Coche(marca, modelo, combustible, cantidad) {  
    this.marca = marca;  
    this.modelo = modelo;  
    this.combustible = combustible;  
    this.cantidad = cantidad;  
    // Cada propiedad toma los valores recibidos por parámetro  
}
```

Objetos definidos por el usuario

Definir propiedades de un objeto:

Crear un objeto vacío (sin propiedades):

```
var cocheVacio = new Coche();
```

Cambiar valores en las propiedades:

```
cocheVacio.marca = "Seat";  
cocheVacio.modelo = "Ibiza";  
cocheVacio.combustible = "Diesel";  
cocheVacio.cantidad = 40;
```

Crear un objeto inicializado (con propiedades):

```
var miCoche = new Coche("Seat", "Ibiza", "Diesel", 40);
```

Acceder a las propiedades:

```
document.write(" Mi coche es un " + miCoche.marca + " " + miCoche.modelo);
```



Objetos definidos por el usuario

Definir métodos de un objeto:

Permiten acceder y modificar las propiedades de los objetos. Empiezan en minúscula.

Método que modifica la cantidad de combustible:

```
function rellenarDeposito(litros) {  
    this.cantidad = litros;  
}
```

Referencia al método fuera del objeto (ino recomendado!)

```
function Coche(marca, modelo, combustible, cantidad) {  
    //Propiedades  
    this.marca = marca;  
    this.modelo = modelo;  
    this.combustible = combustible;  
  
    //Métodos  
    this.rellenardepósito = rellenardepósito;  
}
```

Objetos definidos por el usuario

Definir métodos de un objeto:

Permiten acceder y modificar las propiedades de los objetos. Empiezan en minúscula.

Método que modifica la cantidad de combustible:

```
function rellenarDeposito(litros) {  
    this.cantidad = litros;  
}
```

Referencia al método fuera del objeto (ino recomendado!)

```
function Coche(marca, modelo, combustible, cantidad) {  
    //Propiedades  
    this.marca = marca;  
    this.modelo = modelo;  
    this.combustible = combustible;  
  
    //Métodos  
    this.rellenardepósito = rellenardepósito;  
}
```

Objetos predefinidos - Arrays

Definir métodos de un objeto:

Referencia al método dentro del objeto

```
function Coche(marca, modelo, combustible, cantidad) {  
  //Propiedades  
  this.marca = marca;  
  this.modelo = modelo;  
  this.combustible = combustible;  
  //Métodos  
  this.rellenardepósito = function (litros) {  
    this.cantidad = litros;  
  };  
}
```



Objetos definidos por el usuario

Objetos literales:

Un literal es un valor fijo. Está formado por parejas de tipo nombre:valor.

Ejemplo:

```
var coche = {marca:"Ibiza", modelo:"Seat", combustible:"diesel", cantidad:40};
```

Ejemplo equivalente:

```
var coche = new Object();  
coche.marca = "Ibiza";  
coche.modelo = "Seat";  
coche.combustible = "diesel";  
coche.cantidad = 40;
```

Acceso:

```
coche.marca;  
coche["marca"];
```



Objetos definidos por el usuario

Ejercicio U3T8 – Parte 1:

Necesitamos almacenar en un programa todos los discos de música que tenemos en casa. Ahora que sabemos crear nuestros propios objetos es el mejor modo de guardar esta información.

Crea un objeto “disco” que almacene la siguiente información:

- Nombre del disco.
- Grupo de música o cantante.
- Año de publicación.
- Tipo de música (podrá ser “rock”, “pop”, “punk” o “indie”);
- Localización: almacenará un número de estantería.
- Prestado: almacenará un valor booleano. Por defecto será false.

Además tendrá los siguientes métodos:

- Un “constructor” sin parámetros (las 4 primeras propiedades serán cadenas vacías, la localización será 0 por defecto y prestado estará a false).
- Un método que permita incluir las cinco primeras propiedades; la propiedad prestado seguirá a false.
- Un método que permitirá cambiar el número de estantería en la localización.
- Un método que permitirá cambiar la propiedad Prestado.
- Un método que muestre toda la información de un disco.

Guarda todo el código en un archivo llamado disco.js



Objetos definidos por el usuario

Ejercicio U3T9 – Parte 2:

Carga en tu página el archivo de arrays que hicimos en la práctica anterior.

Crea un array vacío para almacenar los discos.

Cuando el usuario cargue la página, se cargarán las opciones:

- **Mostrar número de discos.**
- **Mostrar listado de discos(y le preguntará si quiere mostrarlos en el orden que se encuentran en el array, del revés u ordenados alfabéticamente).**
- **Mostrar un intervalo de discos(y le pedirá que introduzca el intervalo en formato inicio-fin; luego deberás extraer el valor inicio y fin).**
- **Añadir un disco (y le preguntará si quiere añadir al principio o al final).**
- **Borrar un disco (y le preguntará si quiere borrar al principio o al final).**
- **Consultar un disco (y le preguntará si quiere consultar por posición o por nombre).**

Todas las operaciones que se realicen se irán mostrando en la página con su título.

REUTILIZA EL CÓDIGO DE LA PRÁCTICA ANTERIOR.



END



prof.jduran@iesalixar.org