

UT 4. Programación con funciones, arrays y objetos definidos por el usuario.

3. CREACIÓN DE OBJETOS A MEDIDA

En JavaScript, los objetos se crean empleando una función especial denominada constructor.

3.1 Definición de propiedades

Las propiedades se crean dentro del constructor empleando la palabra reservada `this`.

Ejemplo básico:

```
javascript
```

```
function Coche() {  
    this.marca = "Audi A6";  
    this.combustible = "diesel";  
    this.cantidad = 0; // Cantidad de combustible en el  
depósito  
}  
  
var cocheDeMartin = new Coche();  
var cocheDeSilvia = new Coche();
```

Problema: Todos los coches son iguales. Solución: pasar parámetros.

Mejora - Constructor con parámetros:

```
javascript
```

```
function Coche(marca, combustible) {  
    this.marca = marca;  
    this.combustible = combustible;  
    this.cantidad = 0; // Cantidad de combustible inicial por  
defecto  
}
```

```
var cocheDeMartin = new Coche("Volkswagen Golf", "gasolina");
var cocheDeSilvia = new Coche("Mercedes SLK", "diesel");
```

Acceso a propiedades:

```
javascript
```

```
document.write("<br>El coche de Martin es un: " +
cocheDeMartin.marca +
    " a " + cocheDeMartin.combustible);
// El coche de Martin es un: Volkswagen Golf a gasolina
```

```
document.write("<br>El coche de Silvia es un: " +
cocheDeSilvia.marca +
    " a " + cocheDeSilvia.combustible);
// El coche de Silvia es un: Mercedes SLK a diesel
```

Modificación de propiedades:

```
javascript
```

```
cocheDeMartin.marca = "BMW X5";
cocheDeMartin.combustible = "diesel";
document.write("<br>El coche de Martin es un: " +
cocheDeMartin.marca +
    " a " + cocheDeMartin.combustible);
// El coche de Martin es un: BMW X5 a diesel
```

3.2 Definición de métodos

Los métodos son funciones asociadas a objetos que pueden acceder a las propiedades del mismo.

Ejemplo de método como función global:

```
javascript
```

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

```
function Coche(marca, combustible) {
    this.marca = marca;
    this.combustible = combustible;
    this.cantidad = 0;
    this.rellenarDeposito = rellenarDeposito; // Asignar el
método
}
```

Mejor práctica - Método definido dentro del constructor:

javascript

```
function Coche(marca, combustible) {
    this.marca = marca;
    this.combustible = combustible;
    this.cantidad = 0;

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

cocheDeMartin.rellenarDeposito(35);
document.write("<br>El coche de Martin tiene " +
cocheDeMartin.cantidad +
    " litros de " + cocheDeMartin.combustible +
    " en el depósito.");
// El coche de Martin tiene 35 litros de gasolina en el
depósito.
```

3.3 Clases en ES6

A partir de ES6, JavaScript soporta la sintaxis de clases, que es más similar a otros lenguajes de programación.

Sintaxis ES6:

```
javascript

class Producto {
    constructor(nombre, precio, cantidad) {
        this.nombre = nombre;
        this.precio = precio;
        this.cantidad = cantidad;
    }

    calcularSubtotal() {
        return this.precio * this.cantidad;
    }

    mostrarInfo() {
        return this.nombre + " - $" + this.precio;
    }
}

// Crear instancias
let p1 = new Producto("Laptop", 800, 1);
let p2 = new Producto("Mouse", 25, 3);

console.log(p1.nombre); // Laptop
console.log(p2.calcularSubtotal()); // 75
```

Propiedades y métodos:

```
javascript

class Estudiante {
    constructor(nombre, carrera) {
        this.nombre = nombre;
        this.carrera = carrera;
        this.calificaciones = [];
    }

    agregarCalificacion(calificacion) {
        this.calificaciones.push(calificacion);
    }
}
```

```

        console.log("Calificación agregada");
    }

    calcularPromedio() {
        let suma = this.calificaciones.reduce((a, b) => a + b,
0);
        return suma / this.calificaciones.length;
    }

    mostrarInfo() {
        return this.nombre + " - " + this.carrera;
    }
}

let estudiante = new Estudiante("Ana", "Informática");
estudiante.agregarCalificacion(8.5);
estudiante.agregarCalificacion(9.0);
console.log(estudiante.calcularPromedio()); // 8.75

```

Composición de objetos:

javascript

```

class Dirección {
    constructor(calle, ciudad, codigoPostal) {
        this.calle = calle;
        this.ciudad = ciudad;
        this.codigoPostal = codigoPostal;
    }
}

class Cliente {
    constructor(nombre, email) {
        this.nombre = nombre;
        this.email = email;
        this.dirección = new Dirección("", "", "");
    }
}

```

```
establecerDirección(calle, ciudad, código) {
    this.dirección = new Dirección(calle, ciudad, código);
}

mostrarInfo() {
    return this.nombre + " - " + this.dirección.calle + ",
" +
    this.dirección.ciudad;
}
}

let cliente = new Cliente("Juan", "juan@email.com");
cliente.establecerDirección("Calle Principal 123", "Madrid",
"28001");
console.log(cliente.mostrarInfo());
```
