

Programación Orientada a Objetos en JavaScript

En este ejemplo trabajamos con la Programación Orientada a Objetos (POO) en JavaScript a través de una clase llamada 'Gato' y una subclase llamada 'GatoEspecial'.

1. Clase Gato

- Se define una clase Gato con propiedades: nombre, raza y color.
- El constructor inicializa estos atributos.
- Se definen métodos para que el gato maúlle, salte y para obtener/modificar atributos.
- Se utilizan getters y setters, incluyendo propiedades con 'get' y 'set' personalizados.
- También se define un método estático que se puede usar sin crear una instancia.

Ejemplo de uso:

```
let miGato = new Gato("Bella", "Abisinio", "Marron");  
  
miGato.setRaza("Persa");  
  
console.log(miGato.getRaza());  
  
miGato.maullar();  
  
miGato.guardarColor = "Negro";  
  
console.log(miGato.sacarColor);  
  
Gato.saludar();
```

2. Herencia: Clase GatoEspecial

- GatoEspecial extiende la clase Gato e incorpora un nuevo atributo: habilidad.
- Usa 'super()' para reutilizar el constructor de Gato.
- Agrega métodos propios para manejar la habilidad y ejecutarla.

Ejemplo de uso:

```
let superGato = new GatoEspecial("Max", "Japonesa", "blanco", "Super atrapador de lauchas");  
superGato.ejecutarHabilidad();  
superGato.saltar();
```

Resumen de conceptos aplicados:

- Clases y constructores.
- Métodos normales y estáticos.
- Encapsulamiento con getters y setters.
- Herencia y reutilización de código con 'super()'.

Este ejemplo es ideal para comprender cómo se modela un objeto real (como un gato) en programación usando POO, y cómo se pueden extender las funcionalidades con herencia.

Autor: Nahuel Cejas