

# Ejercitación Métodos del objeto Arrays

## Enunciados:

- 1- Crea dos objetos, uno llamado animal y otro llamado Perro, donde Perro herede propiedades y métodos de animal utilizando `__proto__`.
- 2- Define una función constructora Persona que tenga una propiedad name. Luego, agrega un método `decirHola()` al prototipo de Persona para que todas las instancias de Person puedan usarlo.
- 3- Crea una jerarquía de clases utilizando prototype, donde una clase Figura tiene un método `getArea()`, y luego crea subclases como Circulo y Rectangulo que hereden de Figura y proporcionen sus implementaciones de `getArea()`.
- 4- Crea tres objetos en cadena: padre, hijo, y nieto. Establece parent como prototipo de child, y child como prototipo de grandchild. Agrega propiedades y métodos a cada objeto.
- 5- Extiende el prototipo de Array para agregar un método personalizado `sum()` que sume todos los elementos de un array.
- 6- Reescribe los ejercicios anteriores utilizando la sintaxis de clase de ECMAScript 6.
- 7- Crea un objeto llamado parent usando `Object.create(null)` y luego crea un objeto child que herede de parent utilizando `Object.create(parent)`.
- 8- Crea una función que acepte un objeto y una clase, y devuelva true si el objeto es una instancia de esa clase utilizando el operador `instanceof`.
- 9- Crea un objeto vacío, sin métodos ni atributos y que no herede ninguno tampoco.

## Respuestas:

```
1-
const animal = {
  tipo: "Desconocido",
  hablar() {
    console.log("Algún sonido");
  }
};
```

```
const perro = {
  __proto__: animal,
  tipo: "Perro",
  hablar() {
    console.log("¡Guau!");
  }
};
```

```
2-
function Persona(nombre) {
  this.nombre = nombre;
}

Persona.prototype.decirHola = function() {
  console.log(`Hola, mi nombre es ${this.nombre}`);
};

3-
function Figura() {}

Figura.prototype.calcularArea = function() {
  return 0; // Implementación predeterminada
};

function Circulo(radio) {
  this.radio = radio;
}

Circulo.prototype = Object.create(Figura.prototype);
Circulo.prototype.calcularArea = function() {
  return Math.PI * Math.pow(this.radio, 2);
};

function Rectangulo(ancho, alto) {
  this.ancho = ancho;
  this.alto = alto;
}

Rectangulo.prototype = Object.create(Figura.prototype);
Rectangulo.prototype.calcularArea = function() {
  return this.ancho * this.alto;
};

4-
const padre = { x: 10 };
const hijo = Object.create(padre);
const nieto = Object.create(hijo);

5-
Array.prototype.sumar = function() {
  return this.reduce((acumulador, actual) => acumulador + actual, 0);
};

7-
const padre = Object.create(null);
const hijo = Object.create(padre);

8-
```

```
function esInstanciaDe(obj, clase) {  
  return obj instanceof clase;  
}
```

```
9-  
Let o = {};  
o.__proto__=null;
```