



Programación Orientada a Objetos (P00)

La P00 en JavaScript se basa en la creación y manipulación de objetos. Aunque JavaScript es un lenguaje de programación orientado a objetos, su implementación es prototípica, lo que significa que los objetos pueden heredar directamente de otros objetos.

1. Objetos:

En JavaScript, un objeto es una colección de pares clave-valor, y se puede crear de diversas maneras. Ejemplo:

```
// Crear un objeto literal

const persona = {

  nombre: 'Juan',

  edad: 25,

  saludar: function() {

    console.log(`Hola, soy ${this.nombre}`);

  },

};
```

```
// Acceder a propiedades y métodos

console.log(persona.nombre); // Imprime: Juan

persona.saludar(); // Imprime: Hola, soy Juan
```

2. Funciones Constructoras:

Las funciones constructoras son una forma de crear objetos similares. Se utilizan con la palabra clave new. Ejemplo:

```
// Función constructora

function Persona(nombre, edad) {

  this.nombre = nombre;

  this.edad = edad;

  this.saludar = function() {

    console.log(`Hola, soy ${this.nombre}`);

  };

}
```

```
// Crear instancias

const persona1 = new Persona('Ana', 30);

const persona2 = new Persona('Carlos', 28);
```

```
persona1.saludar(); // Imprime: Hola, soy Ana

persona2.saludar(); // Imprime: Hola, soy Carlos
```

3. Prototipos:

En JavaScript, los objetos pueden heredar propiedades y métodos de otros objetos mediante prototipos. Ejemplo:

```
// Prototipo

Persona.prototype.despedir = function() {

  console.log(`Adiós, ${this.nombre}`);

};
```

```
// Uso del prototipo
```

```
persona1.despedir(); // Imprime: Adiós, Ana
```

```
persona2.despedir(); // Imprime: Adiós, Carlos
```

4. Clases (ES6+):

A partir de ECMAScript 2015 (ES6), JavaScript introdujo la sintaxis de clases, que proporciona una forma más sencilla de crear objetos y manejar la herencia. Ejemplo:

```
class Animal {  
  
    constructor(nombre) {  
  
        this.nombre = nombre;  
  
    }  
  
    sonido() {  
  
        console.log('Haciendo algún sonido');  
  
    }  
  
}
```

```
class Perro extends Animal {  
  
    sonido() {  
  
        console.log('Woof, woof');  
  
    }  
  
}
```

```
// Crear instancias de clases
```

```
const animalGenerico = new Animal('Animal');
```

```
const perro = new Perro('Buddy');
```

```
animalGenerico.sonido(); // Imprime: Haciendo algún sonido
```

```
perro.sonido(); // Imprime: Woof, woof
```

Estos son solo ejemplos básicos de cómo trabajar con objetos en JavaScript, y hay muchos otros conceptos avanzados relacionados con la POO en este lenguaje.