

JavaScript

Soluciones de Programación Orientada a Objetos (POO)

CertiDevs

Índice de contenidos

1. Crear una clase y un objeto	1
2. Utilizar el constructor de una clase	1
3. Crear y utilizar una clase heredada	2
4. Utilizar el polimorfismo en JavaScript	2
5. Crear y utilizar una clase estática	3

1. Crear una clase y un objeto

1. Crea una clase llamada Persona con las propiedades nombre, apellido y edad, y un método llamado saludar que imprima en consola un saludo con el nombre y apellido de la persona.
2. Instancia un objeto de la clase Persona y llama al método saludar.

```
class Persona {
  nombre = "";
  apellido = "";
  edad = 0;

  saludar() {
    console.log(`¡Hola, soy ${this.nombre} ${this.apellido}!`);
  }
}

const juan = new Persona();
juan.nombre = "Juan";
juan.apellido = "Pérez";
juan.saludar();
```

2. Utilizar el constructor de una clase

1. Modifica la clase Persona para agregar un constructor que acepte los valores de las propiedades nombre, apellido y edad.
2. Instancia un objeto de la clase Persona con valores específicos y llama al método saludar.

```
class Persona {
  constructor(nombre, apellido, edad) {
    this.nombre = nombre;
    this.apellido = apellido;
    this.edad = edad;
  }

  saludar() {
    console.log(`¡Hola, soy ${this.nombre} ${this.apellido}!`);
  }
}

const maria = new Persona("María", "García", 28);
maria.saludar();
```

3. Crear y utilizar una clase heredada

1. Crea una clase llamada Estudiante que herede de Persona y tenga una propiedad adicional llamada carrera.
2. Modifica el constructor de Estudiante para aceptar y asignar el valor de la propiedad carrera.
3. Agrega un método llamado estudiar en la clase Estudiante que imprima en consola un mensaje que incluya la carrera y el nombre del estudiante.
4. Instancia un objeto de la clase Estudiante y llama a los métodos saludar y estudiar.

```
class Estudiante extends Persona {  
  constructor(nombre, apellido, edad, carrera) {  
    super(nombre, apellido, edad);  
    this.carrera = carrera;  
  }  
  
  estudiar() {  
    console.log(`${this.nombre} está estudiando ${this.carrera}.`);  
  }  
}  
  
const carlos = new Estudiante("Carlos", "Ramírez", 22, "Ingeniería Informática");  
carlos.saludar();  
carlos.estudiar();
```

4. Utilizar el polimorfismo en JavaScript

1. Crea una clase llamada Profesor que herede de Persona y tenga una propiedad adicional llamada materia.
2. Modifica el constructor de Profesor para aceptar y asignar el valor de la propiedad materia.
3. Sobreescribe el método saludar en la clase Profesor para que imprima un mensaje diferente que incluya la materia que enseña el profesor.
4. Instancia un objeto de la clase Profesor y llama al método saludar.

```
class Profesor extends Persona {  
  constructor(nombre, apellido, edad, materia) {  
    super(nombre, apellido, edad);  
    this.materia = materia;  
  }  
  
  saludar() {  
    console.log(`¡Hola, soy el profesor ${this.nombre} ${this.apellido} y enseño  
    ${this.materia}!`);  
  }  
}
```

```
const ana = new Profesor("Ana", "González", 45, "Matemáticas");
ana.saludar();
```

5. Crear y utilizar una clase estática

1. Crea una clase llamada Calculadora con un método estático llamado sumar que acepte dos números y devuelva su suma.
2. Utiliza el método estático sumar de la clase Calculadora sin instanciar un objeto.

```
class Calculadora {
  static sumar(a, b) {
    return a + b;
  }
}

const resultado = Calculadora.sumar(5, 7);
console.log(`Resultado: ${resultado}`);
```