Ejercicios sobre Objetos

# **Actividad cuenta.html**

Crea un objeto utilizando la **notación literal de objeto** que permita almacenar los siguientes datos referente a una cuenta bancaria: el titular y el saldo. El objeto debe tener también un método para ingresar dinero y otro para extraerlo.

Crea un array de objetos cuenta y ordénalos de menor a mayor saldo.

# **Actividad persona.html**

1. Clase Persona

Implementar clase **Persona** con atributos los \_nombre, \_edad y \_fumador. El constructor debe permitir asignar el nombre y la edad, y opcionalmente si es fumador (que por defecto será false).

Implementar método get y set para el acceso a los atributos de la clase (el método set para cargar la edad garantizará que la edad no es negativa, lanzando un error en dicho caso).

Implementar un método toString que devuelva el nombre, la edad y si es fumador usando plantilla de cadena de caracteres.

Implementa un método estático llamada comparaPersonas que reciba como argumento dos objetos persona y devuelva el de mayor edad.

Implementa un método cumplirEdad que incremente en una unidad el número de años

Implementa una propiedad estática llamada NUMERO\_CROMOSOMAS que devuelva el número de cromosomas de un ser humano (debe devolver 46, y será consultado usando Persona.NUMERO\_CROSOMAS).

Implementa un método de instancia llamado personasConMasEdad al que le pasemos un número variable de objetos personas (usa parámetros Rest en la función) y debe devolver un array conteniendo todas las personas que tengan una edad superior a la de nuestro objeto.

1. Clase Conductor (hereda de clase Persona)

Crea una clase **Conductor** que herede de la clase Persona y que tenga los atributos para almacenar el tipo de carnet (por ejemplo si tiene el de coche almacenaremos "B1", y si tiene varios estarán separados por como, por ejemplo si tiene el de moto y coche almacenaremos "A2,B1"). También debe tener otro atributo que almacene el número de puntos. El constructor permitirá indicar el nombre, la edad, el tipo de carnet y opcionalmente los puntos (que será 15 si se omite) . Implementa un método llamado sancionPuntos que tenga un argumento de entrada con el número de puntos a quitar del carnet (los puntos nunca podrán tener un valor negativo)..

# **Actividad jarras.html**

Crea una clase llamada Jarra que contenga dos atributos, uno llamado capacidad y otro llamado cantidad. El atributo capacidad contendrá los litros de capacidad de la jarra y el atributo cantidad contendrá los litros que actualmente contiene la jarra. Cuando creemos nuevas instancias de objetos Jarra lo haremos de la siguiente manera:

var jarra1 = new Jarra(10,4); → Crearía un objeto jarra1 con capacidad 10 litros y llena con una cantidad de 4 litros.

var jarra2 = new Jarra(15,8); → Crearía un objeto jarra2 con capacidad 15 litros y llena con una cantidad de 8 litros.

La clase Jarra debe tener una propiedad llamada **cantidad** (get/set del atributo \_cantidad) que garantice que no podamos asignar una cantidad mayor a la capacidad de la jarra ni un valor negativo. Si intentamos asignar un valor negativo debe devolver el error **“La cantidad debe ser un número positivo”**

La clase Jarra debe tener los siguientes **métodos de instancia**:

jarra1.**llenar()** : llenará completamente la jarra hasta la capacidad

jarra1.**vaciar()** : vaciará completamente la jarra.

jarra1.**llenarDesde(jarra2)**: se volcarán todos los litros que quepan de jarra2 en jarra1 (si jarra2 contiene más litros de los que caben en jarra1) o todos los que contiene jarra2 quedándose ésta vacía (si jarra2 contiene menos litros de los que caben en jarra1).

También se implementará un **método de clase** llamado comparar que permita comparar qué jarra contiene más líquido. Por ejemplo:

Jarra.**comparar(jarra1,jarra2)** devolverá la jarra que más líquido contiene. Para implementar este método debes utilizar una función arrow (novedad introducida en ES6).

Añade un método de instancia llamado **toString()** que devuelva la capacidad y la cantidad de la jarra (debes usar una plantilla cadena de caracteres).

Crear una función de testeo con las siguientes instrucciones:

var jarra1=new Jarra(10,4); // Crea una jarra de 10 litros de capacidad y llena con 4 litros

var jarra2=new Jarra(15,8); // Crea una jarra de 15 litros de capacidad y llena con 8 litros

Jarra.comparar(jarra1, jarra2); // Debería devolver jarra2, ya que contiene más litros que jarra1

jarra1.llenarDesde(jarra2); // Llena jarra1 con el contenido que quepa de jarra2. A jarra1 le quedan 6 litros para llenarse, por lo que jarra1 debería llenarse completamente (4+6=10 litros) y jarra2 quedar con 2 litros (8-6= 2 litros)

console.log(“cantidad jarra1: “ + jarra1.cantidad); // Debería mostrar 10 litros

console.log(“cantidad jarra2: “ + jarra2.cantidad); // Debería mostrar 2 litros

jarra2.vaciar(); //Debería vaciarse completamente la jarra2

console.log(“cantidad jarra2: “ + jarra2.cantidad); // Debería mostrar 0 litros

|  |
| --- |