

Guía de JavaScript

Variables, Arrays y Funciones

- 1- Imprimir por consola un mensaje guardado en una variable usando la función `console.log()`
- 2- Crear un array llamado meses y que almacene el nombre de los doce meses del año. Mostrar por pantalla los doce nombres utilizando la función `console.log()`
- 3- Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

Métodos de arrays

En esta sección debes resolver todo con los métodos **forEach**, **map**, **some**, **filter**, **every**, **find** y **sort**

- 4- Define una función que dado un array de números nos dice si alguno es par.

```
algunoEsPar([1,2,3])  
> true
```

- 5- Define la función aprobó, que dada la lista de las notas de un alumno devuelve si el alumno aprobó. Un alumno aprobó si todas sus notas son mayores o iguales a 4

```
aprobo([8,6,2,4])  
> False
```

- 6- Define la función quienesAprobaron, que dada la información de un curso devuelve la información de los alumnos que aprobaron. Podés usar la función que hiciste en el ejercicio anterior.

```
quienesAprobaron([[8,6,2,4],[7,9,4,5],[6,2,4,2],[9,6,7,10]])  
> [[7,9,4,5],[9,6,7,10]]
```

7- Define la función hayAlgunNegativo, que dada una lista de números nos dice si hay algún negativo

```
hayAlgunNegativo([2,-3,9])  
> True
```

8- Escribi una función cuantosCumplen que dada una condición y una lista, diga cuantos elementos la cumplen

```
// even es una función que retorna true si un número es par (deben  
crearla)  
cuantosCumplen(even,[7,9,25,42])  
> 1 //Solo 42 es par
```

9- Escribí una función rechazar, que dada una condición y un array, devuelva un array con los elementos que no la cumplen

```
rechazar(even,[7,9,87,42])  
> [7, 9, 87]
```

10- Define la función contiene que dado un elemento y una lista, nos diga si la lista contiene al elemento

```
contiene(8,[7,8,9])  
> True
```

Clases y Objetos

11- Crea una clase Persona que tenga los atributos nombre y edad. Además agregale un método presentarse() que como resultado imprima un mensaje diciendo su nombre y edad.

12- Crea una clase Estudiante (que herede de Persona) que tenga un atributo profesor y tenga dos métodos.

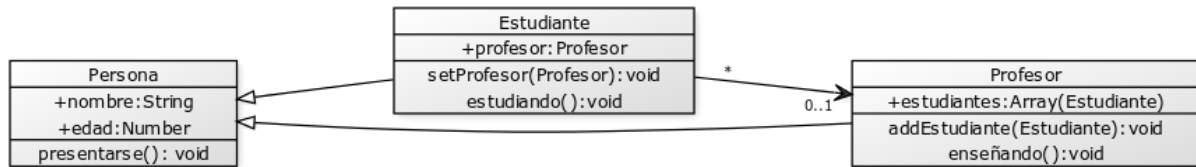
El método estudiando() que imprime el mensaje “Estudiando con ” y el nombre del profesor con el que se encuentra estudiando.

El método setProfesor(profesor) que setea con qué profesor se encuentra estudiando.

13- Crea una clase Profesor (que herede de Persona) que tenga un atributo estudiantes con un array de Estudiante. Además debe tener dos métodos enseñando() que imprima por

consola todos sus estudiantes a los cuales les está enseñando y el método `addEstudiante(unEstudiante)` que agregue a ese estudiante a la lista. Además este método es responsable de manejar de agregar al objeto profesor como profesor del estudiante (manejo de doble referencia)

Debería quedar algo así...



CREATED WITH YUML