

Tenemos el siguiente programa HTML:

```
<html>
<head>
<title>retos JavaScript</title>
</head>
<body>
<h1>retos JavaScript</h1>
<script>
var cars = ["Saab", "Volvo", "BMW",["Honda", "Toyota"],["Lexus",
"Mercedes", "Lada"], "Hyunday"]];

var carrakos = ["Chevrolet", "Tesla", "BMW",["Honda",
"Infiniti"],["Lexus", "Mercedes", "Kia"], "Hyunday"],["Mazda",
"Mercedes", "Jeep",["KTM", "Nissan", "Isuzu"]]]];

var newcars = ["Saab", "Volvo", "BMW",["Honda", "Toyota"],["Lexus",
"Volvo", "Lada"], "Hyunday", "Volvo"]];

buddies = [{firstName:"Jonny", lastName:"Rambo",
car:"BMW"},{firstName:"Arnold", lastName:"Suarchenager",
car:"Honda"},{firstName:"Chuck", lastName:"Norris",
car:"Honda"},{firstName:"Yakie", lastName:"Chang",
car:"BMW"},{firstName:"Bruce", lastName:"Lee",
car:"Honda"},{firstName:"Steven", lastName:"Seagal", car:"Lada"}];

let xungas=["Honda","Lada"]

//EJERCICIO 1

//EJERCICIO 2

//EJERCICIO 3

//EJERCICIO 4

</script>
</body>
</html>
```

Se pide:

Ejercicio 1: ¿Cuántas veces aparece la palabra Volvo en newcars en una línea?
(1,5 PUNTOS)

Ejercicio 2: Ver si una marca es trendy. Una marca es trendy si esta es utilizada por más de un buddy. Crea en este mismo ejercicio un array trendy con dichas marcas. Muestra el array trendy por consola. Se pide que lo realices en 3-4 líneas de código. (3 PUNTOS)

Ejercicio 3: Muestra por consola las marcas del array newcars que usen los buddies, que están obviamente en newcars pero que no estén en el array xungas. Muestra el resultado por consola. Puede aparecer alguna marca repetida (no importa). Realiza este programa en una línea de código.

(3 PUNTOS)

Ejercicio 4: Se quiere conocer la marca más trendy de todas usada por los buddies (la más preferida). Realiza este programa en 2 líneas de código. (2,5 PUNTOS)

Importante. No puedes utilizar en ningún ejercicio el método includes() ni utilizar sets. No puedes utilizar for.

Se valorará de tu programa:

- El programa realiza la funcionalidad requerida.
- Que no haya errores de consola.
- La eficiencia.
- La modularidad (alta cohesión y bajo acoplamiento).
- Que el código sea limpio, flexible, reutilizable y mantenible.
- Sigue los principios SOLID.
- Otros criterios.