



JS Review Basics

Bienvenido a Upgrade-Hub! Por fin ha llegado el momento de ponernos a trabajar. Hoy hemos avanzado mucho con los conceptos de JS y es el momento de ponerlos en práctica.

Antes de comenzar crea un nuevo proyecto en vuestro GitLab o Github, el nombre que debéis utilizar es **upgrade-review-basics** y recuerda realizar un commit por iteración:

```
git add .  
git commit -m "Iteration#1"  
git push origin master
```

Guía de práctica

Iteración #1: Mix for e includes

Dado el siguiente javascript usa `forof` para recorrer el array de películas, genera un nuevo array con las categorías de las películas e imprime por consola el array de categorías. Ten en cuenta que las categorías no deberían repetirse. Para filtrar las categorías puedes ayudarte de la función `.includes()`

```
const movies = [
  {title: 'Madaraspar', duration: 192, categories: ['comedia', 'aventura']},
  {title: 'Spiderpan', duration: 122, categories: ['aventura', 'acción']},
  {title: 'Solo en Whatsapp', duration: 223, categories: ['comedia', 'thriller']},
  {title: 'El gato con guantes', duration: 111, categories: ['comedia', 'aventura', 'animación']},
]
```

Iteración #2: Mix Fors

Dado el siguiente javascript usa forof y forin para hacer la media del volumen de todos los sonidos favoritos que tienen los usuarios.

```
const users = [
  {name: 'Manolo el del bombo',
    favoritesSounds: {
      waves: {format: 'mp3', volume: 50},
      rain: {format: 'ogg', volume: 60},
      firecamp: {format: 'mp3', volume: 80},
    }
  },
  {name: 'Mortadelo',
    favoritesSounds: {
      waves: {format: 'mp3', volume: 30},
      shower: {format: 'ogg', volume: 55},
      train: {format: 'mp3', volume: 60},
    }
  },
  {name: 'Super Lopez',
    favoritesSounds: {
      shower: {format: 'mp3', volume: 50},
      train: {format: 'ogg', volume: 60},
      firecamp: {format: 'mp3', volume: 80},
    }
  },
  {name: 'El culebra',
    favoritesSounds: {
      waves: {format: 'mp3', volume: 67},
      wind: {format: 'ogg', volume: 35},
      firecamp: {format: 'mp3', volume: 60},
    }
  },
]
```

Iteración #3: Mix Fors

Dado el siguiente javascript usa forof y forin para saber cuantas veces ha sido cada sonido agregado por los usuarios a favorito. Para ello recorre la lista de usuarios y usa forin para recoger el nombre de los sonidos que el usuario tenga como favoritos.

Una vez accedas a ellos piensa en la mejor forma de hacer un conteo de cada vez que ese sonido se repita como favorito en cada usuario.

Este ejercicio es un poco complicado con los conocimientos actuales pero...a la vez un buen reto y oportunidad para comprender que hay muchas formas de hacer las cosas en javascript.

```
const users = [  
  {name: 'Manolo el del bombo',  
    favoritesSounds: {  
      waves: {format: 'mp3', volume: 50},  
      rain: {format: 'ogg', volume: 60},  
      firecamp: {format: 'mp3', volume: 80},  
    }  
  },  
  {name: 'Mortadelo',  
    favoritesSounds: {  
      waves: {format: 'mp3', volume: 30},  
      shower: {format: 'ogg', volume: 55},  
      train: {format: 'mp3', volume: 60},  
    }  
  },  
  {name: 'Super Lopez',  
    favoritesSounds: {  
      shower: {format: 'mp3', volume: 50},  
      train: {format: 'ogg', volume: 60},  
      firecamp: {format: 'mp3', volume: 80},  
    }  
  },  
  {name: 'El culebra',  
    favoritesSounds: {  
      waves: {format: 'mp3', volume: 67},  
      wind: {format: 'ogg', volume: 35},  
      firecamp: {format: 'mp3', volume: 60},  
    }  
  },  
]
```

Iteración #4: Métodos findArrayIndex

Crea una función llamada `findArrayIndex` que reciba como parametros un array de textos y un texto y devuelve la posición del array cuando el valor del array sea igual al valor del texto que enviaste como parametro. Haz varios ejemplos y compruebalos.

Sugerencia de función:

```
function findArrayIndex(array, text) {}
```

Ej array:

```
['Caracol', 'Mosquito', 'Salamandra', 'Ajolote']
```

Iteración #5: Función `rollDice`

Crea una función llamada `rollDice()` que reciba como parametro el numero de caras que queramos que tenga el dado que deberá simular el código dentro de la función. Como hemos dicho, que la función use el parametro para simular una tirada de dado y retornar el resultado. Si no se te ocurre como hacer un numero aleatorio no te preocupes! busca información sobre la función de javascript **`Math.random()`**;

Iteración #6: Función `swap`

Crea una función llamada `swap()` que reciba un array y dos parametros que sean indices del array. La función deberá intercambiar la posición de los valores de los indices que hayamos enviado como parametro. Retorna el array resultante.

Sugerencia de array:

```
['Mesirve', 'Cristiano Romualdo', 'Fernando Muralla', 'Ronalguiño']
```

Happy coding 🌟 & Happy Upgrader ❤️