



# JS ES6

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-es6** y recuerda realizar un commit por iteración:

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

## Guía de práctica

### Iteración #1: Arrows

Crea una arrow function que tenga dos parametros a y b y que por defecto el valor de a = 10 y de b = 5. Haz que la función muestre por consola la suma de los dos parametros.

- 1.1 Ejecuta esta función sin pasar ningún parametro
- 1.2 Ejecuta esta función pasando un solo parametro
- 1.3 Ejecuta esta función pasando dos parametros

### Iteración #2: Destructuring

2.1 En base al siguiente javascript, crea variables en base a las propiedades del objeto usando object destructuring e imprimelas por consola. Cuidado, no hace falta hacer destructuring del array, solo del objeto.

```
const game = {title: 'The last us 2', gender: ['action', 'zombie', 'survival'], year: 2020}
```

2.2 En base al siguiente javascript, usa destructuring para crear 3 variables llamadas fruit1, fruit2 y fruit3, con los valores del array. Posteriormente imprimelo por consola.

```
const fruits = ['Banana', 'Strawberry', 'Orange'];
```

2.3 En base al siguiente javascript, usa destructuring para crear 2 variables igualandolo a la función e imprimiendolo por consola.

```
const animalFunction = () => {
  return {name: 'Bengal Tiger', race: 'Tiger'}
};
```

2.4 En base al siguiente javascript, usa destructuring para crear las variables name y itv con sus respectivos valores. Posteriormente crea 3 variables usando igualmente el destructuring para cada uno de los años y comprueba que todo esta bien imprimiendolo.

```
const car = {name: 'Mazda 6', itv: [2015, 2011, 2020] }
```

### Iteración #3: Spread Operator

3.1 Dado el siguiente array, crea una copia usando spread operators.

```
const pointsList = [32, 54, 21, 64, 75, 43]
```

3.2 Dado el siguiente objeto, crea una copia usando spread operators.

```
const toy = {name: 'Bus laiyiar', date: '20-30-1995', color: 'multicolor'};
```

3.3 Dado los siguientes arrays, crea un nuevo array juntandolos usando spread operators.

```
const pointsList = [32, 54, 21, 64, 75, 43];  
const pointsList2 = [54,87,99,65,32];
```

3.4 Dado los siguientes objetos. Crea un nuevo objeto fusionando los dos con spread operators.

```
const toy = {name: 'Bus laiyiar', date: '20-30-1995', color: 'multicolor'};  
const toyUpdate = {lights: 'rgb', power: ['Volar Like a dragon', 'MoonWalk']}
```

3.5 Dado el siguiente array. Crear una copia de él eliminando la posición 2 pero sin editar el array inicial. De nuevo, usando spread operators.

```
const colors = ['rojo', 'azul', 'amarillo', 'verde', 'naranja'];
```

### Iteración #4: Map

4.1 Dado el siguiente array, devuelve un array con sus nombres utilizando .map().

```
const users = [  
  {id: 1, name: 'Abel'},  
  {id:2, name: 'Julia'},  
  {id:3, name: 'Pedro'},  
  {id:4, name: 'Amanda'}  
];
```

4.2 Dado el siguiente array, devuelve una lista que contenga los valores de la propiedad .name y cambia el nombre a 'Anacleto' en caso de que empiece por 'A'.

```
const users = [  
  {id: 1, name: 'Abel'},  
  {id:2, name: 'Julia'},  
  {id:3, name: 'Pedro'},  
  {id:4, name: 'Amanda'}  
];
```

4.3 Dado el siguiente array, devuelve una lista que contenga los valores de la propiedad .name y añade al valor de .name el string ' (Visitado)' cuando el valor de la propiedad isVisited = true.

```
const cities = [  
  {isVisited:true, name: 'Tokyo'},  
  {isVisited:false, name: 'Madagascar'},  
  {isVisited:true, name: 'Amsterdam'},  
  {isVisited:false, name: 'Seul'}  
];
```

### Iteración #5: Filter

5.1 Dado el siguiente array, utiliza .filter() para generar un nuevo array con los valores que sean mayor que 18

```
const ages = [22, 14, 24, 55, 65, 21, 12, 13, 90];
```

5.2 Dado el siguiente array, utiliza `.filter()` para generar un nuevo array con los valores que sean par.

```
const ages = [22, 14, 24, 55, 65, 21, 12, 13, 90];
```

5.3 Dado el siguiente array, utiliza `.filter()` para generar un nuevo array con los streamers que tengan el `gameMorePlayed` = 'League of Legends'.

```
const streamers = [
  {name: 'Rubius', age: 32, gameMorePlayed: 'Minecraft'},
  {name: 'Ibai', age: 25, gameMorePlayed: 'League of Legends'},
  {name: 'Reven', age: 43, gameMorePlayed: 'League of Legends'},
  {name: 'AuronPlay', age: 33, gameMorePlayed: 'Among Us'}
];
```

5.4 Dado el siguiente array, utiliza `.filter()` para generar un nuevo array con los streamers que incluyan el caracter 'u' en su propiedad `.name`. Recomendamos usar la funcion `.includes()` para la comprobación.

```
const streamers = [
  {name: 'Rubius', age: 32, gameMorePlayed: 'Minecraft'},
  {name: 'Ibai', age: 25, gameMorePlayed: 'League of Legends'},
  {name: 'Reven', age: 43, gameMorePlayed: 'League of Legends'},
  {name: 'AuronPlay', age: 33, gameMorePlayed: 'Among Us'}
];
```

5.5 utiliza `.filter()` para generar un nuevo array con los streamers que incluyan el caracter 'Legends' en su propiedad `.gameMorePlayed`. Recomendamos usar la funcion `.includes()` para la comprobación.

Además, pon el valor de la propiedad `.gameMorePlayed` a MAYUSCULAS cuando `.age` sea mayor que 35.

5.6 Dado el siguiente html y javascript, utiliza `.filter()` para mostrar por consola los streamers que incluyan la palabra introducida en el input. De esta forma, si introduzco 'Ru' me debería de mostrar solo el streamer 'Rubius'. Si introduzco 'i', me debería de mostrar el streamer 'Rubius' e 'Ibai'.

```
const streamers = [
  {name: 'Rubius', age: 32, gameMorePlayed: 'Minecraft'},
  {name: 'Ibai', age: 25, gameMorePlayed: 'League of Legends'},
  {name: 'Reven', age: 43, gameMorePlayed: 'League of Legends'},
  {name: 'AuronPlay', age: 33, gameMorePlayed: 'Among Us'}
];
```

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <input type="text" data-function="toFilterStreamers"/>
</body>
</html>
```

5.7 Dado el siguiente html y javascript, utiliza `.filter()` para mostrar por consola los streamers que incluyan la palabra introducida en el input. De esta forma, si introduzco 'Ru' me debería de mostrar solo el streamer 'Rubius'. Si introduzco 'i', me debería de mostrar el streamer 'Rubius' e 'Ibai'.

En este caso, muestra solo los streamers filtrados cuando hagamos click en el button.

```
const streamers = [
  {name: 'Rubius', age: 32, gameMorePlayed: 'Minecraft'},
  {name: 'Ibai', age: 25, gameMorePlayed: 'League of Legends'},
  {name: 'Reven', age: 43, gameMorePlayed: 'League of Legends'},
  {name: 'AuronPlay', age: 33, gameMorePlayed: 'Among Us'}
];
```

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <input type="text" data-function="toFilterStreamers"/>
  <button data-function="toShowFilterStreamers">Filter</button>
</body>
</html>
```

## Iteración #6: Find

```
6.1 Dado el siguiente array, usa .find() para encontrar el número 100.
const numbers = [32, 21, 63, 95, 100, 67, 43];

6.2 Dado el siguiente array, usa .find() para encontrar la película del año 2010.
const movies = [
  {title: 'Madagascar', stars: 4.5, date: 2015},
  {title: 'Origen', stars: 5, date: 2010},
  {title: 'Your Name', stars: 5, date: 2016}
];

6.3 Dado el siguiente javascript, usa .find() para encontrar el alien de nombre
'Cucushumushu' y la mutación 'Porompompero'. Una vez que los encuentres, usa
spread operator para fusionarlos teniendo en cuenta que el objeto de la mutación
lo queremos meter en la propiedad .mutation del objeto fusionado.
const aliens = [
  {name: 'Zalamero', planet: 'Eden', age: 4029},
  {name: 'Paktu', planet: 'Andromeda', age: 32},
  {name: 'Cucushumushu', planet: 'Marte', age: 503021}
];
const mutations = [
  {name: 'Porompompero', description: 'Hace que el alien pueda adquirir la habilidad de tocar el tambor'},
  {name: 'Fly me to the moon', description: 'Permite volar, solo y exclusivamente a la luna'},
  {name: 'Andando que es gerundio', description: 'Invoca a un señor mayor como Personal Trainer'}
];
```

## Iteración #7: Reduce

```
7.1 Dado el siguiente array, haz una suma de todas las notas de los exámenes de
los alumnos usando la función .reduce().

const exams = [
  {name: 'Yuyu Cabeza Crack', score: 5},
  {name: 'Maria Aranda Jimenez', score: 1},
  {name: 'Cristóbal Martínez Lorenzo', score: 6},
  {name: 'Mercedez Reguera Brito', score: 7},
  {name: 'Pamela Anderson', score: 3},
  {name: 'Enrique Perez Lijó', score: 6},
  {name: 'Pedro Benitez Pacheco', score: 8},
  {name: 'Ayumi Hamasaki', score: 4},
  {name: 'Robert Kiyosaki', score: 2},
  {name: 'Keanu Reeves', score: 10}
];

7.2 Dado el mismo array, haz una suma de todas las notas de los exámenes de los
alumnos que estén aprobados usando la función .reduce().

7.3 Dado el mismo array, haz la media de las notas de todos los exámenes .reduce().
```

## Iteración #8: Bonus

```
6.1 Dado el siguiente javascript filtra los videojuegos por gender = 'RPG' usando
.filter() y usa .reduce() para conseguir la media de sus .score.
La función .find() también podría ayudarte para el contrar el genero 'RPG' en el
array .gender.

const videogames = [
  {name: 'Final Fantasy VII', genders: ['RPG'], score: 9.5},
  {name: 'Assasins Creed Valhala', genders: ['Aventura', 'RPG'], score: 4.5},
  {name: 'The last of Us 2', genders: ['Acción', 'Aventura'], score: 9.8},
  {name: 'Super Mario Bros', genders: ['Plataforma'], score: 8.5},
  {name: 'Genshin Impact', genders: ['RPG', 'Aventura'], score: 7.5},
  {name: 'Legend of Zelda: Breath of the wild', genders: ['RPG', 'La cosa más puto bonita que he visto nunca'], score: 10},
]
```

Happy coding 🌟 & Happy Upgrader ❤️