



JS Asynchronous

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

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

Guía de práctica

Iteración #1: Fetch

1.1 Utiliza esta url de la api Agify 'https://api.agify.io?name=michael' para hacer un `.fetch()` y recibir los datos que devuelve. Imprimelo mediante un `console.log()`. Para ello, es necesario que crees un `.html` y un `.js`.

2.1 Dado el siguiente javascript y html. Añade la funcionalidad necesaria usando `fetch()` para hacer una consulta a la api cuando se haga click en el botón, pasando como parametro de la api, el valor del input.

```
const baseUrl = 'https://api.nationalize.io';
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <input type="text">
  <button>Consultar</button>
</body>
</html>
```

2.3 En base al ejercicio anterior. Crea dinamicamente un elemento por cada petición

a la api que diga... 'El nombre X tiene un Y porciento de ser de Z' etc etc.
EJ: El nombre Pepe tiene un 22 porciento de ser de ET y un 6 porciento de ser de MZ.

2.4 En base al ejercicio anterior, crea un botón con el texto 'X' para cada uno de los p que hayas insertado y que si el usuario hace click en este botón eliminemos el parrafo asociado.

Iteración #2: async-await

2.1 Convierte la siguiente promesa para esperar a ejecutar el console usando async-await.

```
const runTimeOut = () => {  
  const promise = new Promise((resolve) => {  
    setTimeout(function () {  
      resolve();  
    }, 2000);  
  })  
  
  promise.then(() => {console.log('Time out!')})  
};  
  
runTimeOut();
```

2.2 Convierte la siguiente función con un fetch utilizando async-await.

Recuerda que para usar .fetch() tendrás que probar el ejercicio en el navegador;

```
function getCharacters () {  
  fetch('https://rickandmortyapi.com/api/character').then(res => res.json()).then(characters => console.log(characters));  
}  
  
getCharacters();
```

Happy coding 🌟 & Happy Upgrader ❤️