**Fetch**

Fetch se utiliza para acceder a peticiones y repuestas

fetch('https://pokeapi.co/api/v2/pokemon/') //peticion a la api de pokemon

    .then( repuesta => repuesta.json()) //aqui se convierte la respuesta a json

    .then( data => { //aqui se guarda la data en una variable

        // console.log(data); //aqui se imprime la data

        // let pokemones = data.results; //aqui se guarda la data en una variable

        // console.log(pokemones)

        data.results.forEach(pokemon => { //aqui se recorre la data

            console.log(pokemon);

        });

    })

    .catch(error => console.log(error)); //aqui se imprime el error si es que falla la peticion

**Async await**

Una función async siempre debe llevar un await

   const obtenerPokemons = async () => {  //aqui se crea una funcion asincrona

        try{

            const url = 'https://pokeapi.co/api/v2/pokemon/'; //aqui se crea la url

            const respuesta = await fetch(url); //aqui se hace la peticion, await, espera a que se realice la peticion

            const data = await respuesta.json(); //aqui se convierte la respuesta a json

            console.log(data.results)

        }

        catch(error){

            console.log(error);

        }

    }

    obtenerPokemons();

**Map**

El map te recorre un array y lo retorna

const resultados = data.results.map( poke => poke.name ) //aqui se recorre la data y se guarda en una variable

      console.log(resultados);

**Filter**

El método filter() crea un nuevo array con todos los elementos que cumplan la condición implementada por la función dada

const resultados = data.results.filter( poke => poke.name === 'bulbasaur') //aquí se filtra la data

            console.log(resultados);