


## Utilidades 2

### Type of

Type of es algo tan útil y tan sencillo como saber el tipo de un dato.... se utiliza tal que así



```
let n = 12;  
  
let str = "Hola";  
  
console.log(typeof n)  
console.log(typeof str)
```

### instalando paquetes de npm

La instalación es super sencilla.... npm i <NombreDelPaquete> -s



### **blocking/io**

Hablemos sobre Block o no blocking IO... Cuando nosotros leemos un archivo con una función nonblocking/IO no podemos obtener el archivo de forma inmediata ni su lectura... por que son acciones que corren en paralelo

### **Sincronia**

una lectura de un archivo síncrona en un archivo corta la ejecución de nuestro programa y espera al resultado, pero a cambio la función retorna el valor



```
const fs = require('fs')

let readBlocking = fs.readFileSync( './pokemon.txt', 'utf8' )
```

## Asincronia

entendamos la asincronia... al no bloquear las acciones estas suceden de forma independiente sin tener en cuenta de si el procesamiento necesario se ha cumplido o no, esclarezcamos esto con un ejemplo



```
const fs = require('fs')

let archivo = "-";

fs.readFile( './pokemon.txt', 'utf8', function(err,data) {
    archivo = data
});

console.log(archivo) // Vacio

setTimeout(() => {
    console.log(archivo) // Despues de la espera el archivo se llena
}, 3000);
```

## **Promesas**

Aquí es donde entran las promesas.... Javascript creo algo para esperar a la lectura del programa sin bloquear la ejecución, esto son las promesas....

Las promesas deben de ejecutarse siempre de un entorno async... si estamos en una función deberemos indicar que es async y si queremos ejecutar una promesa fuera de nuestro código async, tenemos que crear una función anónima.

veamos estos dos casos.



```
//Función asincrona
```

```
async function miFuncion(param1){  
    //Mi await aqui  
}
```

```
//Función anonima asincrona
```

```
(async function(){  
    //Mi await aqui  
})();
```

### **Axios**

Axios es una librería de código que nos permite realizar peticiones http a una determinada dirección, por ejemplo recientemente hemos visto que es una API, una API REST, pero como podemos obtener una api, en nuestro

axios hace uso de los verbos HTTP. el verbo que nos interesa es el get.



```
const axios = require('axios')

async function getUser() {
  const response = await axios.get('http://girlazo.com');
  return response
}

(async function(){
  let data = await getUser()
  console.log(data.data)
})();
```