

Lectura y escritura de archivos

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [writeFileSync\(\)](#)
2. [appendFileSync\(\)](#)
3. [readFileSync\(\)](#)

2 | writeFileSync()

.writeFileSync()

Es un método que trae el paquete nativo **file system** que nos permite escribir archivos. Recibe dos parámetros:

- El primero, el **archivo** en donde queremos escribir.
- El segundo, el **contenido** que queremos escribir.

Algo importante a tener en cuenta es que, si le pasamos el nombre de un archivo que aún no existe, el mismo método se encargará de crearlo.

Otra cosa importante a tener en cuenta es que este método **borra todo el contenido existente** y lo reemplaza por el nuevo.

```
{}
```

```
const fs = require('fs');  
fs.writeFileSync('estrenos-2020.txt', 'Titanic 2');
```

Preparando el contenido

Los **métodos** de **escritura** de archivos que trae **file system** solo pueden recibir contenido que sea de tipo string.

Si lo que queremos guardar es otro tipo de dato de JavaScript, podemos usar el formato JSON. Para convertirlo utilizamos el método `JSON.stringify()`.

```
{  
  {}  
  const fs = require('fs');  
  let pelicula = {  
    titulo: 'Titanic',  
    minutos: 560  
  };  
  let peliculaJson = JSON.stringify(pelicula);  
  fs.writeFileSync('titanic.json', peliculaJson);  
}
```

3 | appendFileSync()

.appendFileSync()

Es un método que trae el paquete nativo **file system** que nos permite agregar contenido a archivos. Recibe dos parámetros:

- El primero, el **archivo** en donde queremos escribir.
- El segundo, el **contenido** que queremos escribir.

Al igual que **writeFileSync()**, si le pasamos el nombre de un archivo que aún no existe, el mismo método se encargará de crearlo.

A diferencia de **writeFileSync()**, este método agrega contenido al final del contenido existente sin borrarlo.

```
{ }
```

```
const fs = require('fs');  
fs.appendFileSync('estrenos-2021.txt', 'Titanic 3');
```

1 | readFileSync()

.readFileSync()

Es un método que trae el paquete nativo **file system** que nos permite **recuperar datos** de un archivo para poder usarlos.

Como **primer parámetro** recibe la **ruta** del archivo que queremos leer.

```
{ } const fs = require('fs');  
let sitcoms = fs.readFileSync('sitcoms.txt');
```

Para poder decodificar los datos que el método devuelve es **fundamental** pasarle un **segundo parámetro** aclarando el tipo de encoding.

```
{ } let sitcoms = fs.readFileSync('sitcoms.txt', 'utf-8');
```

“

Si estamos leyendo un archivo **JSON**, hay que convertir ese string en un **objeto literal** para poder manipular los datos usando el método **JSON.parse()**.



”

JSON.parse()

Pasa los datos en formato JSON a un objeto literal para poder ser manipulado con JavaScript.

```
{ }
```

```
let users = fs.readFileSync('users.json', 'utf-8');  
let usersJson = JSON.parse(users);
```

DigitalHouse>
Coding School