



Leitura e escrita de arquivos



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

readFileSync()

2

writeFileSync()

3

appendFileSync()



1 | **readFileSync()**



.readFileSync()

É um método que traz o pacote nativo **file system** que nos permite **recuperar dados** de um arquivo, para que possamos usá-los.

Como **primeiro parâmetro** recebe a **rota** do arquivo que queremos ler.

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

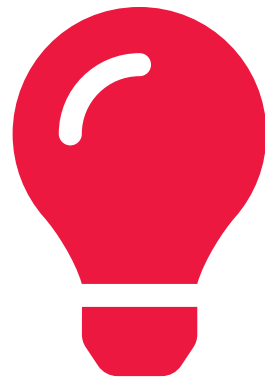
Para poder decodificar os dados que o método devolve é **fundamental** passar um **segundo parâmetro** declarando o tipo de encoding.

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



“

Se estamos lendo um arquivo **JSON**, temos que converter essa string em um **objeto literal** para poder manipular os dados usando o método **JSON.parse()**.



”





JSON.parse()

Passa os dados em formato JSON a um objeto literal para poder ser manipulado com JavaScript.

```
{ }
```

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



2

writeFileSync()



.writeFileSync()

É um método que traz o pacote nativo **file system** que nos permite escrever arquivos. Recebe dois parâmetros:

- O primeiro, o **arquivo** onde queremos escrever.
- O segundo, o **conteúdo** que queremos escrever.

Algo importante a ter em mente é que, se passarmos o nome de um arquivo que ainda não existe, o método **.writeFileSync()** se encarregará de criá-lo.

Outra coisa importante a levar em conta é que este método **apaga todo o conteúdo existente** e o substitui pelo novo.

```
{ }
```

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




Preparando o conteúdo

Os **métodos** de **escrita** de arquivos que traz o pacote **file system** só podem receber conteúdo que seja do tipo string.

Se o que queremos guardar é outro tipo de dado em JavaScript, podemos usar o formato JSON. Para convertê-lo utilizamos o método `JSON.stringify()`.

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

3

| appendFileSync()



.appendFileSync()

É um método que traz o pacote nativo **file system** que nos permite adicionar conteúdo a arquivos. Recebe dois parâmetros:

- O primeiro, o **arquivo** onde queremos escrever;
- O segundo, o **conteúdo** que queremos escrever.

Assim como o **writeFileSync()**, se passarmos o nome de um arquivo que ainda não existe, método **.appendFileSync()** se encarregará de criá-lo.

A diferença do **.appendFileSync** para o **writeFileSync()**, é que ele adiciona conteúdo ao final do conteúdo existente, sem apagá-lo.

```
{ }
```

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

DigitalHouse>
Coding School