

## Primeiro módulo – file system (fs)



O módulo fs permite interagir com o sistema de arquivos do sistema operacional.

Fornece métodos para realizar operações de leitura e gravação de arquivos, criar diretórios, verificar a existência de arquivos, etc.

Aqui estão alguns dos comandos principais e exemplos de uso do módulo fs:

**Leitura de Arquivos:** `fs.readFile()` ou `fs.readFileSync()`

**Escrita de Arquivos:** `fs.writeFile()` ou `fs.writeFileSync()`

**Verificar a Existência de um Arquivo:** `fs.existsSync()`

**Listar Arquivos em um Diretório:** `fs.readdir()`

**Criar um Diretório:** `fs.mkdir()` ou `fs.mkdirSync()`



# Leitura de arquivos

Forma síncrona

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

const data = fs.readFileSync('arquivo.txt', 'utf-8')

console.log(data)
```

Forma assíncrona com callback

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

fs.readFile('arquivo.txt', 'utf8', (erro, data) => {
  if (erro) {
    console.log(erro)
  } else {
    console.log(data)
  }
});
```

# Criação ou modificação de arquivos



Forma síncrona

```
const fs = require('fs')  
  
fs.writeFileSync('arquivo.txt', 'Conteúdo a ser escrito')
```

Forma assíncrona com callback

```
const fs = require('fs')  
  
fs.writeFile('arquivo.txt', 'Conteúdo a ser escrito', () => console.log('sucesso.'));
```

```
const fs = require('fs')  
  
fs.writeFile('arquivo.txt', 'Conteúdo a ser escrito', (erro) => {  
  if (erro) {  
    console.log(erro)  
  } else {  
    console.log('Arquivo foi escrito com sucesso.')  
  }  
});
```



# Verificar a Existência de um Arquivo

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

if (fs.existsSync('arquivo.txt')) {
  console.log('O arquivo existe.')
} else {
  console.log('O arquivo não existe.')
}
```

# Listar arquivos de uma pasta



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

fs.readdir('./pasta', (erro, arquivos) => {
  if (erro) {
    console.log(`Deu erro -- ${erro}`)
  } else {
    console.log('Arquivos na pasta:', arquivos)
  }
});
```



# Criar uma pasta

Forma síncrona

```
const fs = require('fs')  
  
fs.mkdirSync('nova_pasta')
```

Forma assíncrona com callback

```
const fs = require('fs')  
  
fs.mkdir('nova_pasta_2', (erro) => {  
  if (erro) {  
    console.log(`Deu erro -- ${erro}`)  
  } else {  
    console.log('Diretório criado com sucesso.')  
  }  
});
```

# Leitura e edição de um arquivo



```
const fs = require('fs')  
  
let data = fs.readFileSync('arquivo.txt', 'utf-8')  
  
data += '\nNovo conteúdo a ser adicionado.'  
  
fs.writeFileSync('arquivo.txt', data)
```

## Receber dados via terminal



Para conseguirmos receber dados via terminal, semelhante ao prompt, teremos que inicialmente instalar um pacote, minha sugestão é o readline-sync. Para instalar ele, temos que rodar o comando:

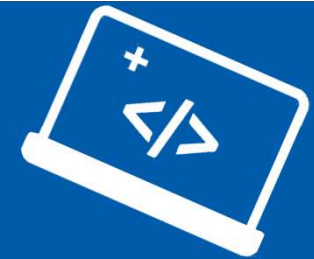
```
npm install readline-sync
```

E para receber dados via terminal, usamos a seguinte sintaxe:

```
const readlineSync = require('readline-sync')  
let nomeInserido = readlineSync.question("Qual é o seu nome?")
```



# Atividades (node fs)



Faça uma aplicação em node.js que permita:

- Fazer a leitura de um arquivo de texto e exibir o resultado no console
- Solicitar a entrada de um texto via terminal e inserir o texto no arquivo, mantendo o que já havia nele.
- Perguntar um nome de arquivo via terminal e verificar se esse arquivo existe na pasta atual ('.')
- Exiba a lista de arquivos da pasta atual