

NODEJS STANDARD LIBRARY

MODULO FS

- ▶ FS è il modulo della libreria standard di NodeJS che permette di interagire con il file system (file e directory). E' agnostico rispetto al file system, quindi funziona allo stesso modo con tutti i file system più utilizzati.
 - ▶ NTFS
 - ▶ FAT32
 - ▶ ext4
- ▶ Permette di creare, cancellare, modificare e interrogare file o directory.

- ▶ Tutti i metodi esposti da FS esistono in doppia versione: sincrona e asincrona:
 - ▶ e.g. `readFileSync()` e `readFile()`
- ▶ Gli argomenti sono gli stessi ma cambia il modo di fruizione del risultato: flusso standard del codice nel primo caso, tramite callback nel secondo.

- ▶ Importiamo il modulo fs

JS

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

TS

```
import * as fs from 'fs';
```

```
import {readFileSync} from 'fs';
```

▶ fs.open() e fs.close()

▶ Input:

- ▶ **path**
String -> Path al file da aprire
- ▶ **flags (opzionale)**
String -> Flags di lettura/scrittura (r/rw)

▶ Restituisce

- ▶ **File descriptor richiesto o un errore**
Il file descriptor è un numero intero che rappresenta il file all'interno del processo. Va usato in tutte le successive operazioni che lo richiedono come parametro. E' importante chiuderlo una volta terminate le operazioni.

▶ Input:

- ▶ **fd**
Integer -> File descriptor da chiudere

▶ Restituisce

- ▶ **undefined**

▶ fs.readFile() e fs.read()

▶ Input:

▶ path

String -> Path al file da leggere

▶ options (opzionale)

Object -> Encoding e flag

▶ Restituisce

▶ data

Viene restituito il contenuto del file come Buffer nel caso di encoding non specificato, o come stringa nel caso di encoding specificato.

▶ Input:

▶ fd

Integer -> File descriptor da leggere

▶ options (opzionale)

Object

▶ Restituisce

▶ bytesRead + buffer

bytesRead è un intero che rappresenta il numero di byte letti. All'interno di buffer sono memorizzati tutti i byte letti. Per ottenere una stringa è necessaria una conversione esplicita.

▶ fs.writeFile() e fs.write()

▶ Input:

- ▶ **path**
String -> Path al file su cui scrivere
- ▶ **data**
String | Buffer -> Encoding e flag
- ▶ **options (opzionale)**
Object -> Encoding, mode e flag

▶ Restituisce

- ▶ **undefined**

▶ Input:

- ▶ **fd**
Integer -> File descriptor su cui scrivere
- ▶ **string**
String -> Stringa da scrivere sul file
- ▶ **position (opzionale)**
Integer -> Offset all'interno del file dove scrivere
- ▶ **encoding (opzionale)**
String -> Codifica del parametro string

▶ Restituisce

- ▶ **undefined**

▶ fs.stat()

▶ Input:

▶ path

String -> Path del file o cartella di cui estrarre informazioni

▶ Restituisce

▶ stat

Un oggetto di tipo fs.Stat che racchiude tutte le informazioni del file o della cartella, come dimensioni, data di creazione, data di ultima modifica, permessi etc.