

NodeJS

- File System - 文件系统模块 - require('fs')

- 该模块是核心模块，需要使用require导入后使用
- 该模块提供了操作文件的一些API
 - fs.open(path, flags, [mode], callback)
 - 异步版的打开一个文件
 - fs.openSync(path, flags, [mode])
 - fs.open() 的同步版
 - fs.read(fd, buffer, offset, length, position, callback)
 - 从指定的文档标识符fd读取文件数据。
 - fs.readSync(fd, buffer, offset, length, position)
 - fs.read 函数的同步版本。返回bytesRead的个数。

妙味译

open 异步

`fs.open(path, flags, [mode], callback)`

`path`: 要打开文件的路径

`flags`: 打开文件的方式 读/写 `'r'` – 以读取模式打开文件。如果文件不存在则发生异常。

■ `'r+'` – 以读写模式打开文件。如果文件不存在则发生异常。

■ `'rs+'` – 以同步读写模

式打开文件。
命令操作系统
绕过本地文件
系统缓存。

mode: 设置文件的模式 读/写/执行

callback: 执行方法——回调

err: 文件打开失败的错误保存在err里，成功为null

fd: 被打开文件的标识

```
fs.open('1.txt', 'r', function (err, fd) {  
  console.log(fd)  
})  
console.log('ok');
```

openSync open的同步版本

```
var fd = fs.openSync('1.txt', 'r');  
console.log(fd)
```

read 方法

```
var fs = require('fs');  
fs.open('1.txt', 'r', function (err, fd) {  
  if(err) {  
    console.log("文件打开失败")  
  }else{  
    console.log("文件打开成功")  
    // fs.read(fd, buffer, offset, length, position, callback)  
    // buffer:buffer对象  
    // offset:新的内容添加到buffer中的其实位置  
    // length:添加到buffer中内容的长度  
    // position:读取的文件中的其实位置
```

```
// callback:回调
//      err
//      lenth
//      buffer
var bf1 = Buffer.alloc(10);
console.log(bf1);
fs.read(fd, bf1, 0, 4, 0, function (err, len, new_bf) {
    console.log(bf1) //buffer
    console.log(len) //4
    console.log(new_bf) //buffer
})
}
})
```

readSync read的同步版本