

## NodeJS

- File System - 文件系统模块 - `require('fs')`
  - `fs.writeFile(filename, data, [options], callback)`
    - 异步的将数据写入一个文件,如果文件不存在则新建, 如果文件原先存在, 会被替换。`data` 可以是一个string, 也可以是一个原生buffer。
  - `fs.writeFileSync(filename, data, [options])`
    - `fs.writeFile`的同步版本。注意: 没有callback, 也不需要
  - `fs.appendFile(filename, data, [options], callback)`
    - 异步的将数据添加到一个文件的尾部, 如果文件不存在, 会创建一个新的文件。`data` 可以是一个string, 也可以是原生buffer。
  - `fs.appendFileSync(filename, data, [options])`
    - `fs.appendFile`的同步版本。



## writeFile

向指定文件中写入数据, 如果不存在会新建一个文件, 如果存在会覆盖原有文件

```
var fs = require('fs');  
var filename = '2.txt';  
fs.writeFile(filename, 'hello', function () {  
    console.log(arguments)  
})
```

## appendFile

向指定文件后添加数据, 如果不存在会新建一个文件, 如果存会在后面添加

```
fs.appendFile(filename, "新加数据", function () {  
    console.log(arguments)  
})
```

## NodeJS

- File System - 文件系统模块 - `require('fs')`
  - `fs.readFile(filename, [options], callback)`
    - 异步读取一个文件的全部内容
  - `fs.readFileSync(filename, [options])`
    - `fs.readFile`的同步版本
  - `fs.exists(path, callback)`
    - 检查指定路径的文件或者目录是否存在
  - `fs.existsSync(path)`
    - `fs.exists`的同步版本
  - `fs.unlink(path, callback)`
    - 删除一个文件
  - `fs.unlinkSync(path)`
    - `fs.unlink()` 的同步版本

## fs.exists

检查指定路径的文件是否存在

## 异步处理

```
fs.exists(filename, function (isExists) {  
    console.log(isExists)  
    if(!isExists) {  
        fs.writeFile(filename, 'miaowei', function (err) {  
            if(err) {  
                console.log('出错了')  
            }else{  
                console.log('创建成功并写入了新内容')  
            }  
        })  
    }else {  
        fs.appendFile(filename, "存在的时候加的内容", function (err) {  
            if(err) {  
                console.log('出错了')  
            }else{  
                console.log('增加了新内容')  
            }  
        })  
    }  
})
```

```
}}
```

## 同步处理

```
if(!fs.existsSync(filename)) {  
    fs.writeFileSync(filename, 'miaowei')  
    console.log("新文件创建成功")  
} else {  
    fs.appendFileSync(filename, "-leo")  
    console.log("新内容追加成功")  
}
```

## readFile 读取文件

```
fs.readFile('2.txt', function (err, data) {  
    console.log(err)  
    console.log(data)  
    if(err) {  
        console.log('文件读取失败');  
    } else {  
        console.log(data.toString());  
    }  
})
```

## unlink 删除文件

```
fs.unlink('2.txt', function (err) {  
    if(err) {  
        console.log('删除失败')  
    } else {  
        console.log('删除成功')  
    }  
})
```

## NodeJS

- File System - 文件系统模块 - require('fs')
  - fs.rename(oldPath, newPath, callback)
    - 重命名
  - fs.renameSync(oldPath, newPath)
    - fs.rename() 的同步版本
  - fs.stat(path, callback)
    - 读取文件信息
  - fs.statSync(path, callback)
    - fs.stat() 的同步版本
  - fs.watch(filename, [options], [listener])
    - 观察指定路径的改变, filename 路径可以是文件或者目录

## rename 重命名文件

```
fs.rename(filename, '2.new.txt', function () {  
    console.log(arguments)  
})
```

## stat 读取文件信息

```
fs.stat(filename, function () {  
    console.log(arguments)  
})
```

返回文件信息

## watch 监听文件改变

```
fs.watch(filename, function (ev, fn) {  
    console.log(ev)  
    if(fn) {  
        console.log(fn+' 发生了改变')  
    } else {  
        console.log('....')  
    }  
})
```