

fs核心模块

Node.js的文件系统的Api,同步方法和异步方法同时出现

读取文件

- 异步读取 `readFile`
- 同步读取 `readFileSync`

写文件

- 异步写入 `writeFile`
- 同步写入 `writeFileSync`

拷贝文件

目录操作

fs核心模块

读取文件

写文件

目录操作

- 创建目录
- 判断一个文件是否存在
- 读取目录下所有的文件
- 查看文件目录信息

路径处理path模块

- `path.join`
- `path.basename`
- `path.extname`
- `path.sep`
- `path.delimiter`
- `path.normalize`
- `resolve`
- 创建目录

创建目录

```
fs.mkdirSync('a/b/c/d');
```

要求父目录必须存在

判断一个文件是否存在

```
fs.existsSync('./a');
```

读取目录下所有的文件

```
fs.readdirSync('');
```

查看文件目录信息

```
fs.statSync('./e/'+item);
```

路径处理path模块

path是node中专门处理路径的一个核心模块

path.join

将多个参数值字符串结合为一个路径字符串

fs核心模块

读取文件

写文件

目录操作

- 创建目录
- 判断一个文件是否存在
- 读取目录下所有的文件
- 查看文件目录信息

路径处理path模块

- path.join
- path.basename
- path.extname
- path.sep
- path.delimiter
- path.normalize
- resolve
- 创建目录

path.basename

获取一个路径中的文件名

path.extname

获取一个路径中的扩展名

path.sep

操作系统规定的文件分隔符

path.delimiter

属性值为系统指定的环境变量路径分隔符

path.normalize

将非标准的路径字符串转化为标准路径字符串 特点：

- 可以解析 . 和 ..
- 多个杠可以转换成一个杠
- 在windows下反杠会转化成正杠
- 如结尾以杠结尾的，则保留斜杠

resolve

fs核心模块

读取文件

写文件

目录操作

- 创建目录
- 判断一个文件是否存在
- 读取目录下所有的文件
- 查看文件目录信息

路径处理path模块

- path.join
- path.basename
- path.extname
- path.sep
- path.delimiter
- path.normalize
- resolve
- 创建目录

取得绝对路径 特点：

- 以应用程序根目录为起点
- 如果参数是普通字符串，则意思是当前目录的下级目录
- 如果参数是.. 回到上一级目录
- 如果是/开头表示一个绝对的根路径

创建目录

- 正向创建

```
function mkdirp(p) {
  var arr = p.split('/');
  var index = 0;
  make(arr[index]);
  function make(p) {
    if(index >= arr.length+1){
      return
    }
    fs.mkdir(p, function (err) {
      make(arr.slice(0, ++index+1).join('/'))
    })
  }
}
```

- 逆向创建

```
function mkdirs(p, fn) {
  fs.exists(p, function (exists) {
    if(exists){
      fn();
    }else{
```

fs核心模块

读取文件

写文件

目录操作

- 创建目录
- 判断一个文件是否存在
- 读取目录下所有的文件
- 查看文件目录信息

路径处理path模块

- path.join
- path.basename
- path.extname
- path.sep
- path.delimiter
- path.normalize
- resolve
- 创建目录

```
    mkdirs(path.dirname(p),function () {  
        fs.mkdir(p,fn);  
    })  
}  
})  
}
```

fs核心模块

读取文件

写文件

目录操作

- 创建目录
- 判断一个文件是否存在
- 读取目录下所有的文件
- 查看文件目录信息

路径处理path模块

- path.join
- path.basename
- path.extname
- path.sep
- path.delimiter
- path.normalize
- resolve
- 创建目录