

# Node.js中的流



## fs 模块读写方法

用途	异步	同步
将文件作为 整体 读入缓存区	readFile	readFileSync
将数据作为 整体 写入文件	writeFile	writeFileSync

## 流的概念

- 流是一组 **有序** 的，有 **起点** 和 **终点** 的 **字节数据传输** 手段
- 不关心文件的整体内容，只关注是否从文件中 **读** 到了数据，以及读到数据之后的 **处理**
- 流是一个 **抽象接口**，被 Node 中的很多对象所实现。比如对一个 HTTP 服务器的请求对象 request 是一个流，stdout 也是一个流。

## stream.Readable可读流

使用实现了stream.Readable接口的对象来将对象数据读取为流数据,在您表明您 **准备好** 接收之前,Readable 流并不会开始发射数据。

# ReadStream文件可读流

```
fs.createReadStream(path,[options]);
```

- path 读取的 **文件路径**
- options
  - flags 对文件采取何种操作,默认为 'r'
  - encoding 指定 **编码** , 默认为null
  - autoClose 读完数据后是否关闭文件描述符
  - start 用整数表示文件 **开始** 读取的字节数的索引位置
  - end 用整数表示文件 **结束** 读取的字节数的索引位置( **包括end位置** )
  - highWaterMark 最高水位线, 停止从底层资源读取前内部缓冲区最多能存放的字节数。缺省为 64kb



## 可读流触发的事件

事件	用途
<b>data</b>	绑定一个 <code>data</code> 事件监听器到会将流切换到 <b>流动模式</b> ，数据会被尽可能快的读出
<b>end</b>	该事件会在 <b>读完</b> 数据后被触发
<b>error</b>	当数据接收时发生 <b>错误</b> 时触发

## 可读流的方法

方法	描述
<b>setEncoding</b>	指定 <b>编码</b>
<b>pause</b>	通知对象 <b>停止</b> 触发data事件
<b>resume</b>	通知对象 <b>恢复</b> 触发data事件
<b>pipe</b>	设置 <b>管道</b> ，将可读流里的内容导入到参数指定的 <b>可写流</b> 里

# Writable可写流

使用各种实现stream.Writable接口的对象来将流数据 **写入** 到对象中





## 可写流的方法

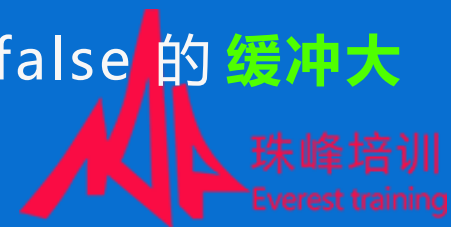
方法	描述
<code>write</code>	写入数据
<code>end</code>	<b>结束写入</b> 数据时触发。迫使缓存区中的数据立即写入目标对象，调用后不能再写入

# WriteStream

在fs模块中使用 `createWriteStream` 方法创建一个将流数据写入文件中的 `WriteStream` 对象

```
fs.createWriteStream(path,[options]);
```

- path 读取的文件路径
- options
  - flags 对文件采取何种 **操作**, 默认为 'w'
  - encoding 指定 **编码**, 默认为 null
  - autoClose 是否 **关闭** 文件描述符
  - start 用整数表示文件 **开始** 字节数的写入位置
  - highWaterMark 最高水位线, write() 开始返回 false 的 **缓冲大小**。缺省为 16kb



## write方法

```
writable.write(chunk,[encoding],[callback]);
```

- 参数
  - chunk 要 **写入** 的数据，Buffer或字符串对象，必须指定
  - encoding 写入 **编码**，chunk为字符串时有用，可选
  - callback 写入成功后的 **回调**
- 返回值为布尔值，系统缓存区定满时为false,未满时为true

## end方法

在写入文件时，当不再需要写入数据时可调用该方法关闭文件。迫使系统缓存区的数据立即写入文件中。

```
writable.end(chunk,[encoding],[callback]);
```

# pipe

流，尤其是 pipe() 方法的初衷，是将数据的 **滞留量** 限制到一个可接受的水平，以使得不同速度的来源和目标不会 **淹没** 可用内存。

```
readStream.pipe(writeStream,[options]);
```

- readStream可读流对象
- writeStream可写流对象
- options
  - end 为true时表示数据读取完毕后立刻将缓存区中的数据写入目标文件并 **关闭** 文件