

# 基本知识

## 四种流类型

序号	名称	操作
1	Readable	可读操作
2	Writable	可写操作
3	Duplex	可读可写操作
4	Transform	操作被写入数据，然后读出信息

所有的Stream对象都是EventEmitter的实例。常用的事件：

序号	名称	操作
1	data	当有数据在可读时触发
2	end	没有更多的数据可读时触发
3	error	在接收和写入过程中发生错误时触发
4	finish	所有数据已被写入到底层系统时触发

# 从流中读取数据

```
1  var fs = require('fs');
2  var data = '';
3
4  //创建可读流
5  var readerStream = fs.createReadStream('input.txt');
6
7  //设置编码为'utf8'
8  readerStream.setEncoding('utf8');
9
10 //处理流事件 -->data, end, and error
11 readerStream.on('data',function (chunk) {
12     data += chunk;
13 });
14
15 readerStream.on('end',function () {
16     console.log(data);
17 });
18
19 readerStream.on('error',function (err) {
20     console.log(err.stack);
21 });
22
```

```
console.log("程序执行完毕");
```

## 写入流

```
1 var fs = require(&apos;fs&apos;);
2 var data = &apos;索拉卡大家拉克斯基的拉萨看得见啊离开家&apos;;
3
4 //创建一个可以写入的流，写到文件 output.txt中
5 var writeStream = fs.createWriteStream(&apos;output.txt&apos;);
6
7 //使用utf8编码写入数据
8 writeStream.write(data,&apos;UTF8&apos;);
9
10 //标记文件末尾
11 writeStream.end();
12
13 //处理流事件
14 writeStream.on(&apos;finish&apos;,function( ) {
15     console.log("写入完成");
16 });
17
18 writeStream.on(&apos;error&apos;,function(err) {
19     console.log(err.stack);
20 });
21
22 console.log("程序执行完毕");
```

## 管道流

```
1 var fs = require(&apos;fs&apos;);
2
3 //创建一个可读流
4 var readerStream = fs.createReadStream(&apos;input.txt&apos;);
5
6 //创建一个可写流
7 var writeStream = fs.createWriteStream(&apos;output.txt&apos;);
8
9 //管道读写操作
10 //读取input.txt 文件内容，并将内容写入到output文件中
11 readerStream.pipe(writeStream);
12
13 console.log("程序执行完毕");
```

## 链式流

链式是通过连接输出流到另外一个流并创建多个流操作链的机制。链式流一般用于管道操作。

### 压缩文件

```
1 var fs = require(&apos;fs&apos;);
```

```
2  var zlib = require('zlib');
3
4  //压缩input.txt文件为input.txt.gz
5  fs.createReadStream('input.txt')
6    .pipe(zlib.createGzip())
7    .pipe(fs.createWriteStream('input.txt.gz'));
8
9  console.log('文件压缩完成');
```

## 解压文件

```
1  var fs = require('fs');
2  var zlib = require('zlib');
3
4  //解压input.txt.gz文件为input.txt
5  fs.createReadStream('input.txt.gz')
6    .pipe(zlib.createGunzip())
7    .pipe(fs.createWriteStream('input.txt'));
8
9  console.log('文件解压完成');
10
```