**Stream 流类型**

- Readable - 可读操作。

- Writable - 可写操作。

- Duplex - 可读可写操作.

- Transform - 操作被写入数据，然后读出结果。

**Stream 对象事件**

所有的 Stream 对象都是 EventEmitter 的实例。常用的事件有：

- data - 当有数据可读时触发。

- end - 没有更多的数据可读时触发。

- error - 在接收和写入过程中发生错误时触发。

- finish - 所有数据已被写入到底层系统时触发。

**从流中读取数据**

创建 input.txt 文件，内容如下：

菜鸟教程官网地址：www.runoob.com

读取示例

var fs = require("fs");

var data = '';

// 创建可读流

var readerStream = fs.createReadStream('input.txt');

// 设置编码为 utf8。

readerStream.setEncoding('UTF8');

// 处理流事件 --> data, end, and error

readerStream.on('data', function(chunk) {

   data += chunk;

});

readerStream.on('end',function(){

   console.log(data);

});

readerStream.on('error', function(err){

   console.log(err.stack);

});

**写入流**

写入示例

var fs = require("fs");

var data = '菜鸟教程官网地址：www.runoob.com';

// 创建一个可以写入的流，写入到文件 output.txt 中

var writerStream = fs.createWriteStream('output.txt');

// 使用 utf8 编码写入数据

writerStream.write(data,'UTF8');

// 标记文件末尾（从文件尾部开始写入）

writerStream.end();

// 处理流事件 --> data, end, and error

writerStream.on('finish', function() {

    console.log("写入完成。");

});

writerStream.on('error', function(err){

   console.log(err.stack);

});

**管道流**

管道流通常用于从一个流中获取数据并将数据传递到另外一个流中

示例，设置 input.txt 文件内容如下：

菜鸟教程官网地址：www.runoob.com

管道流操作实例

示例代码如下：

var fs = require("fs");

// 创建一个可读流

var readerStream = fs.createReadStream('input.txt');

// 创建一个可写流

var writerStream = fs.createWriteStream('output.txt');

// 管道读写操作

// 读取readerStream内容，并将内容写入writerStream中

readerStream.pipe(writerStream);

来看一个有趣的代码，从这里可以看出标准输出可标准错误也是一个流

const { exec } = require('child\_process');

// 执行命令

const cprocess = exec(cmd, options);

// 将输出重定向到主进程

cprocess.stdout.pipe(process.stdout);

cprocess.stderr.pipe(process.stderr);

**链式流**

链式流就是管道流，如下看代码

var fs = require("fs");

var zlib = require('zlib');

// 压缩 input.txt 文件为 input.txt.gz

fs.createReadStream('input.txt')

    // zlib.createGzip()创建一个Gzip流, 执行pipe之后返回Gzip流

    .pipe(zlib.createGzip())

    // 创建一个写入流，并将Gzip流的数据放到写入流中

.pipe(fs.createWriteStream('input.txt.gz'));