

# IN / OUT(I/O)

## NodeJS

- global
  - process 对象
    - stdin、stdout: 标准输入输出流 (IO)
      - 标准输入设备: <http://baike.baidu.com/view/632680.htm>
      - 标准输出设备: <http://baike.baidu.com/view/632676.htm>
      - stdin和stdout提供了操作输入数据和输出数据的方法, 我们也通常称为IO操作
    - stdin
      - 标准输入流
    - stdout
      - 标准输出流

加小点

<https://baike.baidu.com/item/%E8%BE%93%E5%85%A5%E8%AE%BE%E5%A4%87/10823368?fr=aladdin>

★ 收藏 | 936 | 35

## 输入设备 锁定

 本词条由“科普中国”百科科学词条编写与应用工作项目 审核。

输入设备：向计算机输入数据和信息的设备。是计算机与用户或其他设备通信的桥梁。输入设备是用户和计算机系统之间进行信息交换的主要装置之一。键盘，鼠标，摄像头，扫描仪，光笔，手写输入板，游戏杆，语音输入装置等都属于输入设备。输入设备（Input Device）是人或外部与计算机进行交互的一种装置，用于把原始数据和处理这些数的程序输入到计算机中。计算机能够接收各种各样的数据，既可以是数值型的数据，也可以是各种非数值型的数据，如图形、图像、声音等都可以通过不同类型的输入设备输入到计算机中，进行存储、处理和输出。

<https://baike.baidu.com/item/%E8%BE%93%E5%87%BA%E8%AE%BE%E5%A4%87/10823333?fr=aladdin>

★ 收藏 | 941 | 32

## 输出设备 锁定

 本词条由“科普中国”百科科学词条编写与应用工作项目 审核。

输出设备（Output Device）是计算机硬件系统的终端设备，用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。也是把各种计算结果数据或信息以数字、字符、图像、声音等形式表现出来。常见的输出设备有显示器、打印机、绘图仪、影像输出系统、语音输出系统、磁记录设备等。

## process.stdin#

process.stdin 属性返回连接到stdin(fd 0)的流。它是一个net.Socket(它是一个Duplex流)，除非 fd 0指向一个文件，在这种情况下它是一个[可读]流。

举个例子：

```
process.stdin.setEncoding('utf8');
```

```
process.stdin.on('readable', () => {  
  const chunk = process.stdin.read();  
  if (chunk !== null) {
```

```

    process.stdout.write(`data: ${chunk}`);
  }
});

```

```

process.stdin.on('end', () => {
  process.stdout.write('end');
});

```

`process.stdin` 返回的 `Duplex` 流，可以在旧模式下使用，兼容node v0.10。 更多信息查看流的兼容性。

```

process.stdin.on('data', function (data) {
  console.log("用户输入了"+data)
})

```

## .stdin.resume()

注意：在“旧模式下” `stdin`流 默认是暂停的. 所以必须通过执行`.stdin.resume()`来恢复它. 同时

`process.stdin.resume()`会切换到旧模式

## process.stdout#

`process.stdout` 属性返回连接到 `stdout` (`fd 1`)的流。 它是一个`net.Socket` (它是一个`Duplex`流)， 除非 `fd 1` 指向一个文件，在这种情况下它是一个[可写][]流。

例1： 将输入流数据输出到输出流，即输出到终端。

```

process.stdin.pipe(process.stdout);

```

例2： 要求用户输入两个数值，然后把和输出到终端。

```

/*1:声明变量*/
var num1, num2;
/*2: 向屏幕输出，提示信息，要求输入num1*/
process.stdout.write('请输入num1的值: ');
/*3: 监听用户的输入*/
process.stdin.on('data', function (chunk) {
  if (!num1) {
    num1 = Number(chunk);
    /*4: 向屏幕输出，提示信息，要求输入num2*/
    process.stdout.write('请输入num2的值');
  } else {
    num2 = Number(chunk);
    process.stdout.write('结果是: ' + (num1 + num2));
  }
});

```

注意：重要的是`process.stdout`不同于 `Node.js` 的其他流，详情可以参考[note on process I/O](#) .

## 输出（模拟console.log）

```

function log(msg) {

```

```
        process.stdout.write(msg)
    }
    log("hello, everybody")
```