Node.js 有多个内置的事件，我们可以通过引入 events 模块，并通过实例化 EventEmitter 类来绑定和监听事件，如下实例：

// 引入 events 模块

var events = require('events');

// 创建 eventEmitter 对象

var eventEmitter = new events.EventEmitter();

以下程序绑定事件处理程序：

// 绑定事件及事件的处理程序

eventEmitter.on('eventName', eventHandler);

我们可以通过程序触发事件：

// 触发事件

eventEmitter.emit('eventName');

// 引入 events 模块

var events = require('events');

// 创建 eventEmitter 对象

var eventEmitter = new events.EventEmitter();

// 创建事件处理程序

var connectHandler = function connected() {

console.log('连接成功。');

// 触发 data\_received 事件

eventEmitter.emit('data\_received');

}

// 绑定 connection 事件处理程序

eventEmitter.on('connection', connectHandler);

// 使用匿名函数绑定 data\_received 事件

eventEmitter.on('data\_received', function() {

console.log('数据接收成功。');

});

// 触发 connection 事件

eventEmitter.emit('connection');

console.log("程序执行完毕。");