在nodejs中，模块与文件一一对应。分为核心模块、本地文件模块、npm第三方模块

模块流程：创建模块、导出模块（[module.]exports.xxx=xxx，向模块中添加元素）、加载模块（require(模块文件/模块名)，返回模块，其实就是上一步的exports对象）、使用模块（使用上一步返回的模块，模块.xxx）

**http模块**

**http.createServer([requestListener])** 创建服务器，返回一个新的http.server对象。

参数：function(req,res)，req是请求信息(http.ClientRequest对象)，res是应答信息（http.ServerResponse对象）。res.statusCode设置返回状态码，如200；res.end(信息)设置响应信息。

**http.request(options[, callback])** 用于发起一个http请求。返回一个http.ClientRequest对象

参数2：function(res)

**http.get(options[, callback])** 发起一个get请求，它与http.request的唯一区别是它是get方式并且自动执行req.end()

**Class:http.server**

**server.listen(handle[, callback])** server事件监听

**server.listen(path[, callback])**

**server.listen(port[, hostname][, backlog][, callback])**

**Class: http.ServerResponse**

**response.statusCode** 设置状态码，如200

**response.setHeader(name, value)** 设置响应头，如response.setHeader('Content-Type', 'text/html')、response.setHeader('Set-Cookie', ['type=ninja', 'language=javascript'])

**response.writeHead(statusCode[, statusMessage][, headers])** 设置状态码、响应头，如response.writeHead(200, {'Content-Length': body.length,'Content-Type': 'text/plain' });

如果有两个参数，是第1、3个参数

参数2：有可读性的状态信息（文字）

**response.write(chunk[, encoding][, callback])** 设置返回的响应信息

**response.end([data][, encoding][, callback])** 结束响应，返回响应信息

**url模块**

**url.parse(urlStr[, parseQueryString][, slashesDenoteHost])** 将url转化为一个url js对象（其中带有url信息）

参数2：是否将query（？后面的字符串）字符串解析为对象，默认false。

参数3：此参数为true的时候，若不带有协议（如http）解析不会出错，默认false。

**url.format(urlObj)** 将一个url js对象转换为一个url

**url.resolve(from, to)** 连接url的两部分，如url.resolve('http://example.com/', '/one')，返回'http://example.com/one'

**querystring模块**

**querystring.stringify(obj[, sep][, eq][, options])** 将js对象序列化为一个query字符串，默认为url样式

参数2：&号的替换字符串

参数3：=号的替换字符串

**querystring.parse(str[, sep][, eq][, options])** 将query字符串反序列化为一个js对象，默认分隔符为&号和=号

参数2：替换&号的分隔符

参数3：替换=号的分隔符

参数4：最多项数，设为0表示无限制，默认为1000

**Querystring.escape（str）** 转义字符串

**Querystring.unescape（str）** 反转义字符串