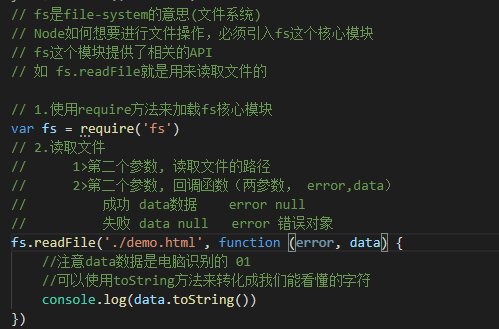
1. Nodejs引入fs核心模块（带实例读文件）-->



1. 运行nodejs



注意：需cd到运行文件所在的文件夹位置运行

1. Js不能读文件, js文件能操作html, node能读文件, node不能操作html。
2. 写入文件-->

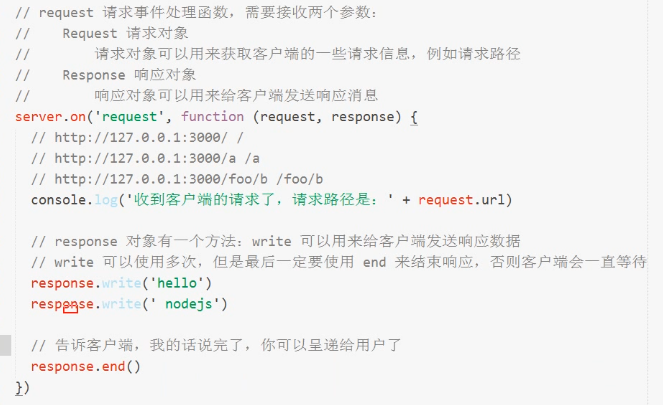


1. 构建服务器基本操作

注意：在浏览器访问设置的地址就能访问



1. 响应(response)-->



1. 响应数据简写(推荐使用)-->

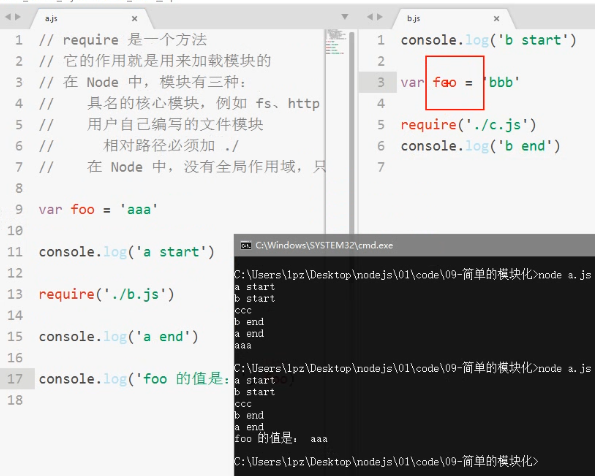


注意：返回东东西必须是字符串(可以使用JSON.strigify方法把数组转化成字符串形式)或者二进制数

1. 端口号80不用输入端口就就能访问， 其他的都要输入端口号

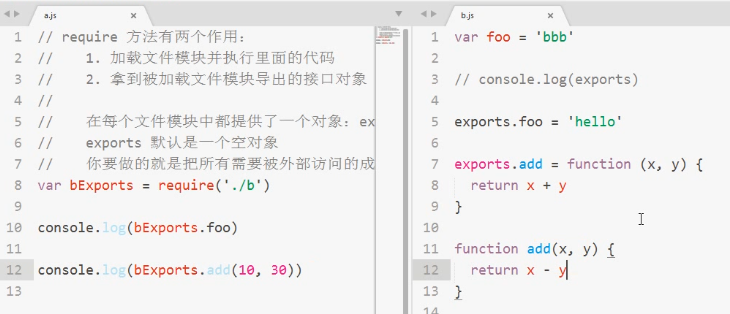
注意: (端口号被占用了就不可使用)

1. 模块化
2. .require只是是加载文件执行里面的代码,可以用readFile读里面的代码(无法访问块里面的成员(无论是方法还是声明的变量)，只执行块里面的代码，默认情况下模块是封闭的) -->



注意：模块没有全局作用域，只有一个块级作用域。

1. .require可以拿到加载文件模块导出对象(请求模块里面的exports)这样就能实现简单的通信了-->



注意：请求里面的模块必须挂载到了exports才能访问。(例子中注释exports.add无法访问add方法)

1. IP地址和端口号的区别。

IP地址定位计算机， 端口号定位具体的引用程序。

(所有联网的应用程序必须占用一个端口号)

1. 响应内容的Content-Type

浏览器不知道服务器响应内容编码类型的时候会按当前操作系统的默认编码去解析内容 --- 中文操作系统是 gbk (出现写的中文和自己想要解析的不符， 不是utf-8)

解决: response.setHeader(‘Content-Type’, ‘text/plain’; charset=’utf-8’);-->



红标(text/plain)就是用来告知对方发送的内容是什么类型渲染成什么样 (具体需要什么内容可查表)-->



1. 发送文件的Content-Type(利用读文件， 读文件是动态的， 修改读读到的文件不需要重启服务器)

在server.on(‘require’, function (req, res) {})里面的内容-->



注意: -->

