# 1: fs.readFile与fs.readFileSync

fs.readFile(filename[, options], callback)#

fs.readFileSync(filename[, options])#

* 返回文件的内容
* 如果在options中指定编码，则返回字符串，如果没有在options中指定编码，则返回buffer。

这两个方法的主要区别是一个是以同步的方式获取文件，一个是用异步的方式获取文件。

由于自己在开始写代码的时候没有弄懂这两个方法的区别，于是用了fs.readFile().后来看到作者用的是fs.readFileSync.在这种情况下就是应该要使用fs.readFileSync.在这里使用这个方法的话，对于代码的可读性更高。

# 2：dns查询在后台可以操作，但是在返回页面的时候就不可以？

**var *dns***=*require*(**'dns'**);  
**var *hostname***=**"www.baidu.com"  
  
*dns***.lookup(***hostname***, **function** (err, address, family ){  
  
 **search\_result**=**"Domain:"**+***hostname***+**" IP:"**+address;  
 ***console***.log(**search\_result**);  
});

Domain:www.baidu.com IP:119.75.218.70

现在可以了，出现这种现象的原因在于异步编程。

req.on(**'data'**,**function**(chunked){  
 hostname=***querystring***.parse(chunked).search;  
  
 ***dns***.lookup(hostname, **function** (err, address, family ){  
  
 search\_result=**"Domain:"**+hostname+**"IP:"**+address;  
  
 ***console***.log(search\_result);  
 res.writeHead(200,{**'content-type'**:**"text/plain"**});  
  
 res.end(search\_result);  
 });  
  
});

原来并没有把dns的查找放到data事件的内部，因此出现了问题。

所以我觉得在nodejs中异步编程是一个难点。自己不注意可能就会出现问题。

3：重构自己写的代码.

在开始写代码的时候，业务逻辑肯定是比较复杂的，如果每个模块各司其职的话，那么整个的代码量就会减少很多。我们需要把各个部分的工作抽象开来。