脚本化HTTP

Comet：web服务器发起通信并异步发送消息到客户端（服务器推、ajax推、HTTP流）

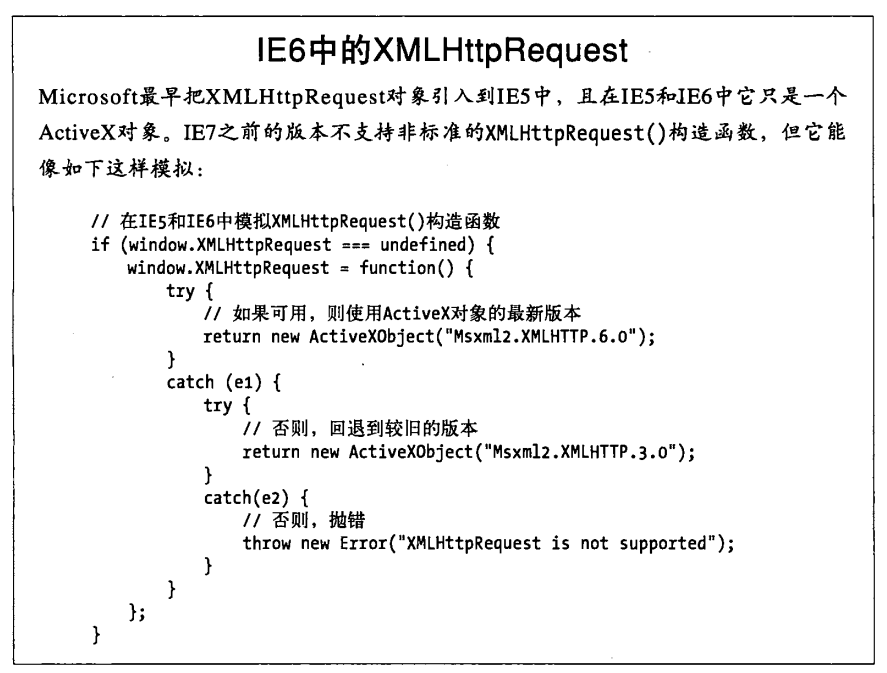
Ajax：客户端重服务器拉数据（操作json格式的数据--->JSONP）

RPC（Remote Procedure Call，远程过程调用）机制或发布/订阅事件系统的基础：在ajax/comet上更高级的通信协议

HTTP API 使用

1：实例化XMLHttpRequest

var request = new XMLHttpRequest();



2、HTTP请求4组成

a）HTTP请求方法或“动作”（verb）

b）正在请求的url

c）一个可选的请求头集合，肯恶搞包括身份验证信息

d）可选的请求主体

3、服务器返回http响应

a）文字+数字的状态码，用来显示请求的成功和失败

b）响应头集合

c）响应主体

4、xmlHttpRequest：同源策略

5、指定请求：open()

request.open(‘GET’//请求方式,’data.csv’//请求主体，‘‘，’用户名‘，’密码‘)

请求方式：GET、POST、DELETE、HEAD、OPRIONS、PUT，不区分大小写的，HTTPCONNECT、TRACE、TRACK安全风险被禁止

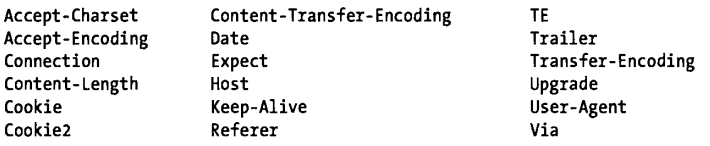
请求主体：相对文档的url

6、设置请求头：setRequestHeader()

request.setRequeestHeader(‘Content-Type’,’text/plain’);

禁止的头信息：Content-Length、Date、Referer、User-Agent，XMLHttpRequest自动添加防止伪造

处理cookie、连接事件、字符集、编码判断，无法传入以下头信息



7、发送消息：send()

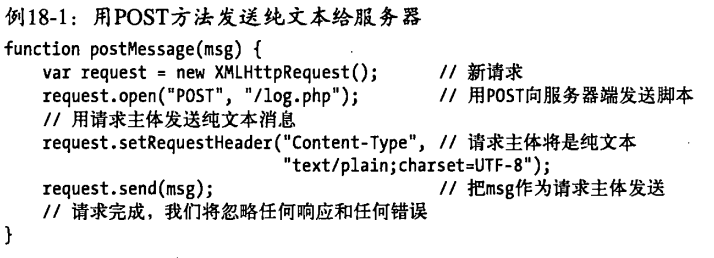
GET：request.send(null)，get请求没有主体，传null或省略这个参数

POST：request.send(data)，POST有请求主体，需要设置头信息

8、信息发送顺序

a）HTTP：请求方法和url先到、然后请求头、最后请求主体

b）XML：调用send()方法才启动网络，api的设计都写进了网络，顺序必须匹配http



9、完整的响应

状态码、响应头集合、响应主体

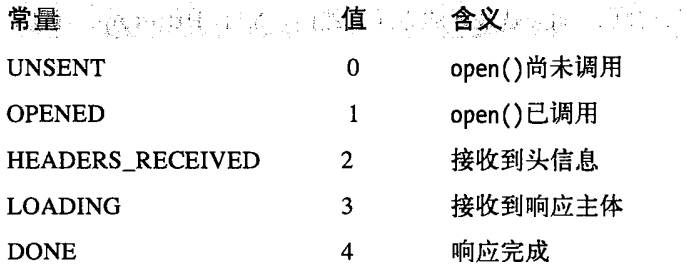
10、响应头

a）状态码：status（数字），statusText（文本）

b）响应头集合：getReponseHeader()，getAllResponseHeaders()【这个会过滤cookie】

c）响应主体：responseText：文本形式，属性只能用于文本，不能妥善处理二进制响应，xhr2可以处理二进制响应的方法

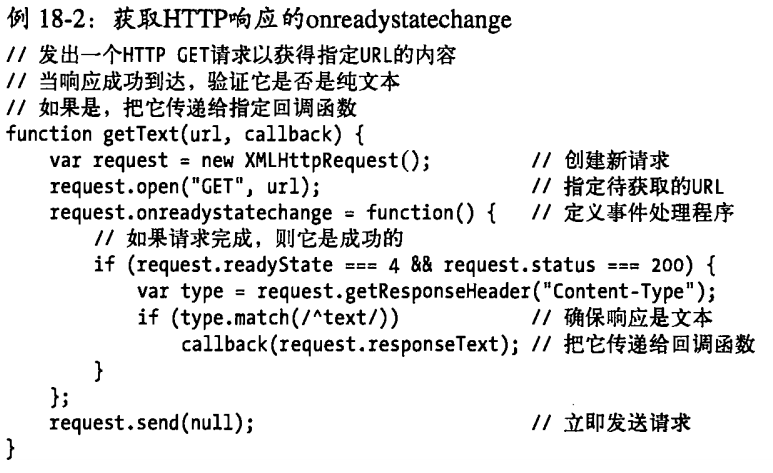
11、readyState：指定HTTP的请求状态，readyState属性改变触发readystatechange事件



0、1：可能没有触发事件，send调用后还是保持1状态

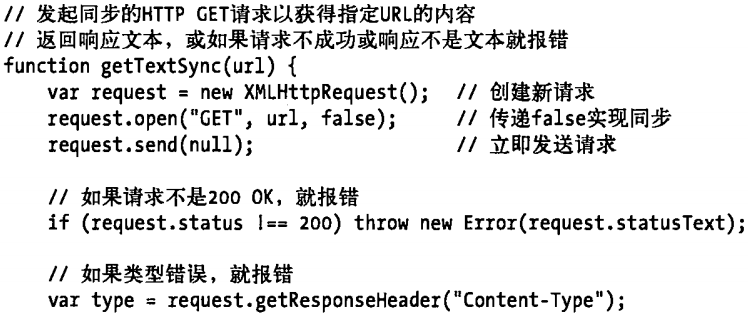
3：触发可以给出事件带来的进度反馈

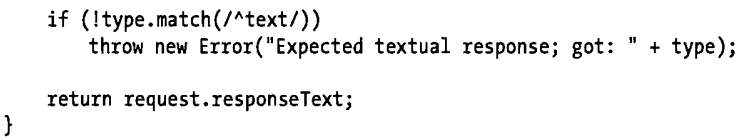
4：浏览器触发readystatechange事件，onreadystatechange可以监听readystatechange事件



12、同步响应

open传入第三个参数false，默认为true（异步），设置为同步响应，send()方法将会一直阻塞到请求完成，send返回后，仅检查status和responseText属性

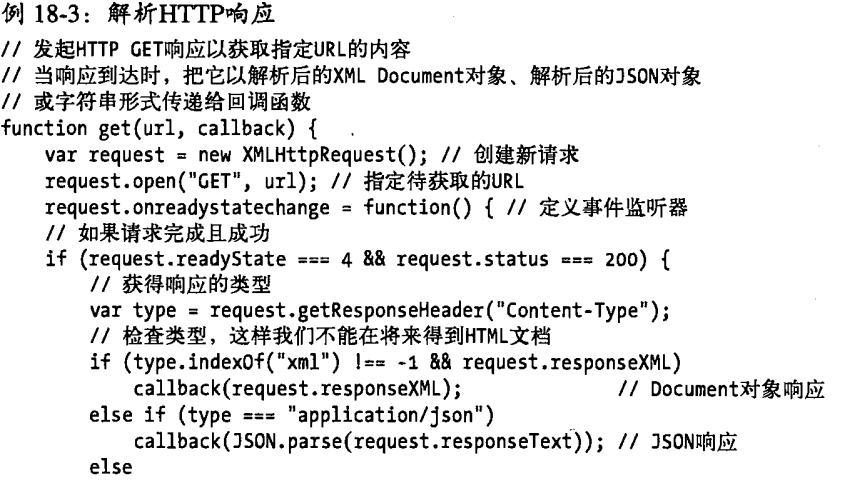


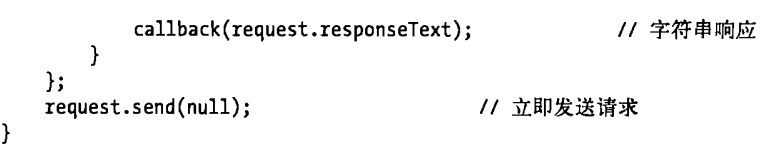


同步请求，浏览器被冻结，客户端JavaScript是单线程的

13、响应解码

如果不是传递文本，可以使用JSON.parse()转义json为对象，也可以使用eval()处理脚本





正常解码是加上响应发送了content-type头和正确的mime类型

a）没有设置适当的mime，xmlhttprequest对象不对解析并且设置responseXML属性

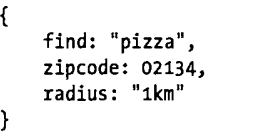
b）如果服务器的content-type头包含了错误的charset参数，会以错误的解码来响应，responseText中的字符可能是错误的，xhr2中overrideMimeType()可以解决，在send发送前设置，忽略默认使用指定



14、编码请求主体

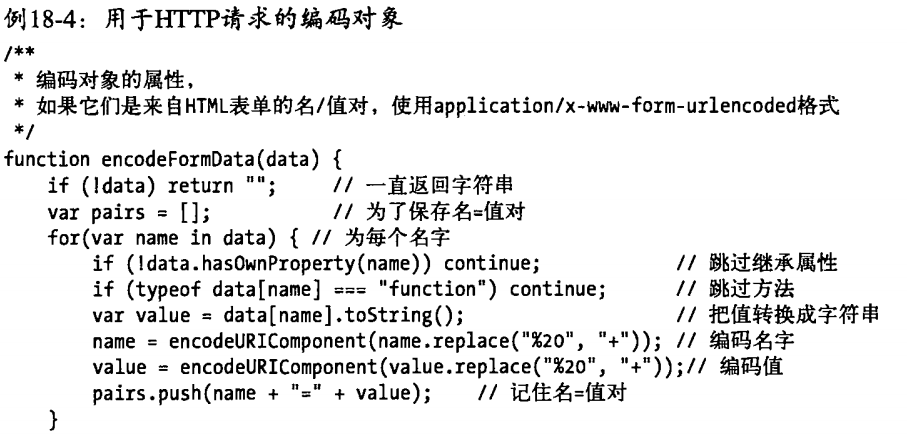
a）表单编码请求：

a=b&c=d，使用&连接，实际发送可以使用对象

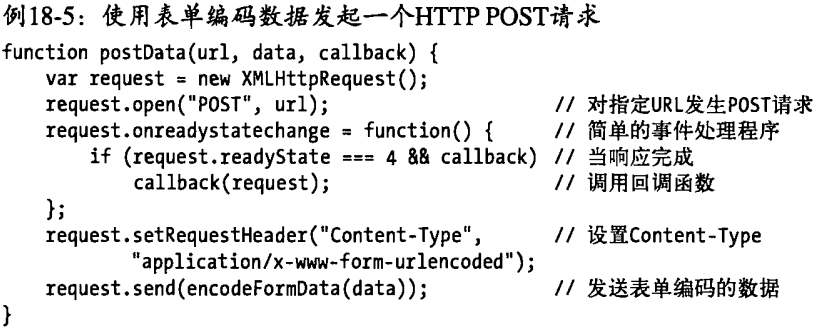


mime设置：application/x-www-form-urlencoded

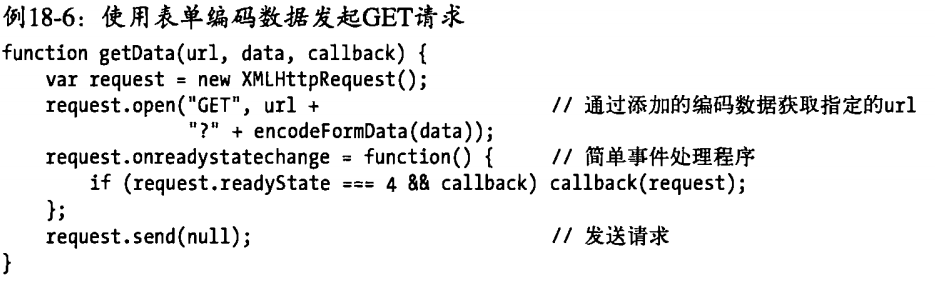
在发送是必须设置Content-Type



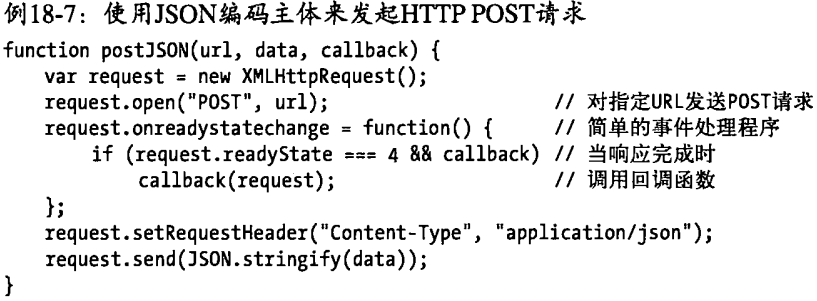




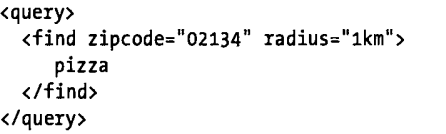
只读查询比get比post效果好

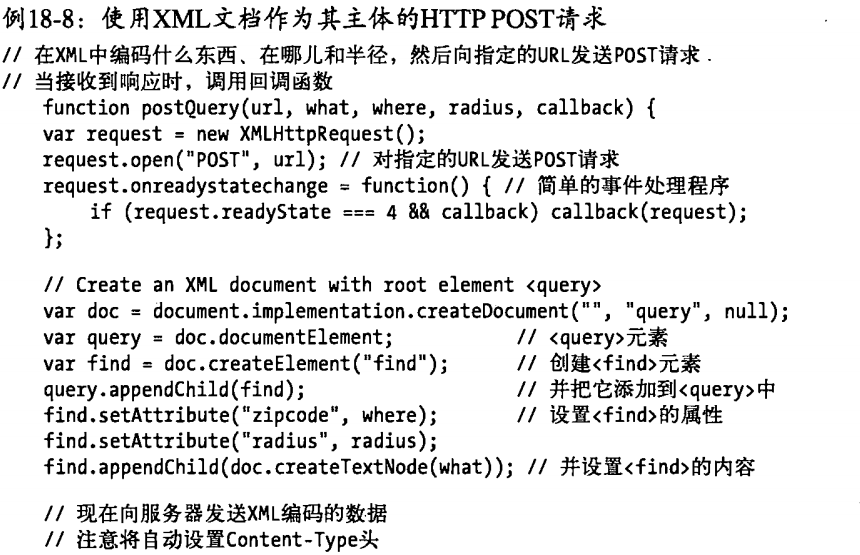


b）json编码请求：使用JSON.stringify()编码请求主体



c）XML编码请求：pizza查询







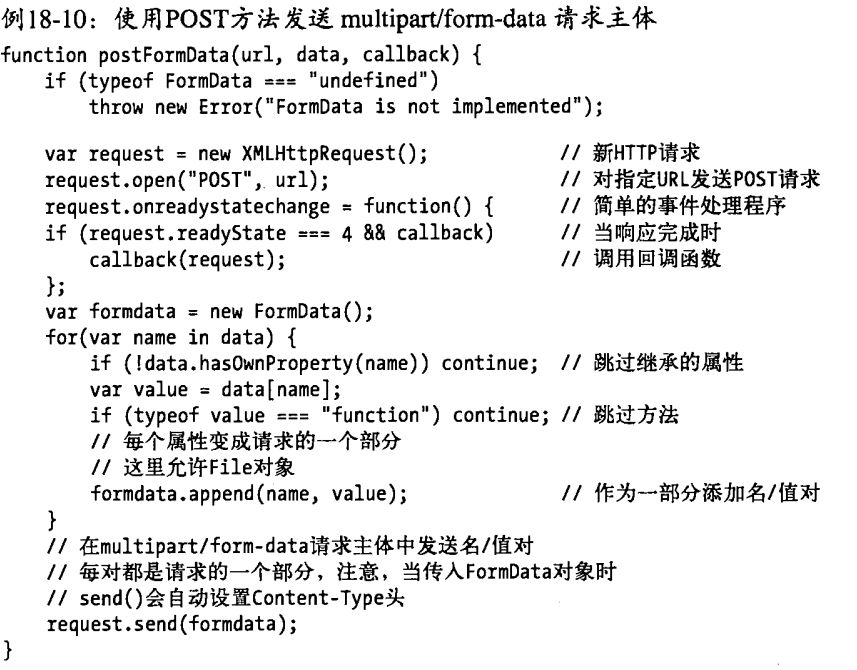
d）上传文件：xhr2允许通过send()方法传入file对象实现上传





e）multipart/form-data请求：同时包含文件上传元素和其他元素，必须使用这个content-type属性用post发放提交表单，xhr2定义了FormData API

FormData：先创建对象，然后调用append()添加元素，最后formdata属性放入send中



f）HTTP进度事件：调用send时，触发loadstart事件，加载的服务器响应，xmlHttpRequest对象回发送progress事件，每隔50毫秒反馈请求进度，事件完成触发load

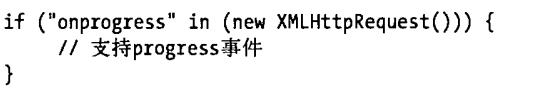
无法完成请求的情况：

请求超时：timeout事件

请求终止：abort事件

网络错误：error事件

判断浏览器是否支持progress事件：



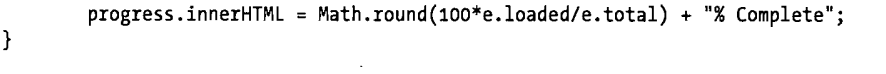
3个有用的属性：

loaded：目前传输的字节数

total：数据的整体长度

lengthComputable：知道内容长度为true反之false



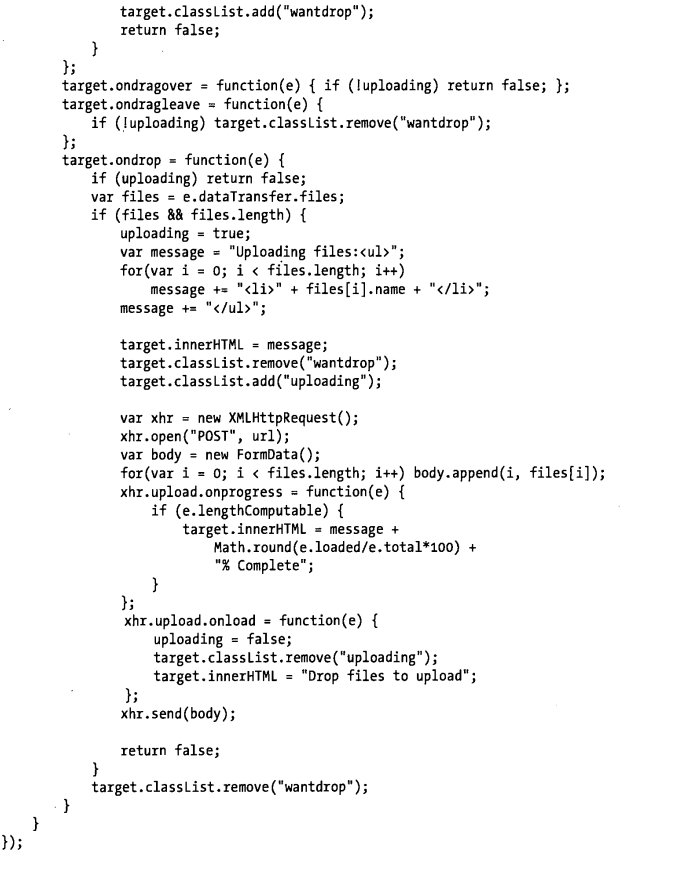


上传进度事件：在一些浏览器中upload属性，定义了addEventListener()方法和整个progress事件集合（onprogress、onload）

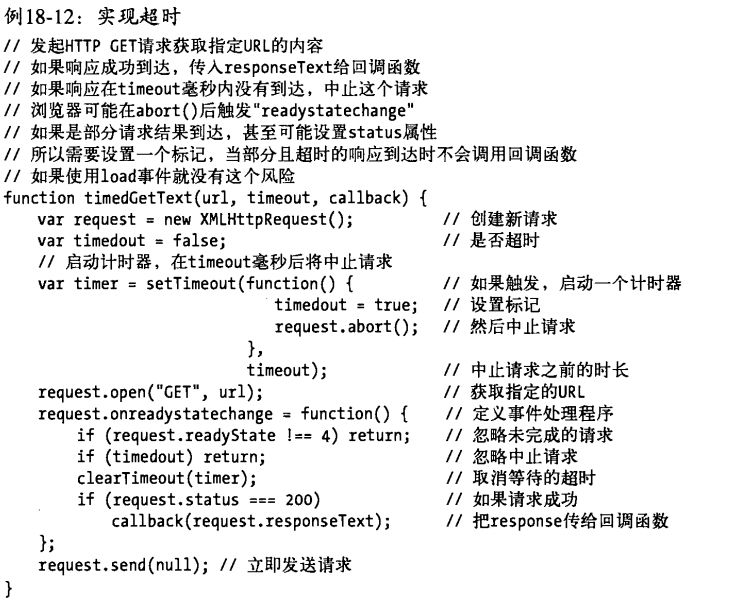
下载进度：x.onprogess

上传进度：x.upload.onprogress





中止请求和超时：abort，timeout，如果浏览器没有timeout可以使用setTimeout()属性设置



跨域请求：同源策略禁止，使用CORS实现，但不能传账号和密码（分布式密码破解攻击），使用withCredentials设置为true，测试浏览器是否支持CORS



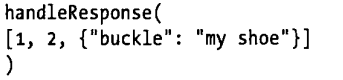


借助<script>发送HTTP请求：jsonp，不受同源影响，src属性指向url，json编码数据会自动解码（达到浏览器执行）

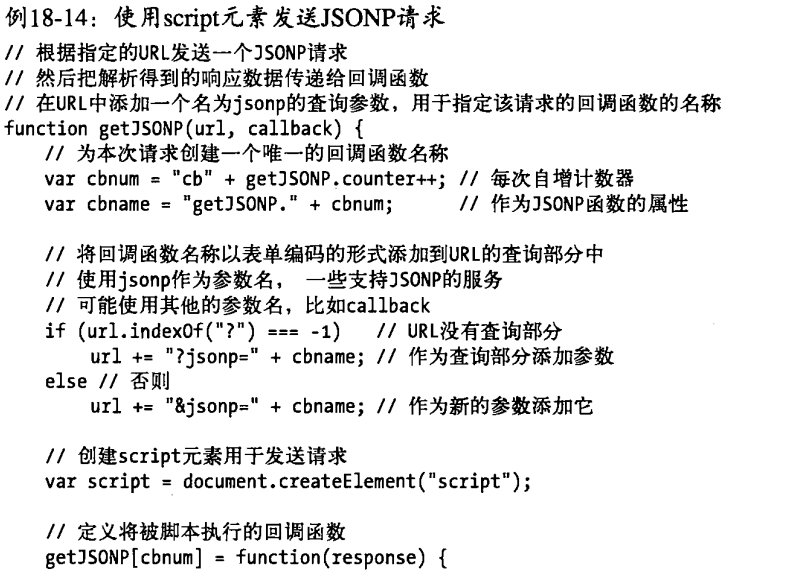
jsonp：<script>元素作为ajax传输的技术

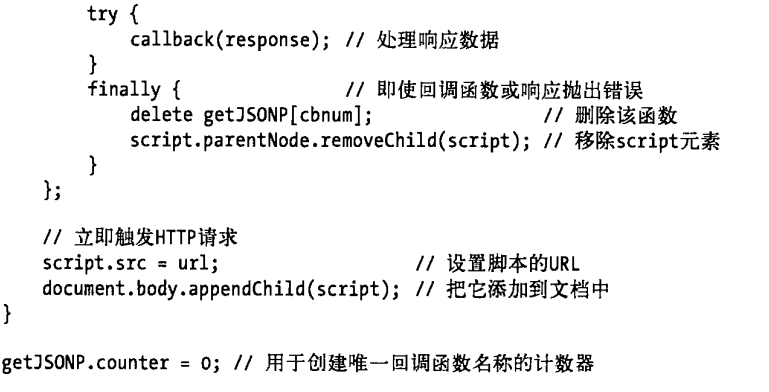
p：填充或前缀

达到浏览器可以执行：使用json的解析出来还是数据，jsonp的数据可以执行

响应内容：json：[1,2,{‘a’:’b’}]、jsonp数据：

得有回调函数





服务器推送Coment技术：EventSource对象，可以传递url给这个对象，返回实例上监听事件，客户端打开，服务端建立长连接，客户端可能会有id

