**珠峰培训第七周随堂测试题目**

**[基础知识]**

1、简述当用户在浏览器地址栏输入一个网址后，到看到网页中的内容，都经历了哪些事情？(10分)

第一步:DNS解析，通过域名到DNS服务器上找到服务器的外网IP地址

第二步:通过服务器的外网IP找到服务器

第三步:通过端口号找到对应的服务和项目文件

第四步:服务器把资源文件中的源代码返回给客户端

第五步:浏览器引擎渲染html代码,在渲染过程中可能还会重复上述的步骤把css/js/img等资源加载回来

2、NODE中的内置模块都有哪些，都是做什么用的？

HTTP模块：创建服务http.createServer，监听端口server.listen，获取客户端传递的内容request.url/request.on(‘data’)/request.on(‘end’)，以及向客户端返回内容response.writeHead/response.write/response.end

URL模块：解析客户端请求的url地址和传递的内容 url.parse

FS模块：操作I/O fs.readFileSync/fs.writeFileSync

在NODE中定义了A模块，A中提供了sum方法，如果想让B模块调取sum，需要做哪些事情？

A模块中module.exports.sum=sum  
 B模块中var a=require(‘./a’); a.sum()

NODE是如何安装第三方模块的(比如安装LESS)？ (10分)

npm install less -g 安装到全局 npm uninstall less -g 从全局下移除

3、AJAX是什么，干什么用的，AJAX同步和异步有什么区别？ (10分)

Ajax：async javascript and xml，在js中我们可以使用ajax向服务器发送请求，把需要的数据获取到，使用ajax请求数据当前的页面不会刷新；

Ajax同步是：当执行send后，ajax请求的操作就算开始了，只有当xhr.readyState===4的时候才结束，在这个过程中不能做其它的事情

Ajax异步是：当ajax开始后，到xhr.readyState===4ajax结束时，中间任意时候，都可以做其它的事情，等到其它的事情做完在返回头看一下ajax是否完成

5、AJAX中常用的HTTP请求方式都有哪些？ GET请求和POST请求的区别？ (10分)

Ajax中常用的请求方式：get、post、put、delete、head...

get和post请求的本质区别在于，传递给服务器数据的方式不同：get请求是通过url问号传参的方式传递给服务器的数据，而post是把传递给服务器的数据放在请求头中传递过去的，由于方式的不同引发出以下三种区别：

->大小问题： get请求传递给服务器的数据存在大小的限制，因为浏览器本身对于url的长度存在限制(谷歌8kb、火狐7kb、ie2kb)；而post请求不存在这个限制，但是真实项目中为了优化页面性能，我们一般设定传递数据内容的大小不超过2mb

->缓存问题：使用get请求的时候浏览器有些时候会记录请求的缓存，这样下一次重新的请求数据，很有可能读取的是上一次缓存的数据，为了解决这个问题，当使用get请求的时候，我们在url的末尾增加一个随机数即可

->安全问题：get请求传递给服务器的数据都在url后面，这样不安全容易被别人进行url劫持，而post传递的内容都在请求主体中，相对于get来说比较的安全；

6、AJAX中经常出现的HTTP STATUES(网络状态码)都是什么意思？ (7分)

200 成功

301 永久重定向

302 临时重定向，一般用于服务器的负载均衡

304 读取的是缓存数据，项目中一般设置不经常更新的css、js为304状态，优化性能

404 请求的地址不存在

500 未知的服务器错误

503 服务器承载超负荷

1. AJAX的状态码都是什么意思？ (7分)

xhr.readyState  
 0 UNSENT 请求还没有发送(AJAX对象刚刚创建完成)  
 1 OPENED 已经把XHR.OPEN执行完成了  
 2 HEADERS\_RECEIVED 响应头信息已经接收,但是主体的内容还没有响应回来  
 3 LOADING 响应主体的内容正在处理加载  
 4 DONE 响应主体内容已经获取成功

8、在客户端AJAX中设置请求头和获取响应头信息使用哪两个方法？ (6分)

设置请求头：xhr.setRequestHeader([key],[value])

获取响应头：xhr.getResponseHeader([key])

**[代码题目]** (每题20分)

1. 编写server模块，当客户端请求index.html的时候，把服务器上index.html文件中的代码返回

var http = require("http"),  
 url = require("url"),  
 fs = require("fs");  
 var server1 = http.createServer(function (request, response) {  
 var urlObj = url.parse(request.url, true),  
 pathname = urlObj.pathname,  
 query = urlObj.query;  
 //->资源文件的请求处理:资源文件的请求都是客户端浏览器自己通过URL地址向服务器发送的,服务器返回的是对应文件中的源代码,客户端的浏览器接收到源代码后用自己的引擎渲染出对应的页面;真实项目中这部分的操作都是由工具(IIS、APACHE...)完成的  
 var reg = /\.(HTML|CSS|JS|ICO)/i;  
 if (reg.test(pathname)) {  
 var suffix = reg.exec(pathname)[1].toUpperCase();  
 var suffixMIME = suffix === 'HTML' ? 'text/html' : (suffix === 'CSS' ? 'text/css' : 'text/javascript');  
 try {  
 var conFile = fs.readFileSync('.' + pathname, 'utf-8');  
 response.writeHead(200, {'content-type': suffixMIME + ';charset=utf-8;'});  
 response.end(conFile);  
 } catch (e) {  
 response.writeHead(404, {'content-type': 'text/plain;charset=utf-8;'});  
 response.end('request file is not found!');  
 }  
 return;  
 }  
});  
server1.listen(80, function () {  
 console.log("server is success,listening on 80 port!");  
});

1. 编写AJAX代码，获取服务器端的时间var xhr = new XMLHttpRequest;  
   xhr.open("head", "/qianduan/temp.txt", true);//->只能是异步不能是同步 最好使用HEAD请求  
   xhr.onreadystatechange = function () {

if(!/^2\d{2}$/.test(xhr.status)) return;  
 if (xhr.readyState === 2) {  
 var serverTime = xhr.getResponseHeader("Date");//->这样直接获取的时间是格林尼治时间(GMT) 北京时间是GMT+0800，并且数据格式是一个字符串  
 console.log(new Date(serverTime));

}  
};  
xhr.send(null);