**1 cookie和session区别?**

**1 session保存到服务器,客户端不知道其中的信息,cookie保存到客户端,服务器能够知道其中的信息.**

**2 session保存的是对象,cookie保存的是字符串.**

**3 session不能区分路径.同一个用户在访问一个网站期间,所有session在任何一个地方都可以访问到,**

**而cookie中设置了路径参数那么同一个网络中不同路径下面cookie互相访问不到的.**

**2 position:absolute和float区别?**

**position:absolute:设置绝对定位从文档流完全删除,删除后不占用原来的位置**

**float 浮动的元素可以向左向右移动,直到它的边缘碰到包含浮动另外一个边框为止.**

**3 null 和 undefind的区别 ?**

**undefined用来表示变量尚未定义或者变量定义但是还没有初始化数据.**

**unll 其实就是一种赋值.用来表示尚未存在的对象.也表示当前变量属于一种没有值得状态.可以理解为站位.**

**4 document.write和innerHTML区别?**

**documen.write 是直接写入页面内容流.如果在写之前没有调用document.open浏览器会自动调用,会导致页面被重写**

**innerHTML是DOM页面元素的一个属性 代表该元素html内容 . 你可以精确到某一个具体元素来进行更改.**

**5 列举IE和其他浏览器的区别?**

**1 ie是微软新版本windows操作一个组成部分.其他浏览器一ie为核心,如果没有IE就运行不了,其他浏览器本质上就是ie.**

**6 get 和 post 区别**

**1 get 是从服务器获取数据 post是向服务器传送数据.**

**2 在客户端get方式通过url提交数据.数据在url中可以看到;post方式,数据放置在HTML header内提交.**

**3 get方式提交数据最多只能有1024字节.而post是没有限制的.**

**4 安全性问题,就像第二条说的,使用get参数会直接暴露在url地址上,而post就不会.**

**所以如果有敏感的字符还是用post方式友好.**

**第二次的面试题**

**7 xml 和 json区别?**

**1 json 是一种轻量级的数据交换格式.他是基于js的一个子集.是js原生格式,json格式简单 都是压缩的所有占用带宽小**

**, json的格式能直接为服务器代码使用 . 节省了大量的代码,也更易于维护.**

**2 xml定义是扩展性标记语言. 它文件庞大文件格式复杂,带宽大.服务器和客户端话费大量代码来解析.**

**导致代码量过于复杂 不容易维护.**

**8 你如何对网站文件和资源进行优化? 你有那些性能优化方法?**

**1 文件合并 目的就是减少http请求次数.**

**2 文件压缩 目的是直接减少文件下载的体积.**

**3 使用CDN (内容分发网络)来托管资源: 基本思路就是为了避开网络上影响数据传输速度的稳定的瓶颈和环节,**

**是内容传输更快,更稳定.通过在网络各处放置节点构成互联网上一层只能虚拟网络.**

**CDN能够实现流量和节点的连接,负载时候可以将数据导向用户最近的服务节点上.**

**4 缓存的使用( 设置多个域名进行缓存)**

**5 利用GZIP压缩你的js和css文件**

**9 js操作获取样式和设置cookie?**

**1**

**10 js延时加载方式有那些?**

**1 使用setTimeout延迟方法加载时间,延迟加载代码,给网页加载流出更多时间**

**2 让js最后加载.假如引入外部js脚本文件时候,如果放入html的head中.页面加载前js就会被加入页面,而放在body中则会按**

**着从上到下加载顺序运行javasctipt代码,所以可以把外部引入js代码放到文件底部.让他最后加载.从而提高加载速度.**

**11 ajax过程?**

**1、创建一个http请求对象**

**new XMLHttpRequest()**

**在IE6需要通过以下方式：**

**new ActiveXObject("microsoft.xmlhttp");**

**2、初始化HTTP请求参数，还没有发送**

**open(method, url, async)**

**method - 请求方式（GET/POST）**

**url - 请求地址**

**async - 是否异步（true/false）**

**3、发送请求**

**send()**

**4、等待并接收响应**

**通过一个事件来监听http的响应**

**xhr.onreadystatechange = function(){**

**}**

**通过属性readyState，代表http请求的状态**

**onreadystatechange事件会在readyState每次发生改变的时候，就会被触发一次**

**readyState状态：**

**1 - 初始化**

**2 - 发送数据**

**3 - 服务器正在处理数据**

**4 - 完成（已经收到响应）**

**属性status 返回结果的状态（是否成功）**

**200 - 成功**

**404 - 找不到地址**

**500 - 服务器错误**

**5、获取响应结果**

**xhr.response**