**请你谈谈cookie**

**Cookie定义：Cookie就是服务器暂存在客服端的资料。好让服务器辨认你的计算机。Cookie是当你浏览某网站时，网站存储在你机器上的一个小文本文件，它记录了你的用户ID，密 码、浏览过的网页、停留的时间等信息，当你再次来到该网站时，网站通过读取Cookie，得知你的相关信息，就可以做出相应的动作，如在页面显示欢迎你的 标语，或者让你不用输入ID、密码就直接登录等。**

Cookie 虽然在持久的保存客服端数据提供了方便，分担了服务器的负担，但是还有很多的局限性；每个特定的域名下最多有20个cookie

1. IE6或者更早的版本最多有20个cookie
2. IE7或者之后可以最多有50个cookie
3. Opera最多为30个cookie
4. Firefox最多有50个cookie
5. Chrome和safari没有硬件的限制
6. IE和Opera会清理最近最少使用的cookie，firefox会随机清理cookie。

Cookie最大的大约为4096个字节，但是为了兼容一般不会超过4095个字节。IE提供了一种持久化用户数据，叫做userData,每个数据最大为128k，每个特定的域名下最多为1M，这种持久化的数据放在缓存中，如果缓存不清理，会一直存在。

**优点：极高的扩张性和可用性**

1：通过良好的编程，控制保存在cookie中的session对象的大小。

2：通过安全加密技术，可以减少cookie 被破解的可能性。

3：只要cookie中存放不敏感数据，即是被盗也是没有关系的。

4：控制cookie的生命周期，使之不会永久有效。

**缺点：**

1. ”cookie”数量和长度的限制。每个特定的域名下最多20个cookie，每个cookie最大为4KB，否则会被裁掉。
2. 安全性问题。如果cookie被人拦截了，那么会得到所有的seesion信息。即是加密也是于事无补。
3. 有的状态不能保存在客服端。例如：为防止表单的重复提交。

**JS对cookie的操作**

JavaScript是运行在客户端的脚本，一般是不能设置session的，因为session是保存在服务器端的。而cookie是保存在客户端的。所以可以使用js了设置cookie。

1. JS设置cookie

document.cookie =”username=”+username+”;possword=”+possword;

**function setCookie(name,value)**

**{**

**var Days = 30;**

**var exp = new Date();**

**exp.setTime(exp.getTime() + Days\*24\*60\*60\*1000);**

**document.cookie = name + "="+ escape (value) + ";expires=" + exp.toGMTString();**

**}**

1. JS读取cookie

var username = document.cookie.split(“;”)[0].split(“=”)[1];

**function getCookie(name)**

**{**

**var arr,reg=new RegExp("(^| )"+name+"=([^;]\*)(;|$)");**

**if(arr=document.cookie.match(reg))**

**return unescape(arr[2]);**

**else**

**return null;**

**}**

3.删除cookie，设置cookie 的过期时间。

**function delCookie(name)**

**{**

**var exp = new Date();**

**exp.setTime(exp.getTime() - 1);**

**var cval=getCookie(name);**

**if(cval!=null)**

**document.cookie= name + "="+cval+";expires="+exp.toGMTString();**

**}**

JS escape(String)函数可以对字符串进行编码。不会对字母和数字进行编码

JS unescape(String)函数可以对escape()函数编码的字符串进行解码。ECMAScript v3 已从标准中删除了 unescape() 函数，并反对使用它，因此应该用 decodeURI() 和 decodeURIComponent() 取而代之。

JS encodeURI(URIString)可以把字符串作为URI进行编码

JS decodeURI(URIString)可以对encodeURI进行解码。

JS encodeURIComponent(URIString)函数可以把字符串进行URI组件进行编码。

JS decodeURIComponent(URIString)函数可以对encodeURIComponent函数编码的URI进行解码