**一.什么是cookies？**

　　大家都知道，浏览器与WEB服务器之间是使用HTTP协议进行通信的，当某个用户发出页面请求时，WEB服务器只是简单的进行响应，然后就关闭 与该用户的连接。因此当一个请求发送到WEB服务器时，无论其是否是第一次来访，服务器都会把它当作第一次来对待，这样的不好之处可想而知。为了弥补这个 缺陷，Netscape开发出了cookie这个有效的工具来保存某个用户的识别信息，因此人们昵称为“小甜饼”。cookies是一种WEB服务器通过 浏览器在访问者的硬盘上存储信息的手段：Netscape Navigator使用一个名为cookies.txt本地文件保存从所有站点接收的Cookie信息；而IE浏览器把Cookie信息保存在类似于 C://windows//cookies的目录下。当用户再次访问某个站点时，服务端将要求浏览器查找并返回先前发送的Cookie信息，来识别这个用 户。

　　cookies给网站和用户带来的好处非常多：

　　1、Cookie能使站点跟踪特定访问者的访问次数、最后访问时间和访问者进入站点的路径  
　　2、Cookie能告诉在线广告商广告被点击的次数，从而可以更精确的投放广告

　　3、Cookie有效期限未到时，Cookie能使用户在不键入密码和用户名的情况下进入曾经浏览过的一些站点

　　4、Cookie能帮助站点统计用户个人资料以实现各种各样的个性化服务

　　在JSP中,我们也可以使用Cookie,来编写一些功能强大的应用程序。

　　下面,我想介绍一下如何用JSP创建和处理Cookie。

**二．如何创建Cookie**   
  
**import="javax.servlet.http.Cookie"**

　　说了这么多，大家一定很想知道JSP是如何创建cookie了。JSP是使用如下的语法格式来创建cookie的：

**Cookie cookie\_name =new Cookie("Parameter","Value");**

　　例如：  
  
Cookie username\_Cookie =new Cookie("username","waynezheng");   
response.addCookie(username\_Cookie);

　　解释：JSP是调用Cookie对象相应的构造函数Cookie(name,value)用合适的名字和值来创建Cookie，然后Cookie可以通过HttpServletResponse的addCookie方法加入到Set-Cookie应答头，  
  
本例中Cookie对象有两个字符串参数：username,waynezheng。注意，名字和值都不能包含空白字符以及下列字符：@ : ;? , " / [ ] ( ) =   
  
**处理Cookie的属性**  
　　看到这里，有的朋友又要问了：我光知道如何创建Cookie有什么用呀？是呀，光知道如何创建Cookie而不知道怎么使用是不够的。  
  
在JSP中，程序是通过cookie.setXXX设置各种属性，用cookie.getXXX读出cookie的属性，现在把Cookie的主要属性，及其方法列于下，供大家参考：

|  |  |  |
| --- | --- | --- |
| 类型 | 方法名 | 方法解释 |
| String | getComment() | 返回cookie中注释,如果没有注释的话将返回空值. |
| String | getDomain() | 返回cookie中Cookie适用的域名. 使用getDomain() 方法可以指示浏览器把Cookie返回给同 一域内的其他服务器,而通常Cookie只返回给与发送它的服务器名字完全相同的服务器。注意域名必须以点开始（例如.yesky.com） |
| int | getMaxAge() | 返回Cookie过期之前的最大时间，以秒计算。 |
| String | getName() | 返回Cookie的名字。名字和值是我们始终关心的两个部分，笔者会在后面详细介绍 getName/setName。 |
| String | getPath() | 返回Cookie适用的路径。如果不指定路径，Cookie将返回给当前页面所在目录及其子目录下 的所有页面。 |
| boolean | getSecure() | 如果浏览器通过安全协议发送cookies将返回true值，如果浏览器使用标准协议则返回false值。 |
| String | getValue() | 返回Cookie的值。笔者也将在后面详细介绍getValue/setValue。 |
| int | getVersion() | 返回Cookie所遵从的协议版本。 |
| void | setComment(String purpose) | 设置cookie中注释。 |
| void | setDomain(String pattern) | 设置cookie中Cookie适用的域名 |
| void | setMaxAge(int expiry) | 以秒计算，设置Cookie过期时间。 |
| void | setPath(String uri) | 指定Cookie适用的路径。 |
| void | setSecure(boolean flag) | 指出浏览器使用的安全协议，例如HTTPS或SSL。 |
| void | setValue(String newValue) | cookie创建后设置一个新的值。 |
| void | setVersion(int v) | 设置Cookie所遵从的协议版本。 |

**读取客户端的Cookie**   
  
　　在Cookie发送到客户端前，先要创建一个Cookie，然后用addCookie方法发送一个HTTP Header。JSP将调用request.getCookies()从客户端读入Cookie，getCookies()方法返回一个HTTP请求头中的内容对应的**Cookie对象数组**。你只需要用循环访问该数组的各个元素,调用getName方法检查各个Cookie的名字，直至找到目标Cookie，然后对该Cookie调用getValue方法取得与指定名字关联的值。  
　　例如  
　　<％  
　　　//从提交的HTML表单中获取，用户名  
　　　String userName=request.getParameter("username");  
  
　　　//以"username",　userName　值/对　创建一个Cookie  
　　　Cookie theUsername=new Cookie("username",userName);  
  
　　　response.addCookie(theUsername);  
　　％>  
　　..............  
　　<％  
　　　Cookie myCookie[]=request.getCookies();//创建一个Cookie对象数组  
  
　　　for(int n=0;n=cookie.length-1;i++);//设立一个循环，来访问Cookie对象数组的每一个元素  
  
　　　Cookie newCookie= myCookie[n];  
  
　　　if(newCookie.getName().equals("username")); //判断元素的值是否为username中的值  
　　　　{％>  
　　　　　你好,<％=newCookie.getValue()％>!//如果找到后，向他问好  
　　　　<％}  
　　％>  
  
**设置Cookie的存在时间，及删除Cookie**  
  
　　　在JSP中，使用setMaxAge(int expiry)方法来设置Cookie的存在时间，参数expiry应是一个整数。正值表示cookie将在这么多秒以后失效。注意这个值是cookie将要存在的最大时间，而不是cookie现在的存在时间。负值表示当浏览器关闭时，Cookie将会被删除。零值则是要删除该Cookie。如：　　  
　　<％  
　　　Cookie deleteNewCookie=new Cookie("newcookie",null);  
　　　deleteNewCookie.setMaxAge(0);　//删除该Cookie  
　　　deleteNewCookie.setPath("/");   
　　　response.addCookie(deleteNewCookie);  
　　％>    
  
**一、 前言**   
  
说起来，Cookie应该是一种应用较久的技术了。早在HTML刚刚出现的时候，在每个独立的页面之间没有办法记录和标识不同的用户。后来人们就发明了 Cookie技术，当用户访问网页时，它能够在访问者的机器上创立一个文件，我们把它叫作Cookie，写一段内容进去，来标识不同的用户。如果下次用户 再访问这个网页的时候，它又能够读出这个文件里面的内容，这样网页就知道上次这个用户已经访问过该网页了。

虽然现在网页的制作技术比起几年以前已经发展了许多。不过有些时候，Cookie还是能够帮我们很多忙的。接下来，我们就来看看，如何在写JSP文件的时候，用JSP操作Cookie。  
＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

**二、 保存写入Cookie**

其实用JSP操作Cookie是非常简单的，我们来看下面一段JSP程序：

........(中间略)

//保存**写入Cookie**   
＜%   
String cookieName="Sender";   
Cookie cookie=new Cookie(cookieName, "Test\_Content");   
cookie.setMaxAge(10);   //存活期为10秒  
response.addCookie(cookie);   
%＞   
........(其他内容)

这样我们就设置了一个Cookie，很简单吧？

我们来仔细研究一下这段代码：

　　Cookie cookie=new Cookie(cookieName, "Test\_Content");

这一行建立了一个Cookie对象，初始化有两个参数，第一个参数cookieName定义了Cookie的名字，后一个参数，也是一个字符串，定义了Cookie的内容。也就是我们希望网页在用户的机器上标识的文件内容。

接下来一行：cookie.setMaxAge(10)，调用了Cookie中的setMaxAge方法，设定Cookie在用户机器硬盘上的存活 期为10秒。一个Cookie在用户的硬盘里面存在的时间并不是无限期的，在建立Cookie对象的时候，我们必须制定Cookie的存活期，超过了这个 存活期后，Cookie文件就不再起作用，会被用户的浏览器自行删除。如果我们希望用户在下次访问这个页面的时候，Cookie文件仍然有效而且可以被网 页读出来的话，我们可以将Cookie的存活期设得稍微长一些。比如cookie.setMaxAge(365\*24\*60\*60)可以让Cookie文件在一年内有效。

**三、 读取出Cookie**

Cookie文件创建好后，自然还需要我们把它读出来，否则我们不是白费力气吗？接下来我们看看如何读出在用户硬盘上的Cookie。

........(中间略)

Name value

＜%   
Cookie cookies[]=request.getCookies(); //读出用户硬盘上的Cookie，并将所有的Cookie放到一个cookie对象数组里面  
Cookie sCookie=null;   
String svalue=null;   
String sname=null;   
  
for(int i=0;i<cookies.length-1;i++{    //用一个循环语句遍历刚才建立的Cookie对象数组  
sCookie=cookies[i];   //取出数组中的一个Cookie对象  
  
sname=sCookie.getName(); //取得这个Cookie的名字  
svalue=sCookie.getValue(); //取得这个Cookie的内容  
%＞

＜%   
}   
%＞

　　name 　　　value  
＜%=name%＞ ＜%=svalue%＞

........(其他内容)

这一小段JSP文件可以读出用户硬盘上的所有有效的Cookie，也就是仍然在存活期内的Cookie文件。并用表格的形式列出每个Cookie的名字和内容。  
  
我们来逐行分析一下这段代码：   
Cookie cookies[]=request.getCookies() 我们用request.getCookies()读出用户硬盘上的Cookie，并将所有的Cookie放到一个cookie对象数组里面。   
接下来我们用一个循环语句遍历刚才建立的Cookie对象数组，我们用sCookie=cookies[i]取出数组中的一个Cookie对象，然后我们 用sCookie.getValue()和sCookie.getName()两个方法来取得这个Cookie的名字和内容。   
  
通过将取出来的Cookie的名字和内容放在字符串变量中，我们就能对其进行各种操作了。在上面的例子里，可通过循环语句的遍历，将所有Cookie放在一张表格中进行显示。   
＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝  
  
**四、 需要注意的一些问题**

通过上面两个简单的例子，可以看到，用JSP进行Cookie的操作，是非常简单的。不过我们在实际操作中还要注意一些问题：

**1. Cookie的兼容性问题**

Cookie的格式有2个不同的版本，第一个版本，我们称为Cookie Version 0，是最初由Netscape公司制定的，也被几乎所有的浏览器支持。而较新的版本，Cookie Version 1，则是根据RFC 2109文档制定的。为了确保兼容性，JAVA规定，前面所提到的涉及Cookie的操作都是针对旧版本的Cookie进行的。而新版本的Cookie目 前还不被Javax.servlet.http.Cookie包所支持。

**2. Cookie的内容**

同样的Cookie的内容的字符限制针对不同的Cookie版本也有不同。在Cookie Version 0中，某些特殊的字符，例如：空格，方括号，圆括号，等于号（=），逗号，双引号，斜杠，问号，@符号，冒号，分号都不能作为Cookie的内容。这也就 是为什么我们在例子中设定Cookie的内容为"Test\_Content"的原因。

虽然在Cookie Version 1规定中放宽了限制，可以使用这些字符，但是考虑到新版本的Cookie规范目前仍然没有为所有的浏览器所支持，因而为保险起见，我们应该在Cookie的内容中尽量避免使用这些字符