

Cookie 是存储在客户端计算机上的文本文件，并保留了各种跟踪信息。

Servlet Cookie 处理需要对中文进行编码与解码，方法如下：

```
String str = java.net.URLEncoder.encode("中文", "UTF-8"); //编码
String str = java.net.URLDecoder.decode("编码后的字符串", "UTF-8"); // 解码
```

Cookie 剖析

Cookie 通常设置在 HTTP 头信息中，设置 Cookie 的 Servlet 会发送如下的头信息到客户端浏览器：

```
HTTP/1.1 200 OK
Date: Fri, 04 Feb 2000 21:03:38 GMT
Server: Apache/1.3.9 (UNIX) PHP/4.0b3
Set-Cookie: name=xyz; expires=Friday, 04-Feb-07 22:03:38
GMT;
                path=/; domain=runoob.com
Connection: close
Content-Type: text/html
```

Set-Cookie 头包含了一个名称值对、一个 GMT 日期、一个路径和一个域。名称和值会被 URL 编码。expires 字段是一个指令，告诉浏览器在给定的时间和日期之后“忘记”该 Cookie。

如果用户的浏览器指向任何匹配该 Cookie 的路径和域的页面，它会重新发送 Cookie 到服务器。浏览器的头信息可能如下所示：

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.6 (X11; I; Linux 2.2.6-15apmac ppc)
Host: zink.demon.co.uk:1126
Accept: image/gif, */*
Accept-Encoding: gzip
Accept-Language: en
```

```
Accept-Charset: iso-8859-1,*,utf-8
```

```
Cookie: name=xyz
```

Cookie 方法

1	public void setDomain(String pattern) 该方法设置 cookie 适用的域，例如 runoob.com。
2	public String getDomain() 该方法获取 cookie 适用的域，例如 runoob.com。
3	public void setMaxAge(int expiry) 该方法设置 cookie 过期的时间（以秒为单位）。如果不这样设置，cookie 只会在当前 session 会话中有效。
4	public int getMaxAge() 该方法返回 cookie 的最大生存周期（以秒为单位），默认情况下，-1 表示 cookie 将持续下去，直到浏览器关闭。
5	public String getName() 该方法返回 cookie 的名称。名称在创建后不能改变。
6	public void setValue(String newValue) 该方法设置与 cookie 关联的值。
7	public String getValue() 该方法获取与 cookie 关联的值。
8	public void setPath(String uri) 该方法设置 cookie 适用的路径。如果您不指定路径，与当前页面相同目录下的（包括子目录下的）cookie。
9	public String getPath() 该方法获取 cookie 适用的路径。
10	public void setSecure(boolean flag) 该方法设置布尔值，表示 cookie 是否应该只在加密的（即 SSL）连接上发送。
11	public void setComment(String purpose) 设置cookie的注释。该注释在浏览器向用户呈现 cookie 时非常有用。
12	public String getComment() 获取 cookie 的注释，如果 cookie 没有注释则返回 null。

设置 Cookie

(1) 创建一个 Cookie 对象：cookie 名称和 cookie 值都是字符串。

```
Cookie cookie = new Cookie("key", "value");
```

(2) 设置最大生存周期：（以秒为单位）。下面将设置一个最长有效期为 24 小时的 cookie。若设置为负数，为临时Cookie，不会被写进 Cookie文件中，窗口关闭即失效

```
cookie.setMaxAge(60*60*24);
```

(3) 发送 Cookie 到 HTTP 响应头: `response.addCookie` 添加 HTTP 响应头中的 Cookie, 如下所示:

```
response.addCookie(cookie);
```

读取 Cookie

通过调用 `HttpServletRequest` 的 `getCookies()` 方法创建一个 `javax.servlet.http.Cookie` 对象的数组。然后循环遍历数组, 并使用 `getName()` 和 `getValue()` 方法来访问每个 cookie 和关联的值。

删除 Cookie

cookie机制没有提供删除cookie的方法。

- 读取一个现有的 cookie, 并把它存储在 Cookie 对象中。
- 使用 `setMaxAge()` 方法设置 cookie 为0, 来删除现有的 cookie。
- 把这个 cookie 添加到响应头。(必须执行)

或者覆盖要删除的cookie (只修改value, maxage属性)。