### Cookie概念：在浏览某些 网站 时,这些网站会把一些数据存在 客户端 , 用于使用网站 等跟踪用户,实现用户自定义 功能.

    是否设置过期时间:

              如果不设置 过期时间,则表示这个 Cookie生命周期为 浏览器会话期间 , 只要关闭浏览器,cookie就消失了.

               这个生命期为浏览会话期的cookie,就是会话Cookie;

     存储:    一般保存在 内存,不在硬盘;

             如果设置了过期时间, 浏览器会把cookie保存在硬盘上,关闭再打开浏览器, 这些cookie 依然有效直到 超过的设置过期时间;

存储在硬盘上的Cookie可以在不同的浏览器进程间共享，比如两个IE窗口。  
而对于保存 在内存的Cookie，不同的浏览器有不同的处理方式。  
原  
 如果浏览器使用的是 cookie，那么所有的数据都保存在浏览器端，  
 比如你登录以后，服务器设置了 cookie用户名(username),那么，当你再次请求服务器的时候，浏览器会将username一块发送给服务器，这些变量有一定的特殊标记。  
 服务器会解释为 cookie变量。  
 所以只要不关闭浏览器，那么 cookie变量便一直是有效的，所以能够保证长时间不掉线。  
 如果你能够截获某个用户的 cookie变量，然后伪造一个数据包发送过去，那么服务器还是认为你是合法的。所以，使用 cookie被攻击的可能性比较大。  
 如果设置了的有效时间，那么它会将 cookie保存在客户端的硬盘上，下次再访问该网站的时候，浏览器先检查有没有 cookie，如果有的话，就读取该 cookie，然后发送给服务器。  
 如果你在机器上面保存了某个论坛 cookie，有效期是一年，如果有人入侵你的机器，将你的 cookie拷走，然后放在他的浏览器的目录下面，那么他登录该网站的时候就是用你的的身份登录的。  
 所以 cookie是可以伪造的。  
 当然，伪造的时候需要主意，直接copy cookie文件到 cookie目录，浏览器是不认的，  
 他有一个index.dat文件，存储了 cookie文件的建立时间，以及是否有修改，所以你必须先要有该网站的 cookie文件，并且要从保证时间上骗过浏览器，  
 曾经在学校的vbb论坛上面做过试验，copy别人的 cookie登录，冒用了别人的名义发帖子，完全没有问题。

cookie 用法:  
  
 setcookie("user","zy",time()+3600); 设置user为zy,一小时之后失效;  
 $\_COOKIE['user']; 取回user值(名字)  
 setcookie("user","",time()-3600); 删除cookie,第二个参数为空,第三个时间设置为小于系统的当前时间即可.  
 或在浏览器设置  
 在使用Cookie时,Cookie自动生成一个文本文件存储在IE浏览器的Cookie临时文件夹中,应用浏览器删除Cookie文件的具体操作步骤为  
 >选择IE浏览器中的工具/internet选项命令,打开Internet选项对话框,  
 >在常规选项卡中单击删除Cookie按钮,在弹出的对话框中单击确定按钮,即可成功删除全部Cookie文件.

Session的概念

Session 是存放在服务器端的类似于HashTable结构（每一种web开发技术的实现可能不一样，下文直接称之为HashTable）来存放用户数据;  
 作用：实现网页之间数据传递，是一个存储在服务器端的对象集合。  
 原理：当用户请求一个Asp.net页面时，系统将自动创建一个Session;退出应用程序或关闭服务器时，该Session撤销。系统在创建Session时将为其分配一个长长的字符串标识，以实现对Session进行管理与跟踪。

        session机制是一种服务器端的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息。

       保存:

            存储在Server段的内存进程中的，而这个进程相当不稳定，经常会重启，这样重启的话，就会造成Session失效，用户就必须要重新登录，用户体验相当差，比如用户在填写资料，快要结束的时候Session失效，直接跳到登录页面;

是否已经创建过session:  
当程序需要为某个客户端的请求创建一个session时，服务器首先检查这个客户端的请求里是否已包含了一个session标识（称为session id），

如果已包含则说明以前已经为此客户端创建过session，服务器就按照session id把这个session检索出来....使用（检索不到，会新建一个），

如果客户端请求不包含session id，则为此客户端创建一个session并且生成一个与此session相关联的session id，

session id的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个session id将被在本次响应中返回给客户端保存。

(总结: 创建一个session时,服务器看这个客户端 是否包含session标识, 是的话按照session id把session检索出来,否则就得 新建一个.)

     Session的客户端实现形式（即Session ID的保存方法）:

一般浏览器提供了两种方式来保存，还有一种是程序员使用html隐藏域的方式自定义实现：  
  
[1] 使用Cookie来保存，这是最常见的方法，本文“记住我的登录状态”功能的实现正式基于这种方式的。  
  
 服务器通过设置Cookie的方式将Session ID发送到浏览器。  
  
 如果我们不设置这个过期时间，那么这个Cookie将不存放在硬盘上，当浏览器关闭的时候，Cookie就消失了，这个Session ID就丢失了。  
  
 如果我们设置这个时间为若干天之后，那么这个Cookie会保存在客户端硬盘中，即使浏览器关闭，这个值仍然存在，下次访问相应网站时，同 样会发送到服务器上。  
  
  
[2] 使用URL附加信息的方式，也就是像我们经常看到JSP网站会有aaa.jsp?JSESSIONID=\*一样的。这种方式和第一种方式里面不设置Cookie过期时间是一样的。(URL重写，就是把session id直接附加在URL路径的后面。)  
  
  
[3] 第三种方式是在页面表单里面增加隐藏域，这种方式实际上和第二种方式一样，只不过前者通过GET方式发送数据，后者使用POST方式发送数据。但是明显后者比较麻烦。

表单隐藏字段就是服务器会自动修改表单，添加一个隐藏字段，以便在表单提交时能够把session id传递回服务器。比如：   
<form name="testform" action="/xxx">   
<input type="hidden" name="jsessionid" value="ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764">   
<input type="text">   
</form>   
实际上这种技术可以简单的用对action应用URL重写来代替。

 session 用法:  
          用户信息保存到session前,先启动;  
             session\_start();                     启动session  
             $\_SESSION['user']="zy";      设置用户名  
             unset($\_SESSION['user']);    销毁用户名  
             session\_destory();                失去已经存储的session的数据

cookie 和session 的区别：

1. cookie数据存放在客户的浏览器上，session数据放在服务器上.

简单的说，当你登录一个网站的时候，如果web服务器端使用的是session,那么所有的数据都保存在服务器上面，

客户端每次请求服务器的时候会发送 当前会话的session\_id，服务器根据当前session\_id判断相应的用户数据标志，以确定用户是否登录，或具有某种权限。

由于数据是存储在服务器 上面，所以你不能伪造，但是如果你能够获取某个登录用户的session\_id，用特殊的浏览器伪造该用户的请求也是能够成功的。

session\_id是服务 器和客户端链接时候随机分配的，一般来说是不会有重复，但如果有大量的并发请求，也不是没有重复的可能性，我曾经就遇到过一次。

登录某个网站，开始显示的 是自己的信息，等一段时间超时了，一刷新，居然显示了别人的信息。  
  
Session是由应用服务器维持的一个服务器端的存储空间，用户在连接服务器时，会由服务器生成一个唯一的SessionID,用该SessionID 为标识符来存取服务器端的Session存储空间。而SessionID这一数据则是保存到客户端，用Cookie保存的，用户提交页面时，会将这一 SessionID提交到服务器端，来存取Session数据。这一过程，是不用开发人员干预的。所以一旦客户端禁用Cookie，那么Session也会失效。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能考虑到减轻服务器性能方面，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。(Session对象没有对存储的数据量的限制，其中可以保存更为复杂的数据类型)

  注意:

     1. session很容易失效,用户体验很差;

     2.虽然cookie不安全,但是可以加密 ;

     3.cookie也分为永久 和暂时 存在的;

     4.浏览器 有禁止cookie功能 ,但一般用户都不会设置;

     5.一定要设置失效时间,要不然浏览器关闭就消失了;

      例如:

            记住密码功能就是使用永久cookie写在客户端电脑，下次登录时，自动将cookie信息附加发送给服务端。

            application是全局性信息，是所有用户共享的信息，如可以记录有多少用户现在登录过本网站，并把该信息展示个所有用户。

 两者最大的区别在于生存周期，一个是IE启动到IE关闭.(浏览器页面一关 ,session就消失了)

一个是预先设置的生存周期，或永久的保存于本地的文件。(cookie)