

第13章 COOKIE与SESSION技术

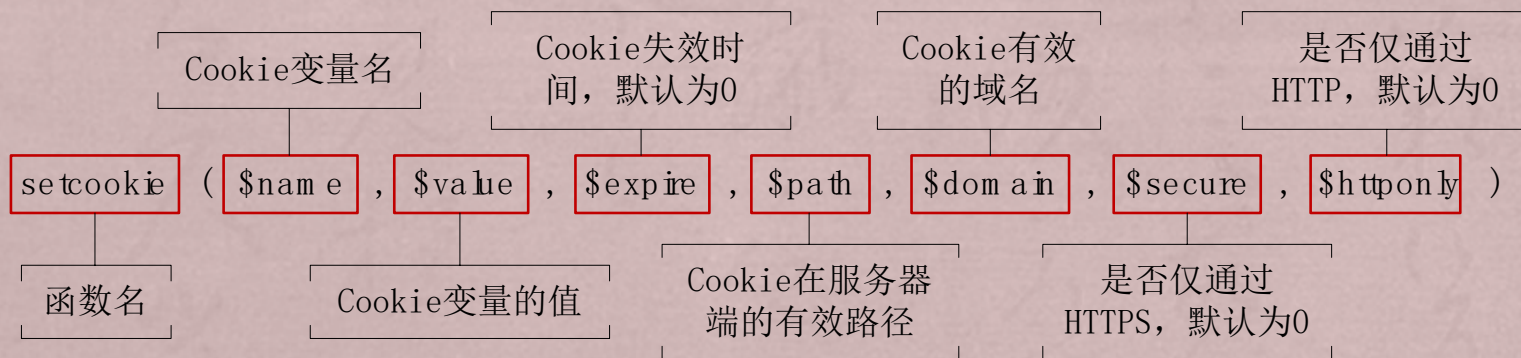
- + Cookie和Session是两种不同的存储机制，Cookie是从一个Web页面到另一个页面的数据传递方法，它被存储在客户端。Session是让数据在页面中持续有效的方法，它存储在服务器端。掌握Cookie和Session技术对于Web页面间的信息传递是必不可少的。

13.1 COOKIE技术

- + Cookie就是将浏览器浏览网页的特定信息以文件的形式保存在客户端硬盘中。当我们再次访问该网站的时候，浏览器会首先尝试读取Cookie文件，读取到的数据供浏览器使用。我们常见的应用就是在登陆一些类似空间、博客类网页，这些网页通常每个人都有个性化的设置，通过Cookie技术就可以很快将它们展示出来。Cookie常用于以下三个方面：
- + 记录访客的某些信息。就像我们前面说的访问空间或者博客。
- + 在页面之间传递数据。通常Cookie不会用来保存密码之类的数据。但是在类似在一个页面设置的数据要在另一个页面使用，这就可以通过Cookie技术来实现。
- + 将查看的Web网页存储在Cookie临时文件中，可以加快网页的访问速度。
- + 下面我们就来开始学习Cookie技术。

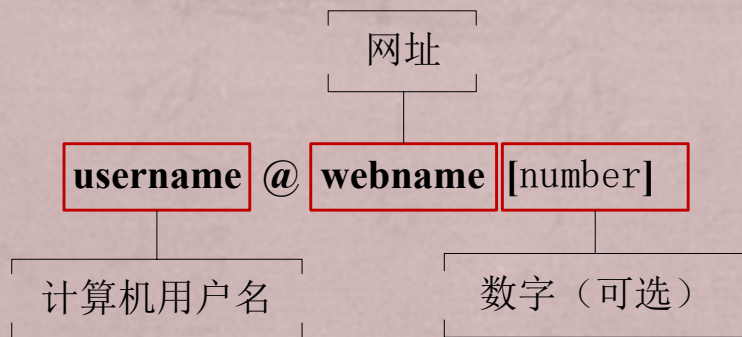
13.1.1 创建COOKIE

- + 在PHP中是通过setcookie()创建一个Cookie。在创建之前需要注意的是Cookie是HTTP标头的组成部分，而标头必须要在其他内容之前发送。因此在创建Cookie之前不可以有任何输出，包括空格和空行。setcookie()的语法如图所示。
- + 在setcookie()的语法中，除了第一个\$name参数值是必须的以外，其他值都是可以选的。我们通常使用的是前三个参数。下面我们就来演示创建一个Cookie。



13.1.1 创建COOKIE

- + (1)演示通过setcookie()创建一个Cookie。
- + 前面我们也说过Cookie是保存在客户端硬盘上的。我们当然可以从硬盘上找到它们。
Cookie文件名通常是如图所示的格式。



13.1.2 读取COOKIE

- + 在PHP中可以使用超全局变量`$_COOKIE`来读取浏览器端的Cookie值。`$_COOKIE`是一个存储着通过`setcookie()`设置的键值对，我们可以通过输出数组详细信息来了解这个全局变量的结构。
- + (1)演示使用`print_r()`输出超全局变量`$_COOKIE`的信息。
- + (2)演示使用`setcookie()`设置一个时间变量，并通过超全局变量`$_COOKIE`来获取时间变量的值。
- + (3)演示利用Cookie记录访客访问次数。

13.1.3 删除COOKIE

- + 在前面我们已经学习了如何设置Cookie并且通过几个简单的示例加深了读者的理解。Cookie在没有设置失效时间的情况下会随着浏览器的关闭而删除。如果我们需要在浏览器关闭之前就删除Cookie，有两种方法可以使用。

1.在浏览器手动删除COOKIE

- + 我们知道Cookie是以文件的形式存储在硬盘中的，浏览器为我们提供了删除这些Cookie的方法。

2. 使用 SETCOOKIE() 删除 COOKIE

- + 在浏览器端删除是由用户控制而不是程序控制的。要通过程序控制 Cookie 的有效时间我们还是要通过 `setcookie()` 来实现的。我们执行将要删除的 Cookie 有效时间设为当前时间之前的时间即可。
- + (1) 演示了通过 `setcookie()` 删除 Cookie。

13.1.4 COOKIE的生命周期

- + 如果Cookie不设置过期时间则表示它的生命周期会随着浏览器关闭而结束。这种Cookie被称为会话Cookie，一般不会保存在硬盘中而是保存在内存中。
- + 如果设置了过期时间则浏览器会把Cookie保存在硬盘中，再次打开浏览器会依然生效，直到它的有效时间超时。但是有的情况下是不遵循这个说法的，例如现在的浏览器通常会限制每个域名的Cookie数量和Cookie的大小。因此在超过这个限制后，Cookie很可能会被删除即使没有超过时间限制。

13.2 SESSION技术

- + Session技术也是Web页面间传递数据必不可少的技术。对比Cookie，Session文件保存的数据是PHP程序中以变量的形式创建的，创建的Session变量在生命周期（20分钟）中可以被跨页面的请求所使用。另外Session文件是存储在服务器端的，相对Cookie存储在客户端安全，而且也没有类似Cookie的存储长度限制。

13.2.1 SESSION 简介

- + 在计算机术语中，Session是指一个终端用户与交互系统进行通信的时间间隔，通常指从注册进入系统到注销退出系统所经过的时间。因此可以说Session是一个特定的时间概念。

1.SESSION的工作原理

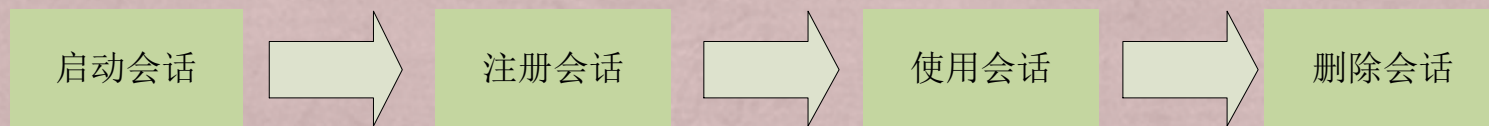
- + 当启动一个Session会话时，程序会随机生成一个唯一的session_id，它也是Session的文件名。此时session_id存储早服务器内存中。当关闭页面关闭后，id会自动注销，重新登录次页面会再次生成一个随机且唯一的id。

2.SESSION的功能

- + Session在Web技术中非常重要。由于网页是一种无状态的连接程序，因此无法得知用户的浏览状态。我们可以想象在网上购物时候，把很多商品加入了购物车，而在结账时候网页却不知道你购物车有哪些物品。通过Session就可以记录这些信息供用户再次使用。
- + Session适用于存储信息量比较少的情況。如果用户需要存储的信息量相对较少，并且对于存储内容不需要长期存储，那么使用Session把信息存储到服务器端比较合适。

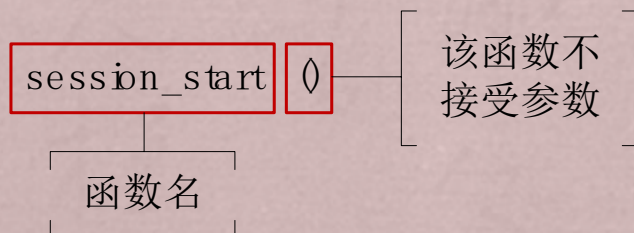
13.2.2 SESSION控制

+ 创建一个完整的Session控制需要通过如图所示的四个步骤。



1. 启动SESSION

- + 在PHP中我们使用session_start()来开启一个新的会话，它的语法如图所示。
- + 需要注意的是，在使用session_start()启动一个新的会话之前不可以有其他的输出。否则会出现错误。这类似于创建Cookie。



2. 注册SESSION

- + 在启动Session之后，所有的Session变量都会被保存在数组\$_SESSION中。要想创建一个变量只需给\$_SESSION添加一个元素即可。
- + (1)演示创建一个Session变量。
- + 该程序在执行之后，创建的两个变量就会服务器端的某个文件中，这个文件是由php.ini配置文件指定的，我们可以通过查看配置文件来中找到这个文件。
- + 这个文件可以使用文本编辑器直接打开，该文件的内容结构如图所示，这从图中所示的文件内容也可以体现出来。

变量名|类型：长度：变量值；

3. 使用 SESSION

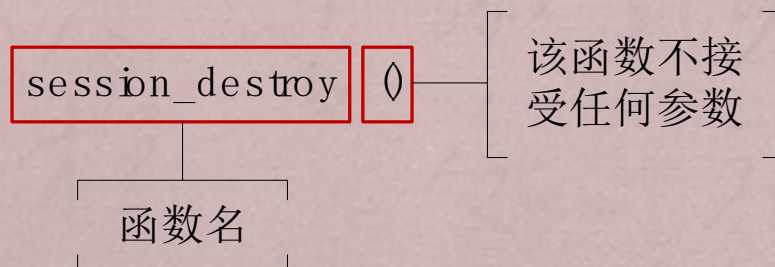
- + 在注册了 Session 变量之后我们就可以使用超全局变量 `$_SESSION` 来使用该变量了。它的使用方法类似于读取 Cookie。
- + (1) 演示将已经注册的 Session 变量值赋值给一个变量。

4. 删除SESSION

- + 删除Session的方法主要分为删除Session、删除多个Session以及结束当前Session三种。如果要删除单个Session，只要将制定的变量注销即可。
- + (1) 演示删除单个Session。
- + (2) 演示注销所有Session。

4. 删除SESSION

- + 如果Session已经完全使用完毕。那么首先应该注销所有Session变量，然后使用 `session_destroy()` 清除结束当前的Session，并释放Session中所有的资源，彻底将Session销毁，它的语法如图所示。
- + (1) 演示使用 `session_destroy()` 完全销毁Session。

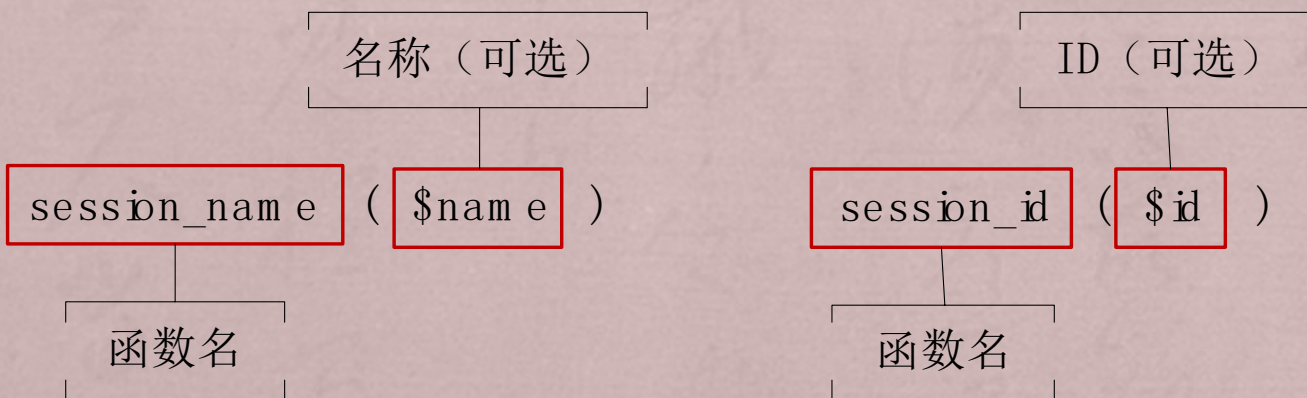


13.2.3 传递SESSION ID

- + 前面我们可以使用Cookie来实现跟踪用户，使用Session也同样可以实现的。实现跟踪用户利用的是Session生成的唯一ID。实现的方法有两种：
- + 第一种是基于Cookie的方式来传递Session。这种方法是有限制的，因为客户端可以禁用Cookie。
- + 第二种是通过URL参数传递，直接将Session ID嵌入到URL中去。

1. 基于COOKIE传递COOKIE ID

- + 基于URL的传递方式就是将当前Session的名称和ID保存在Cookie中。当用户再次请求服务器时候，就会把Session ID发送回来。当脚本中再次使用Session_start()时候，就会根据Cookie中的Session ID返回已经存在的Session。
- + 在将Session保存在Cookie中之前我们首先需要学习两个函数session_name()和session_id()。Session_name()用于获得Session的名称，Session_id()用于获得Session的ID，它们的语法如图所示。
- + 在如图所示的语法中，如果可选的名称或者ID被传入参数，则它们会替换当前Session的名称和ID。



1. 基于COOKIE传递COOKIE ID

+ (1) 演示将Session保存在Cookie中。

2. 通过URL传递SESSION

- + 前面我们学习的是基于Cookie的传递方式，而Cookie在客户端是可以禁用的，那么这种方式就不再行得通。这就需要我们使用URL来传递Session。
- + (1) 演示通过URL传递Session。
- + 从上面的代码中我们可以看到，我们在没有使用Cookie的情况下成功将Session的变量值传递。
- + 在PHP中，为我们提供了一个预定义常量“SID”，“SID”是包含着Session名以及Session ID的常量，格式为“name=ID”，或者如果会话ID已经在适当的会话cookie中设定时则为空字符串。
- + (2) 演示使用常量SID替代“session_name()=session_id()”。

13.3 小结

- + 本章我们主要学习的是Web页面非常重要的Cookie和Session技术。在本章的学习中，核心思想是让读者明白Session与Cookie的不同之处以及工作原理，这是非常重要的。语法方面就比较次要了，本章我们新接触的只有简单的几个，使用是没有任何困难的。