验证码：



一、

问题说明：点击更换验证码 部分浏览器 加载不成功；

加载方式：

Jquery --- attr 操作 即 (‘.img’).attr(‘src’,’图片地址请求’)；经测试后发现部分浏览器（360、 ie8一下、火狐30.0） 都未成功。本人感觉是浏览器缓存记忆导致的，即 每次点击图片 attr 更新的图片地址请求永远是唯一的，浏览器不认为有何改变，而不进行加载（个人想法）。

可在地址请求中加上随机任意字符即可解决此问题。要求每次请求的随机字符都不一致即可。

Php session 时间：

二、

问题说明：控制php session 的有效时间；

如何严格限制session在30分钟后过期！

1.设置客户端cookie的lifetime为30分钟；

2.设置session的最大存活周期也为30分钟；

3.为每个session值加入时间戳，然后在程序调用时进行判断；

至于为什么，我们首先来了解下php中session的基本原理：

PHP中的session有效期默认是1440秒（24分钟），也就是说，客户端超过24分钟没有刷新，当前session就会失效。当然如果用户关闭了浏览器，会话也就结束了，Session自然也不存在了！

大家知道，Session储存在服务器端，根据客户端提供的SessionID来得到这个用户的文件，然后读取文件，取得变量的值，SessionID可以使用客户端的Cookie或者Http1.1协议的

Query\_String（就是访问的URL的“?”后面的部分）来传送给服务器，然后服务器读取Session的目录……

要控制Session的生命周期，首先我们需要了解一下php.ini关于Session的相关设置（打开php.ini文件，在“[Session]”部分）：

1、session.use\_cookies：默认的值是“1”，代表SessionID使用Cookie来传递，反之就是使用Query\_String来传递；

2、session.name：这个就是SessionID储存的变量名称，可能是Cookie，也可能是Query\_String来传递，默认值是“PHPSESSID”；

3、session.cookie\_lifetime：这个代表SessionID在客户端Cookie储存的时间，默认是0，代表浏览器一关闭SessionID就作废……就是因为这个所以Session不能永久使用！

4、session.gc\_maxlifetime：这个是Session数据在服务器端储存的时间，如果超过这个时间，那么Session数据就自动删除！

还有很多的设置，不过和本文相关的就是这些了，下面开始讲如何设置Session的存活周期。

前面说过，服务器通过SessionID来读取Session的数据，但是一般浏览器传送的SessionID在浏览器关闭后就没有了，那么我们只需要人为的设置SessionID并且保存下来，不就可以……

如果你拥有服务器的操作权限，那么设置这个非常非常的简单，只是需要进行如下的步骤：

1、把“session.use\_cookies”设置为1，使用Cookie来储存SessionID，不过默认就是1，一般不用修改；

2、把“session.cookie\_lifetime”改为你需要设置的时间（比如一个小时，就可以设置为3600，以秒为单位）;

3、把“session.gc\_maxlifetime”设置为和“session.cookie\_lifetime”一样的时间；

在PHP的文档中明确指出，设定session有效期的参数是session.gc\_maxlifetime。可以在php.ini文件中，或者通过ini\_set()函数来修改这一参数。问题在于，经过多次测试，修改这个

参数基本不起作用，session有效期仍然保持24分钟的默认值。

由于PHP的工作机制，它并没有一个daemon线程，来定时地扫描session信息并判断其是否失效。当一个有效请求发生时，PHP会根据全局变量

session.gc\_probability/session.gc\_divisor（同样可以通过php.ini或者ini\_set()函数来修改）的值，来决定是否启动一个GC（Garbage Collector）。

默认情况下，session.gc\_probability ＝ 1，session.gc\_divisor ＝100，也就是说有1%的可能性会启动GC。GC的工作，就是扫描所有的session信息，用当前时间减去session的最后修

改时间（modified date），同session.gc\_maxlifetime参数进行比较，如果生存时间已经超过gc\_maxlifetime，就把该session删除。

到此为止，工作一切正常。那为什么会发生gc\_maxlifetime无效的情况呢？

在默认情况下，session信息会以文本文件的形式，被保存在系统的临时文件目录中。在Linux下，这一路径通常为\tmp，在 Windows下通常为C:\Windows\Temp。当服务器上有多个PHP应

用时，它们会把自己的session文件都保存在同一个目录中。同样地，这些PHP应用也会按一定机率启动GC，扫描所有的session文件。

问题在于，GC在工作时，并不会区分不同站点的session。举例言之，站点A的gc\_maxlifetime设置为2小时，站点B的 gc\_maxlifetime设置为默认的24分钟。当站点B的GC启动时，它会扫

描公用的临时文件目录，把所有超过24分钟的session文件全部删除掉，而不管它们来自于站点A或B。这样，站点A的gc\_maxlifetime设置就形同虚设了。

找到问题所在，解决起来就很简单了。修改session.save\_path参数，或者使用session\_save\_path()函数，把保存session的目录指向一个专用的目录，gc\_maxlifetime参数工作正常了。

还有一个问题就是，gc\_maxlifetime只能保证session生存的最短时间，并不能够保存在超过这一时间之后session信息立即会得到删除。因为GC是按机率启动的，可能在某一个长时间内

都没有被启动，那么大量的session在超过gc\_maxlifetime以后仍然会有效。

解决这个问题的一个方法是，把session.gc\_probability/session.gc\_divisor的机率提高，如果提到100%，就会彻底解决这个问题，但显然会对性能造成严重的影响。另一个方法是自己

在代码中判断当前session的生存时间，如果超出了 gc\_maxlifetime，就清空当前session。