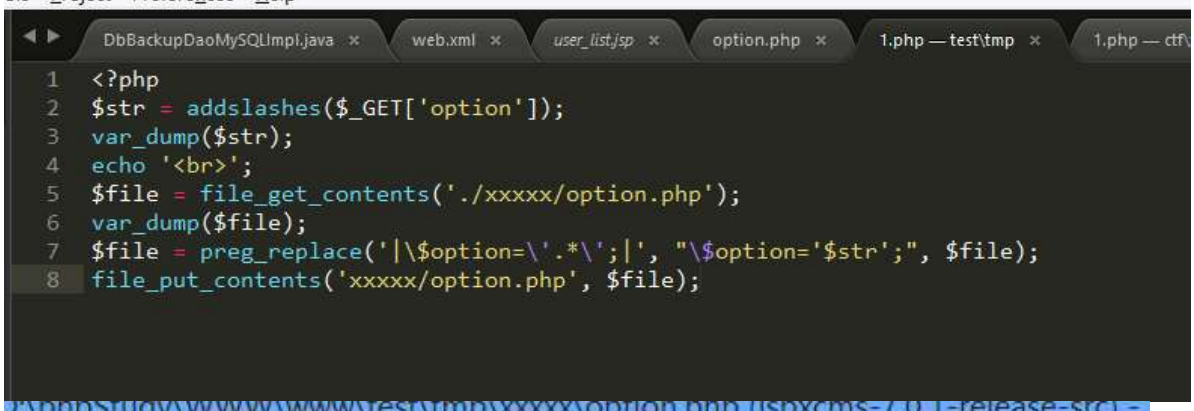
preg-replace 一些奇葩的性质

笔记本： 学习记录

创建时间： 2017/2/27 0:19 **更新时间：** 2017/3/1 17:08

作者： 546325574@qq.com

p牛小窝圈的一题



++

我测试时候是这样的

123

但是其实原理有点不清楚。

++++++++++++++++++++++++++++++++
查看preg_replace函数。

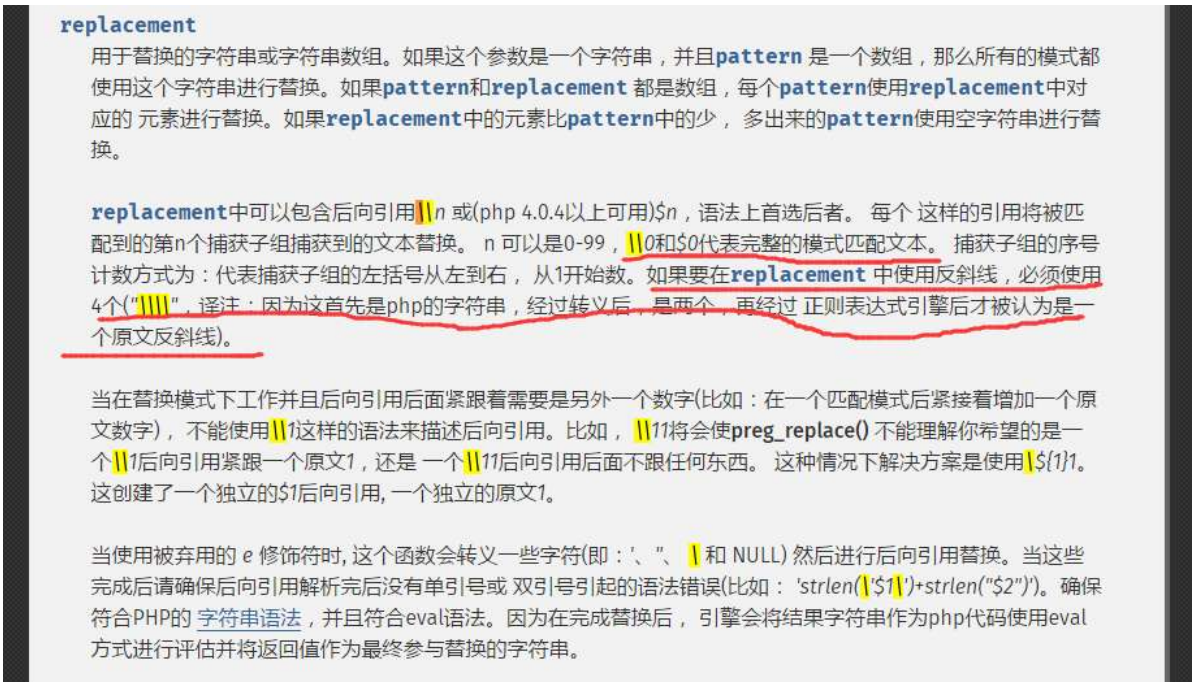
http://php.net/manual/zh/function.preg-replace.php

(PHP 4, PHP 5, PHP 7)
preg_replace — 执行一个正则表达式的搜索和替换

说明

```
mixed preg_replace ( mixed $pattern , mixed $replacement , mixed $subject [, int $limit = -1 [,
int &$count ]] )
```

搜索**subject**中匹配**pattern**的部分， 以**replacement**进行替换。



replacement

用于替换的字符串或字符串数组。如果这个参数是一个字符串，并且**pattern** 是一个数组，那么所有的模式都使用这个字符串进行替换。如果**pattern**和**replacement** 都是数组，每个**pattern**使用**replacement**中对应的 元素进行替换。如果**replacement**中的元素比**pattern**中的少， 多出来的**pattern**使用空字符串进行替换。

replacement中可以包含后向引用*n* 或(php 4.0.4以上可用)*$n*，语法上首选后者。 每个 这样的引用将被匹配到的第n个捕获子组捕获到的文本替换。 n 可以是0-99，*0*和*$0*代表完整的模式匹配文本。 捕获子组的序号计数方式为：代表捕获子组的左括号从左到右， 从1开始数。如果要在**replacement** 中使用反斜线，必须使用4个("\\\\"，译注：因为这首先是php的字符串，经过转义后，是两个，再经过 正则表达式引擎后才被认为是一个原文反斜线)。

当在替换模式下工作并且后向引用后面紧跟着需要是另外一个数字(比如：在一个匹配模式后紧接着增加一个原文数字)， 不能使用*1*这样的语法来描述后向引用。比如， *11*将会使**preg_replace()** 不能理解你希望的是一个*1*后向引用紧跟一个原文*1*，还是 一个*11*后向引用后面不跟任何东西。 这种情况下解决方案是使用*${1}*1。这创建了一个独立的*$1*后向引用, 一个独立的原文*1*。

当使用被弃用的 *e* 修饰符时, 这个函数会转义一些字符(即：'、"、 \ 和 NULL) 然后进行后向引用替换。当这些完成后请确保后向引用解析完后没有单引号或 双引号引起的语法错误(比如： *'strlen(\'$1\')+strlen("$2")'*)。确保符合PHP的 字符串语法，并且符合eval语法。因为在完成替换后， 引擎会将结果字符串作为php代码使用eval方式进行评估并将返回值作为最终参与替换的字符串。

所以p牛小蜜圈里面除了该隐大牛的解法1，其他方法其实参考以上就有答案了。之前解出来也是瞎猫碰到死耗子。搞安全的，还是不能只有表面功夫，还得继续深入研究。结合php c源码分析等。