|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 2018/5/19 | 地点 | 会议室 |
| 主持人 | 黎晓峰 | | |
| 人员 | 黎晓峰, 余林, 陈喆，李彬，罗鑫 | | |
| 主题 | 周会记录 | | |
| 会议内容   1. 如何判断字符串中包含某个字符串   Strpos  查找字符串首次出现的位置  int **strpos** ( string $haystack , [mixed](http://php.net/manual/zh/language.pseudo-types.php#language.types.mixed) $needle [, int $offset = 0 ] )  返回 **needle** 在 **haystack** 中首次出现的数字位置  **Haystack**  在该字符串中进行查找。  **Needle**  如果 **needle** 不是一个字符串，那么它将被转换为整型并被视为字符的顺序值。  **Offset**  如果提供了此参数，搜索会从字符串该字符数的起始位置开始统计。 如果是负数，搜索会从字符串结尾指定字符数开始.    当前结果输出 2  注意：字符串的起始位置是从0开始的，而不是从1开始的  第二种情况： | | | |

当前结果输出 1

Strstr

查找字符串首次出现的位置

string **strstr** ( string $haystack , [mixed](http://php.net/manual/zh/language.pseudo-types.php#language.types.mixed) $needle [, bool $before\_needle = FALSE ] )

返回 **haystack** 字符串从 **needle** 第一次出现的位置开始到 **haystack** 结尾的字符串

该函数区分大小写。如果想要不区分大小写，请使用 [stristr()](http://php.net/manual/zh/function.stristr.php)。

**haystack**

输入字符串。

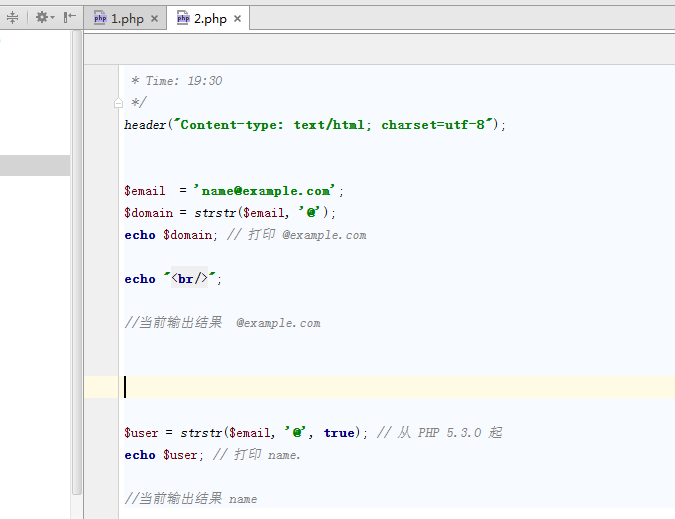
**needle**

如果 **needle** 不是一个字符串，那么它将被转化为整型并且作为字符的序号来使用。

**before\_needle**

若为 **TRUE**，**strstr()** 将返回 **needle** 在 **haystack** 中的位置之前的部分。

返回字符串的一部分或者 **FALSE**（如果未发现 **needle**）



Arrar\_merge 和array()+array() 的区别

array\_merge()如果是merge的以数字为键值的数组，键值会重新排列，+不会

遇到这个坑的背景重现：

当时是做包裹状态的合并，在发的包裹列表的时候，需要展示当前包裹的状态，而在正常情况下，状态的来源是C('dict/status', 'status')的

，但是包裹异常的时候，异常的状态来源是C('dict/status', 'service\_status')，在列表的时候我的状态列表需要这两个来源的总和，所以第

一次的时候，我想到的是array\_merge函数，但是发现key值被重新索引了，相当于我根本就不能在使用的时候用键值来索引到状态名称了，这

和预期不符， 于是我就开始犯难了，未必还不得不把异常的几个状态塞到正常的那里面包起迈？这么搞的话在其他并不需要异常状态的地方不

是每一次都加载了多余的数据了迈？而且最不爽的地方是，索引的码还需要纠结半天去定出来，代码展示的时候又怎么匹配，想想决定放弃。问

度娘，很快就搜到一个分析array\_merge 与 array +的帖子，发现原来array\_merge在处理以数字为索引的数组的时候，会重新索引，而array+则

不会。

以下为代码实例

//定义三个数组

$arr\_1 = array(

'a' => '我是a',

'b' => '我是b',

'c' => '我是c'

);

$arr\_2 = array(

1 => '我是1',

2 => '我是2'

);

$arr\_3 = array(

1 => '我是一',

2 => '我是二',

3 => '我是三'

);

var\_dump(array\_merge($arr\_1,$arr\_2));

结果为：array(5) {

["a"]=>

string(7) "我是a"

["b"]=>

string(7) "我是b"

["c"]=>

string(7) "我是c"

[0]=>

string(7) "我是1"

[1]=>

string(7) "我是2"

}

可见：以数字为键的那部分被重置了索引

执行var\_dump($arr\_1+ $arr\_2);

结果为：

array(5) {

["a"]=>

string(7) "我是a"

["b"]=>

string(7) "我是b"

["c"]=>

string(7) "我是c"

[1]=>

string(7) "我是1"

[2]=>

string(7) "我是2"

}

可见：以数字为索引的部分并没有被重置

执行var\_dump($arr\_2 + $arr\_3);

结果为：

array(3) {

[1]=>

string(7) "我是1"

[2]=>

string(7) "我是2"

[3]=>

string(9) "我是三"

}

可见，array +的方式，如果有相同所以的值，以第一个参数对应键的值为主

项目进度

1. EDK项目修复bug以及测试
2. FR

邮件的内容

提出bug整改

Base\_url的处理

导入奶粉渠道备案商品功能，导入失败