



链滴

# PHP 没有块级作用域

作者: [Tristana](#)

原文链接: <https://ld246.com/article/1604320653115>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<?php
    $arr = range(0, 9);
    foreach ($arr as $k => $v) {
        if ($v/2 == 0) {
            $flag = true;
        } else if ($v > 6) {
            $flag = true;
        }
        if (isset($flag)) {
            // do something
            echo "I Love PHP!\n";
        }
    }
}
```

以上代码除了 `$arr` 中的值为偶数和大于 6 时会输出 " I Love PHP!" ，其它情况也会输出，一开始，是有点困惑的，搞了好些时间才搞明白。

简单分析下解决的过程：这个代码片段是在最近排查系统的 bug 是看到的，流程做了些简化。代码我本人写的 `sweat`，系统在最开始并没有出现问题，上线一段时间后，就出 bug 了。有的人可能一眼就看出来问题在哪了，但是如果你对 PHP 的变量的作用域不太了解的话，就容易栽进这坑里。

**PHP 有函数作用域，但没有像 c/c++、java 等语言的块级作用域。** 例如在 c++ 里：

```
for (int i = 0; i < 10; i++) {
    cout << i << endl;
}
cout << i << endl; // 编译错误 identifier "i" is undefined
```

而在 PHP 里一下几种情况变量则会一直存在：

```
<?php
// demo01: 单层循环
for ($i = 0; $i < 5; $i++) {
    ;
}
echo $i; // 输出 5

// demo02: 多层循环
for ($i = 0; $i < 5; $i++) {
    for ($j = 0; $j < 5; $j++) {
        $k = 5;
    }
}
echo $i, $j, $k; // 输出 555, 循环块的外部仍就可以访问这些变量
```

那么如果我要保证这个变量能够在代码块中保证局部的特性怎么做呢？

第一种方式是在每次循环的开始位置重新初始化这个变量：

```
<?php
    $arr = range(0, 9);
    foreach ($arr as $k => $v) {
        $flag = false; // init $flag
```

```
    // change $flag value ...  
}
```

第二种方式是使用完之后直接 `unset` 掉:

```
<?php  
$arr = range(0, 9);  
foreach ($arr as $k => $v) {  
    // set $flag value ...  
    if ($v/2 == 0) {  
        $flag = true;  
    } else if ($v > 6) {  
        $flag = true;  
    }  
    if (isset($flag)) {  
        // do something  
        echo "I Love PHP!\n";  
        unset($flag); // 直接释放这个变量  
    }  
}
```

起初我写这段代码的时候，想当然的认为 PHP 在每次循环后都会回收并重新初始化 `$flag` 这个变量，然后在单元测试的时候，`$arr` 这个变量并没有出现条件满足和不满足数据混合的情况，最后的输出结果也符合预期。所以这里也叮嘱下大家，谁说语法不重要了，而且在做单元测试时，一定要注意代码的覆盖率，尽可能的覆盖所有会出现的情况，确保代码的正确和完备性，减少这种类型 bug 的出现。

[php坑系列之块级作用域](#)