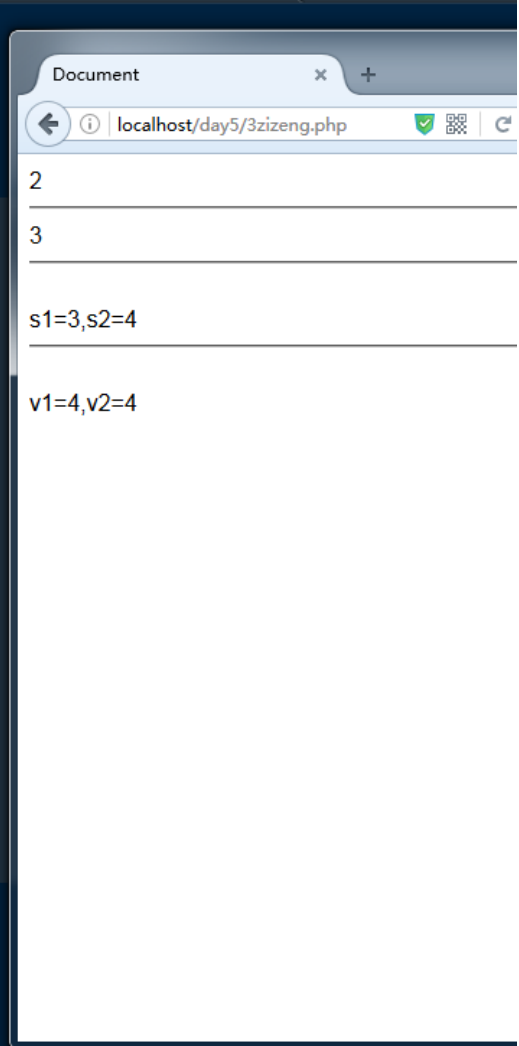


Day7-2自加运算

前自增和后自增的区别（自减类似）

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body>
8   <?php
9     $v1 = 1;
10    $v2 = 1;
11    $v1++; // 此行后, v1为2
12    ++$v2; // 此行后, v2为2
13    // 说明: 此时前自增和后自增效果一样!
14
15    echo $v1++; // 输出2, 此行后, v1为3
16    echo "<hr />";
17    echo ++$v2; // 输出3, 此行后, v2为3
18
19    $s1 = $v1++; // $s1为? 3, 此行后v1为4
20    $s2 = ++$v2; // $s2为? 4, 此行后v2为4
21
22    echo "<hr />";
23    echo "<br/>s1=$s1,s2=$s2";
24    echo "<hr />";
25    echo "<br/>v1=$v1,v2=$v2";
26
27    // 可见, 在有加加运算的其他语句中
28    // 前加加和后加加会有区别:
29    // 影响其他语句的执行结果:
30    // 前加加是先对自加变量加!, 然后做其他运算
31    // 后加加是先做其他运算, 然后对自加变量加1
32  ?>
33 </body>
34 </html>
```



通常, 我们在循环中, 推荐使用前加加, 比如:

```
for($i=1; $i<10000; ++$i){
    .....
}
```

演示前加加和后加加进行1千万次的“效率比较”:

```
// 可见, 在有加加运算的其他语句中
// 前加加和后加加会有区别:
// 影响其他语句的执行结果:
// 前加加是先对自加变量加!, 然后做其他运算
// 后加加是先做其他运算, 然后对自加变量加1

$t1 = microtime(true); // 获得当前时间, 精确到万分之一秒
for ($i=1; $i<10000000; ++$i){
}
$t2 = microtime(true); // 获得当前时间, 精确到万分之一秒
for ($i=1; $i<10000000; $i++){
}
$t3 = microtime(true); // 获得当前时间, 精确到万分之一秒

echo "<p>前加加耗时: ". ($t2-$t1);
echo "<p>后加加耗时: ". ($t3-$t2);

?>
body>
html>
```

3

s1=3,s2=4

v1=4,v2=4

前加加耗时: 0.31931090354919

后加加耗时: 0.34411406517029

