

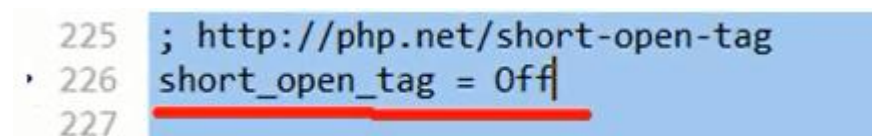
Php 基本语法形式

Php 的标记符

推荐使用：

```
<?php
    这里要写符合 php 语法的语句
?>
```

依赖于 php.ini 中的一个设置项



```
225 ; http://php.net/short-open-tag
226 short_open_tag = Off
227
```

将其改为 on,就可以用了

Php 的区分大小写特征

- 1、变量区分大小写
- 2、常量通常默认也区分，但可以认为设定为不区分
- 3、其他场合的关键字都不区分

一条语句使用一个分号结束

- 1，在一个 php 的语句最后一个分号可省略
- 2，php 结束标记省略则不能省略最后一个分号

Php 的注释

单行注释

形式 1: //注释内容

形式 2: #注释内容

多行注释：

```
/* 注释内容，可以多行 */
```

变量

基本理解

变量可以理解为只是一个代表一定储存空间及其中的数据的一个标识符

使用形式：每个变量名前必须以 **s** 开头

变量的基本操作

4 个操作：

赋值

取值

判断

删除

变量的命名规则

保持程序的正确性

以字母或下划线开头

后跟任意数量的字母，数字和下划线

行业规则 保证程序的可读性

3 种常见命名法

1，骆驼命名

2，帕斯卡命名

3, 下划线分割法

变量的传值方式

单对单

值传递, 引用传递

```
$v1 = 1;
$v2 = $v1; //这就是值传递
```

简单理解：将\$v1 的值取出来（注意：\$v1 中的值还在），然后再用该值给\$v2 赋值。

图示理解：

可见，值传递，就是变量 \$v1 的值进行了复制，然后在给另一个变量 \$v2 赋值。

注意：

- 1 这两个变量此时是值相等的；
- 2 这两个变量又是互相独立的——互不影响；

```
php 中，只有一种语法形式可以实现变量的引用传值方式：&符号。
举例如下：
$m1 = 1;
$m2 = &$m1; //引用传值方式
```

简单理解：将变量\$m1 跟其数据值之间的“引用关系”，复制一份，再给予变量\$m2，即，此时，变量\$m2 也同样具有跟原来数据的一个“引用关系”（指向关系）；

图示理解：

可变变量

```
$s1 = "abc"; //这是一个变量，里面存储的是字符串"abc"
$abc = 10; //
echo $$s1; //???, 输出 10
```

理解：

- 1, 在 php 中，一个"\$" 后面，总是跟着一个变量名！
- 2, 这里，echo 输出的这个变量(以第一个\$为标识)的名字是：\$s1，即"abc"
- 3, 所以，这里输出的是 \$abc，即 10
- 4, 这种连续出现"\$" 的变量形式，就是所谓的“可变变量”；

```
14 $s1 = 'abc'; //这是一个变量，里面存储的是字符串"abc"
15 $abc = 10; //
16 echo $$s1; //???, 输出10
```

10

```
18 //以下演示“可变变量”的灵活性:
19 //所谓可变变量, 其实就是变量的名字是可以“动态变化”以获取不同的数据值
20 $v1 = 1;
21 $v2 = 12;
22 $v3 = 33;
23 $v4 = 44;
24 $v5 = 115;
25 //求这5个变量的和:
26 //其他语言, 只能“一个一个”加起来。
27 $sum = 0; //用于存储总和
28 for($i = 1; $i <= 5; $i++){
29     $v = "v" . $i; //这里, 结果其实只是一个“字符串”, 比如“v1”, “v2”...
30     $sum += ${$v};
31 }
32 echo "<br />sum = $sum";
```

sum = 205