

# PHP 基本语法形式

Php 的标记符

```
<?php
```

这里写 php 语法内容

```
?>
```

## Php 区分大小写特性

- 1.变量区分大小写
- 2.常量默认也区分，但可以设定不区分
- 3.其他场合的关键字都不区分，比如函数名，系统关键字（for.if.return ...）

## 一条语句使用一个分号结束

- 1.在一个 php 标记中的最后一个分号可以省略
- 2.php 结束标记省略则不能省略最后一个分号

## Php 注释

单行注释

```
//注释内容
```

```
#注释内容
```

多行注释

```
/*多行内容，可以注释*/
```

```
14  /*
15  echo "<br />代码1";      全部注释了！
16  echo "<br />代码2";
17  echo "<br />代码3,代表多行需要注释的代码";
18  /**/
```

反注释：

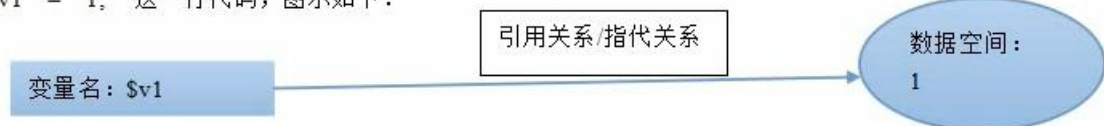
```
14  /**/
15  echo "<br />代码1";      这样就有全部取消注释了
16  echo "<br />代码2";
17  echo "<br />代码3,代表多行需要注释的代码";
18  /**/
```

## 变量

### 基本理解

变量可以理解为只是一个代表一定存储空间及其中的数据的一个“标识符”——也就是一个名字。

`$v1 = 1;` 这一行代码，图示如下：



这表明，使用该变量，就是在使用该数据空间的数据值！比如：

```
echo $v1;      //输出数据 1
```

```
$v2=$v1+3;    //此时，进行的是 1+3 的计算
```

使用形式：每个变量名前边必须以\$开头

定义形式：php 中不支持“单纯定义”一个变量，而是，在第一次变量赋值的时候，就定义变量！

## 变量的基本操作

变量的 4 个操作：

### 赋值

等号 (=) 的左边放变量名，右边放“数据”，就是赋值

```
$v1=1;
```

### 取值

任何需要一个数据的语句中使用一个变量，此时就是指从该变量取得该变量的值

```
echo $v1;      //取出 v1 的值并输出
$v2 = $v1;     //取出 v1 的值并给其他变量赋值；
$v2 = $v1 + 3; //取出 v1 的值并与 3 进行运算！
$v2 = round($v1); //取出 v1 的值并并使用函数 round() 对其进行四舍五入运算
```

### 判断 isset(变量名)

作用:判断该变量是否存在，或该变量是否有数据值，存在或有数据值就是 true，否则就是 false

```
$v1 = isset($s1);» //判断$s1是否存在，结果是false
$s2 = 2;
$v2 = isset($s2);» //true

$s3 = false;
$v3 = isset($s3);» //true

$s4 = "";
$v4 = isset($s4);» //true

$s5 = null;»//特别注意: null这个“值”（数据）的含义就是“没有数据”
$v5 = isset($s5);» //false;
```

### 删除

含义：删除一个变量，并不是指将该变量从变量程序中删掉，而是“断开”该变量名和原有数据间的“引用联系”，此时就会有这样的结果：

- 1.变量名已经不指向（引用）任何数据，此时其 `isset()` 判断的结果为 false
- 2.该数据（可能）没有任何指向它了，就成为“无法使用的数据”--通常就被自动回收了



```
$v1=1;
unset($v1)
$s1=isset($v1);      //false
```

### 变量命名规则

基本规则：保证程序的正确性

以字母或下划线开头

后跟任意数量（含 0 个）的字母，数字和下划线

行业规则：保证程序的可读性

有 3 中命名法：

- 1.骆驼命名法：首单词小写，其后每个单词首字母大写

举例：\$name \$myName \$myFatherName

- 2.帕斯卡命名法：每个单词首字母都大写

举例: \$name     \$MyName     \$MyFatherName

3.下划线分割法: 每个单词小写, 但此件用下划线分开

举例: \$name     \$my\_name     \$my\_father\_name

## 变量的传值方式

1.变量的传值方式指“一个变量传给另一个变量”的内部细节方式---单对单

2.变量的传值方式只有 2 中: 值传递, 引用传递

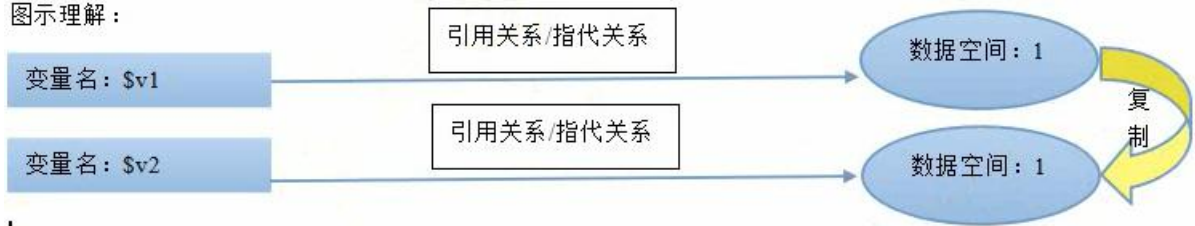
### 值传递

```
$v1=1;
```

```
$v2=$v1;     //这就是值传递
```

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

图示理解:



v1 的值进行了复制, 然后赋值给 v2

\$v1 和 \$v2 值相同, 相互独立, 互不影响

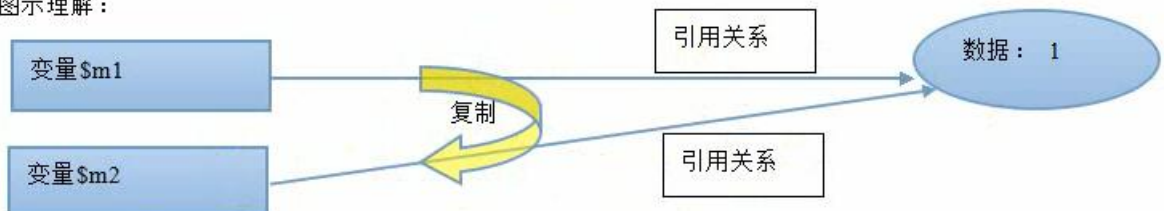
### 引用传递:

Php 中, 只有一种语法实现变量的引用传值方式: &符号

```
$m1=1;
```

```
$m2=&m2;     //引用传值方式
```

图示理解:



结果: 有 2 个变量, 只有 1 个值 (数据空间), 对其任何一个操作, 就是在操作数据值

```
14 $v1 = 1;
15 $v2 = $v1; //值传递;
16 $v1 = 10;
17 echo "<br />v1 = $v1"; //10
18 echo "<br />v2 = $v2"; //1
19 unset($v1); //断开$v1跟其数据的关系
20 $r1 = isset($v2); //??
21 echo "<br />"; var_dump($r1); //true
22
23 echo "<hr />";
24
25 $m1 = 1;
26 $m2 = &$m1; //引用传值方式
27 $m1 = 10;
28 echo "<br />m1 = $m1"; //10
29 echo "<br />m2 = $m2"; //10
30 unset($m1); //断开$m1跟其数据的关系
31 $r2 = isset($m2); //??
32 echo "<br />"; var_dump($r2); //true
```

## 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; //???
```

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”；

```
$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
```

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