

1. 12

变量

变量——名字。

```
echo $v1;    ///输出的是数据 1
$v2 = $v1 + 3;    //此时，其实进行的是 1+3 的计算！
```

使用形式：每个变量名前面必须以\$开头。

定义形式：php 中，不支持“单纯定义”一个变量，而是，在第一次一个变量赋值的时候，就算是定义变量！

变量的基本操作

只有 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——取值；

概括来说，大约有如下情形：

```
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

```

14 $v1 = isset($s1);» //判断$s1是否存在，结果是false
15 $s2 = 2;
16 $v2 = isset($s2);» //true
17
18 $s3 = false;
19 $v3 = isset($s3);» //true
20
21 $s4 = "";
22 $v4 = isset($s4);» //true
23
24 $s5 = null;»//特别注意：null这个“值”（数据）的含义就是“没有数据”
25 $v5 = isset($s5);» //false;

```

删除 unset(变量名):

含义：删除一个变量，并不是指将该变量从程序中删掉，而是，“断开”该变量名跟该变量原有的数据值之间的“引用关系”（联系）！，此时，会有这样的结果：

- 1，该变量名已经不指向（引用）任何数据了，则此时其 `isset()` 判断的结果就是 `false`
- 2，该数据（可能）没有任何变量指向它了，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种常见的命名法：

- 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: 每个单词首字母大写

举例: `$Name` `$MyName` `$MyFatherName`

3, 下划线分割法: 每个单词小写, 并且之间用下划线分开

举例: `$name` `$my_name` `$my_father_name`

变量的传值方式

1, 变量的传值方式, 是指“一个变量, 传给另一个变量”的内部细节形式——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 `$v1` 的值取出来 (注意: `$v1` 中的值还在), 然后再用该值给 `$v2` 赋值。

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;

即 `$v1 = 10;` 则 `echo $v2 ;` //输出 1

引用传递

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);

结果是:

1, 此时仍然有了 2 个变量, 但只有一个数据值 (数据空间), 2 个变量都共同指向该数据空间。

2, 对其中任何一个变量的操作, 其实都是在操作数据值 (空间);

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
17
```

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