

Php 的基本语法形式

PHP 的标记符

形式:

1, (推荐)

```
<?PHP
```

这里要写符合 PHP 语法的语句;

```
?>
```

2,

```
<script language="php">
```

这里要写符合 PHP 语法的语句;

```
<script>
```

3,(不推荐)

```
<?
```

这里要写符合 PHP 语法的语句;

```
?>
```

*标签缩写;

它依赖于 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, 则上述形式就可以用了:

PHP 的结束标记(比如?>),在如下情况可以省略

Php 的语句之后,再没有 HTML 代码部分

php 区分大小写特性

1,变量区分大小写;

2,常量默认也区分,但可以人为设定不区分(不推荐);

3,其他场合的关键字都不区分,比如函数名,系统关键字(for.return ...);

一条语句使用一个分号结束

1. 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 的注释小技巧

技巧 1

```
/*    (/**就注释掉了)
```

```
    代码....
```

```
/**/
```

技巧 2

```
if(1 == 0){
```

```
    代码.....
```

```
}
```

反注释就改成

```
(1==1)
```

变量

不支持单纯定义一个变量,必须要有值;

变量的判断

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`.

变量的删除

`unset`(变量名)

变量命名规则

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
- 2, 帕斯卡命名法: 每个单词首字母大写
- 3, 下划线分割法: 每个单词小写, 并且之间用下划线分开

变量在""中可以被识别.

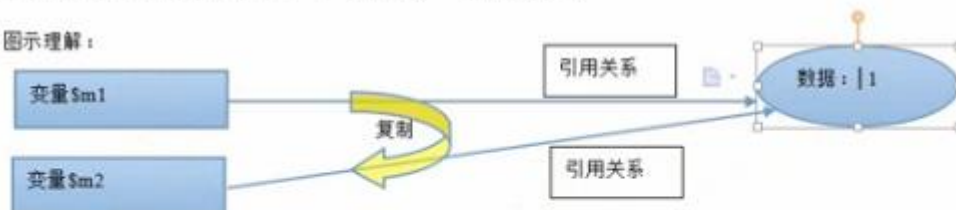
引用传递:

PHP 中只能一种语法形式可以实现变量的引用全传值方式:'&'符号;

例: `$m1 = 1`

`$m2 = &$m1` //引用传值方式;

图示理解:



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);

- `Var dump`(变量名) //可以输出任意函数的详细信息;

可变变量:

- 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

I