

## 4-7-8 学习回顾

### php 运行环境

web 软件运行模式

web 运行原理

php 网页运行原理

安装 apache

配置 apache 以运行 php 程序

配置 apache 以运行 php 程序

目标 1: `echo'abc';`

在 apache 的主配置文件 `httpd.conf` 中, 添加两行

`LoadModule php5_module 'php 语言包所在目录/php5apache2.2.dll'`

`AddType application/x-httpd-php .php .php3 .phtml`

补充一个检测 apache 的配置语法命令

`Httpd.exe -t` 这个命令

其所在位置: apache 安装目录/bin/

### Php 的配置

首先, php 的配置, 依赖于一个前提:

让 apache 知道 `php.ini` 文件在哪里!

在 apache 的配置文件 `httpd.conf`, 添加一行:

`PHPIniDir 'php 语言包所在目录/'`

### 时区配置

`Data.timezone=PRC`

### php 模块配置:

1. 指定模块位置: `extension_dir="php 语言包所在目录/ext"`

2. 打开所需要的模块

`extension=php_mysql.dll`

### mysql 的安装

#### 主机 (站点) 配置

一个站点 2 个核心的信息为:

`ServerName` 站点名称/服务器名/域名

`DocumentRoot` '站点的真实目录'

端口监听 `Listen` 端口号

设置主机 (站点) 的名字, 服务器名, 站点名

设置主机 (站点) 的物理路径

目录访问权限

`<Directory '要设置权限的物理路径'>`

`Options Indexes` #设置目录显示'文件列表'

`Order Deny,Allow` #x 先拒绝后允许或先允许后拒绝, 不管哪个顺序, 都是后都后

者覆盖前者

`Allow from All`

#下一行, 可以上我们在文件夹中设定权限

```
AllowOverride all
</Directory>
```

目录访问权限中的‘分布式权限’

一个站点的任何一个文件夹，都可以对其进行‘单独权限设置’；

- 1, 需要在该文件夹中放入一个特殊名字的文件：.htaccess
- 2, 该文件夹的内容，几科可以跟 Directory 中的设置一样
- 3, 该文件中的设置，优先于 Directory 中的设置，即如果有同样设置项但值不同，以.htaccess 中的为准
- 4, .htaccess 中的设置项，无需重启 apache,就可以立即生效

单站点配置

多站点配置

分 3 步：

1. 在 httpd.conf 中，打开‘虚拟主机配置文件’,apache 安装位置/conf/extra/httpd-vhosts.conf

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

2. 在该虚拟主机配置文件（httpd-vhosts.conf）,添加一行表示要进行多站点配置的 IP 端口

```
NameVirtualHost *:80
```

3. 然后，在其中，一个站点一个站点进行配置，第个站点基本形式如下：

```
<VirtualHost>
  ServerName ...
  DocumentRoot
<Directory ...>
</Directory>
```

```
</VirtualHost>
```

站点别名的设置

```
ServerAlias 别名 1 别名 2 ...
```

注意:它是写在 ServerName 并列位置

目录别名（虚拟目录）的设置

就是设置一个“虚拟的目录名字”，以供外界访问‘访问之外的内容’；

4-9

作业 问题

>>,

在 httpd-vhost.conf 文件中配置至少 3 个虚拟主机:

要求: 一个可以访问到基础班做的“呱呱网”, 一个可以访问到基础班做的“当当网”, 还有一个就是 localhost, 可以访问到 apache 原始安装后之后的初始页面 (显示 it works!)

>>, |

再创建一个虚拟主机 (站点), 域名用自己的名字, 其中只有一个文件: myinfo.php, 其中用 php 输出一些自己的信息, 比如: 姓名, 性别, 年龄, 爱好, 在哪里——当然你可以把网页做的漂亮一点。该站点一打开 (即类似输入 http://www.luodihua.com 后回车) 就显示该网页。——此站点可作为此后几天 php 核心知识学习的测试网站, 并还可以按天建立目录, 每天的东西放入该天。类似这样:

姓名:	罗弟华
性别:	男
生日:	02-10
学历:	大学本科
<a href="#">2014-07-07</a> <a href="#">2014-07-08</a> <a href="#">2014-07-09</a>	

>>,

有些网站中显示 (或使用) 的是 phtml 文件, 其实是 php 文件。如果你将你的 php 文件后缀保存为 phtml, 你该怎样做配置使之能够访问并执行? ——试着在第二题的基础上加一个链接, 链接到一个 phtml 文件, 该文件中用 php 输出当前时间 (或别的一些内容)。

```
<Directory 'c:/www/'>
```

```
DirectoryIndex    myinfo.php
```

```
</Directory>
```

← → ↻ ⓘ localhost/zuoye.php

姓名: 张三

[day1](#) [day2](#)

## PHP 基本语法形式

Php 的标记

有以下形式:

形式 1 (推荐)

```
<?php
```

这里要写符合 php 语法语句

```
?>
```

形式 2:

```
<script language='php'>
```

这里要写符合 php 语法语句

```
</script>
```

形式 3(不推荐)

```
<?
```

这里要写符合 php 语法语句

```
?>
```

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，则上述情式就可以用了

```
; http://php.net/short-open-tag
short_open_tag = On
```

Php 的结束标记（比如 ?>），在如下情况下可以省略：

Php 的语句之后，没有其他 html 代码部分

**Php 的区分大小写的特性：**

1. 变量区分大小写；
2. 常量通常默认也区分，但可以人为设定为不区分（这种不推荐）
3. 其他场合的关键字都不区分，比如函数名，系统关键字（for,if,return...）

**一条语句使用一个分号结束**

1. 在一个 php 的语句标记中最后一个分号可省略

```
12 <body>
13 <?php
14 echo "<br />abc11";
15 echo "<br />abc12";
16 echo "<br />abc13"
17 ?>
18 <hr />
```

不能省略

可以省略

2. Php 结束标记省略则不能省略最后一个分号；

```
</html>
<?php
echo "<br />这是最后的php代码，其后没有其他代码了";
echo "<br />这是最后的php代码，其后没有其他代码了";
echo "<br />这是最后的php代码，其后没有其他代码了"
```

但：这里不能省略，因为其后没有php的结尾标记

PHP 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 \*/

2 个多行注释的技巧：

有时候，我们因为测试或别的原因，需要将一大段代码多次进行‘注释’或‘反注释’，则此是可以使用如下 2 个技巧来实现

技巧 1：

```
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

```
4  /**/
5  echo "<br />代码1";    这样就有全部取消注释了
6  echo "<br />代码2";
7  echo "<br />代码3,代表多行需要注释的代码";
8  /**/
```

技巧 2：

```
if( 1 == 0){
    echo "<br />代码1";
    echo "<br />代码2";
    echo "<br />代码3,代表多行需要注释的代码";
}
```

反注释

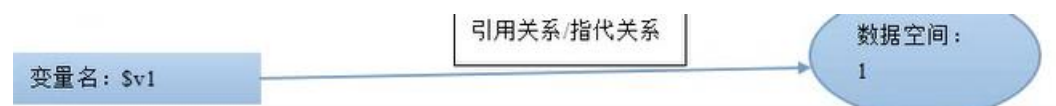
```
if( 1 == 1){
    echo "<br />代码1";
    echo "<br />代码2";
    echo "<br />代码3,代表多行需要注释的代码";
}
```

## 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极其中数据的一个‘标识符’——也就是一个名字。

\$v1= 1;这一行代码，图示如下：



这表明，使用该变理，就是在使用该数据的数据值。比如：

echo \$v1; //输出的是数据 1；

\$2=\$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’一个变理，而是在第一次变量赋值的进候，就算是定义变量。

## 变量的基本操作

只有 4 件操作：

赋值：

等号 (=) 的左边放变量名，右边放的‘数据’，就是赋值。 Eg \$v1=1;

取值：

任何需要一个数据的语句使用一个变量，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 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’，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



Eg \$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

A 以字母或下划线开头

B 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：首个单词小写，其后每个单词的首字母大写

- Eg \$name \$myName \$myFatherName
- 帕斯卡命名法：每个单词首字母大写  
Eg \$Name \$MyName \$MyFatherName
  - 下划线分割当：每个单词小写，并且之前用\_下划线分开  
Eg \$name \$my\_name \$my\_father\_name

## 变量的传值方式

- 变量的传值方式，是指‘一个变量，传给另一个变量’的内部细节形成 ---单对单；
- 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
```

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：

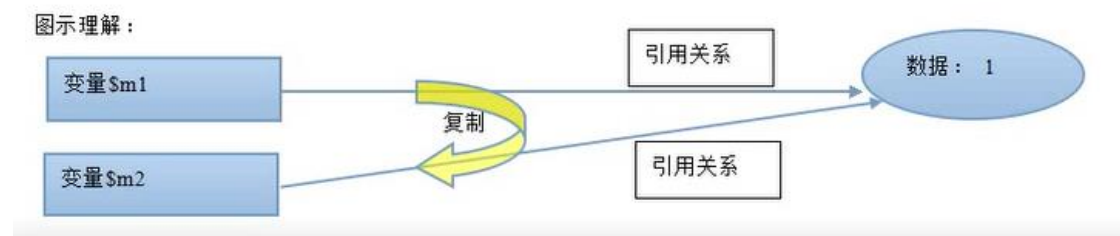
- 这两个变量此进是值相等的；
- 这两个变量又是互相---互不影响  
即\$v1=10; 则 echo \$v2; //输出 1

### 引用传递

Php 中,只有一种语法形式可以实现引用传值方式：&符合

```
Eg:$m1=1;
$m2=& $m1; // 引用方式
```

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，此进，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：

- 此时仍然有了 2 个变量，但吸有一个数据值（数据空间），2 个变量都共同指向该数据空间
  - 对其中任何一个变量操作，其实都是在操作数据值（空间）；
- 代码对比如下：



```
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
33 ?>
34 </body>
35 </html>
```

```
$v1 = 1;
$v2 = $v1; //值传递;
$v1 = 10;
echo "<br />v1 = $v1"; //10
echo "<br />v2 = $v2"; //1
unset($v1); //断开$v1跟其数据的关系
$r1 = isset($v2); //??
echo "<br />"; var_dump($r1); //true

echo "<hr />";

$m1 = 1;
$m2 = &$m1; //引用传值方式
$m1 = 10;
echo "<br />m1 = $m1"; //10
echo "<br />m2 = $m2"; //10
unset($m1); //断开$m1跟其数据的关系
$r2 = isset($m2); //??
echo "<br />"; var_dump($r2); //true
?>
```

v1 = 10  
v2 = 1  
bool(true)

m1 = 10  
m2 = 10  
bool(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; // 输出 10

1. 在 php 中一个'\$'后面，总是跟着一个变量名
2. 这里 echo 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

```
<?php
$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:  
//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  
$v1 = 1;  
$v2 = 12;  
$v3 = 33;  
$v4 = 44;  
$v5 = 115;  
//求这5个变量的和;  
//其他语言, 只能“一个一个”加起来。  
$sum = 0; //用于存储总和  
for($i = 1; $i <= 5; $i++){  
    $v = "v" . $i; //这里, 结果其实只是一个“字符串”, 比如“v1”, “v2”..  
    $sum += $$v;  
}  
echo "<br />sum = $sum";
```

10  
sum = 205

编辑图片