

第二章---PHP 语法基础（黑 p16）

2.1.1.php 标记

```
<?php ?>
```

2.1.2.注释

- (1) 单行注释 //
- (2) 多行注释 /* */

2.1.3.语句结束符 ;

2.2.1 变量与常量（黑 p18）

变量是可变的可以保存数据的容器，变量由符号\$组成

```
//2.1
//传值赋值，直接赋值使用
$age = 12; //定义变量，赋值为12
$num = $age; //将age的值赋给num
$age = 100; //修改age的值
echo $num; //输出num，num值为12
//注意$num = $age是对变量的引用赋值，$num值依然为12

//引用赋值，其中一个值发生改变另一个也会改变
$age = 12;
$num = &$age; //将age的值赋给num
$age = 100; //修改age的值
echo $num; //输出num，num值为100
```

注：标识符

- (1) 标识符只能由字母、数字和下划线组成。
- (2) 标识符可以由一个或多个字符组成，必须以字母或下划线开头。
- (3) 当标识符用作变量名时，区分大小写。
- (4) 如果标识符由多个单词组成，那么应使用下划线进行分隔，如：user_name。

```
//2.1.2
$text; //合法
$_test; //合法
$text66; //合法
$66text; //非法
$123; //非法
$te st; //非法
$*text; //非法
```

2.2.2 可变变量（黑 p19）

通过可变变量，将另外一个变量的值作为该变量的名称，在变量前多加一个\$符号即可

```
//2.2.2
$a = "say";
$say = "hello";
$hello = "kucy";
echo $a; //say
echo $$a; //hello
echo $ $a; //kucy
```

2.2.3 常量（黑 p19）

常量为保存在脚本运行中始终保持不变的量，特点为一但被定义就**不能被修改或重新定义**。

- (1) 使用 define () 定义常量（第三个参数可忽略）
第一个参数表示常量名称，第二个参数表示常量值，第三个参数，true 表示区分大小写，false 区分大小写
- (2) 使用 const 关键字定义常量

```
define('con','hello',true); //false为默认值
echo con; //hello

const PI = 3.14;
echo PI; //3.14
```

2.2.4 预定义常量（黑 p20）

PHP 预先定义好的常量，用来获取 PHP 中的信息。

常量名	功能描述
PHP_VERSION	获取 PHP 的版本信息，如 7.3.19
PHP_OS	获取解析 PHP 的操作系统类型，如：WINNT
PHP_INT_MAX	获取 PHP 中整型数的最大值 2147483647
PHP_INT_SIZE	获取 PHP 中整型数的字长，如：4
E_ERROR	表示运行时致命性错误，使用 1 表示
E_WARNING	表示运行时警告错误（非致命），使用 2 表示
E_PARSE	表示编译时解析错误，使用 4 表示
E_NOTICE	表示运行时提醒信息，使用 8 表示

注：关键字

编程语言里预先定义好并赋予了特殊含义的单词，也称作保留字。

注意：关键字不能作为常量、函数名或类名使用，关键字不推荐作为变量名使用，容易混淆。

常见的关键字						
if	and	as	break	callable	case	catch
class	clone	const	continue	declare	default	echo
else	extends	final	finally	for	foreach	function
new	abstract	include	namespace	global	or	print
use	protected	public	require	return	static	switch
throw	trait	try	private	var	while	yield

📖案例：显示服务器信息

```
<title>服务器信息</title>
</head>
<body>
  <table>
    <tr><th colspan="2">服务器信息展示</th></tr>
    <tr>
      <td>当前PHP版本号:</td><td><?php echo PHP_VERSION; ?></td>
    </tr>
    <tr>
      <td>操作系统的类型:</td><td><?php echo PHP_OS; ?></td>
    </tr>
  </table>
</body>
```

服务器信息展示

当前PHP版本号: 7.4.26
操作系统的类型: WINNT

2.3.1 数据类型 (黑 p21)

1. 布尔型

用于逻辑判断，只有 true 和 false 两个值，表示“真”和“假”，不区分大小写。
特殊情况下，0 表示 false，1 表示 true

2. 整形

用于表示整数，由八进制、十进制和十六进制表示，前面加上“+”或“-”符号表示正数或负数。
八进制数：使用 0~7 表示，且数字前必须加上 0。
十六进制数：使用 0~9 和 A~F 表示，数字前必须加上 0x。

3. 浮点型

用于保存浮点数或整数，浮点数在程序中用来表示小数。浮点数通常有两种表示方式：标准格式和科学记数法格式。

4. 字符串型

由连续的字母、数字或字符组成的字符序列。字符串通常使用单引号或双引号作为定界符。

注意：“.”是字符串连接符，用于连接字符串、数字、变量或常量等；“,”仅适用于 echo。

```
//3.1.1
//布尔型
$flag1 = true; // 将true赋值给变量$flag1
$flag2 = false; // 将false赋值为变量$flag2
//整形
$oct = 073; // 八进制数值59
$dec = 59; // 十进制数
$hex = 0x3b; // 十六进制数值59
//浮点数
$num1 = 1.759; // 标准格式
$num2 = -4.382; // 标准格式
$num3 = 3.14E5; // 科学记数法格式
$num4 = 7.469E-3; // 科学记数法格式
//字符串型
echo '生命在于运动！';
echo '生命在于' . '运动！';
echo '生命在于', '运动！'; // 输出结果：生命在于运动！
//注意：“.”是字符串连接符，用于连接字符串、数字、变量或常量等；“,”仅适用于echo。
```

2.3.2 字符串解析 (黑 p23)

1.单引号字符串和双引号字符串的区别

单引号字符串中书写的变量名会原样输出，双引号字符串中会被解析。

```
$name = 'Tom';
$area = 'China';
echo $name . " from $area"; // 输出结果: Tom from China
echo $name . ' from $area'; // 输出结果: Tom from $area
```

2. 界定变量

在双引号字符串中，使用“{}”对变量进行界定。

```
$ap = 'ma';
$apple = 'test';
echo "$apple"; // 输出结果: test
echo "{$ap}ple"; // 输出结果: maple
```

3.转义字符

在双引号中：双引号用“\” 换行符\n 制表符\t 等单引号只支持使用\和\\

```
echo "嵌套\"双引号\""; //嵌套"双引号"
echo '嵌套\'单引号\''; //嵌套'单引号'
```

2.3.3 数据类型检测 (黑 p23)

检测数据类型：使用内置的 is_*()函数检测，如果符合检测的数据类型，返回 true，否则返回 false。
为了查看数据类型，通常使用 var_dump()来检测

函数名称	功能描述
is_bool()	检测是否为布尔类型
is_string()	检测是否为字符串类型
is_float()	检测是否为浮点类型
is_int()	检测是否为整型
is_null()	检测是否为空值
is_array()	检测是否为数组
is_resource()	检测是否为资源
is_object()	检测是否为对象类型
is_numeric()	检测是否为数字或由数字组成的字符串

```
//数据类型检测示例
var_dump(is_bool('1')); // 输出结果: bool(false)
var_dump(is_string('php')); // 输出结果: bool(true)
var_dump(is_float('23')); // 输出结果: bool(false)
var_dump(is_int('23.0')); // 输出结果: bool(false)
var_dump(is_numeric('45.6')); // 输出结果: bool(true)
```

2.3.4 数据类型转换（黑 p24）

1. 自动类型转换

当运算与期望的结果类型不匹配时，PHP 将自动进行类型转换，无需开发人员做任何操作。
常见的自动类型转换有 3 种，分别为转换成布尔型、转换成整型和转换成字符串型。

数据类型转换

转换成布尔型

整型值0、浮点型值0.0、空字符串、字符串“0” → false
其他值 → true

转换成整型

布尔值true → 1；布尔值false → 0
浮点型转换成整型向下取整
若字符串以数字开始，则使用该数值，否则转换为0

转换成字符串型

布尔值true → “1”；布尔值false → 空字符串
整型、浮点型转换成字符串，直接将数字转换成字符串

```
//自动类型转换示例
var_dump(0 == false);           // 运行结果: bool(true)
var_dump(NULL == false);        // 运行结果: bool(true)
var_dump('888php' == 888);      // 运行结果: bool(true)
var_dump(true + 1);              // 运行结果: int(2)
var_dump('3');
```

2.强制类型转换

在要转换的数据或变量之前加上“(目标类型)”即可

强制类型	功能描述	强制类型	功能描述
(boolean) 或(bool)	转为布尔型	(float)	转为浮点型
(string)	转为字符串型	(array)	转为数组
(integer)或 (int)	转为整型	(object)	转为对象

```
//强制类型转换
var_dump((bool)-5.9);           // 运行结果: bool(true)
var_dump((int)'hello');         // 运行结果: int(0)
var_dump((float>false);        // 运行结果: float(0)
var_dump((string)12);           // 运行结果: string(2) "12"
```

2.4 运算符（黑 p25）

2.4.1 算术运算符

运算符	作用	范例	结果
+	加	echo 5 + 5;	10
-	减	echo 6 - 4;	2
*	乘	echo 3 * 4;	12
/	除	echo 5 / 5;	1
%	取模（即算术中的求余数）	echo 7 % 5;	2

注：错误抑制运算符：用来抑制错误的输出，在可能出现错误的代码前加@符号。

```
$num1 = 10 / 0; // 输出结果会报错，除数不能为0
$num2 = @(10 / 0); // 输出结果不会报错
```

2.4.2 赋值运算符

运算符	作用	范例	结果
+	加	echo 5 + 5;	10
-	减	echo 6 - 4;	2
*	乘	echo 3 * 4;	12
/	除	echo 5 / 5;	1
%	取模 (求余数)	echo 7 % 5;	2
运算符	作用	范例	结果
+	加	echo 5 + 5;	10

案例：商品价格计算（黑 p26）

商品名称	购买数量(斤)	商品价格(元/斤)
香蕉	2	7.99
苹果	1	6.89
橘子	3	3.99
商品折扣: 0.8		
打折后购买商品总价格: 27.872 元		

```
1  <?php
2  //设定字符集
3  header('Content-type:text/html;charset=utf-8');
4  //定义一个常量，保存所有商品的折扣
5  const DISCOUNT = 0.8;
6  //定义变量，保存所有商品的名称
7  $fruit1 = '香蕉';
8  $fruit2 = '苹果';
9  $fruit3 = '橘子';
10 //对应商品的购买数量（斤）
11 $fruit1_num = 2;
12 $fruit2_num = 1;
13 $fruit3_num = 3;
14 //对应商品的价格（元/斤）
15 $fruit1_price = 7.99;
16 $fruit2_price = 6.89;
17 $fruit3_price = 3.99;
18 //依次计算每件商品的总价格
19 $fruit1_total = $fruit1_num * $fruit1_price;
20 $fruit2_total = $fruit2_num * $fruit2_price;
21 $fruit3_total = $fruit3_num * $fruit3_price;
22 //计算所有商品总价格
23 //计算公式：所有商品价格 = （香蕉总价格+苹果总价格+橘子总价格）*商品折扣
24 $total = ($fruit1_total + $fruit2_total + $fruit3_total)*DISCOUNT;
25 ?>
26 <title>服务器信息</title>
27
28 <table border="1" cellpadding="1" cellspacing="0">
29     <tr>
30         <td>商品名称</td>
31         <td>购买数量(斤)</td>
32         <td>商品价格(元/斤)</td>
33     </tr>
34     <tr>
35         <td><?php echo $fruit1; ?></td>
36         <td><?php echo $fruit1_num; ?></td>
37         <td><?php echo $fruit1_price; ?></td>
38     </tr>
39     <tr>
40         <td><?php echo $fruit2; ?></td>
41         <td><?php echo $fruit2_num; ?></td>
42         <td><?php echo $fruit2_price; ?></td>
43     </tr>
44     <tr>
45         <td><?php echo $fruit3; ?></td>
46         <td><?php echo $fruit3_num; ?></td>
47         <td><?php echo $fruit3_price; ?></td>
48     </tr>
49     <tr>
50         <td colspan="3">商品折扣: <span><?php echo DISCOUNT; ?></span></td>
51     </tr>
52     <tr>
53         <td colspan="3">打折后购买商品总价格: <?php echo $total; ?> 元</td>
54     </tr>
55 </table>
```

2.4.5 比较运算符

比较两个变量或表达式，其结果是一个布尔类型的 true 或 false

运算符	运算	范例	结果
==	等于	5 == 4	false
!=	不等于	5 != 4	true
<>	不等于	5 <> 4	true
===	全等	5 === 5	true
!==	不全等	5 !== '5'	true
>	大于	5 > 5	false
>=	大于或等于	5 >= 5	true
<	小于	5 < 5	false
<=	小于或等于	5 <= 5	true

2.4.6 合并运算符

用于简单的数据存在性判定

```
// <条件表达式> ?? <表达式>
$age = $age ?? 18;
// 如果$age存在，则使用$age的值，如果$age不存在，则将$age的值设置为18。
```

2.4.7 三元运算符

```
//<条件表达式> ? <表达式1> : <表达式2>
$age = 10;
echo $age >= 18 ? '已成年' : '未成年';
// 如果变量$age的值大于或等于18，输出结果为“已成年”，如果小于18，则输出结果为未成年。
```

2.4.8 逻辑运算符

用于逻辑判断的符号，其返回值类型是布尔类型。

运算符	运算	范例	结果
&&	与	\$a && \$b	\$a 和 \$b 都为 true，结果为 true，否则 false
	或	\$a \$b	\$a 和 \$b 中至少有一个为 true，结果为 true，否则 false
!	非	!\$a	若 \$a 为 false，结果为 true，否则相反
xor	异或	\$a xor \$b	\$a 和 \$b 一个为 true，一个为 false，结果 true，否则 false
and	与	\$a and \$b	与 “&&” 相同，但优先级较低
or	或	\$a or \$b	与 “ ” 相同，但优先级较低

2.4.9 递增递减运算符

也称为自增自减运算符，是一种特定形式的复合赋值运算符。

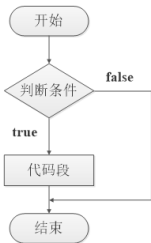
运算符	运算	范例	结果
++	自增（前）	\$a = 2; \$b = ++\$a;	\$a = 3; \$b = 3;
++	自增（后）	\$a = 2; \$b = \$a++;	\$a = 3; \$b = 2;
--	自减（前）	\$a = 2; \$b = --\$a;	\$a = 1; \$b = 1;
--	自减（后）	\$a = 2; \$b = \$a--;	\$a = 1; \$b = 2;

2.5 流程控制

1.if 语句

也称为单分支语句，用于判断当满足某种条件时就行某种处理。

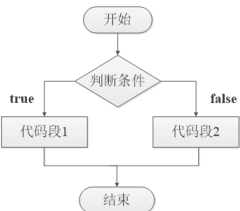
```
if (判断条件) {
    // 代码段
}
```



2. if...else 语句

也称为双分支语句，当满足某种条件时就进行某种处理，否则进行另一种处理。

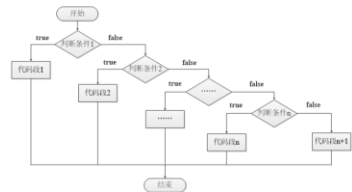
```
if (判断条件) {
    // 代码段1
} else {
    // 代码段2
}
```



3. if...else if...else 语句

也称为多分支语句，用于对多种条件进行判断，并进行不同处理

```
if (判断条件1) {
    // 代码段1
} else if (判断条件2) {
    // 代码段2
}
...
else if (判断条件n) {
    // 代码段n
} else {
    // 代码段n + 1
}
```



4. switch 语句

也是多分支语句，使代码更加清晰简洁、便于阅读

```
switch (表达式) {  
    case 值1:  
        代码块1;  
        break;  
    case 值2:  
        代码块2;  
        break;  
    ...  
    default:  
        代码块n;  
}
```

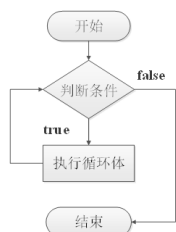
```
switch ($num) {  
    case 10:  
        echo "num的值是10";  
        break;  
    case 20:  
        echo "num的值是20";  
        break;  
    // 此处可以添加更多case  
    default:  
        echo "num的值是" . $num;  
}
```

5. while 语句 (黑 p35)

根据循环条件来判断是否重复执行某一段代码，先判断条件后执行循环体

while (循环条件)

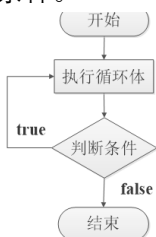
```
{  
    循环体  
}
```



6. do...while 语句

先无条件执行一次循环体后再判断条件。

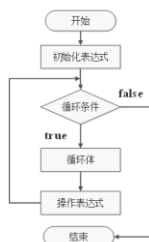
```
do {  
    执行语句  
} while (循环条件);
```



7. for 循环语句

是最常用的循环语句，适合循环次数已知情况

```
for (初始化表达式; 循环条件; 操作表达式) {  
    循环体  
}
```



8. 跳转语句

用于实现循环执行过程中程序流程的跳转。

常用的跳转语句：break 语句和 continue 语句。

break 用于终止当前循环，跳出循环体。

continue 用于结束本次循环的执行，开始下一轮循环的执行

案例：判断闰年 (黑 p34)

闰年的判断

输入的年份：2020

判断的结果：2020年是闰年

```
1  <?php  
2  $year = 2020;  
3  if (($year % 4 == 0) && ($year % 100 != 0) || ($year % 400 == 0)) {  
4      $result = $year . '年是闰年';  
5  } else {  
6      $result = $year . '年不是闰年';  
7  }  
8  ?>  
9  <h2>闰年的判断</h2>  
10 <p>输入的年份: <?php echo $year; ?></p>  
11 <p>判断的结果: <?php echo $result; ?></p>  
12  
13 <!-- 使用三元运算符判断闰年  
14  
15 <?php  
16 $year = 2015;  
17 $result = ($year % 4 == 0) && ($year % 100 != 0) || ($year % 400 == 0);  
18 ?>  
19 <h2>闰年的判断</h2>  
20 <p>输入的年份: <?php echo $year; ?>年  
21 <?php echo $result ? '是闰年' : '不是闰年'; ?></p>  
22
```

案例：金字塔循环 (黑 p37)

```
*  
***  
*****  
*****  
*****  
*****
```

```
1  <?php  
2  // 金字塔的总行数  
3  $n = 5;  
4  // 控制金字塔的行  
5  for($i = 1; $i <= $n; $i++){  
6      // 输出每行的空格  
7      for($j = 1; $j <= $n-$i; $j++){  
8          echo "&nbsp;";  
9      }  
10     // 输出每行的*  
11     for($k = 1; $k <= (2*$i)-1; $k++){  
12         echo "<font color='yellow'> . " . " . "</font>";  
13     }  
14     echo "<br/>";  
15 }
```

案例：九九乘法表 (黑 p38)

1×1=1									
1×2=2	2×2=4								
1×3=3	2×3=6	3×3=9							
1×4=4	2×4=8	3×4=12	4×4=16						
1×5=5	2×5=10	3×5=15	4×5=20	5×5=25					
1×6=6	2×6=12	3×6=18	4×6=24	5×6=30	6×6=36				
1×7=7	2×7=14	3×7=21	4×7=28	5×7=35	6×7=42	7×7=49			
1×8=8	2×8=16	3×8=24	4×8=32	5×8=40	6×8=48	7×8=56	8×8=64		
1×9=9	2×9=18	3×9=27	4×9=36	5×9=45	6×9=54	7×9=63	8×9=72	9×9=81	

```
13 <table class="bordered">  
14 <?php  
15 // 循环九九乘法表的层数  
16 for($i=1; $i<10; ++$i){  
17     echo "<tr>";  
18     // 循环每层中单元格的个数  
19     for($j=1; $j<=$i; $j++){  
20         // 计算每层中的乘法  
21         echo "<td>{$j}×{$i}={$j*$i}";  
22     }  
23     echo "</tr>";  
24 }  
25 ?>  
26 </table>
```


📖案例：判断学生成绩等级（黑 p33）

学生成绩等级

学生姓名：小明

学生分数：78分

成绩等级：C级

```
1  <?php
2  // 定义变量$name保存学生的名字
3  $name = '小明';
4  // 定义变量$score保存学生的分数
5  $score = 78;
6  // 判断$score是否为一个有效数值
7  if (is_int($score) || is_float($score)) {
8      // 根据分数所在区间，显示相应的得分等级
9      if ($score >= 90 && $score <= 100) {
10         $str = 'A级';
11     } else if ($score >= 80 && $score < 90) {
12         $str = 'B级';
13     } else if ($score >= 70 && $score < 80) {
14         $str = 'C级';
15     } else if ($score >= 60 && $score < 70) {
16         $str = 'D级';
17     } else if ($score >= 0 && $score < 60) {
18         $str = 'E级';
19     } else {
20         $str = '学生成绩范围必须在0~100之间!';
21     }
22 } else {
23     $str = '输入的学生成绩不是数值!';
24 }
25 ?>
26 <h2>学生成绩等级</h2>
27 <p>学生姓名：<?php echo $name; ?></p>
28 <p>学生分数：<?php echo $score; ?>分</p>
29 <p>成绩等级：<?php echo $str; ?></p>
30
```

3.1 函数（黑 p42）

静态变量：使用 static 关键字在函数中声明，函数执行完成后保留局部变量的值。（相当于 javascript 中的 let）

3.1.6 字符串函数（黑 p44）

是 PHP 的内置函数，用于操作字符串

函数名称	功能描述
strlen()	获取字符串的长度
strpos()	在指定字符串中查找目标字符串首次出现的位置
strrpos()	获取指定字符串在目标字符串中最后一次出现的位置
str_replace()	用于对字符串中的某些字符进行替换操作
substr()	用于获取字符串中的子串
explode()	将指定字符串拆分成数组
implode()	用指定的分隔符将数组中的键值拼接成字符串
trim()	去除字符串首尾处的空白字符（或指定的字符串）
str_repeat()	重复字符串
strcmp()	比较两个字符串的大小

3.1.7 数学函数

PHP 的内置函数，方便处理程序中的数学运算

函数名	功能描述
abs()	绝对值
ceil()	向上取最接近的整数
floor()	向下取最接近的整数
fmod()	返回除法的浮点数余数
is_nan()	判断是否为合法数值
max()	返回最大值
min()	返回最小值
pi()	返回圆周率的值
pow()	返回 x 的 y 次方
sqrt()	返回一个数的平方根
round()	对浮点数进行四舍五入
rand()	返回随机整数

3.1.8 时间和日期函数

PHP 的内置函数，用于处理日期和时间

函数名	功能描述
time()	获取当前的 UNIX 时间戳
date()	格式化一个本地时间/日期
mktime()	获取指定日期的 UNIX 时间戳
strtotime()	将字符串转化成 UNIX 时间戳
microtime()	获取当前 UNIX 时间戳和微秒数

📖案例：获取文件扩展名（黑 p46）

获取文件拓展名

文件路径：C:\images\apple.jpg

文件拓展名jpg

```
1  <?php
2  function getFileExt($path)
3  {
4      // 获取文件拓展名
5      $ext = substr($path, strrpos($path, '.') + 1);
6      // 获取返回结果
7      return $ext;
8  }
9  // 变量保存的文件路径
10 $path = 'C:\images\apple.jpg';
11 // 调用函数getFileExt() 获取文件拓展名
12 $ext = getFileExt($path);
13 ?>
14 <title>获取文件拓展名</title>
15 <h1>获取文件拓展名</h1>
16 <p>文件路径：<?php echo $path; ?></p>
17 <p>文件拓展名<?php echo $ext; ?></p>
```

3.2 数组 (黑 p48)

1. 定义数组

定义数组的两种方式: array()语言结构、短数组定义法。

```
-
3 // 使用array()定义数组
4 $info = array('id' => 1, 'name' => 'Tom');
5 $fruit = array(1 => 'apple', 3 => 'pear');
6 // 使用短数组定义数组
7 $info = ['id' => 1, 'name' => 'Tom'];
8 $num = [1, 4, 7, 9];
9
```

2. 新增数组元素

创建一个数组变量, 使用赋值运算符直接给变量赋值

```
$arr[] = 'PHP'; // 存储结果: $arr[0] = 'PHP'
$arr[] = 'Java'; // 存储结果: $arr[1] = 'Java'
$arr[3] = 'C语言'; // 存储结果: $arr[3] = 'C语言'
$arr[5] = 'C++'; // 存储结果: $arr[5] = 'C++'
$arr['sub'] = 'iOS'; // 存储结果: $arr['sub'] = 'iOS'
$arr[] = '网页平面'; // 存储结果: $arr[6] = '网页平面'
```

3. 访问数组

通过数组的键来访问数组元素

```
$info = ['id' => 1, 'name' => 'Tom'];
echo $info['name']; // 输出结果: Tom

print_r($info); // 输出结果: Array ( [id] => 1 [name] => Tom )

var_dump($info);
// 输出结果: array(2) { ["id"]=> int(1) ["name"]=> string(3) "Tom" }
```

4. 删除数组

使用 unset 语言结构可以删除数组中的某个元素, 也可以删除整个数组。

```
$fruit = ['apple', 'pear'];
unset($fruit[1]);
print_r($fruit); // 输出结果: Array ( [0] => apple )
unset($fruit);
print_r($fruit); // 输出结果: Notice: Undefined variable: fruit...
```

5. 判断数组元素是否存在

使用 isset()语言结构判断数组中的元素是否存在, 返回结果为布尔值。

3.2.3 数组遍历

依次访问数组中的每个元素, 通常使用 foreach 语句

```
$fruit = ['apple', 'pear'];
foreach ($fruit as $key => $value) {
    echo $key . '-' . $value . ' '; // 输出结果: 0-apple 1-pear
}
foreach ($fruit as $value) {
    echo $value; // 输出结果: apple pear
}
```

注: php 中使用“.”来拼接字符串

3.2.4 数组和字符串之间的转换

1. explode()函数

使用字符串分割另一个字符串, 返回数组。

第 1 个参数是分隔符,

第 2 个参数是要分割的字符串,

第 3 个是可选参数, 表示返回结果最多包含的元素个数, 当为负数 m 时, 返回除最后 m 个元素外的所有元素, 当为 0 时, 则把它当做 1 处理。

```
// ① 使用“n”作为分隔符对字符串进行分割
var_dump(explode('n', 'banana'));
// 输出结果: array(3) { [0]=> string(2) "ba" [1]=> string(1) "a" [2]=> string(1) "a" }

// ② 第3个参数为正数, 表示规定最多分割成几个元素
var_dump(explode('n', 'banana', 2));
// 输出结果: array(2) { [0]=> string(2) "ba" [1]=> string(3) "ana" }

// ③ 第3个参数为负数, 表示返回除最后几个元素外的所有元素
var_dump(explode('n', 'banana', -2));
// 输出结果: array(1) { [0]=> string(2) "ba" }
```

2. implode()函数

通过指定的连接符将数组拼接成一个字符串,

第 1 个参数是连接符,

第 2 个参数是待处理的数组。

```
$arr = [1, 2, 3]; // 定义一个数组
echo implode('-', $arr); // 输出结果: 1-2-3
```

案例: 订书单 (黑 p50)

商品订书单				
商品名称	价格(元)	产地	数量(件)	总价(元)
主板	379	广东	3	1137
显卡	799	上海	2	1598
硬盘	589	北京	5	2945
小计: 5680元				

```
1 <?php
2 header('Content-type:text/html;charset=utf-8');
3 //定义数组, 存储订书单中商品信息
4 $goods = array(
5     array('name'=>'主板','price'=>'379','producing'=>'广东','num'=>3),
6     array('name'=>'显卡','price'=>'799','producing'=>'上海','num'=>2),
7     array('name'=>'硬盘','price'=>'589','producing'=>'北京','num'=>5)
8 );
9 //商品价格总计
10 $total = 0;
11 //拼接订书单中信息
12 $str = "<title>商品订书单</title>";
13 $str .= "<table border='1' cellpadding='1' cellspacing='0'>";
14 $str .= "<tr><td colspan='5'>商品订书单</td></tr>";
15 $str .= "<tr><td>商品名称</td><td>价格(元)</td><td>产地</td><td>数量(件)</td><td>总价(元)</td></tr>";
16 //循环数组中商品信息
17 foreach($goods as $values){
18     $str .= "<tr>";
19     foreach($values as $v){
20         $str .= "<td>".$v."</td>";
21     }
22     //计算并拼接每件商品的总价格
23     $sum = $values['price']*$values['num'];
24     $str .= "<td>".$sum."</td>";
25     $str .= "</tr>";
26     //计算订书单中所有商品总价格
27     $total += $sum;
28 }
29 $str .= "<tr><td colspan='5'>小计: ".$total."</td></tr></table>";
30 //显示订书单信息
31 echo $str;
32
```

3.3 常用数组函数

1.count()函数 用于统计数组元素个数、合并等操作

```
$stu = [
    ['Tom', 'male', 18],
    ['Alice', 'female', 15],
    ['Julia', 'female', 14]
];
echo count($stu); // 输出结果: 3
echo count($stu, 1); // 输出结果: 12 3+9
```

2.range()函数 建立一个包含指定范围元素的数组

```
$arr = range("a","c");
print_r($arr); //Array ( [0] => a [1] => b [2] => c )
var_dump($arr);
/* D:\php39\课程资料\chapter03\1.php:22:
array (size=3)
  0 => string 'a' (length=1)
  1 => string 'b' (length=1)
  2 => string 'c' (length=1)
*/
```

3.array_merge() 合并一个或多个数组

```
$arr1=["a","b","c"];
$arr2=[1,2,3];
$arr3 = array_merge($arr1,$arr2);
print_r($arr3)
//Array ( [0] => a [1] => b [2] => c [3] => 1 [4] => 2 [5] => 3 )
```

4.array_chunk() 将一个数组分割成多个数组

第一个参数为分隔数组名

第二个参数为指定分隔后数组元素的个数

第三个为决定分割后的数组是否保留原来的键名

```
$arr4 = [1,2,3,4,5,6,7,8];
$arr5= array_chunk($arr4,5);
$arr6 = array_chunk($arr4,5,true);
$arr7 = array_chunk($arr4,5,false);//默认
print_r($arr5);
//Array ( [0] => Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 ) [1] => Array ( [0] => 6 [1] => 7 [2] => 8 ) )
echo '<br>';
print_r($arr6);
//Array ( [0] => Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 ) [1] => Array ( [5] => 6 [6] => 7 [7] => 8 ) )
echo '<br>';
print_r($arr7);
//Array ( [0] => Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 ) [1] => Array ( [0] => 6 [1] => 7 [2] => 8 ) )
```

3.3.2 数组排序函数 (黑 p52)

函数名	功能描述
sort()	对数组排序
rsort()	对数组逆向排序
ksort()	对数组按照键名排序
krsort()	对数组按照键名逆向排序
asort()	对数组进行排序并保持索引关系
arsort()	对数组进行逆向排序并保持索引关系
shuffle()	打乱数组顺序
array_reverse()	返回一个单元顺序相反的数组

```
$arr = ['dog', 'lion', 'cat'];
sort($arr, SORT_NATURAL);
print_r($arr);
// 输出结果: Array ( [0] => cat [1] => dog [2] => lion )
```

3.3.3 数组检索函数

用于对数组中的键、值进行查找、获取

函数名	功能描述
array_search()	在数组中搜索给定的值
array_unique ()	移除数组中重复的值
array_column ()	返回数组中指定的一列
array_keys ()	返回数组中键名
array_rand ()	从数组中随机取出一个或多个单元
key ()	从关联数组中取得键名
in_array ()	检查数组中是否存在某个值
array_values ()	返回数组中所有的值

案例：双色球 (黑 p53)

09 01 32 33 26 22 14

```
1 <?php
2 //创建一个1~33的红色球号码区数组
3 $red_num = range(1,33);
4 //随机从红色球号码区数组中获取6个键
5 $keys = array_rand($red_num,6);
6 //打乱键顺序
7 shuffle($keys);
8 //根据键获取红色球号码区数组中相应的值
9 foreach($keys as $v){
10     //判断：当红球号码是一位数时，在左侧补零
11     $red[] = $red_num[$v]<10 ? ('0'.$red_num[$v]) : $red_num[$v];
12 }
13 //随机从1~16的蓝色球号码区中取一个号码
14 $blue_num = rand(1,16);
15 //判断：当蓝球号码是一位数时，在左侧补零
16 $blue = $blue_num<10 ? ('0'.$blue_num) : $blue_num;
17 echo "<title>双色球</title>";
18 foreach($red as $v){
19     //输出红球号码
20     echo "<font color='red'>" . $v . "</font> ";
21 }
22 //输出蓝球号码
23 echo "<font color='blue'>" . $blue. "</font>";//查看运行结果
24
```

3.4 文件包含语句

include、require、include_once 和 require_once

// 第一种写法:

include '完整路径文件名';

// 第二种写法:

include('完整路径文件名');

注：require 语句与 include 语句的区别:

文件未找到：include 语句发生警告信息，程序继续运行；

require 语句发生致命错误，程序停止运行。

注：被包含文件的加载时间:

程序执行到 include 语句才会调用，require 语句在程序开始执行时则立即调用。

3.5 替代语法 (黑 p54)

在 HTML 模板中嵌入 PHP 代码时，将一些语句替换成可读性更好的语法。

把 if、while、for、foreach 和 switch 的左花括号“{”换成冒号“:”

把右花括号“}”分别换成“endif;”“endwhile;”“endfor;”“endforeach;”和“endswitch;”。

var_dump()

函数显示关于一个或多个表达式的结构信息，包括表达式的类型与值

print_r()

函数用于打印变量，以更容易理解的形式展示

sset()

函数用于检测变量是否已设置并且非 NULL。指定变量存在且不为 NULL，则返回 TRUE，否则返回 FALSE。测试一个被设置成 NULL 的变量，将返回 FALSE。

4.3 会话控制

4.3.1 cookie:

记录用户信息，行为，保存在浏览器里面，
保存本地，自己的电脑里面的。明文的形式。

创建 Cookie: 使用 `setcookie()`函数创建或修改 Cookie

```
bool setcookie(
    string $name,           // Cookie的名称 (必须)
    string $value = "",     // Cookie的值 (可选)
    int $expire = 0,        // Cookie的有效期 (可选)
    string $path = "",      // Cookie在服务器端的路径 (可选)
    string $domain = "",    // Cookie的有效域名 (可选)
    bool $secure = false,   // 指定是否通过安全的HTTPS连接来传输 (可选)
    bool $httponly = false  // 指定Cookie只能通过HTTP协议访问 (可选)
)
```

读取 Cookie: 使用超全局变量 `$_COOKIE` 读取 Cookie

Cookie 存储多个值: 在 Cookie 名后添加“[]”进行设置

```
// Cookie存储复杂数据
setcookie('user[name]', 'tom');
setcookie('user[age]', 30);
var_dump($_COOKIE);
// 输出结果: array(["user"]=>array(["name"]=>"tom"["age"]=>"30"))
```

Cookie 高级使用: 通过第 3 个参数设置 Cookie 的生命周期，通过第 4 个参数设置 Cookie 的访问路径，通过第 5 个参数设置访问网站。

```
// 第3个参数设置Cookie有效期
setcookie('name', 'value', time() + 1800); // 30分钟后过期
setcookie('name', 'value', time() + 60 * 60 * 24); // 一天后过期
setcookie('name', '', time() - 1); // 立即过期 (相当于删除Cookie)
// 第4个参数设置Cookie的访问路径
setcookie('name', 'value', 0, '/'); // 当前整个网站都可访问
// 第5个参数设置Cookie的有效域名
setcookie('name', 'value', 0, '/', '.com'); // 所有.com的网站都可以访问
```

4.3.4 session

用于身份标识，控制页面访问。保存在服务器上，一般的话无法直接访问

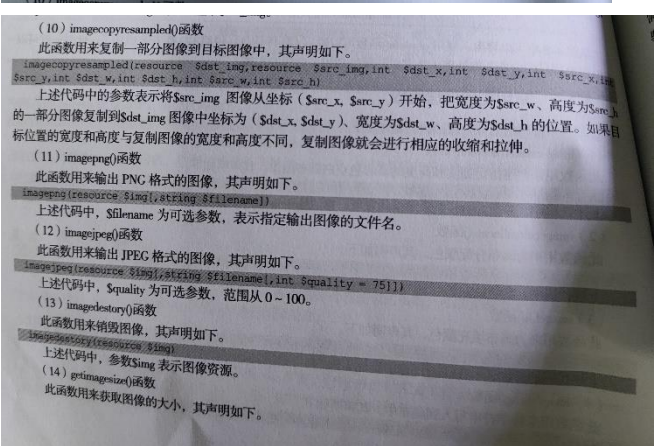
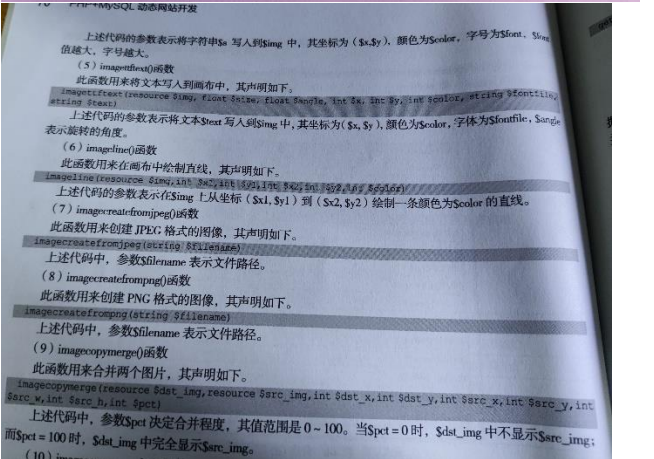
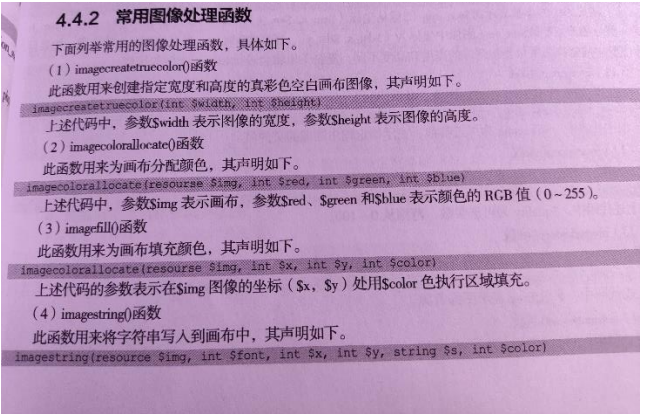
```
session_start(); // 开启Session
$_SESSION['name'] = 'tom'; // 向Session添加数据 (字符串)
$_SESSION['id'] = [1,2,3]; // 向Session添加数据 (数组)
unset($_SESSION['name']); // 删除单个数据
$_SESSION = []; // 删除所有数据
session_destroy(); // 结束当前会话
```

4.4.1 开启 GD 拓展

在 PHP 的配置文件 `php.ini`，找到
“`extension=gd2`”，去掉分号“;”。

查看开启结果: 重启 Apache 使配置生效，通过 `phpinfo()`函数查看 GD 库是否开启成功。

函数	作用
<code>imagecreatetruecolor()</code>	创建指定宽高的真彩色空白画布图像
<code>imagecolorallocate()</code>	为画布分配颜色
<code>imagefill()</code>	为画布填充颜色
<code>imagestring()</code>	将字符串写入到画布中
<code>imagettftext()</code>	将文本写入到画布中
<code>imageline()</code>	在画布中绘制直线
<code>imagecreatefromjpeg()</code>	创建 JPEG 格式的图像
<code>imagecreatefrompng()</code>	创建 PNG 格式的图像
<code>imagecopymerge()</code>	合并两个图片
<code>imagecopyresampled()</code>	复制一部分图像到目标图像中
<code>imagepng()</code>	输出 PNG 格式的图像
<code>imagejpeg()</code>	输出 JPEG 格式的图像
<code>imagedestroy()</code>	销毁图像
<code>getimagesize()</code>	获取图像的大小



案例：模拟登录

用户名:

密码:

验证码:

☐ 记住用户名 ☐ 7天免登录

[个人中心](#)

```
1 <!-- 登录页login.php -->
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>登录</title>
9 </head>
10 <body>
11 <?php
12 if(isset($_COOKIE["username"])){
13     $user = $_COOKIE["username"];
14     $check = "checked";
15 }
16 else{
17     $user = "";
18     $check = "";
19 }
20 >
21 <!-- 表单 -->
22 <form action="t_login.php" method="post">
23 <p>用户名: <input type="text" name="user" id="user" value=""<?php echo $user>"/></p>
24 <p>密码: <input type="password" name="pwd" id="pwd"/></p>
25 <p>验证码: <input type="text" name="captcha" id="captcha"/></p>
26 
27 <p>
28 <input type="checkbox" name="remember" id="remember" value="remember" <?php echo $check>>记住用户名
29 <input type="checkbox" name="login" id="login" value="login">7天免登录
30 </p>
31 <p><input type="submit" value="登录" name="btn" id="btn"></p>
32 </form>
33 <a href="welcome.php">个人中心</a>
34 </body>
35 </html>
```

```
1 <!-- 登录处理页t_login.php -->
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>后台登录</title>
9 </head>
10 <body>
11 <?php
12 //isset(变量或者数组元素): 有的返回true, 没有的返回false
13 //empty(数组元素): 判断数组元素是否为空, 空返回true, 非空返回false
14 if (isset($_POST["user"])) {
15     //判断验证码是否正确
16     session_start();
17     if (strtolower($_POST["captcha"]) != strtolower($_SESSION["captcha"])) {
18         echo "<script>alert('验证码不正确');location.href='login.php'</script>";
19         exit();
20     }
21     //得到用户名, 密码
22     $user = $_POST["user"];
23     $pwd = $_POST["pwd"];
24
25     //判断用户名和密码是否正确, admin,111
26     if ($user == "admin" && $pwd == "111") {
27         //P192. session怎么使用的??? 步骤
28         //怎么把值写到session里面.
29         session_start(); //启动会话
30         $_SESSION["user"] = $user; //登录成功的用户名存到session里面
31         //跳转到后面的页面
32         //echo "登录成功, 欢迎".$_SESSION["user"];
33
34         if (isset($_POST["remember"])) { //判断一下这个记住用户名的复选框是否有选中
35             setcookie("username", $user, time() + 60 * 60 * 24 * 30); //有选中设置cookie
36         } else {
37             setcookie("username", $user, time() - 1); //没有选中, 删除相应的cookie
38         }
39         //七天免登录
40         if (isset($_POST["login"])) {
41             setcookie("login", $user, time() + 60 * 60 * 24 * 7); //设置cookie
42         } else {
43             setcookie("login", $user, time() - 1); //删除cookie
44         }
45         echo "<script>alert('登录成功');location.href='welcome.php'</script>";
46     } else {
47         //直接输出javascript的代码
48         echo "<script>alert('用户名或者密码不正确');history.go(-1)</script>";
49     }
50 } else {
51     echo "<script>alert('不要直接访问后面网页, 非法操作');location.href='login.php'</script>";
52 }
53 >
```

```
1 <!-- 后台欢迎页面welcome.php -->
2 <?php
3 include("check.php");
4 >
5 <!DOCTYPE html>
6 <html lang="en">
7 <head>
8 <meta charset="UTF-8">
9 <meta http-equiv="X-UA-Compatible" content="IE=edge">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <title>后台欢迎页面</title>
12 </head>
13 <body>
14 <?php
15 echo "登录成功, 欢迎".$_SESSION["user"];
16 echo '<a href="exit.php">退出</a><br>';
17 echo date("H:i:s");
18 >
19 .png" alt="">
20 <?php
21 $week = date("N");//得到星期几
22 switch($week){
23     case 1:
24         $str = "新的一周开始啦";//操作
25         break;
26     case 2:
27         $str = "周二了";//操作
28         break;
29     case 3:
30         $str = "怎么才周三";//操作
31         break;
32     case 4:
33         $str = "熬一下, 快了快了";//操作
34         break;
35     case 5:
36         $str = "小周末了";//操作
37         break;
38     case 6:
39         $str = "终于可以睡懒觉了";//操作
40         break;
41     case 7:
42         $str = "明天又要上班了";//操作
43         break;
44 }
45 echo $str;
46 >
```

```
1 <!-- 限制登录页面check.php. 防止直接进入后台 -->
2 <?php
3 session_start();//启动session
4 if(isset($_COOKIE["login"])){
5     $_SESSION["user"] = $_COOKIE["login"];
6 }
7 if(isset($_SESSION["user"])==false){ //判断$_SESSION["user"]是否有数据
8     echo "<script>alert('先登录再使用');location.href='login.php'</script>";//没有登录, 给出提示, 返回登录的界面.
9 }
10 >
```

```
1 <!-- 退出页exit.php -->
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Document</title>
9 </head>
10 <body>
11 <?php
12 session_start();//启动session
13 //清除相应的session. 或者把所有的session清空, 书P194
14 unset($_SESSION["user"]); //清除指定的session
15 //session_destroy();//把所有的session都清空.
16 echo "<script>alert('退出成功');location.href='login.php'</script>";//给出提示, 返回登录页面.
17 >
18 </body>
19 </html>
```

```
1 <?php
2 function getCapcha($width, $height, $lines = 5, $length = 6)
3 {
4     $img = imagecreatetruecolor($width, $height);
5     $bg_color = imagecolorallocate($img, mt_rand(200, 255), mt_rand(200, 255), mt_rand(200, 255));
6     imagefill($img, 0, 0, $bg_color);
7     // $str_arr = range('A', 'Z');
8     $str_arr = array_merge(range("A", "Z"), range(0, 9));
9     shuffle($str_arr);
10    $captcha = "";
11    for ($i = 0; $i < $length; $i++) {
12        $captcha .= $str_arr[$i];
13    }
14    session_start();
15    $_SESSION["captcha"] = $captcha;
16    setcookie("captcha", $captcha);
17    $str_color = imagecolorallocate($img, mt_rand(0, 80), mt_rand(0, 80), mt_rand(0, 80));
18    // imagestring($img, 5, 10, 10, $captcha, $str_color);
19    imagegettext($img, 12, 0, 10, 20, $str_color, realpath("./font/cambria.ttf"), $captcha);
20    for ($i = 0; $i < $lines; $i++) {
21        $line_color = imagecolorallocate($img, mt_rand(100, 160), mt_rand(100, 160), mt_rand(100, 160));
22        imageLine($img, mt_rand(0, $width), mt_rand(0, $height), mt_rand(0, $width), mt_rand(0, $height), $line_color);
23    }
24    //防止因图形出错无法显示的问题
25    // ob_clean();
26    header('content-type:image/png');
27    imagepng($img);
28    imagedestroy($img);
29 }
30 getCapcha(100, 30, 5, 4);
```


一. PHP 与 Web 页面数据交互 P132

- 1.ajax + json。
- 2.表单：POST，把数据打包之后发送出去，数据量大，安全性高
- 3.附加 URL 来传输数据。GET，把数据以明文的方式附加到网址上来发送，数据量小，不安全

案例：模拟注册功能

```
12 <!-- 1、接收数据的网页。以相对路径的写法来写
13 2、如果没有填写，数据会发送给自己本身来处理 -->
14 <!-- 提交数据方法post/get -->
15 <form action="trade.php" method="get">
16 <p>用户名: <input type="text" name="user" id="user"></p>
17 <p>密码: <input type="password" name="pwd" id="pwd"></p>
18 <p>性别: <input type="radio" name="sex" id="sex1" value="男" checked>男
19 <input type="radio" name="sex" id="sex2" value="女">女
20 </p>
21 <p>爱好: <input type="checkbox" name="aihao[]" id="aihao1" value="游戏">游戏
22 <input type="checkbox" name="aihao[]" id="aihao2" value="电影">电影
23 <input type="checkbox" name="aihao[]" id="aihao3" value="逛街">逛街
24 <input type="checkbox" name="aihao[]" id="aihao4" value="学习">学习
25 </p>
26 <input type="submit" value="注册" name="btn" id="btn">
27 </form>
28 </body>
```

```
9 <body>
10 <!-- post提交的属性使用$_POST接收
11 get提交的属性使用$_GET接收
12 根据表单属性的name属性来接收 -->
13 <?php
14 /* echo"用户名: ".$_POST["user"]."<br>";
15 echo"密码: ".$_POST["pwd"]."<br>";
16 echo"性别: ".$_POST["sex"]."<br>";
17 print_r($_POST["aihao"]); */
18 var_dump($_POST);
19 var_dump($_GET);
20 echo"用户名: ".$_GET["user"]."<br>";
21 echo"密码: ".$_GET["pwd"]."<br>";
22 echo"性别: ".$_GET["sex"]."<br>";
23 print_r($_GET["aihao"]);
24 ?>
```

二. PHP 会话控制 P191

- a)session：用于身份标识，控制页面访问。保存在服务器上，一般的话无法直接访问。
- b)cookie：记录用户信息，行为，保存在浏览器里面，保存本地，自己的电脑里面的。明文的形式。

案例：模拟登录

```
12 <form action="table.php" method="post">
13 <p>用户登录</p>
14 <p>用户名:<input type="text" name="user" id="user"></p>
15 <p>密码:<input type="password" name="pwd" id="pwd"></p>
16 <p><input type="checkbox" name="usname" id="usname" value="usname">记住用户名
17 <input type="checkbox" name="denglu" id="denglu" value="denglu">七天免登录</p>
18 <input type="submit" value="登录" name="btn" id="btn">
19 </form>
20 </body>
<!-- empty (数据元素)，用来判断数组是否为空，空为true，非空为false -->
<!-- isset (变量\数组元素)，用来判断变量或数组元素是否存在，存在为true，不存在为false -->
<?php
if (isset($_POST["user"])) { /* 判断数组元素，变量是否有数据 */
// 得到前台传来的用户名和密码
$user = $_POST["user"];
$pwd = $_POST["pwd"];
if ($user == "admin" && $pwd == "123") {
session_start(); /* 启动会话 */
$_SESSION["users"] = $user; /* session赋值，把数据保存到session里面 */
// 给出提示
echo "登录成功，欢迎" . $_SESSION["users"];
} else {
echo "<script>alert('用户名或密码不正确');history.go(-1);</script>";
}
} else {
echo "<script>alert('无效操作');location.href = 'denglu.php';</script>";
}
?>
```