

第二章系统前台界面设计

PHP 网站开发实战项目式教程



学习目标

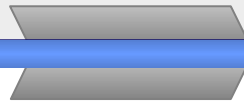
❖ 【知识目标】

- 通过 **Dreamweaver** 创建站点
- 掌握 **PHP** 语法结构、输出结果、注释
- 掌握 **PHP** 语言的常量、变量、数据类型、运算符及表达式
- 掌握 **PHP** 的流程控制语句
- 掌握 **PHP** 语言的数组
- 掌握 **PHP** 语言的函数（常用内置函数、时间日期函数、字符串函数等等）与自定义函数
- 掌握表单的处理

❖ 【技能目标】

- 掌握 **PHP** 基本的语法
- 会利用 **PHP** 开发工具进行简单的 **PHP** 程序编写





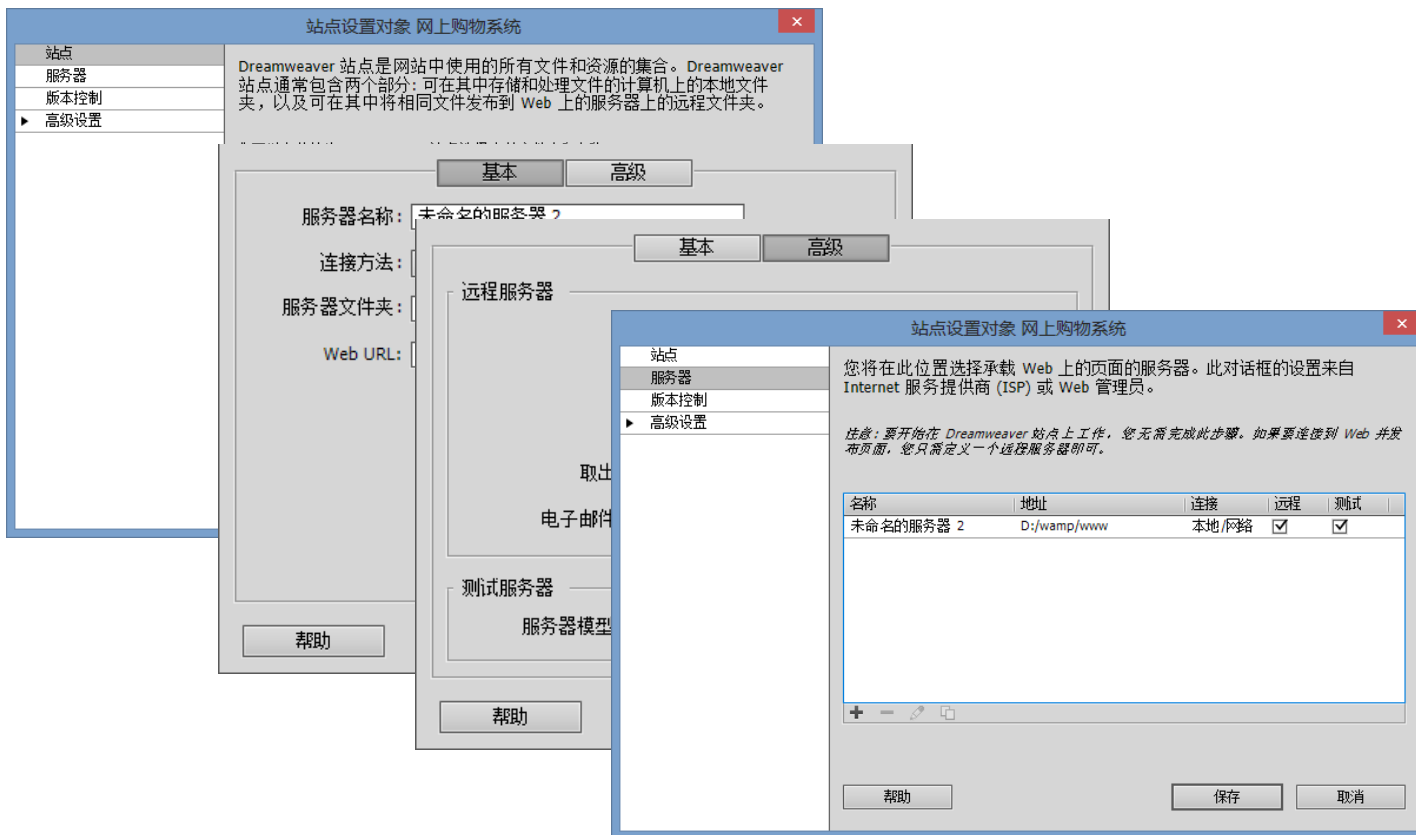
目录

- ❖ 子任务一：创建 **PHP** 动态网站站点
- ❖ 子任务二：商品订单页面设计
- ❖ 子任务三：商品计算功能实现





子任务一：创建 PHP 动态网站站点





子任务二：商品订单页面设计

❖ 3.2.1 PHP 标记



1. XML标记风格

```
<?php  
... //PHP代码  
?>
```

2. 短标记风格

```
<?  
... //PHP代码  
?>
```

3. ASP标记风格

```
<%  
... //PHP代码  
%>
```

4. script标记风格

```
<script language="php">  
... //PHP代码  
</script>
```





3.2.2 PHP 输出语句

❖ 格式:

```
echo “显示内容” ;
```

❖ 例如:

```
<?php
```

```
echo "<h2>PHP is fun!</h2>";
```

```
echo 'Hello world!<br>';
```

```
echo "I'm about to learn PHP!<br>";
```

```
echo "This", " string", " was", " made", " with multiple parameter  
s.";
```

```
?>
```

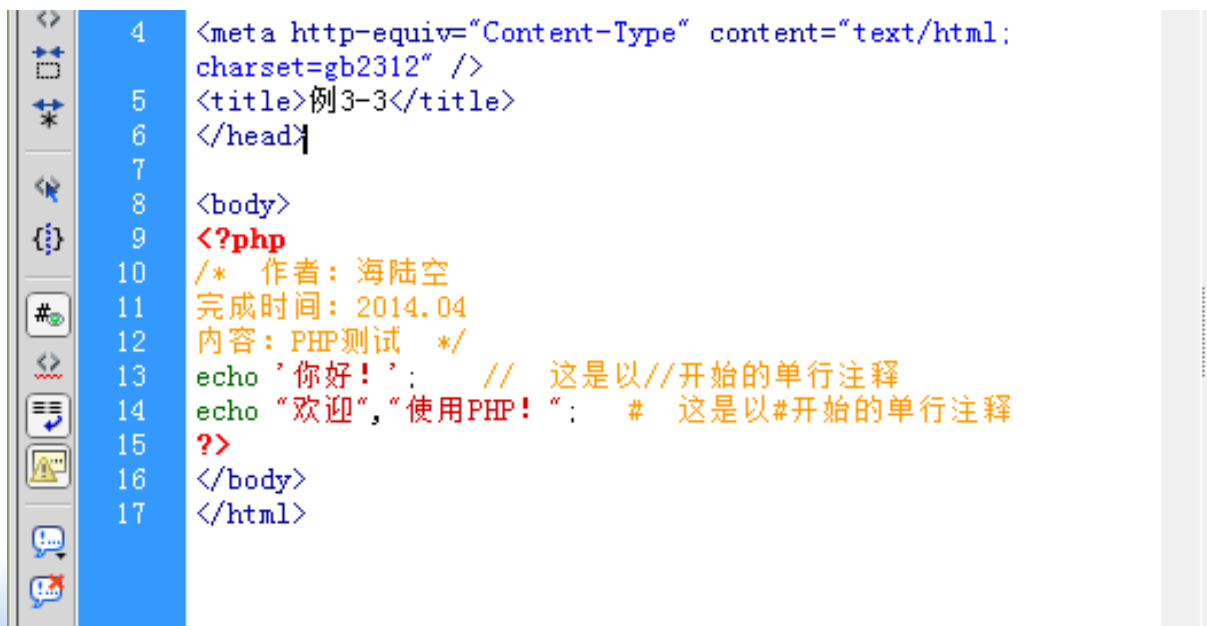
- php 语句结构严谨，每条语句后需要加分号结束；
- echo 输出的语句可以用双引号界定，也可以用单引号界定；
- echo 语句可以同时输出多个内容时，内容之间用逗号分隔





3.2.3 PHP 注释语句

1. 多行注释。以 “/*” 开始，“*/” 结束。
2. 单行注释。以 “//” 或 “#” 开始，所在行结束时结束。



```
4 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
5 <title>例3-3</title>
6 </head>
7
8 <body>
9 <?php
10 /* 作者：海陆空
11 完成时间：2014.04
12 内容：PHP测试 */
13 echo '你好!'; // 这是以//开始的单行注释
14 echo "欢迎","使用PHP! "; # 这是以#开始的单行注释
15 ?>
16 </body>
17 </html>
```





3.2.4 变量

- ❖ **定义** 变量是什么？为什么叫变量？变量就是一个储存数据的容器。因为这个容器里的数据可能随时都会改变（看你的程序怎么去运作），所以叫变量。





1. 变量的命名与赋值

- ❖ **命名** 变量的命名必须符合以下：
 - 变量必须由一个美元符号 “\$” 开头，例如： `$abc` ；
 - 变量名的第二个符号必须是字母或下划线，后面可以是字母、数字或者下划线组合；
 - 变量名严格区分大小写，如果两个变量只是大小写不同，则被视为两个变量。
 - **给变量命名最好做到见名知意。**





2. 变量的使用

- ❖ 双引号或单引号均可以定义字符变量，但两者绝不等价。使用单引号时，程序不会判断字符串中是否包含变量，也就是说，即使字符串中包含变量，也只输出变量名，不输出变量值；而使用双引号时，则输出变量值。

例如：

```
<?php
```

```
    $str=" 和平 ";
```

```
    echo " 世界 $str";
```

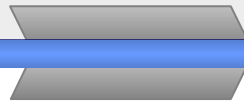
```
    echo "<br>";
```

```
    echo ' 世界 $str';
```

```
?>
```

世界和平
世界\$str

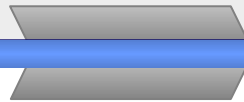




3. 数据类型

分类	类型	类型名称
标量类型	boolean	布尔型
	integer	整型
	float/double	浮点型
	string	字符串
复合类型	array	数组
	object	对象
特殊类型	resource	资源
	NULL	空





4. 转义字符

- ❖ 如果要输出引号或双引号，则需要使用转义字符” \”。
PHP 中还有一些特殊字符的转义字符。

符号	含义
\"	双引号
\\	反斜线
\n	换行
\r	回车
\t	跳位（ TAB ）
\\$	美元符号





5. 数据类型之间的转换

```
<?php
    $str1="1";
    $str2="ab";
    echo $num1=$str1+$str2;
        // $num1 的结果为整型 ( 1 )
    echo $num2=$str1+5;
        // $num2 的结果为整型 ( 6 )
    echo $num3=$str1+2.56;
        // $num3 的结果为浮点型
        ( 3.56 )
?>
```

❖ 自动转换

```
<?php
    $a=10;
    $b='string';
    echo $a.$b;
        // 输出 " 10string"
    echo (int)$b;
        // 显示 0
?>
```

❖ 强制转换





3.2.5 常量

- ❖ **定义** 常量是指在程序运行中无法修改的值。
- ❖ 1. 自定义常量

```
define(“ 常量名” ,” 常量值” );
```

- ❖

```
<?php  
define(“BUF_SIZE”,1000);  
echo BUF_SIZE;  
?>
```

常量一般用大写





2. 预定义常量

- 经常使用的预定义常量有 **5** 个，这些特殊的常量是不区分大小写的。

名称	说明
__FILE__	常量所在的文件的完整路径和文件名
__LINE__	常量所在文件中的当前行号
__FUNCTION__	常量所在的函数名称
__CLASS__	常量所在的类的名称
__METHOD__	常量所在的类的方法名





3.2.6 运算符和表达式

❖ 1. 运算符

类型	运算符
算术运算符	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>%</code> , <code>++</code> , <code>--</code>
字符串运算符	<code>.</code>
赋值运算符	<code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> , <code>.=</code>
位运算符	<code>&</code> , <code> </code> , <code>^</code> , <code>~</code> , <code><<</code> , <code>>></code>
比较运算符	<code>></code> , <code><</code> , <code>>=</code> , <code><=</code> , <code>==</code> , <code>===</code> , <code>!=</code> , <code>!==</code>
逻辑运算符	<code>&&</code> , <code> </code> , <code>!</code> , <code>xor</code>
其它运算符	<code>Condition?Value if True:value if False</code>
运算符的优先级和结合性	





3.2.6 运算符和表达式

❖ 2. 表达式

- ❖ **定义** 操作数和操作符组合在一起即组成表达式。表达式是由一个或者多个操作符连接起来的操作数，用来计算出一个确定的值。





3.2.7 流程控制语句

❖ 1. if 条件结构

```
if( 条件表达式) {  
...  
}
```

```
if( 条件表达式) {  
    语句块 1;}  
else{  
    语句块 2; }
```

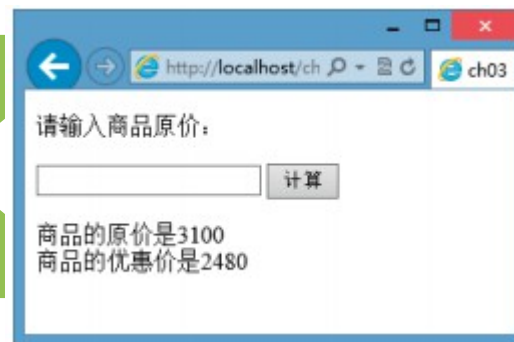
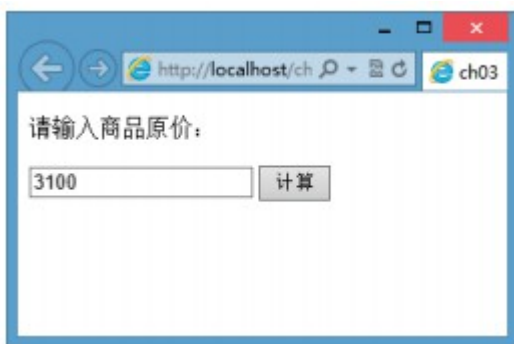
```
if( 条件表达式 1 ) {  
    语句块 1;}  
else if( 条件表达式 2){  
    语句块 2; }  
.....  
else{  
    语句块 n; }
```

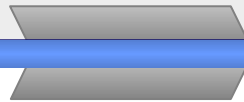




利用 **if...else** 语句求商品的优惠价

- ❖ 已知商品的原价，利用 **if...else** 语句求商品的优惠价，原价大于等于 1000 且小于 3000，9 折优惠，原价大于等于 3000，8 折优惠。





3.2.7 流程控制语句

❖ 1. if 条件结构

```
switch (表达式)
{
case 常量表达式 1: 语句块 1 ; break;
case 常量表达式 2: 语句块 2 ; break;
.....
case 常量表达式 n: 语句块 n ; break;
[default: 语句块 n+1;break;]
}
```





3.2.7 流程控制语句

❖ 2. 循环结构

```
while( 条件表达式) {  
    循环体语句;  
}
```

```
do{  
    循环体语句;  
}while( 条件表达式) ;
```

```
for( 初始值; 执行条件; 执行动作 ){  
    循环体语句;  
}
```





子任务二：商品订单页面设计

- 在此任务中需要完成一个简单的网上购物系统的商品订单程序，当用户输入相应商品数量后，点击“提交”，出现另一个页面，上面详细列明了该订单的明细，包括商品的总量、总价格和折扣等。

书名	数量
网页设计与制作	10
CSS网页设计标准教程	20
PHP开发与制作技术	30
请选择您在哪个区域	广州
提交 重置	

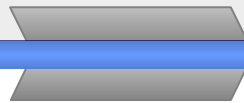
您在2018年3月12日在广州的订单如下：

您购买图书总量是60本，详细如下：
网页设计与制作10本，每本35元
CSS网页设计标准教程20本，每本40元
PHP开发与制作技术30本，每本45元
您的折扣为10%
您需要支付人民币2,250.00元





实施与测试



- ❖ **1、创建两个页面： buy.php 和 order.php 。**
前者用于用户输入订单数据，后者用于计算并显示用户提交的订单信息。
- ❖ **2、在 buy.php 页面，将静态页面创建完毕，并使表单跳转至 order.php 页面。**
- ❖ **3、在 order.php 页面，负责接收 buy.php 页面传递过来的数据，并且进行计算。**





3.2.8 数组

❖ **定义** 数组是一组数据的集合，这组数据的类型可以相同，也可以不同，数组将它们结合在一起形成一个可操作的整体。数组本身也是变量，其命名与变量命名规则一致。组成数组的元素称为数组元素，。每个数组元素对应一个编号，这个编号称为数组的键（**key**），每个键对应一个值（**value**）。**PHP** 中有两种数组，索引数组和关联数组。索引数组的键是整数，且从 **0** 开始标注。关联数组以字符串作为键。





1. 创建数组

❖ 格式:

```
array([key=>]value,...)
```

```
<?php
    $stuinfo1=array('2014022201', '张小欣','女','20');
                                                    //使用 array() 函数创建索引数组

    $stuinfo2[0]='2014022201';
                                                    //使用直接赋值方式创建数组
    $stuinfo2[1]='张小欣';
    $stuinfo2[2]='女';
    $stuinfo2[3]='20';
    $stuinfo3=array('stu_no'=>'2014022201','stu_name'=>'张小欣','stu_
sex'=>'女','stu_age'=>'20');
                                                    //使用 array() 函数创建关联数组

?>
```





2. 遍历数组

- ❖ (1) **for** 循环结构
- ❖ 只有当数组是索引数组且该数组的索引 (**key**) 是连续整数时, 方能使用 **for** 循环结构进行遍历。

【例 3-19】

```
<?php
    $stuiinfo1=array('2014022201','张小欣','女','20');
    for($i=0;$i<count($stuiinfo1);$i++)
        echo $stuiinfo1[$i]."<br>";
?>
```





2. 遍历数组

❖ foreach 循环结构仅能用于数组

```
foreach(array as [$key]=>$value)
```

【例 3-20】

```
<?php
    $stuinfo3=array('stu_no'=>'2014022201','stu_name'=>'张小欣','stu_
sex'=>'女','stu_age'=>'20');
    foreach($stuinfo3 as $key=>$value)
        echo $key.":".$value."<br>";
?>
```





3. 常见数组函数

❖ (1) 数组排序函数

```
sort( $array , $sort_flags)  
resort( $array, $sort_flags)
```

- ❖ 对关联数组进行排序时，可以使用 **asort()** 函数（升序排序）和 **arsort()** 函数（降序排序），以保持数组键名与元素值的对应关系。

```
asort( $array, $sort_flags)  
arsort( $array, $sort_flags)
```





3. 常见数组函数

- ❖ (1) 数组排序函数
- ❖ 如果希望按照数组的键名进行排序，而并非按照数组元素值来进行排序的话，可以使用 **ksort()** 函数和 **krsort()** 函数。

```
ksort( $array, $sort_flags)  
krsort( $array, $sort_flags)
```





3. 常见数组函数

❖ （2）数组查找函数

`array_search($needle, $array)`

【例 3-24】

```
<?php
    $abc=array('one'=>'apple','two'=>'orange','three'=>'pear');
    $find='orange';
    $index=array_search($find,$abc);
    echo $index;
    echo"<br>";
    echo $abc[$index];
?>
```





子任务三：显示商品信息

- ❖ 在此任务中需要完成一个简单的商品信息显示页面，将商品名称、商品价格、商品数量显示在表格中，具体如图所示。



The image shows a web browser window with a single tab titled 'info'. The browser's address bar is empty. The main content area displays a table with three columns: '商品名称' (Product Name), '商品价格' (Product Price), and '商品数量' (Product Quantity). The table contains three rows of data:

商品名称	商品价格	商品数量
ACA面包机	500元	2
格拉斯耐热玻璃保鲜盒	50元	10
堂彩随心杯子	10元	20





3.3.2 PHP 表单处理

1. HTML 表单组成

(1) 表单

```
<form name="form1" action="index.php" method="get"  
>...</form>
```

(2) 表单元素

```
<input type="text" name="..." maxlength="..." value="...">
```





3.3.2 PHP 表单处理

2. 表单传值

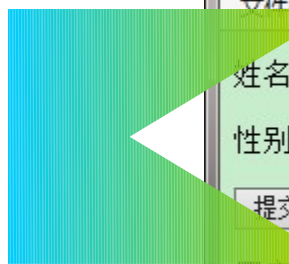
页面中表单数据传送方式有两种，一种是 **get**，另一种是 **post**，同样页面中接收表单数据的方式也有两种，一种是 **\$_GET**，另一种是 **\$_POST**，它们属于 **PHP** 中的全局变量，在 **PHP** 中任何地方均可以调用这些变量。





子任务四：获取个人信息

- ❖ 在发送页面中输入姓名与性别后，单击“提交”按钮，然后在接收页面中显示出来。具体如图所示。



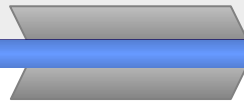


子任务四：商品计算功能实现

❖ 3.3.1 函数

- ❖ **定义** 函数是一段完成指定任务的已命名代码，函数可以遵照给它的一组值或参数完成任务。**PHP** 中的函数有两种，一种是标准的程序内置函数，该类函数在 **PHP** 中已经预定义过，有数百种，用户可以不定义而直接使用。另一种是用户自定义函数，完全由用户根据实际需要而定义。





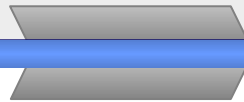
1. 常见内置函数

(1) **die()** 和 **exit()** 函数

(2) **empty()** 函数

(3) 格式化 **number_format()** 函数

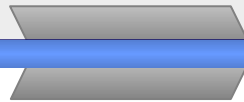




2. 字符串函数

- (1) 统计字符串长度 **strlen()** 函数
- (2) 截取字符串 **substr()** 函数
- (3) 字符串分割 **explode()** 函数
- (4) 字符串合并 **implode()** 函数





3. 自定义函数

❖ (1) 函数定义

```
function 函数名 ([ 参数 1, 参数 2, 参数  
3.....])  
{  
    函数体 ;  
    return 函数返回值 ;  
}
```





3. 自定义函数

❖ （2）函数调用

函数名（实参列表）；

【例 3-31】自定义函数，实现从 0 到 n 的累加和；使用该函数，计算 0 到 10 的累加和。代码如下：

```
<?php
    function jiafa($n)                //定义函数
    {
        $sum=0;
        for($i=0;$i<=$n;$i++)
            $sum+=$i;
        echo $sum;
    }
    jiafa(10);                        //调用函数，运行结果为 55
?>
```





3. 自定义函数

(3) 参数传递

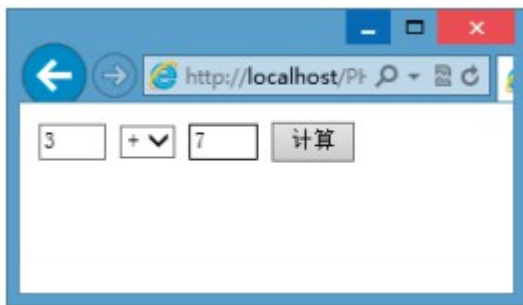
函数调用过程中，需要向函数传递参数，被传入的参数称为实参（如例 3-31 中的 10 ），而函数定义的参数称为形参（如上例中的 \$n ），参数传递的方式主要有值传递和引用传递。





子任务四：商品计算功能实现

- ❖ 利用表单与函数功能制作简单计算器。





实施与测试

- ❖ 1、新建页面，在此页面中利用表单与函数功能制作程序。
- ❖ 2、静态代码，在创建的页面中，创建静态页面效果。
- ❖ 3、动态代码，在页面中插入 **PHP** 代码。

