2-5函数

1. 课堂案例

课堂讲解

一、上节回顾

流程控制

if

switch

while

do...while

for

foreach

运算符

二、学习目标

了解什么是函数

了解常用的函数

三、教学过程描述

函数

1、自定义函数

定义一个功能，在全局调用实现相同效果！提高代码的重用性！

```
function 函数名(){
 函数体;
 [return;]
```

}

声明了函数，要想执行，必须调用。

函数的声明可以在调用前，也可以在调用之后，这与函数在内存中的存储有关。可执行代码段中。

【函数的命名规范】

1. 不能用中文，使用英文

2. 可以使用数组，但是不能以数字开头

3. 不可以使用特殊字符 _除外 ,并且可以在任何地方使用 4. 命名要有意义

5. 大小写不敏感 不区分大小写 6. 函数命名绝对不可以冲重复，重载

2、return

return 有返回值 可以被变量接收。

如果函数没有retutrn 就表示他没有返回值，不能被变量接收。

return 本身有终止函数执行的作用，函数遇到return就不在继续往下执行了,跳出函数!

return 如果需要使用多个 一般是 分直结构来使用

有return 待返回值的函数。 没有return的函数 执行过程函数。

3、参数

1、有参数就必须传参（没有默认值）

2、有默认值的时候，调用时候可以不穿参数，如果传了实参 按照实参的值进行计算。

3、如果存在多个参数，调用函数的实参和函数的形参之间必须一一对应。

4、如果想设置一个参数必须填写，设置形参的时候不给默认值就OK啦。

5、形参既有有默认值的时候也有没有默认值的时候。没有默认值的放在前面，这是逻辑问题而非代码问题。

function 函数名(参数名[=默认值],参数名[=默认值],参数名[=默认值,。。。。])｛

函数体;

[return]

}

4、函数的变量作用域

4-1、全局变量

全局变量就是在函数外面定义的变量。不能在函数中直接使用。因为它的作用域不会到函数内部。

所以在函数内部使用的时候常常看到类似global $a;

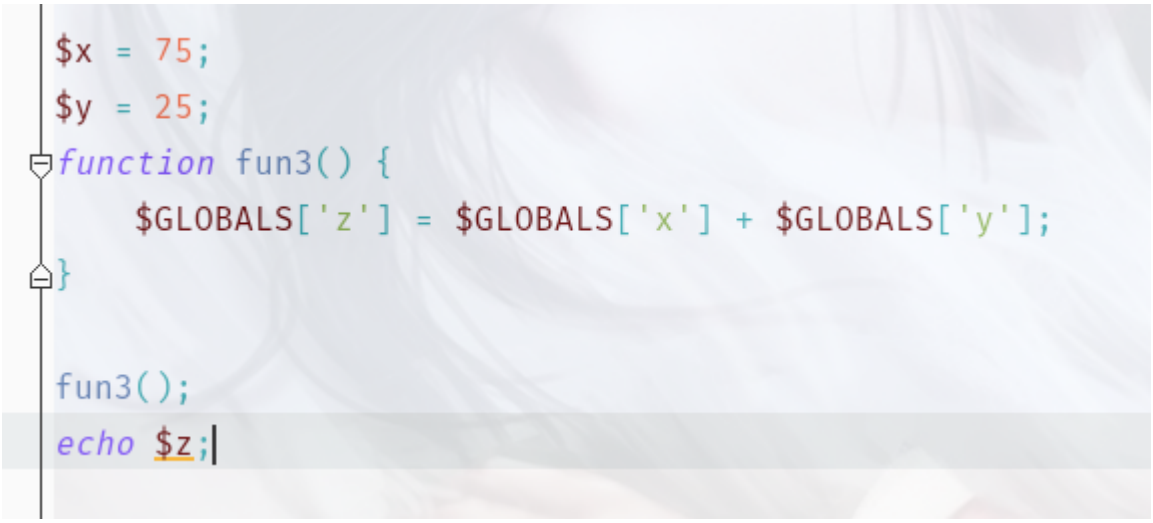
4-1-1、global

global 声明函数内外相通的变量名为一个变量(以外部变量为准。)

4-1-2、超全局变量

它们在一个脚本的全部作用域中都可用。在函数或方法中无需执行 global $variable; 就可以访问它们。

1、$GLOBALS数组的方式



```php
$x = 75;
$y = 25;
function fun3() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

fun3();
echo $z;
```

2、$_GET

PHP $_GET 也可用于收集提交 HTML 表单 (method="get") 之后的表单数据。

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>

<?php
$name = $_GET['fname'];
echo $name;
?>

</body>
</html>
```

3、$_POST

PHP $_POST 广泛用于收集提交 method="post" 的 HTML 表单后的表单数据。$_POST 也常用于传递变量。

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>

<?php
$name = $_POST['fname'];
echo $name;
?>

</body>
</html>
```

4、$_FILES

通过 HTTP POST 方式上传到当前脚本的项目的数组。

5、$_SERVER

$_SERVER['HTTP_ACCEPT_LANGUAGE']//浏览器语言

$_SERVER['REMOTE_ADDR'] //当前用户 IP 。

$_SERVER['REMOTE_HOST'] //当前用户主机名

$_SERVER['REQUEST_URI'] //URL

$_SERVER['REMOTE_PORT'] //端口。

$_SERVER['SERVER_NAME'] //服务器主机的名称。

$_SERVER['PHP_SELF']//正在执行脚本的文件名

$_SERVER['argv'] //传递给该脚本的参数。

$_SERVER['argc'] //传递给程序的命令行参数的个数。

$_SERVER['GATEWAY_INTERFACE']//CGI 规范的版本。

$_SERVER['SERVER_SOFTWARE'] //服务器标识的字串

$_SERVER['SERVER_PROTOCOL'] //请求页面时通信协议的名称和版本

$_SERVER['REQUEST_METHOD']//访问页面时的请求方法

$_SERVER['QUERY_STRING'] //查询(query)的字符串。

$_SERVER['DOCUMENT_ROOT'] //当前运行脚本所在的文档根目录

$_SERVER['HTTP_ACCEPT'] //当前请求的 Accept: 头部的内容。

$_SERVER['HTTP_ACCEPT_CHARSET'] //当前请求的 Accept-Charset: 头部的内容。

$_SERVER['HTTP_ACCEPT_ENCODING'] //当前请求的 Accept-Encoding: 头部的内容

$_SERVER['HTTP_CONNECTION'] //当前请求的 Connection: 头部的内容。例如:“Keep-Alive”。

$_SERVER['HTTP_HOST'] //当前请求的 Host: 头部的内容。

$_SERVER['HTTP_REFERER'] //链接到当前页面的前一页面的 URL 地址。

$_SERVER['HTTP_USER_AGENT'] //当前请求的 User_Agent: 头部的内容。

$_SERVER['HTTPS']//如果通过https访问,则被设为一个非空的值(on), 否则返回off

$_SERVER['SCRIPT_FILENAME'] #当前执行脚本的绝对路径名。

$_SERVER['SERVER_ADMIN'] #管理员信息

$_SERVER['SERVER_PORT'] #服务器所使用的端口

$_SERVER['SERVER_SIGNATURE'] #包含服务器版本和虚拟主机名的字符串。

$_SERVER['PATH_TRANSLATED'] #当前脚本所在文件系统(不是文档根目录)的基本路径。

$_SERVER['SCRIPT_NAME'] #包含当前脚本的路径。这在页面需要指向自己时非常有用。

$_SERVER['PHP_AUTH_USER'] #当 PHP 运行在 Apache 模块方式下, 并且正在使用 HTTP 认证功能, 这个变量便是用户输入的用户名。

$_SERVER['PHP_AUTH_PW'] #当 PHP 运行在 Apache 模块方式下, 并且正在使用 HTTP 认证功能, 这个变量便是用户输入的密码。

$_SERVER['AUTH_TYPE'] #当 PHP 运行在 Apache 模块方式下, 并且正在使用 HTTP 认证功能, 这个变量便是认证的类型

6、$_COOKIE

通过 HTTP Cookies 方式传递给当前脚本的变量的数组。

后面学习再详细介绍

7、$_SESSION

当前脚本可用 SESSION 变量的数组。

后面学习再详细介绍

8、$_REQUEST

默认情况下包含了 $_GET, $_POST 和 $_COOKIE 的数组。

用法:$_REQUEST['参数名']

9、$_ENV

$_ENV 是一个包含服务器端环境变量的数组。它是 PHP 中一个超级全局变量，我们可以在 PHP 程序的任何地方直接访问它。

$_ENV 为空的原因及解决办法：

如果打印输出 $_ENV 为空，可以检查一下 php.ini 的配置：variables_order = "EGPCS"。

上述配置表示 PHP 接受的外部变量来源及顺序，EGPCS 是 Environment、Get、Post、Cookies 和 Server 的缩写。如果 variables_order 的配置中缺少 E ，则 PHP 无法接受环境变量，那么 $_ENV 也就为空了。

4-1-3、引用传参

引用传参 和引用变量类似

```php
function fun4(&$a){
    $a=$a+100;
}
$b=1;
fun4($b);
echo $b;
echo '<br>';
//要注意的是，在这里test(1);的话就会出错，原因是：
function &fun5(){
    static $b=0;//申明一个静态变量
    $b=$b+1;
    echo $b;
    return $b;
}
$a=fun5();//这条语句会输出  $b的值  为1
echo '<br>';
$a=5;
$a=fun5();//这条语句会输出  $b的值  为2
echo '<br>';
$a=&fun5();//这条语句会输出  $b的值  为3
echo '<br>';
$a=5;
$a=fun5();//这条语句会输出  $b的值  为6
```

4-2、局部变量

任何用于函数内部的变量将被限制在局部函数范围内。

```php
function test()
{
    $a = 1;
    echo $a;
}
test();
//或者
$a = 1;
    echo $a;
```

4-3、静态变量

在函数退出时，这个变量始终存在，不被销毁，但不能被其它函数使用，当再次进入该函数时，将保存上次的结果。

```php
function test()
{
    static $test;
    $test .= '哈';
    echo $test.'<br>';
}
test();
test();
test();
```

5、可变函数

这意味着如果一个变量名后有圆括号，PHP
将寻找与变量的值同名的函数，并且尝试执行它。可变函数可以用来实现包括回调函数，函数表在内的一些用途

6、匿名函数

7、常用的一些函数

7-1、is系列

is_array—检测变量是否是数组

is_int — 检测变量是否是整数

is_float — 检测变量是否是浮点型

is_bool — 检测变量是否是布尔

7-2、in_array

in_array — 检查数组中是否存在某个值

7-3、strlen

用于计算字符串的长度

四、小结

怎么自定义函数

函数的作用域

2. 课堂练习

练习

自己定义下函数

练习函数的变量作用域

3. 课后练习

1、写出加减乘除的函数封装

2、封装一个在全局变量函数

3、查找手册:重复一个字符串次数的函数是什么

4. 资料扩展

1、匿名函数

匿名函数(Anonymous functions)，也叫闭包函数(closures)，允许 临时创建一个没有指定名称的函数。

最经常用作回调函数(callback)参数的值。

当然，也有其它应用的情况。

匿名函数目前是通过 Closure 类来实现的。