

Day6 学习报告

1. 概述：今天我们主要学习的内容有：

- (1) 函数基础
- (2) 可变函数及其应用
- (3) 匿名函数
- (4) 常用系统函数及递归思想

以下分别进行具体介绍

2. 函数基础

(1) 函数的定义

形式为：

```
function 函数名(【$行参1】 【, $行参2】 【, ...】){  
    函数体  
};
```

(2) 函数的调用

形式为:函数名(\$实参1, \$实参2,...);

(3) 函数调用的详细过程

略

(4) 函数的参数问题

- 1) 函数形参默认值问题，默认值必需放右边，左边都是必传的形参。
- 2) 函数形参的传值问题，包括值传递和引用传递
- 3) 函数参数的数量问题，可以0-n个，利用function相关系统函数可以获得具体的所有形参

(5) 函数的返回值问题

非必需、return即终止函数进行

3. 可变函数及其应用

- (1) 类比可变变量，函数的名是一个变量，调用的时候\$变量名(函数参数)
- (2) 可变函数的灵活性

当有个一个操作可能执行特定n个函数中的某个函数时，可将那些函数名设置为变量，采用可变函数的方式来只用一行代码进行任意调用。当然，前提要求是变量值要和对应那个函数名一致！

```
test6.php
1 <?php
2 function jpg(){
3     echo "<br>". "进行jpg图像处理";
4 }
5
6 function png(){
7     echo "<br>". "进行png图像处理";
8 }
9
10
11 $n = "abc.1.jpg";
12 $nn = strchr($n, ".");
13 $nnn = strrchr($nn, ".");
14 $nnnn = substr($nnn, 1);
15 echo $nn."<br>".$nnn."<br>".$nnnn;
16
17 $nnnn();
18 ?>
```

localhost:8888/test6.php

.1.jpg
.jpg
jpg
进行jpg图像处理

4. 匿名函数

没有函数名的函数，可以作为实参，由此来达到访问某个函数里的局部变量的目的。

```
//匿名函数
//形式一
$s = function(){
    echo "string."<br>;
};

$s();

//形式二
/*
 * 异步请求定义
 *
 * $m -> 请求参数1
 * $n -> 请求参数2
 * $completion -> 数据回调
 */
function requestWithCompletion($m, $n, $completion){
    $mn = $m * $n; //mn是12
    $mn = $completion($mn); //mn在回调后被外界修改了，为24
    echo "mn = m * n = $m * $n = " . ($m * $n);
    echo "<br>由于先回调，mn被修改了值，所以mn不是". ($m*$n). "而是$mn";
}

requestWithCompletion(3, 4, function($mn){
    return $mn * 2;
});
```

.1.jpg
.jpg
jpg
进行jpg图像处理

string
mn = m * n = 3 * 4 = 12
由于先回调，mn被修改了值，所以mn不是12而是24

5. 变量作用域

(1) 局部作用域：

指一个函数的内部范围

对应这样的变量，就称为“局部变量”

(2) 超全局作用域：

指所有的代码范围 对应这样的变量，就称为“超全局变量” 其实就只有几个：\$_GET, \$_POST, \$_SERVER, \$_REQUEST, \$GLOBALS

\$_COOKIES, \$_FILES, \$_SESSION

(3) 全局作用域：

不在函数内部的范围——函数外部 对应这样的变量，就称为“全局变量”

(4) 局部作用域使用全局作用域变量方法

可以使用global修饰和全局作用域变量名一样的局部变量，这时候局部变量指向和全局变量同一块内存区，则可以进行修改
也可以使用超全局变量\$_GLOBALS[“变量名”]来访问

6. 常用系统函数及递归思想

(1) 常用系统函数：

主要列举了：

函数相关的函数：略

字符串处理相关的函数：略

时间相关函数：略

数学相关函数：略

(2) 有关函数的编程思想

1) 递归思想——递归函数

2) 递推思想