基于PHP制作验证码

# 开发环境

1、环境搭建：Windows 7+Apache 2.4.18+MySQL 5.7.11+PHP 7.1.0。

2、文本编辑器：Sublime 3。

# 二、主要技术

本案例主要基于PHP中的GD库来实现验证码的制作。

GD（graphic device）库是PHP处理图形的扩展库，GD库提供了一系列用来处理图片的api，使用GD库可以处理图片，或者生成图片。 在网站上GD库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表 。

# 三、效果图展示

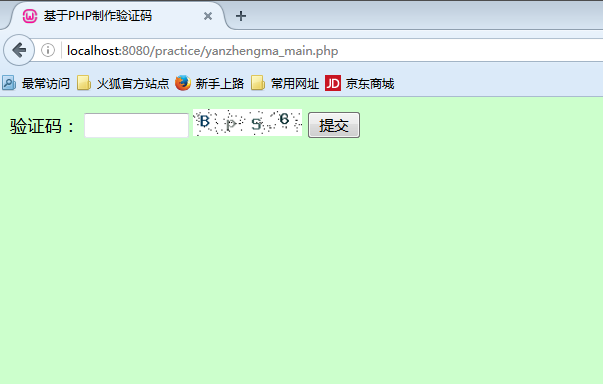
实现效果图，如图1所示。

图1 效果图

# 四、具体步骤

## 4.1、验证码的制作

### （1）制作验证码图片

①创建一个名为“yanzhengma.php”的PHP文件用来制作验证码图片。

②开启session，方便传值来验证用户的输入的内容是否正确。

$lifeTime = 600;

session\_set\_cookie\_params($lifeTime); // 设置session过期时间

session\_start(); //启动session

③输出图片格式

header('Content-type:image/png'); //输出头信息，图片格式为png

④创建图片画布

$image\_w=100;

//验证码图片的宽

$image\_h=25;

//验证码图片的高

//创建图片对象

$check\_image=imagecreatetruecolor($image\_w, $image\_h);

//定义白色

$white=imagecolorallocate($check\_image,255, 255, 255);

//填充图片背景颜色为白色

imagefill($check\_image,0,0,$white);

### （2）生成验证字符串

//验证字符由数字和大小写字母构成

//使用range()函数来创建并返回一个包含指定范围的元素的数组

//定义一个纯数字的数组

$number=range(0,9);

//定义一个成员为大写字母的数组

$character=range("Z","A");

//定义一个成员为小写字母的数组

$bigcharacter=range("z","a");

//合并数组

$result=array\_merge($number,$character, $bigcharacter);

//记录验证码字符串并初始化

$string="";

//数组的长度

$len=count($result );

//随机生成验证码字符串（验证码字符串由4个字符组成）

for($i=0;$i<4;$i++){

//从$result中随机取出一个元素

$new\_number[$i]=$result[rand(0,$len-1)];

//将数据进行拼接起来

$string=$string.$new\_number[$i];

}

//使用$\_SESSION变量传值

$\_SESSION['string']=$string;

### （3）生成干扰元素

//定义黑色

$black=imagecolorallocate($check\_image,0,0,0);

//在验证码图片上加入100个干扰的黑点

for($i=0;$i<100;$i++){

imagesetpixel($check\_image,rand(0,$image\_w),rand(0,

$image\_h，$black);

}

### （4）循环输出验证码

for($i=0;$i<count($new\_number);$i++){

//设置字符所在x坐标（mt\_rand函数和rang()函数类似，都是

产生随机数，只不过mt\_rand()产生随机数的平均速度比rand()

快四倍）

$x=mt\_rand(1,8)+$image\_w\*$i/4;

//设置字符所在y坐标

$y=mt\_rand(1,$image\_h/4);

//随机设定字符颜色

$color=imagecolorallocate($check\_image, mt\_rand(0,200),mt\_rand(0,200), mt\_rand(0,200));

//输入字符到图片中

imagestring($check\_image,5,$x,$y,$new\_number[$i],$color);

}

### （5）输出图片,释放资源

imagepng($check\_image); //输出图片

imagedestroy($check\_image); //清除资源

### （6）测试图片是否生成成功

开启Apache服务，打开浏览器，在地址栏中输入文件存储地址（根据自己文件存放位置），例如“localhost:8080/practice/yanzhengma.php”，可以看到程序运行结果如图2所示。

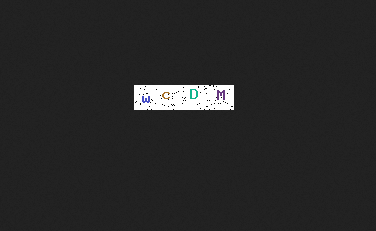


图2 生成验证码图片

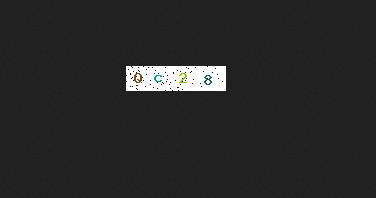
刷新页面，将生成一个新的验证码图片，则说明验证码重新刷新生成，验证码图片制作成功，如图3所示。

图3 刷新验证码

## 4.2、进行用户输入验证

### （1）创建PHP验证文件

创建一个名为“yanzhengma\_main.php”的PHP文件，用来验证用户输入的验证码是否正确。

### （2）页面布局

<!DOCTYPE html>

<html>

<head>

<title>基于PHP制作验证码</title>

</head>

<body bgcolor="#CCFFCC">

<form action="" method="post">

<table><tr >

<td>验证码：</td>

<td><input type="text" size="10" name="check"></td>

<td><a href="yanzhengma\_main.php">

<image src="yanzhengma.php"></a></td>

<td><input type="submit" name="ok"

value="提交 "></td></tr>

</table>

</form>

</body>

</html>

### （3）进行用户输入验证

<?php

//开启session

session\_start();

//判断是否提交表单

if(isset($\_POST['ok'])){

//获取验证码字符串

$checkstr=$\_SESSION['string'];

//获取文本框中用户输入的字符

$str=$\_POST['check'];

//不区分大小写进行比较验证码是否输入正确

if(strcasecmp($str, $checkstr)==0){

echo "<script>alert('验证码输入正确！');</script>";

}

else{

echo "<script>alert('验证码输入错误！');</script>";

}

}

?>

# 五、网页测试

①开启Apache服务后打开浏览器，在浏览器地址栏中输入文件地址，例如：“localhost:8080/pratice/yanzhengma\_main.php”，可以看到程序运行结果如图4所示。

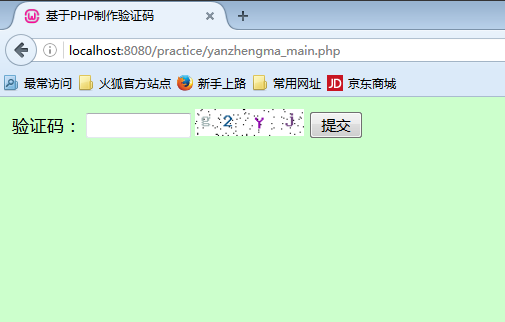


图4 案例结果图（用户输入验证页面）

②在文本框中输入“g2yj”，点击【提交】进行测试，可以看到验证码输入正确，如图5所示。

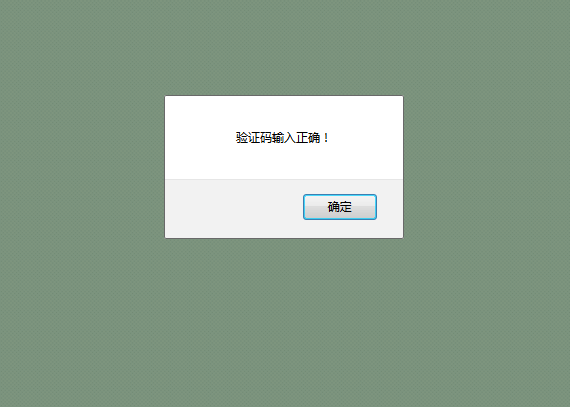


图5案例结果图（验证码输入正确信息提示）

③进行多次测试，如果每次验证码都能够更新并且浏览器能够根据用户的输入情况作出正确的判断，则说明测试成功！