

主讲教师：传智.神龙教主

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 回顾

1. 通过用户名和密码去获取记录，如果获取到记录就登陆成功，如果获取不到就登陆失败

1、方法一：

select \* from 表 where 用户名='' and 密码=''

2、方法二：

select 密码 from 表 where 用户名=''

将获取到的密码和输入的密码匹配，匹配成功就登陆成功，否则登陆失败

1. 防止注入：通过拼接SQL语句达到不用用户名和密码就登陆的漏洞，解决方法：给特殊字符转义。
2. 防止翻墙：登陆成功后，给用户一个登陆凭证（会话），在每一次操作之前都判断一下是否有登陆凭证，如果没有就跳转到登陆页面。
3. 记住用户名和密码（永久性cookie）,将用户的id和md5加密的用户名和密码的组合字符串保存到永久性cookie中。
4. $\_SERVER[‘HTTP\_REFERER’]：获取从哪个页面跳转到本页面上。
5. 安全退出：退出的时候销毁登陆凭证。

## 验证码

用来防止木马病毒自动登陆。验证码的值每次都变化，这样就能有效的防止病毒自动登陆。



脚下留心：要使用验证码必须要打开gb2库扩展



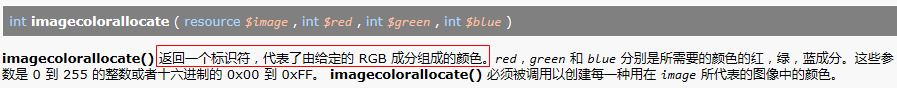
## 创建最简单的图片

### 知识点

1. imagecreate(宽度，高度)，返回一个空白图像



1. imagecolorallocate(),为一幅图像分配颜色



1. imagejpeg()：将图像按jpeg格式显示
2. imagegif()：将图像按gif格式显示
3. imagepng()：将图像按png格式显示
4. 告知浏览器通过什么格式解析代码，书写格式： “类型/格式”



1. imagedestroy()



1. 保存图片



脚下留心：如果要保存图片不能使用header()头。

代码

<?php

$img=imagecreate(200,100); //创建了200\*100的图片

//var\_dump($img); //resource(2) of type (gd)

imagecolorallocate($img,255,0,0); //给图片分配一个颜色

//header('content-type:image/jpeg'); //告诉浏览器通过jpeg的格式去解析此图片

//imagejpeg($img); //图片按jpeg格式显示

imagepng($img,'./aa.png'); //将图片保存为aa.png

imagedestroy($img); //销毁图片资源

## imagefill()：给图片填充颜色



代码

<?php

$img=imagecreate(200,100);

$color1=imagecolorallocate($img,255,0,0); //给图片分配红色

$color2=imagecolorallocate($img,0,255,0); //给图片分配绿色

$color3=imagecolorallocate($img,0,0,255); //给图片分配蓝色

$num=rand(1,10);

switch($num%3) {

case 0:

imagefill($img,0,0,$color1); //给图片填充颜色

break;

case 1:

imagefill($img,0,0,$color2);

break;

case 2:

imagefill($img,0,0,$color3);

break;

}

header('content-type:image/png');

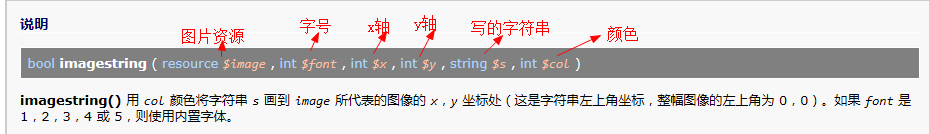
imagepng($img);

imagedestroy($img);

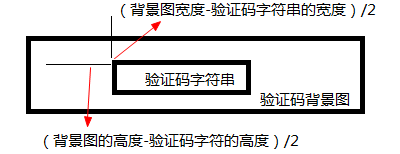
## 自定义验证码

### 知识点

1. imagestring()：将字符串写到图片上



1. imagesx()：测定图片的宽度
2. imagesy()：测定图片的高度
3. imagefontwidth — 取得字体宽度
4. imagefontheight — 取得字体高度
5. 将验证码字符串居中



### 完整代码

<?php

$char\_array=array\_merge(range('A','Z'),range('a','z'),range(0,9));//生成一个字母和数字的数组

$index=array\_rand($char\_array,4); //随机取4个字符，返回的是字符下标

shuffle($index); //打乱下标

//拼接字符串

$str='';

foreach($index as $i) {

$str.=$char\_array[$i];

}

//创建图片

$img=imagecreate(145,20);

imagecolorallocate($img,0xFF,0x00,0x00); //背景色

$color=imagecolorallocate($img,0xFF,0xFF,0xFF); //再次给图片分配一个颜色，用来做前景色

$font=5; //字号

$x=(imagesx($img)-imagefontwidth($font)\*strlen($str))/2; //文字的起始位置的x轴

$y=(imagesy($img)-imagefontheight($font))/2; //文字的起始位置的y轴

imagestring($img,$font,$x,$y,$str,$color); //将字符串写到图片上

header('content-type:image/png');

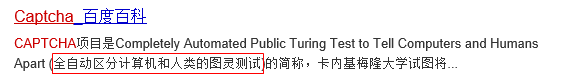
imagepng($img);

imagedestroy($img);

## 模拟ecshop验证码

### 知识点

1. captcha



1. imagecreatefromgif()：打开gif图
2. imagecreatefromjpeg()：打开jpg图
3. imagecreatefrompng()：打开png图

### 完整代码

<?php

$char\_array=array\_merge(range('A','Z'),range('a','z'),range(0,9));//生成一个字母和数字的数组

$index=array\_rand($char\_array,4); //随机取4个字符，返回的是字符下标

shuffle($index); //打乱下标

//拼接字符串

$str='';

foreach($index as $i) {

$str.=$char\_array[$i];

}

//打开背景图

$bg\_path='./captcha/captcha\_bg'.rand(1,5).'.jpg'; //背景图地址

$img=imagecreatefromjpeg($bg\_path); //打开图片

//定义前景色

$color=imagecolorallocate($img,0,0,0);

if(rand(1,2)==2)

$color=imagecolorallocate($img,255,255,255);

//将字符串写到图片上

$font=5;

$x=(imagesx($img)-imagefontwidth($font)\*strlen($str))/2;

$y=(imagesy($img)-imagefontheight($font))/2;

imagestring($img,$font,$x,$y,$str,$color);

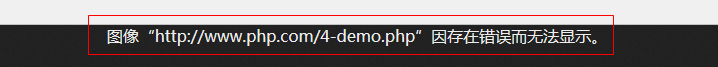
header('content-type:image/png');

imagepng($img);

imagedestroy($img);



## 验证码改错



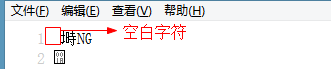
验证码一旦出错，修改起来比较麻烦，按如下方式去做：

第一步：注释掉header()头



第二步：再次运行，就会提示错误，按照错误信息更改后再去掉header注释。

第三步：如果注释掉header头后没有提示错误，很可能是图片二进制代码前面有其他字符，通过查看源文件的方式看一下图片代码前有没有其他字符（包括空白字符），如果有删除无用字符即可。



脚下留心：在图片输出函数（imagepng、imagegif、imagejpeg）之前的所有字符对图片都有影响。

## 封装ecshop验证码

<?php

/\*\*

\*验证码封装类

\*/

class CaptchaLib {

private $length; //验证码长度

private $font=5; //内置字体大小 1,2,3,4,5

//通过构造函数赋值

public function \_\_construct($length=4,$font=5) {

$this->length=$length;

$this->font=$font;

}

//生成随机字符串

private function generalCode() {

$char\_array=array\_merge(range('A','Z'),range('a','z'),range(0,9));//生成一个字母和数字的数组

$index=array\_rand($char\_array,$this->length); //随机取4个字符，返回的是字符下标

shuffle($index); //打乱下标

//拼接字符串

$str='';

foreach($index as $i) {

$str.=$char\_array[$i];

}

return $str;

}

//生成验证码

public function generalCaptcha() {

$str=$this->generalCode();

//打开背景图

$bg\_path='./captcha/captcha\_bg'.rand(1,5).'.jpg'; //背景图地址

$img=imagecreatefromjpeg($bg\_path); //打开图片

//定义前景色

$color=imagecolorallocate($img,0,0,0);

if(rand(1,2)==2)

$color=imagecolorallocate($img,255,255,255);

//将字符串写到图片上

$x=(imagesx($img)-imagefontwidth($this->font)\*strlen($str))/2;

$y=(imagesy($img)-imagefontheight($this->font))/2;

imagestring($img,$this->font,$x,$y,$str,$color);

header('content-type:image/png');

imagepng($img);

imagedestroy($img);

}

}

//测试

$captcha=new CaptchaLib(8,4);

$captcha->generalCaptcha();

## 思考：如何制作中文验证码？

第一步：定义一个数组，此数组中存放若干个中文字符

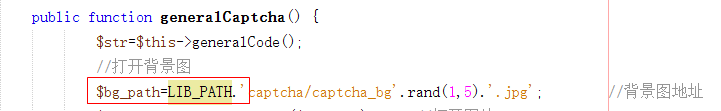
第二步：生成随机中文字符串

第三步：通过ImageTTFText()将中文写到图片上



## 将验证码封装到项目中

1. 将CaptchaLib.class.php和captcha文件夹拷贝到Lib文件夹下
2. 更改CaptchaLib中的代码

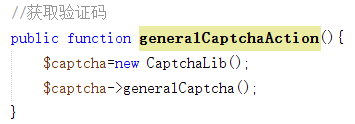


### 显示验证码

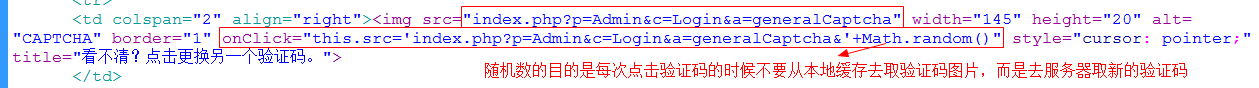
模型

无

控制器

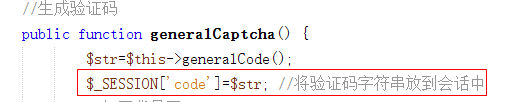


视图

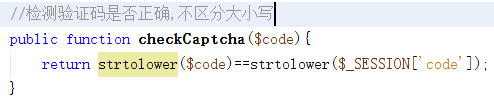


### 检验验证码是否正确

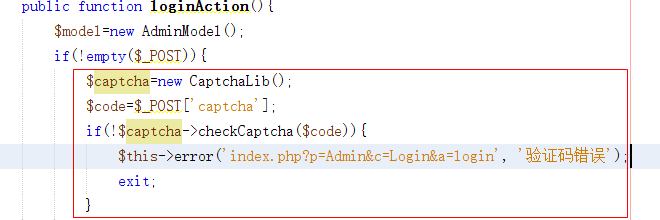
更改CaptchaLib类，将验证码字符串写到会话中



在CaptchaLib类中，添加一个验证验证码是否正确的方法



在用户登录的时候，判断验证码



多学一招：如果验证码不变化，有可能是没有清空输出缓存，手动清空一下即可



多学一招：输入的字符串都显示为大写

