

## PHP的几个常用加密函数

在网站的开发过程中，常常需要对部分数据（如用户密码）进行加密，本文主要介绍PHP的几个常见的加密函数

### MD5加密：

```
string md5 ( string $str [, bool $raw_output = false ] )
```

1.md5()默认情况下以 32 字符十六进制数字形式返回散列值，它接受两个参数，第一个为要加密的字符串，第二个为raw\_output的布尔值，默认为false，如果设置为true，md5()则会返回原始的 16 位二进制格式报文摘要

2.md5()为单向加密，没有逆向解密算法，但是还是可以对一些常见的字符串通过收集，枚举，碰撞等方法破解

```
<?php
$username='jellybool';
$password='jellybool.com';
/*简单地对字符串进行md5加密*/
echo md5($username);
echo "<hr>";
echo md5($password);
echo "<hr>";
/*更推荐的做法是对重要的敏感数据进行多次加密，以防被轻易破解*/
echo md5(md5($password));

/*以上输出：
    username: 4f5436e5d72608fb647b691e8edcf42e
    password: 7bf02cf0f4af6da4accbc73d2a175476
    password(两次加密): 864704bb35754f8cd0232cba6b91521b
*/
```

## Crypt加密：

string crypt ( string \$str [, string \$salt ] )

1.crypt()接受两个参数，第一个为需要加密的字符串，第二个为盐值（就是加密干扰值，如果没有提供，则默认由PHP自动生成）；返回散列后的字符串或一个少于 13 字符的字符串，后者为了区别盐值。

2.crypt()为单向加密，跟md5一样。

```
<?php
$password='jellybool.com';
echo crypt($password);
```

```
//输出:$1$Fe0.qr5.$W0hkI4/5VPo7n7TnXHh5K
/*第二个$与第三个$之间的八个字符是由PHP生成的，每刷新一次就变一次
*/
echo "<hr>";
echo crypt($password,"jellybool");
//输出: je7fNiu1KNaEs
/*当我们要加自定义的盐值时，如例子中的jellybool作为第二个参数直接加入，
超出两位字符的会截取前两位*/
echo "<hr>";
echo crypt($password,'$1$jellybool$');
//输出: $1$jellyboo$DxH7wF7SygRpWb6XBBgfH/
/* crypt加密函数有多种盐值加密支持，以上例子展示的是MD5散列作为盐值，该方式下
盐值以$1$$的形式加入，如例子中的jellybool加在后两个$符之间，
超出八位字符的会截取前八位，总长为12位;crypt默认就是这种形式。
*/
echo "<hr>";
//crypt还有多种盐值加密支持,详见手册
```

## Sha1加密:

```
string sha1 ( string $str [, bool $raw_output = false ]
```

1.跟md5很像，不同的是sha1()默认情况下返回40个字符的散列值，传入参数性质一样，第一个为加密的字符串，第二个为raw\_output的布尔值，默认为false，如果设置为true，sha1()则会返回原始的20位原始格式报文摘要

2.sha1()也是单行加密，没有逆向解密算法

```
<?php
```

```
$my_intro="jellybool";  
echo sha1($my_intro);  
//输出: c98885c04c1208fd4d0b1dadd3bd2a9ff4d042ca  
echo "<hr>";  
//当然, 可以将多种加密算法混合使用  
echo md5(sha1($my_intro));  
//输出: 94f25bf9214f88b1ef065a3f9b5d9874  
//这种方式的双重加密也可以提高数据的安全性
```

## Urlencode加密:

string urlencode ( string \$str )

1. 一个参数, 传入要加密的字符串 (通常应用于对URL的加密),
2. urlencode为双向加密, 可以用urldecode来解密 (严格意义上来说, 不算真正的加密)
3. 返回字符串, 此字符串中除了 \_ . 之外的所有非字母数字字符都将被替换成百分号 (%) 后跟两位十六进制数, 空格则编码为加号 (+) 。

```
<?php  
//urlencode() 通常用于URL中明文数据的隐藏  
$my_urlencode="jellybool.com?jellybool=true + 4-3%5= \& @!";  
echo urlencode($my_urlencode);  
//输出: jellybool.com%3Fjellybool%3Dtrue+%2B+4-3%25%3D+%5C%26+%40%21  
echo "<hr>";  
$my_urldecode="jellybool.com%3Fjellybool%3Dtrue+%2B+4-3%25%3D+%5C%26+%40%21";  
echo urldecode($my_urldecode);  
//输出: jellybool.com?jellybool=true + 4-3%5= \& @!
```

```
//还原了$my_urlencode的输出
echo "<hr>";
$my_urldecode="http://www.baidu.com/s?word=jellybool+%E8%A7%89%E7%B4%AF%E4%B8%8D%E7%88%B1&tn=9i";
echo urldecode($my_urldecode);
/*输出: http://www.baidu.com/s?word=jellybool 觉累不爱&tn=98236947_hao_pg&ie=utf-8
没错, 这就是在百度搜索jellybool 觉累不爱
*/

/*
=====
解决第二个经典问题
=====
*/
$pre_url_encode="jellybool.com?username=jellybool&password=jelly";
//在实际开发中, 我们很多时候要构造这种URL, 这是没有问题的
$url_decode    ="jellybool.com?username=jelly&bool&password=jelly";
/*注意上面两个变量的差别: 第一个的username=jellybool,
                           第二个为username=jelly&bool
这种情况下用$_GET()来接受是会出问题的, 这是可以用下面的方法解决
*/
$username="jelly&bool";
$url_decode    ="jellybool.com?username=".urlencode($username)."&password=jelly";
//这是可以很好的解决问题

/*
总结一下常见的urlencode()的转换字符
? => %3F
= => %3D
% => %25
& => %26
\ => %5C
+ => %2B
空格 => +
*/
```

## base64编码加密：

string base64\_decode ( string \$encoded\_data )

1.base64\_encode()接受一个参数，也就是要编码的数据（这里不说字符串，是因为很多时候base64用来编码图片）

2.base64\_encode()为双向加密，可用base64\_decode()来解密

```
<?php
$my_intro="JellyBool是一个身材有高度,肩膀有宽度,胸肌有厚度,思想有深度的
国家免检五A级优质伪前端IT男屌丝";

echo base64_encode($my_intro);
echo "<hr>";

/*输出: SmVsbHlCb29s5piv5LiA5Liq6Lqr5p2Q5pyJ6auY5bqmLOiCqeiGgOacieWuveW
6pizog7jogozmnInLjprLuqYs5oCd5o0z5pyJ5rex5bqm55qE5Zu95a625YWN5qOA5Lq
UQee6p+S8m0i0qOS8quWJjeerr0LU55S35bGM5Lid
*/

echo base64_decode('SmVsbHlCb29s5piv5LiA5Liq6Lqr5p2Q5pyJ6auY5bqmLOiCqeiGg
OacieWuveW6pizog7jogozmnInLjprLuqYs5oCd5o0z5pyJ5rex5bqm55qE5Zu95a6
25YWN5qOA5LqUQee6p+S8m0i0qOS8quWJjeerr0LU55S35bGM5Lid');

/*输出: JellyBool是一个身材有高度,肩膀有宽度,胸肌有厚度,思想有深度的国家免检五A
级优质伪前端IT男屌丝
*/
```

一个图片的例子：

```
<?php
```

```
/*  
一个图片的应用例子  
*/  
$filename="https://worktile.com/img/index/index_video.png";  
  
$data=file_get_contents($filename);  
echo base64_encode($data);  
/*然后你查看网页源码就会得到一大串base64的字符串，  
再用base64_decode()还原就可以得到图片  
*/
```

本文由 JellyBool 创作，采用 署名-非商业性使用 2.5 中国大陆 进行许可。  
转载、引用前需联系作者，并署名作者且注明文章出处。

[上一篇](#)[关于JustWriting源码的一些解读](#)[下一篇](#)[在Coding中部署Laravel项目演示](#)

分享到: 微博 QQ QQ空间 腾讯微博 微信 豆瓣 Facebook Twitter

0评论

微信 微博 QQ 人人 [更多»](#)



说点什么吧...



发表评论

Copyright © 2014 -- 2016 JellyBool,蜀ICP备15005290号-4