



PHP特性&缺陷对比函数&CTF考点

原创 大飞先生 于 2022-09-13 11:27:09 发布 阅读量2k 收藏 11 点赞数 2

版权

分类专栏: 基础的渗透测试 文章标签: php 开发语言



基础的渗透测试 专栏收录该内容

2 订阅 26 篇文章

订阅专栏

#详细点:

==与===

md5

intval

strpos

in_array

preg_match

str_replace

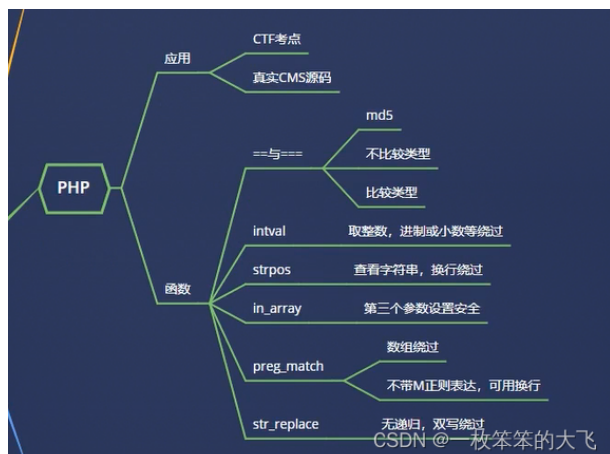
PHP对比规则

== 和 ===

=: 赋值

==: 对比, 但是不会对比数据类型

===: 对比, 同时对数据类型



1.== 只要赋值变量是1开头, 都可以正常访问

```

1 <?php
2 /** Created by PhpStorm */
3
4
5
6
7
8
9 header( string: "Content-Type: text/html; charset=utf-8");
10 $flag="xiaodi ai chi xigua!";
11 $a=1;
12 if ($a=$_GET[ 'x' ]) {
13     echo $flag;
14 }

```

127.0.0.1/ctf.php?x=1asdasdasd

xiaodi ai chi xigua!

CSDN @一枚笨笨的大飞

2.===



大飞先生

关注

2

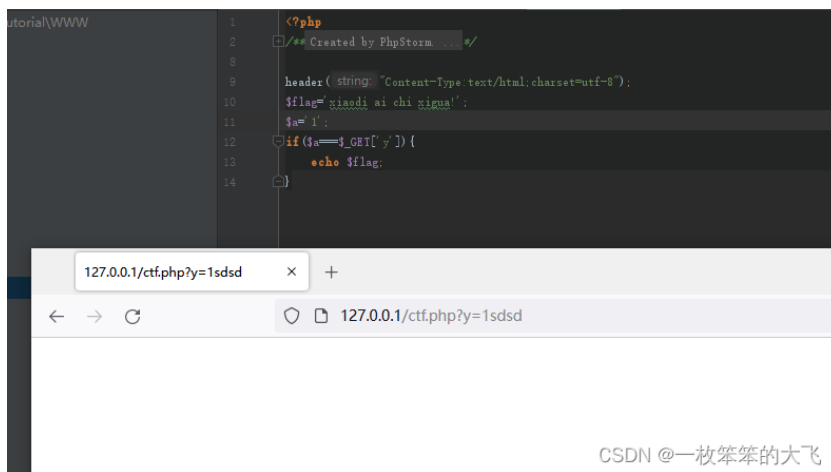


11

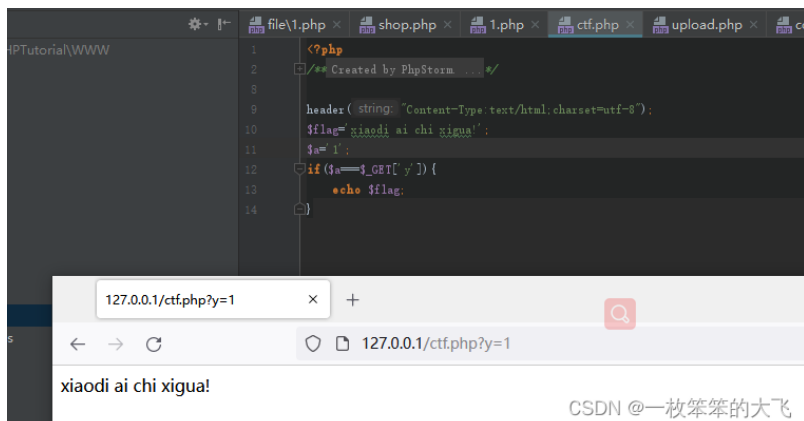


2

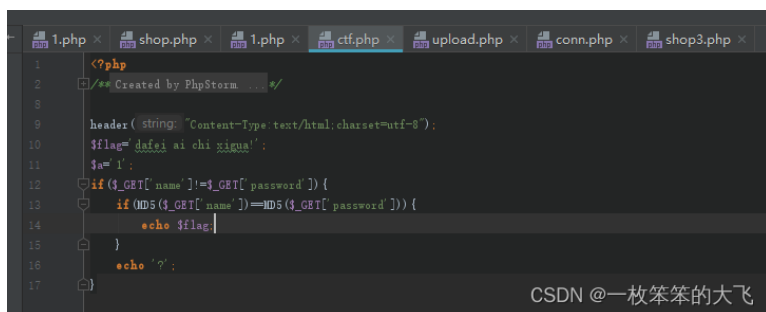




CSDN @一枚笨笨的大飞



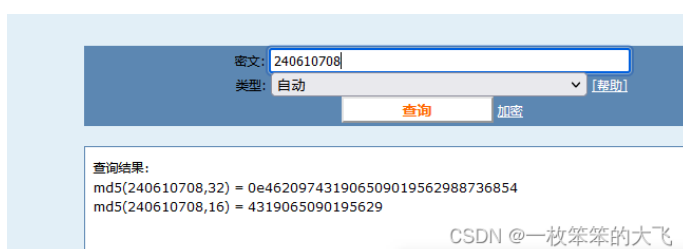
CSDN @一枚笨笨的大飞



CSDN @一枚笨笨的大飞

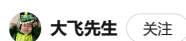


CSDN @一枚笨笨的大飞



CSDN @一枚笨笨的大飞

可以看到这两个MD5值都是0开头，而==符号存在的时



大飞先生

关注

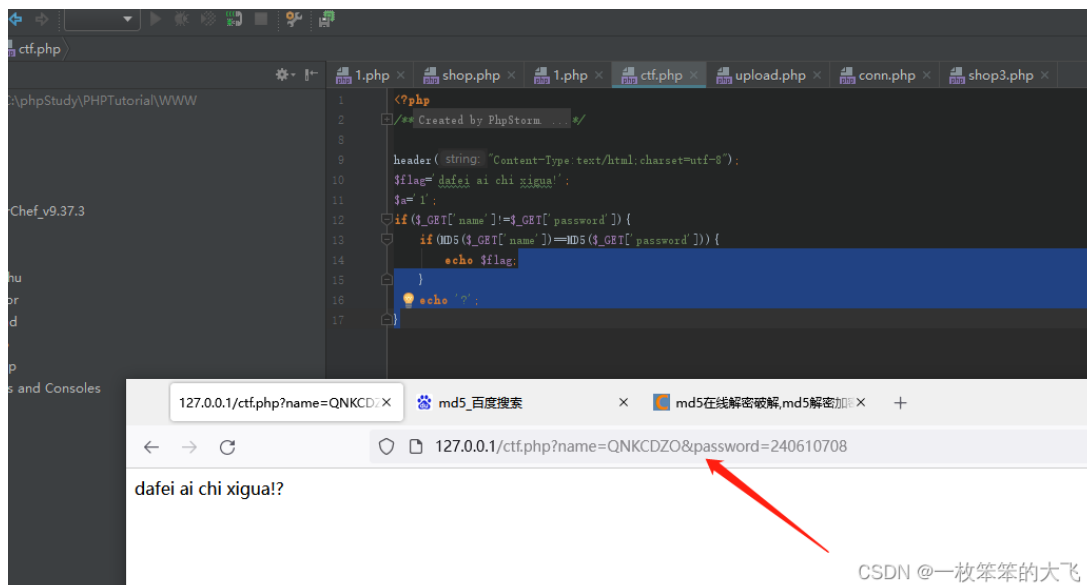


2

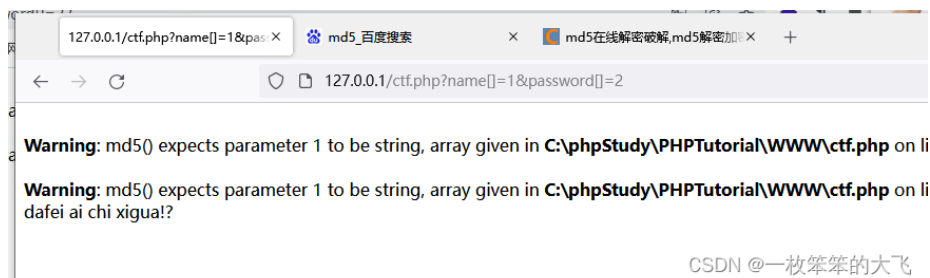
11

2

http://127.0.0.1/ctf.php?name=QNKCDZO&password=240610708



当遇到==的PHP验证代码时



2.intval

intval() 函数用于获取变量的整数值。

intval() 函数通过使用指定的进制 base 转换（默认是十进制），返回变量 var 的 integer 数值。intval() 不能用于 object，否则会产生 E_NOTICE 错误并返回 1。

语法

int intval (mixed \$var [, int \$base = 10])

参数说明:

- \$var: 要转换成 integer 的数量值。
- \$base: 转化所使用的进制。

如果 base 是 0，通过检测 var 的格式来决定使用的进制:

- 如果字符串包括了 "0x" (或 "0X") 的前缀，使用 16 进制(hex); 否则，
- 如果字符串以 "0" 开始，使用 8 进制(octal); 否则，
- 将使用 10 进制(decimal)。

1. 绕过方式1.0 +1.0

<?php

```
header("Content-Type:text/html;charset=utf-8");
```

```
$flag='dafei ai chi xigua!';
```

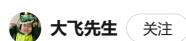
```
$a='1';
```

```
$i=$_GET['n'];
```

```
if(intval($i==$a)){
```

```
    echo $flag;
```

```
}
```





CSDN @一枚笨笨的大飞

2.绕过方式 666 0x29a (16进制)

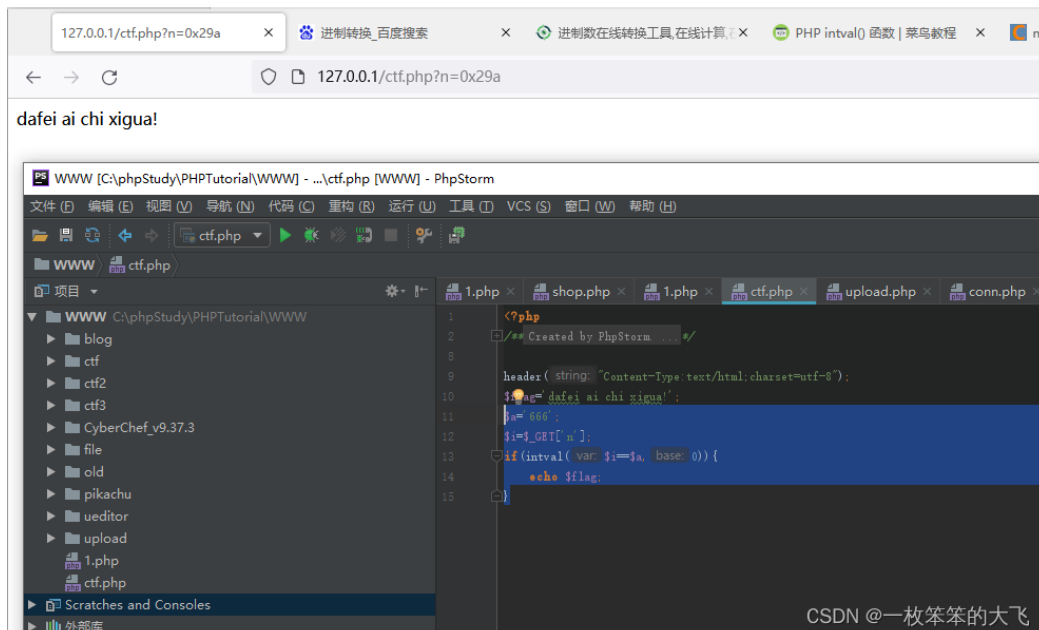
```
$a='666';
```

```
$i=$_GET['n'];
```

```
if(intval($i==$a,0){
```

```
    echo $flag;
```

```
}
```



CSDN @一枚笨笨的大飞

PHP strpos() 函数

查找 "php" 在字符串中第一次出现的位置:

定义和用法

strpos() 函数查找字符串在另一字符串中第一次出现的位置。

注释: strpos() 函数对大小写敏感。

注释: 该函数是二进制安全的。

相关函数:

- [stripos\(\)](#) - 查找字符串在另一字符串中第一次出现的位置 (不区分大小写)
- [strripos\(\)](#) - 查找字符串在另一字符串中最后一次出现的位置 (不区分大小写)
- [strpos\(\)](#) - 查找字符串在另一字符串中最后一次出现的位置 (区分大小写)

语法

```
strpos(string, find, start)
```

string

必需。规定要搜



大飞先生

关注

👍 2



★ 11



💬 2



| | |
|--------------------|---------------|
| <code>find</code> | 必需。规定要查找的字符串。 |
| <code>start</code> | 可选。规定在何处开始搜索。 |

1. 对于strpos()函数，利用换行绕过 (%0a)

0a: 换行

```
test.php x ctf.php x
<?php
/** Created by PhpStorm. ... */

header( string: "Content-Type:text/html;charset=utf-8");

//赋值 == 不会对比类型 ===类型也会对比
$flag='xiaodi ai chi xigua!';

//4. 对于strpos()函数，我们可以利用换行进行绕过 (%0a) |
$i='666';
$ii=$_GET['h'];
if(strpos($i,$ii, offset: "0")){
    echo $flag;
}

//?num=%0a666
```

CSDN @一枚笨笨的大飞

PHP in_array() 函数

in_array() 函数搜索数组中是否存在指定的值。

```
bool in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] )
```

| | |
|-----------------------|--|
| <code>needle</code> | 必需。规定要在数组搜索的值。 |
| <code>haystack</code> | 必需。规定要搜索的数组。 |
| <code>strict</code> | 可选。如果该参数设置为 TRUE，则 in_array() 函数检查搜索的数据与数组的值的类型是否相同。 |

127.0.0.1/ctf.php?n=2

dafei ai chi xigua!

CSDN @一枚笨笨的大飞

127.0.0.1/ctf.php?n=1e

dafei ai chi xigua!

大飞先生 关注

2 11 2

设置true属性时

```

1 <?php
2 /* Created by PhpStorm */
3
4
5
6
7
8
9 header( string: 'Content-Type: text/html; charset=utf-8' );
10 $flag= 'd4f0j ai chi 8:iguu';
11 $a=[1, 2, 3];
12 $i=$_GET['n'];
13 if (in_array($i, $a, strict: true)) {
14     echo $flag;
15 }

```

CSDN @一枚笨笨的大飞

PHP preg_match() 函数

`preg_match` 函数用于执行一个正则表达式匹配。

语法

```
int preg_match ( string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0 ]]] )
```

搜索 subject 与 pattern 给定的正则表达式的一个匹配。

参数说明:

- \$pattern: 要搜索的模式，字符串形式。
- \$subject: 输入字符串。
- \$matches: 如果提供了参数matches，它将被填充为搜索结果。\$matches[0]将包含完整模式匹配到的文本，\$matches[1]将包含第一个捕获子组匹配到的文本，以此类推。
- \$flags: flags 可以被设置为以下标记值:
 1. PREG_OFFSET_CAPTURE: 如果传递了这个标记，对于每一个出现的匹配返回时会附加字符串偏移量(相对于目标字符串的)。注意：这会改变填充到matches参数的数组，使其每个元素成为一个由第0个元素是匹配到的字符串，第1个元素是该匹配字符串 在目标字符串subject中的偏移量。
- offset: 通常，搜索从目标字符串的开始位置开始。可选参数 offset 用于 指定从目标字符串的某个未知开始搜索(单位是字节)。
- isset () 函数用于检测变量是否已设置并且非NULL。如果已经使用unset () 释放了一个变量之后，再通过isset () 判断将返回FALSE。
- 如果使用isset () 测试一个被设置成NULL的变量，将返回FALSE。同时要注意的是null字符 (“\0”) 并不相等于PHP的NULL常量。

//模式分隔符后的"i"标记这是一个大小写不敏感搜索

```
preg_match('/^php$/im')
```

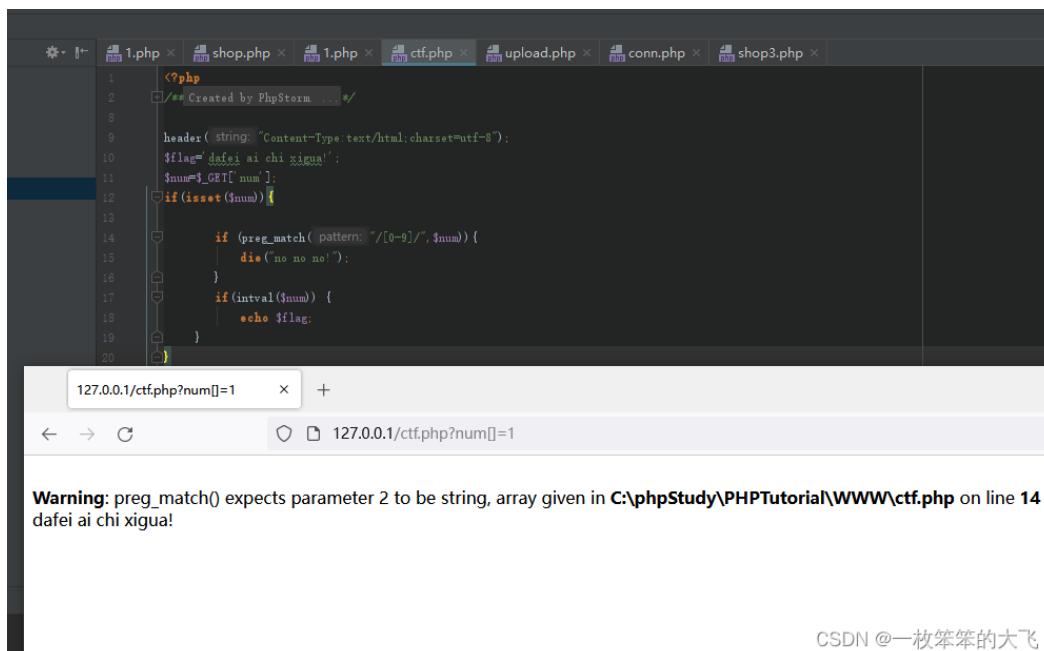
^:前边随意

\$: 结尾随意

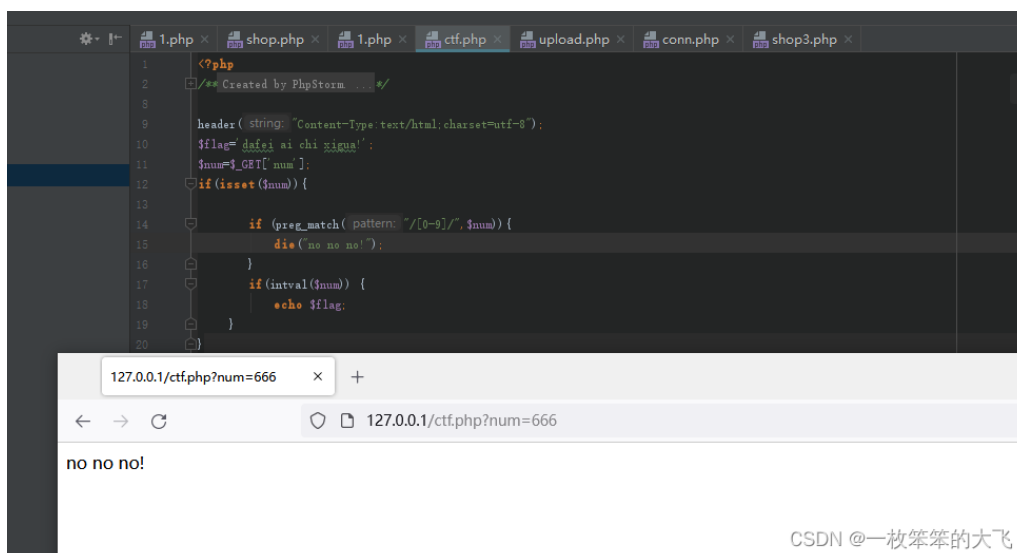
/i: 不区分大小写

/m: 换行匹配

绕过函数方法：利用数组，pre_match处理不了数组数据



CSDN @一枚笨笨的大飞



CSDN @一枚笨笨的大飞

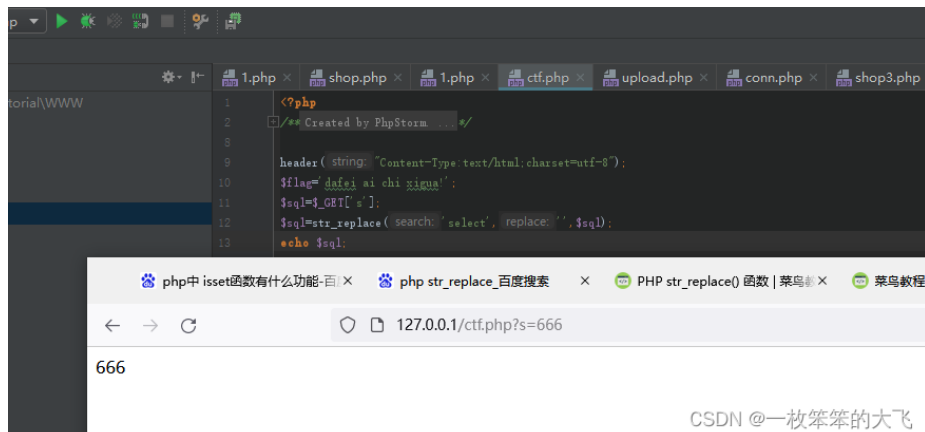
PHP str_replace() 函数

实例

把字符串 "Hello world!" 中的字符 "world" 替换成 "Peter":

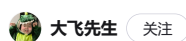
```
<?php
echostr_replace("world","Peter","Helloworld!");
?>
```

正常情况

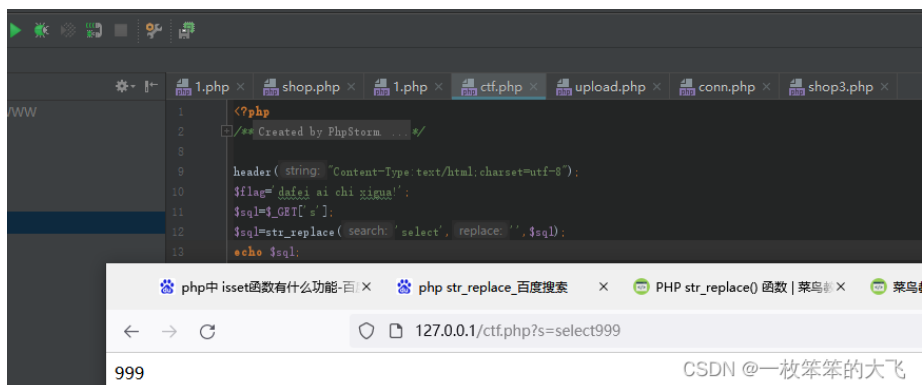


CSDN @一枚笨笨的大飞

过滤掉select时

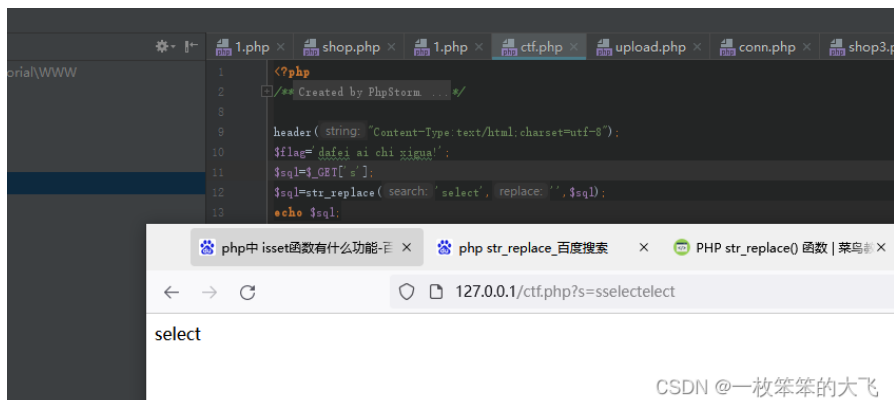


2 11 2



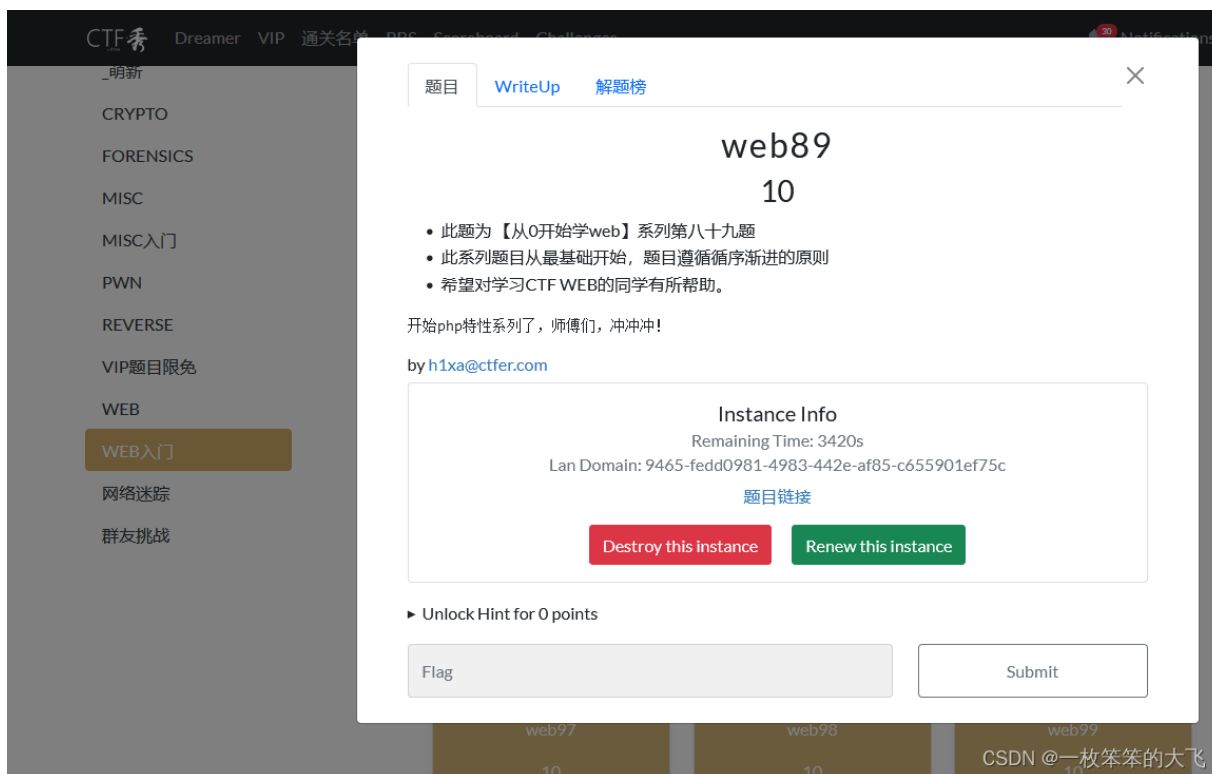
CSDN @一枚笨笨的大飞

过滤方法，只能过滤一次select

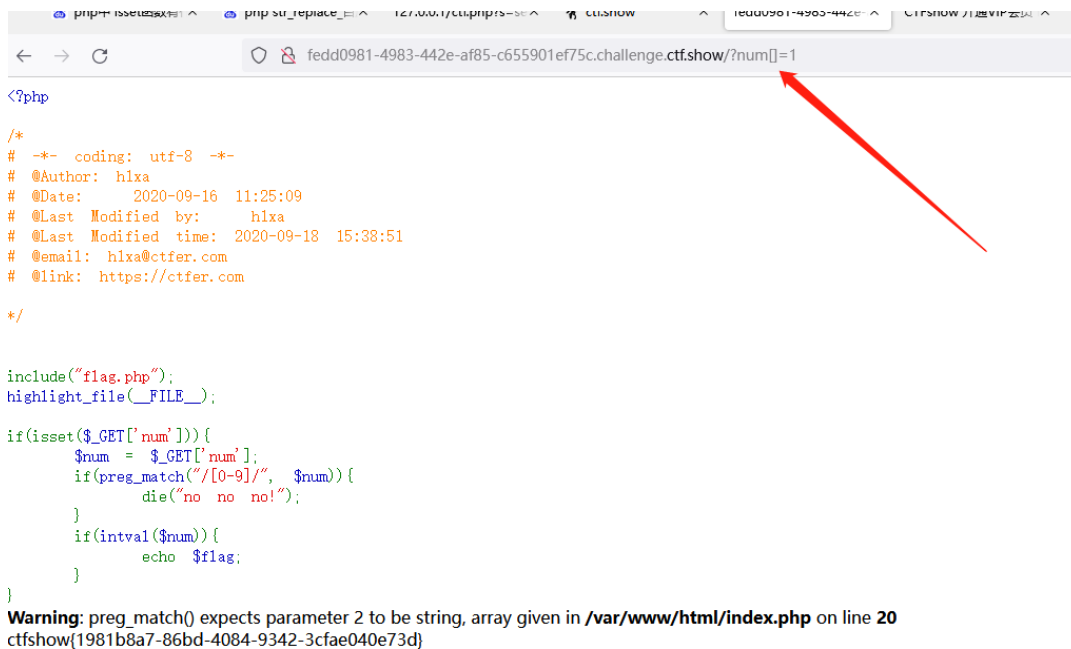


CSDN @一枚笨笨的大飞

CTFshow实战演练



CSDN @一枚笨笨的大飞



```

<?php
/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 15:38:51
# @email: hlxa@ctfer.com
# @link: https://ctfer.com
*/

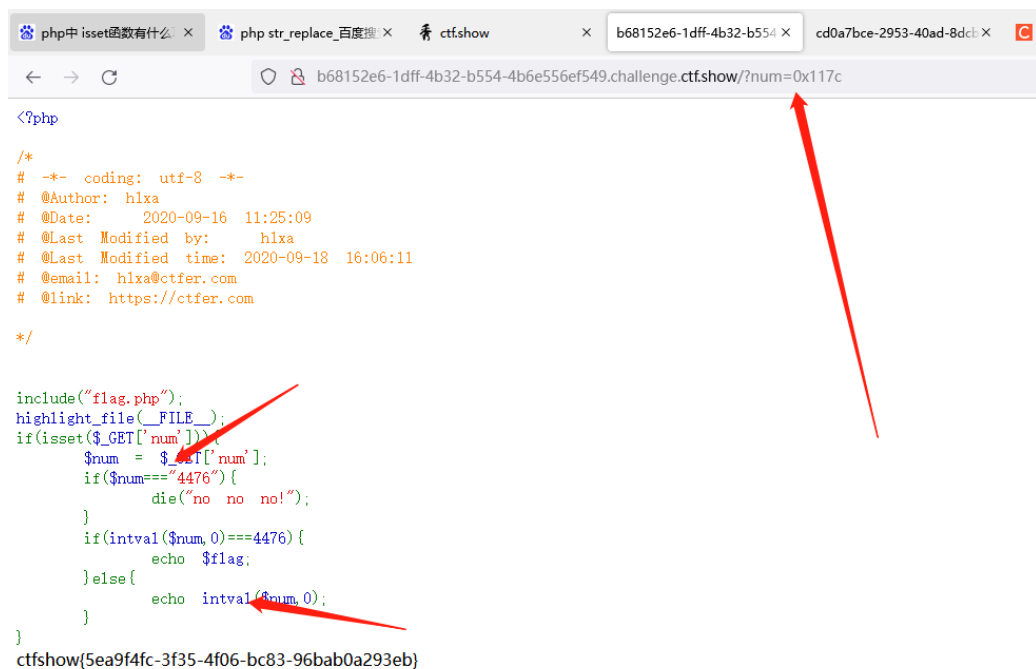
include("flag.php");
highlight_file(__FILE__);

if(isset($_GET['num'])){
    $num = $_GET['num'];
    if(preg_match("/[0-9]/", $num)){
        die("no no no!");
    }
    if(intval($num)){
        echo $flag;
    }
}

Warning: preg_match() expects parameter 2 to be string, array given in /var/www/html/index.php on line 20
ctfshow{1981b8a7-86bd-4084-9342-3cfae040e73d}

```

CSDN @一枚笨笨的大飞



```

<?php
/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 16:06:11
# @email: hlxa@ctfer.com
# @link: https://ctfer.com
*/

include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num=="4476"){
        die("no no no!");
    }
    if(intval($num,0)===4476){
        echo $flag;
    }else{
        echo intval($num,0);
    }
}

ctfshow{5ea9f4fc-3f35-4f06-bc83-96bab0a293eb}

```

CSDN @一枚笨笨的大飞

```

<?php
/*
# -*- coding: utf-8 -*-
# @Author: Firebasky
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 16:32:58
# @link: https://ctfer.com
*/

include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){ == 不限制数据类型
        die("no no no!");
    }
    if(preg_match("/[a-z]/i", $num)){ pre_match限制了a-z的输入，所以使用不了0x开头的16进制，只可以用0开头的8进制进行绕过
        die("no no no!");
    }
    if(intval($num,0)==4476){
        echo $flag;
    }else{
        echo intval($num,0); intval base=0 代表可以使用8、16进制绕过
    }
}
ctfshow(7ee6faaa-f5c2-4af0-b1e4-f0ddaab0926e)

//模式分隔符后的"i"标记这是一个大小写不敏感搜索

```

```

<?php
/*
# -*- coding: utf-8 -*-
# @Author: Firebasky
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 16:16:09
# @link: https://ctfer.com
*/

show_source(__FILE__);
include('flag.php');
$a=$_GET['cmd'];
if(preg_match("/php/im", $a)){ /i 不区分大小写 /m 匹配换行
    if(preg_match("/php/i", $a)){
        echo 'hacker';
    }
    else{
        echo $flag; %0A 换行符
    }
}
else{
    echo 'nonononono';
}
ctfshow(21d53cec-98ab-43b4-bcea-7a2e8620a005)

```

```

<?php
/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 16:46:19
# @link: https://ctfer.com
*/

include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){
        die("no no no!");
    }
    if(preg_match("/[a-z]/i", $num)){
        die("no no no!");
    }
    if(!strpos($num, "0")){ 过滤0 使用换行符%0a进行绕过
        die("no no no!");
    }
    if(intval($num,0)==4476){
        echo $flag;
    }
}
ctfshow(792f479e-7182-46d1-8933-7e81984a0dc6)

```

```

<?php

/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 16:53:59
# @link: https://ctfer.com
*/

include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){
        die("no no no!");
    }
    if(preg_match("/[a-z]|\./i", $num)){
        die("no no no!");
    }
    if(!strpos($num, "0")){
        die("no no no!!!");
    }
    if(intval($num,0)==4476){
        echo $flag;
    }
} ctfshow{10d6225c-ab8a-47af-9464-285c3796503b}

```

CSDN @一枚笨笨的大飞

```

<?php

/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 19:21:24
# @link: https://ctfer.com
*/

highlight_file(__FILE__);

if(isset($_GET['u'])){
    if($_GET['u']=='flag.php'){
        die("no no no");
    }else{
        highlight_file($_GET['u']);
    }
}

<?php

/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:24:37
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-16 11:25:00
# @email: hlxa@ctfer.com
# @link: https://ctfer.com
*/

$flag="ctfshow{29823bd9-52e9-4039-afb9-c1e4cbebd79}";

```

绕过方式: 利用文件目录的格式./flag.php != flag.php

CSDN @一枚笨笨的大飞

SQL BASICS- UNION BASED- ERROR/DOUBLE QUERY- TOOLS- WAF BYPASS- ENCODING- HTML- ENCRYPTION- OTHER- XSS- LFI-

Load URL <http://d558b132-e3e9-4d26-88bf-d3dc525d25c2.challenge.ctf.show/>

Split URL

Execute

Post data Post data Referrer OxHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

a[]=1&b[]=2

```
<?php
/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-18 19:36:32
# @link: https://ctfer.com
*/

include("flag.php");
highlight_file(__FILE__);
if (isset($_POST['a']) and isset($_POST['b'])) {
if ($_POST['a'] != $_POST['b'])
if (md5($_POST['a']) == md5($_POST['b']))
echo $flag;
else
print 'Wrong.';
}
?>
```

Warning: md5() expects parameter 1 to be string, array given in /var/www/html/index.php on line 17

Warning: md5() expects parameter 1 to be string, array given in /var/www/html/index.php on line 17

ctfshow[d7b4d8a8-3cc9-42f3-b3cc-6dbe16e6a84a]

CSDN @一枚笨笨的大飞

==绕过方式: name=QNKCDZO&password=240610708

===绕过方式: 数组的方式 a[]=1&b[]=1

name=QNKCDZO&password=240610708

成功的拿到了flag

php使用strpos判断字符串中数字类型子字符串出错的方法 原创 12-19
本文实例讲述了php使用strpos判断字符串中数字类型子字符串出错的方法。分享给大家供大家参考，具体如下：一、问题：最近的开发中在程序代码里有一个随机数是否在给定字符串里的判断，我...

mysql&&sql注入+ctf+web安全 10-08
mysql&&sql注入+ctf+web安全

2条评论 wlogghost 热评 好家伙，正好在上小迪的课就能刷到你这篇文章 写评论

利用回溯绕过正则表达式_正则回溯绕过正则匹配 5-5
可以看到成功的拿到了flag,到这里利用回溯绕过war的实验就已经全部完成了 总结一下 1、我们利用strpos函数会将非字符串的值当做null+!=来绕过了最基本的正则 2、利用回溯我们也可以绕过文件上...

PHP绕过strpos()_php strpos函数怎么绕过 4-28
PHP绕过strpos() 可以结合ctfshowweb94 strpos(string,find,start)有三个参数,string是被检查的字符串,find是要被搜索的字符串,start是开始检索的位置,从0开始。该函数返回查找到这个find字符串的位置...

bugku strpos数组绕过 m0_52432374的博客 574
strpos数组绕过 <?php \$flag = "flag"; if (isset(\$_GET['ctf'])) { if (@ereg ("^[1-9]+\$", \$_GET['ctf']) === FALSE) echo '必须输入数字才行'; else if (strpos(\$_GET['ctf'], "#biubiubiu") !== FALSE) die('Flag: '.\$fl...

关于 CTF 中 php 考点与绕过那些事的总结 最新发布 Welcome To Myon'Blog! 1605
_POST //获取post数据, 是一个字典\$_GET //获取get数据, 是一个字典\$_COOKIE //获取cookie数据\$_SESSION //获取session数据\$_FILES //获取上传的文件\$_REQUEST //获取\$_GET, \$_POST, \$_C...

攻防世界-warmup_mb_substr()+mb_strpos()绕过_mb_substr绕过 4-26
mb_strpos()*-返回要查找的字符串在另一个字符串中首次出现的位置// mb_strpos (haystack ,needle // haystack:要被检查的字符串。// needle:要搜索的字符串mb_substr()*函数返回字符串的一部...

PHP代码审计总结(2) 5-10
1, strpos数组绕过NULL与ereg正则%00截断 <?php \$flag = "flag"; if (isset(\$_GET['nctf'])) { if (@ereg ("^[1-9]+\$", \$_GET['nctf']) === FALSE) echo '必须输入数字才行'; ...

php中的strpos使用示例 10-26
strpos()函数返回字符串在另一个字符串中第一次出现的位置。如果没有找到该字符串,则返回 false, 下面看示例使用方法

php中字符串查找函数strpos、strchr与strpbrk用法 12-18
本文实例讲述了php中字符串查找函数strpos、strchr与strpbrk用法。分享给大家供大家参考。具体如下：① strpos() 函数返回字符串在另一个字符串中第一次出现的位置,如果没有找到该字符串,则返回 fal...

代码审计-strpos数组绕过 5-6
strpos() 函数查找字符串在另一字符串中第一次出现的位置。参数ctf要输入数字,同时又要输入#biubiubiu 但是两个函数都可以用数组进行绕过 payload http://123.206.87.240:9009/15.php?ctf[]=1 ...

strpos数组绕过NULL、密码md5比较绕过、MD5函数===绕过 giun的博客 3290
<?php \$flag = "flag"; if (isset(\$_GET['nctf'])) { if (@ereg ("^[1-9]+\$", \$_GET['nctf']) === FALSE) echo '必须输入数字才行'; else if (strpos(\$_GET['nctf'], "#biubiubiu") ...

Bugku——strpos数组绕过 热门推荐 王哪哪的博客 1万+
0x00 前言不想学习, 简单的做一做CTF的题目。看到了CTF里的排名, 然后好像又有了动力。题目 0x01 start 题目说的很清楚。数组绕过。那我们来看一下源码, 然后来进行测试。 <?php \$flag = "fla...

封神台ctf练习靶场——一月靶场 AlunXZhou的博客 1374
封神台刷题

PHP代码安全4-- ==逻辑与函数缺陷 ThegreatHaige的博客 531
strpos(string,find,start)有三个参数, string是被检查的字符串, find是要被搜索的字符串, start是开始检索的位置, 从0开始。php代码中的老熟人了, 经常混迹于各大检测与ctf代码审计中,preg_match限...

php 绕过 strpos,PHP的两个特性导致waf绕过注入 (有趣的知识点) weixin_29692851的博客 1501
1、HPP HTTP参数污染HTTP参数污染指的是, 在URL中提

CTF 离线C PHP python 汇编函数查询 chm电子书合: 大飞先生 关注 2 11 2

CTF 离线C PHP python 汇编函数查询 chm电子书合集 离线比赛时使用 学习

strpos() 函数判断字符串中是否包含某字符串的方法
用php的strpos() 函数判断字符串中是否包含某字符串的方法判断某字符串中是否包含某字符串的方法 if(strpos('www.idc-gz.com','idc-gz') !== false){ echo '包含'; }else{ echo '不包含'; } ... 12-20

CTF PHP离线chm文档 08-09
CTF PHP离线chm文档

CTF Attack & Defense线下攻防赛比赛技巧.pdf 01-15
不错的资源哦!

CTF-Web-Challenge:使用PHP在Web中进行CTF挑战 03-25
CTF-网络挑战使用PHP在Web中进行CTF挑战链接: :

php弱类型比较及绕过 weixin_45349299的博客 2602
字符串和数字比较使用==时,字符串会先转换为数字类型再比较,若字符串以数字开头,则取开头数字作为转换结果,不能转换为数字的字符串(例如"aaa"是不能转换为数字的字符串,而"123"或"123aa"...

刷题学习记录(封神台) qq_73861475的博客 151
PHP 过滤器用于对来自非安全来源的数据(比如用户输入)进行验证和过滤。

php一些特性函数(ctfshow) qq_62046696的博客 1825
preg_match()返回 pattern 的匹配次数。它的值将是0次(不匹配)或1次,因为preg_match()在第一次匹配后 将会停止搜索。preg_match_all()不同于此,它会一直搜索subject直到到达结尾。如果发生...

eval函数读取文件 ctf 09-02
引用中提到了通过使用file()函数将文件内容存入数组,然后通过var_dump()和eval()函数将数组内容输出到页面的方法。eval()函数的参数是一个字符串,该字符串末尾必须有分号,并且在最后还要...

“相关推荐”对你有帮助么?

非常没帮助 没帮助 一般 有帮助 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文[2020] 1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心
家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照
©1999-2024北京创新乐知网络技术有限公司

大飞先生 码龄3年 暂无认证
38 39万+ 27万+ 3万+ 原创 周排名 总排名 访问 等级
438 38 22 16 69 积分 粉丝 获赞 评论 收藏
私信 关注

大额流量券送不停
多发多得,流量翻倍!
去查看

搜博主文章

热门文章

- Ubuntu搭建ARL资产灯塔系统(适用所有版本,5分钟迅速搭建成功) 2058
- PHP特性&缺陷对比函数&CTF考点 2031
- Python-shellcode免杀分离 1913
- CDN绕过篇&漏洞回链&接口探针&全网扫描&反向邮件 1775
- PHP-nginx-ctfShow文件上传漏洞 1715

分类专栏

- 基础的渗透测试 26篇
- 渗透测试 2篇
- http 1篇

最新评论

gophish平台搭建
常微笑:请问下,大佬,那个邮件报告是收集的哪方面的信息呢?
Python-shellcode免杀分离
Young洋475:为啥我换成32位了,还是报错呀

大飞先生 关注 2 11 2

PHP特性&缺陷对比函数&CTF考点

wlogghost: 好家伙，正好在上小迪的课就能刷到你这篇文章

XSS学习笔记（一）、CTFSHOW-316到...

shdkfbv: 331 借助修改密码重置管理员权限 POST 发送ajax数据包时不用加127.0 ...

PHP-nginx-ctfShow文件上传漏洞

zhuo21hahaha: 咋跟小迪课程的文档内容那么像呢 🤔

您愿意向朋友推荐“博客详情页”吗？



强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

内网安全-隧道技术&SSH实现通信&DNS上线与通信&CS上线Linux主机

Python-shellcode免杀分离

vulnhub-RAVEN:2(MYSQL-UDF提权，手工提权/工具自动提权)

2023年 14篇

2022年 24篇

目录

定义和用法

相关函数:

语法

PHP in_array() 函数

PHP preg_match() 函数

语法

PHP str_replace() 函数

实例