

2016



4.0

XiunoPHP 开发手册

保持 PHP 原来的味道

www.xiuno.com

目 录

XiunoPHP 入门

什么是 XiunoPHP ?

关于 URL 格式

编码规范

全局变量

\$starttime

\$time

\$conf

\$ip

\$longip

\$ajax

\$method

\$db

\$cache

\$errno

\$errstr

数据库函数

DB 配置

db_insert()

db_create()

db_replace()

db_update()

db_delete()

db_find_one()

db_find()

db_count()

db_maxid()

db_connect()

db_truncate()

db_sql_find()

db_sql_find_one()

db_exec()

db_new()

db_close()

缓存函数

CACHE 配置

cache_set()

cache_get()

cache_delete()

cache_truncate()

cache_new()

数组增强

array_value()

array_filter_empty()

array_addslashes()

array_stripslashes()

array_htmlspecialchars()

array_trim()

array_diff_value()

array_assoc_slice()

arrlist_multisort()

arrlist_cond_orderby()

arrlist_key_values()

arrlist_values()

arrlist_change_key()

arrlist_chunk()

杂项函数

xn_strlen()

xn_substr()

xn_urlencode()

xn_urldecode()

xn_json_encode()

xn_json_decode()

xn_encrypt()

xn_decrypt()

xn_message()

xn_error()

xn_log()

xn_txt_to_html()

xn_rand()

xn_is_writable()

humandate()

humannumber()

humansize()

param()

lang()

url()

pagination()

is_robot()

http_get()
http_post()
https_get()
https_post()
http_multi_get()
file_replace_var()
file_get_contents_try()
file_put_contents_try()
in_string()
file_ext()
file_pre()
file_name()
http_url_path()
glob_recursive()
rmdir_recursive()
copy_recursive()
_GET() _POST() _COOKIE() ...

XiunoPHP 入门

什么是 XiunoPHP 4.0 ?



XiunoPHP - 保持 PHP 原来的味道！ - Keep the PHP native style

XiunoPHP 并不是一个框架，它只是初始化了一些常用的全局变量，增加了一些常用的函数，可以认为是对 PHP 的一些功能的补充。合并后，它只有一个文件 xiunophp.min.php，可以不做任何配置，直接 include 后就可以使用。

你可以用任意方式（过程式、面向对象、函数式）使用它，甚至和其他框架混用。在你需要某个函数和变量的时候，它刚好存在，用它就 OK 了。

如何获取？

XiunoPHP 4.0 基于宽松的 MIT 协议发布，所以您可以最大限度的使用它。

下载地址：http://bbs.xiuno.com/download/xiunophp_4.0.zip

解压后会得到一个目录 xiunophp，里面包含一个合并后的文件：xiunophp.min.php，这个文件可以存放在任何地方。

如何使用？

include 后直接使用里面的函数即可。

```
<?php
include './xiunophp.min.php';

$s = xn_encrypt('hello, world!');

echo $s;

?>
```

什么是 XiunoPHP ?

什么是 XiunoPHP 4.0 ?



XiunoPHP - 保持 PHP 原来的味道 !

XiunoPHP 并不是一个框架，它只是初始化了一些常用的全局变量，增加了一些常用的函数，可以认为是对 PHP 的一些功能的补充。合并后，它只有一个文件 xiunophp.min.php，可以不做任何配置，直接 include 后就可以使用。

你可以用任意姿势（过程式、面向对象、函数式）使用它，甚至和其他框架混用。在你需要某个函数和变量的时候，它刚好存在，用它就 OK 了。

如何获取？

XiunoPHP 4.0 基于宽松的 MIT 协议发布，所以您可以最大限度的使用它。

下载地址：http://bbs.xiuno.com/down/xiunophp_4.0.tar.gz

解压后会得到一个目录 xiunophp，里面包含一个合并后的文件：xiunophp.min.php，这个文件可以存放在任何地方。

如何使用？

include 后直接使用里面的函数即可。

```
<?php
include './xiunophp.min.php';

$s = xn_encrypt('hello, world!');

echo $s;

?>
```

关于 URL 格式

关于 URL 格式

XiunoPHP 4.0 支持 2 种 URL 格式：

<http://domain.com/user-login.htm>

<http://domain.com/user/login>

非 URL-Rewrite 需要加一个？

<http://domain.com/?user-login.htm>

<http://domain.com/?user/login>

如何获取参数（参数是按照位置，从 0 开始计数）：

```
<?php
include './xiunophp.min.php';
$route = param(0);          // user
$action = param(1);         // login
?>
```

当然也可以直接用默认的参数格式：

<http://domain.com/?route=user&action=login>

```
<?php
include './xiunophp.min.php';
$route = param('route');
$action = param('action');
?>
```

如何配置 URL-Rewrite ?

Nginx :

打开 nginx 配置文件 /usr/local/nginx/conf/nginx.conf 找到对应的虚拟主机配置处，追加：

```
location / {
    rewrite "^(.*)/(.+?).htm(.*?)$" $1/index.php?$2.htm$3 last;
    index    index.html index.htm index.php;
    root     /data/wwwroot/xiuno.com;
}
```

然后重新启动 nginx: service nginx restart

Apache .htaccess

如果Apache 支持 .htaccess，那么可以编辑 .htaccess 文件放置于根目录下：

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteRule ^(.?)([^\/*]*)\.htm(.*)$ $1/index.php?$2.htm$3 [L]
</IfModule>
```

更多的细节请访问：<http://bbs.xiuno.com/thread-2.htm>

编码规范

编码规范

XiunoPHP 提倡 Linux 的编码风格（小写 + 下划线），与 PHP 默认的命名风格保持一致，但如果您喜欢匈牙利和骆驼也没有关系。

1. 变量名，函数名全部为：小写 + 下划线，比如：

```
<?php
$uid = 0;
$username = '';
mysql_connect();
mysql_query();
mysql_fetch_assoc();
?>
```

2. 常量全部大写：

```
<?php
define('DEBUG', 1);           // 1: 开发模式, 2: 线上调试: 日志记录, 0: 关闭
define('APP_NAME', 'www');
?>
```

3. 空格，缩进，换行，参考以下格式：

一个 TAB = 8 个空格，尽量减少 TAB 缩进。

```
<?php
function array_addslashes(&$var) {
    if(is_array($var)) {
        foreach($var as $k=>$v) {
            array_addslashes($v);
        }
    } else {
        $var = addslashes($var);
    }
    return $var;
}
?>
```

4. 单引号、双引号：

在 PHP 当中，尽量使用单引号，解析速度比双引号快。

如果里面包含变量，为了代码的易读性，可以使用双引号。

在双引号中的数组 key 不应该加单引号。

在单引号中仅仅转义 \，其他字符都不转义，如 \t\r\n \$。

以下为正确用例：

```
<?php
$sitename = '我在北京吸雾霾';
```

```
$info = "站点名称:$sitename";
$info = "用户名:$user[name]";
?>
```

5. 类、继承、接口、构造、析构、魔术方法：

尽量不要使用 PHP 高级特性。

高级特性往往不利于底层的优化，需要更多的学习和沟通成本。

比如 **get()** **set()** **__call()** 会让代码变得很难读，IDE 提示也不友好。

不是刚需，不要用。

6. 正则表达式：

尽量使用单引号，分隔符为 # 。

禁止使用 e 修饰符，如果刚需，请使用 preg_replace_callback() 代替。

尽量使用 \w \s \S 内置的表示方法，不要啰嗦的去写 [0-9a-zA-Z] 。

为什么不用 / 作为分隔符？

因为 WEB 开发过程中，字符串中出现 / 的概率太高。

以下正则格式符合标准：

```
preg_match('#\w+@\w+.\w+#is', $email);
```

7. include include_once require require_once：

尽量使用 include，速度快，并且不会中断业务逻辑。

require 在文件不存在或不可读的时候，会暴力终止业务逻辑。

8. error_reporting：

在本地开发环境下使用，使用 E_ALL，消灭所有 NOTICE。

线上环境使用 0，并且配置 php.ini error_log 记录到服务器日志，避免错误信息外泄。

9. 模板：

不要用 Smarty 等任何类型的模板“引擎”，他们不时真正意义上的引擎，只是一堆正则替换而已。而且效率低下，学习的时间成本高，浪费脑细胞。

直接使用 PHP 的 原生标签，比如：

```
<?php include "../view/htm/header.inc.htm"; ?>
Hello, <?php echo $username; ?>!
<?php include "../view/htm/footer.inc.htm"; ?>
```

10. 目录约定：

为了便于部署和排查，约定以下目录用途（非强制）：

```
Web 目录：/home/wwwroot/xxx.com
Web 日志：/home/wwwlog
MySQL 数据：/home/mysql
备份目录：/home/backup
Nginx 配置文件：/usr/local/nginx/conf/nginx.conf
MySQL 配置文件：/etc/my.cnf
```

```
PHP 配置文件：/usr/local/php/etc/php.ini  
PHP-CGI 配置文件：/usr/local/php/etc/php-fpm.conf
```

11. 生产环境推荐：

CentOS + PHP7 + MySQL + Opcache + Yac

全局变量

\$starttime

\$time

\$conf

\$ip

\$longip

\$ajax

\$method

\$db

\$cache

\$errno

\$errstr

\$starttime

\$starttime

脚本执行的开始时间，单位为毫秒，格式：1448335223.1877。

一般用来计算程序执行时间。

【定义】 文件：xiunophp/xiunophp.php 大约 40 行：

```
$starttime = microtime(1);
```

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
echo microtime(1) - $starttime;    // 0.24460697174072  
  
?>
```

【注意】

XiunoPHP 在被 include 后，会初始化十几个常用的全局变量，请注意命名不要冲突。

\$time

\$time

\$time

脚本执行的开始时间，单位为秒，格式：1448336550。
通用的 UNIX 时间戳。

【定义】 文件：xiunophp/xiunophp.php 大约 41 行：

```
$time = time();
```

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
echo "当前时间：" . date('Y-n-j', $time); // 当前时间：2015-11-24  
  
?>
```

\$conf

\$conf

应用的全局配置，格式为数组，一般通过配置文件包含返回。
非强制要求，开发较大型的项目时建议设置。

【定义】 文件：xiunophp/xiunophp.php 大约 63 行：

```
empty($conf) AND $conf = array('db'=>array(), 'cache'=>array(), 'tmp_path'=>'./', 'log_path'=>'./', 'timezone'=>'Asia/Shanghai');
```

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

?>
```

【conf.php】

```
<?php
return array (
    // -----> xiunophp 依赖的配置
    'db'=>array(
        'type'=>'pdo_mysql',
        'pdo_mysql' => array (
            'master' => array (
                'host' => 'localhost',
                'user' => 'root',
                'password' => 'root',
                'name' => 'test',
                'charset' => 'utf8',
                'tablepre'=>'bbs_',
                'engine'=>'myisam', // innodb
            ),
            'slaves' => array()
        )
    ),
    'tmp_path' => './', // 可以配置为 linux 下的 /dev/shm , 通过内存缓存临时文件
    'log_path' => './'
);
?>
```

\$ip

\$ip

\$ip

客户端的 IP 地址，格式为：202.106.0.20。

在开启 CDN 后，它会获取 CDN 转发过来的 IP。

【定义】 文件：xiunophp/misc.func.php：

```
$ip = ip();
```

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
echo $ip; // 127.0.0.1  
  
?>
```

【ip()】

// 不安全的获取 IP 方式，在开启 CDN 的时候，如果被人猜到真实 IP，则可以伪造。

```
function ip() {  
    global $conf;  
    $ip = '127.0.0.1';  
    if(empty($conf['cdn_on'])) {  
        $ip = $_SERVER['REMOTE_ADDR'];  
    } else {  
        if(isset($_SERVER['HTTP_CDN_SRC_IP'])) {  
            $ip = $_SERVER['HTTP_CDN_SRC_IP'];  
        } elseif(isset($_SERVER['HTTP_CLIENT_IP'])) {  
            $ip = $_SERVER['HTTP_CLIENT_IP'];  
        } elseif(isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {  
            $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];  
            $arr = array_filter(explode(',', $ip));  
            $ip = end($arr);  
        } else {  
            $ip = $_SERVER['REMOTE_ADDR'];  
        }  
    }  
    return long2ip(ip2long($ip));  
}
```


\$longip

\$longip

\$longip

客户端的 IP 地址的 long 格式：2130706433。
一般用 4 个字节的 unsigned int 记录到数据库。

【定义】 文件：xiunophp/xiunophp.php 大约 50 行：

```
$longip = ip2long($ip);
```

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
echo $longip; // 2130706433  
  
?>
```

\$ajax

\$ajax

\$ajax

是否为 AJAX 请求，值为 TRUE/FALSE。

浏览器在发送 AJAX 请求的时候，会发送特定的头信息，会被当做判断依据。

【定义】 文件：xiunophp/xiunophp.php 大约 55 行：

```
$ajax = isset($_SERVER['HTTP_X_REQUESTED_WITH']) && strtolower(trim($_SERVER['HTTP_X_REQUESTED_WITH'])) == 'xmlhttprequest';
```

【用例】

```
<?php
include './xiunophp/xiunophp.php';

if($ajax) {
    // ajax 请求时，服务端一般返回 json 格式数据
    echo '{"code":1, "message":"ok"}';
} else {
    echo 'ok';
}

?>
```

\$method

\$method

\$method

判断 HTTP 请求的方法，默认值为 GET。

【定义】 文件：xiunophp/xiunophp.php 大约 77 行：

```
$method = $_SERVER['REQUEST_METHOD'];
```

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo $method;

// GET

?>
```

\$db

\$db

\$db

默认数据库实例

如果配置文件设置了数据库相关的配置，则框架会自动实例化一个 DB 类。

一般不用直接使用这个变量，除非在升级，转换需要多个连接的时候。

一般用 db_find(), db_find_one(), db_exec(), db_count() 函数来操作数据库。

【定义】 文件：xiunophp/xiunophp.php 大约 76 行：

```
$db = !empty($conf['db']) ? db_new($conf['db']) : NULL;
```

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arr = $db->find_one("SELECT * FROM bbs_user LIMIT 1");

?>
```

【conf.php】

```
<?php
return array (
    // -----> xiunophp 依赖的配置
    'db'=>array(
        'type'=>'pdo_mysql',
        'pdo_mysql' => array (
            'master' => array (
                'host' => 'localhost',
                'user' => 'root',
                'password' => 'root',
                'name' => 'test',
                'charset' => 'utf8',
                'engine'=>'myisam', // innodb
            ),
            'slaves' => array()
        )
    ),
    'tmp_path' => './', // 可以配置为 linux 下的 /dev/shm，通过内存缓存临时文件
    'log_path' => './'
);
?>
```

\$db

\$cache

\$cache

默认缓存实例

如果配置文件设置了缓存相关的配置，则框架会自动实例化一个 Cache 类。

一般不用直接使用这个变量，除非在升级，转换需要多个连接的时候。

一般用 `cache_set()`, `cache_get()` 函数来操作缓存。

使用时，需要配置好 Cache 服务和 PHP 相关的 Cache 扩展，目前支持：

`apc|xcache|memcached|redis|mysql`。

如果为单机，建议使用 `xcache`，速度比较快，相关文档：<http://bbs.xiuno.com/thread-8762.htm>。

【定义】 文件：`xiunophp/xiunophp.php` 大约 77 行：

```
$cache = !empty($conf['cache']) ? cache_new($conf['cache']) : NULL;
```

【用例】

```
<?php
$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$cache->set('key1', 'value1');
$cache->get('key1');

?>
```

【conf.php】

```
<?php
return array (
    // -----> xiunophp 依赖的配置
    'cache'=> array(
        'enable' => TRUE,
        'type'=> 'xcache', // apc|xcache|memcached|redis|mysql
        'memcached'=> array (
            'host'=>'localhost',
            'port'=>'11211',
        ),
        'redis'=> array (
            'host'=>'localhost',
            'port'=>'6379',
        ),
    ),
    'tmp_path' => './', // 可以配置为 linux 下的 /dev/shm，通过内存缓存临时文件
    'log_path' => './'
);
?>
```

\$cache

\$errno

\$errno

错误号，为 0 时，表示正常，非 0 时表示错误。

在某些函数被调用后发生错误，该全局变量会被设置，有点类似于 Linux C 的 `errno`。

因为 PHP 是同步的（区别于异步）并且线程安全的，所以使用这种方法来返回错误会很方便。

一般配合 `errstr` 一起使用，函数内通过 `xn_error($errno, $errstr)` 设置错误。

相比起抛出异常来，这种处理方式会轻量级很多。

【定义】 文件：`xiunophp/xiunophp.php` 大约 58 行：

```
$errno = 0;
```

【用例】

```
<?php
include './xiunophp/xiunophp.php';
include './xiunophp/xn_send_mail.php';

$r = xn_send_mail(array(), 'username', 'test@gmail.com', '标题', '内容');
if($r === FALSE) {
    echo "Errno: ".$errno.", Errstr: ".$errstr;
}

?>
```

【相关函数】

`xn_error()`

\$errstr

\$errstr

错误字符串，与 \$errno 配合使用。

【定义】 文件：xiunophp/xiunophp.php 大约 58 行：

```
$errstr = '';
```

【相关函数】

xn_error()

数据库函数

DB 配置

`db_insert()`

`db_create()`

`db_replace()`

`db_update()`

`db_delete()`

`db_find_one()`

`db_find()`

`db_count()`

`db_maxid()`

`db_connect()`

`db_truncate()`

`db_sql_find()`

`db_sql_find_one()`

`db_exec()`

`db_new()`

`db_close()`

DB 配置

DB 配置

在使用 DB 相关函数的时候，需要配置数据库相关信息。一般存放在文件中。

新建配置文件 conf.php：

```
<?php

return array (
    'db' => array (
        'type' => 'pdo_mysql',
        'pdo_mysql' => array (
            'master' => array (
                'host' => 'localhost',
                'user' => 'root',
                'password' => 'root',
                'name' => 'test',
                'tablepre' => 'bbs_',
                'charset' => 'utf8',
                'engine' => 'myisam',
            ),
            'slaves' => array (),
        ),
    ),
);

?>
```

include 配置文件：

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

// 下面就可以使用 db_xxx() 系列函数了，具体请参看文档。
db_create(...);
db_update(...);

?>
```

db_insert()

db_insert()

```
db_insert($table, $arr, $d = NULL)
```

【功能】

插入一条记录，成功返回插入的 ID，失败返回 FALSE。

【参数】

\$table：表名
\$arr：数组
\$d：\$db 实例，一般可以省略，使用配置文件指定的参数实例化的实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arr = array(
    'username'=>'Jack',
    'email'=>'jack@email.com',
    'gid'=>123,
);

$uid = db_insert('user', $arr);

if($uid === FALSE) {
    echo $errstr;
} else {
    echo "创建成功, uid : $uid";
}

?>
```

db_create()

db_create()

```
db_create($table, $arr, $d = NULL)
```

【功能】

创建一条记录，成功返回插入的 ID，失败返回 FALSE。

功能与 db_insert() 等价。

【参数】

\$table：表名
\$arr：数组
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arr = array(
    'username'=>'Jack',
    'email'=>'jack@email.com',
    'gid'=>123,
);

$uid = db_create('user', $arr);

if($uid === FALSE) {
    echo $errstr;
} else {
    echo "创建成功, uid : $uid";
}

?>
```

db_replace()

db_replace()

```
db_replace($table, $arr, $d = NULL)
```

【功能】

替换的方式插入一条记录，成功返回插入的 ID，失败返回 FALSE。

【参数】

\$table：表名
\$arr：数组
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arr = array(
    'username'=>'Jack',
    'email'=>'jack@email.com',
    'gid'=>123,
);

$uid = db_replace('user', $arr);

if($uid === FALSE) {
    echo $errstr;
} else {
    echo "替换创建成功, uid : $uid";
}

?>
```

db_update()

db_update()

```
db_update($table, $cond, $update, $d = NULL)
```

【功能】

更新记录，成功返回受影响的行数，失败返回 FALSE。

【参数】

\$table：表名
\$cond：条件
\$update：更新的数据
\$d：\$db 实例

【\$cond】

条件数组的格式：

```
// 单个条件：
array('gid'=>1)

// 多条：
array('gid'=>1, 'uid'=>3)

// 大于
array('create_date'=>array('>'=>100))

// 小于
array('create_date'=>array('<'=>1000))

// LIKE
array('subject'=>array('LIKE'=>'关键字'))

// 组合：
array('gid'=>1, 'create_date'=>array('>'=>100, '<'=>1000))
```

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

// 将 uid=123 所在的行的 username 字段值更新为 Mike
$r = db_update('user', array('uid'=>123), array('username'=>'Mike'));

if($r === FALSE) {
    echo $errstr;
} else {
    echo "更新成功";
}
```

db_update()

?>

db_delete()

db_delete()

```
db_delete($table, $cond, $d = NULL)
```

【功能】

删除记录，成功返回受影响的行数，失败返回 FALSE。

【参数】

\$table：表名
\$cond：条件
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = db_delete('user', array('uid'=>123));

if($r === FALSE) {
    echo $errstr;
} else {
    echo "删除成功";
}

?>
```

db_find_one()

db_find_one()

【函数原型】

```
db_find_one($table, $cond = array(), $orderby = array(), $col = array(), $d = NULL)
```

【功能】

查询数据库，成功返回一维数组（一条记录），失败返回 FALSE，并且自动记录到错误日志。

\$table：表名
\$cond：条件
\$orderby：排序方式
\$col：查询哪些列
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

// 查找 uid=1 的用户记录，返回关联数组
$arr = db_find_one('user', array('uid'=>123));

print_r($arr);

?>
```

db_find()

db_find()

【函数原型】

```
db_find($table, $cond = array(), $orderby = array(), $page = 1, $pagesize = 10, $key = '', $col = array(), $d = NULL)
```

【功能】

查询数据库，成功返回二维数组（多条记录），失败返回 FALSE，并且自动记录到错误日志。

\$table：表名
\$cond：条件
\$orderby：排序方式
\$page：页数
\$pagesize：每页记录条数
\$key：返回的数组用那一列的值作为 key
\$col：查询哪些列
\$d：\$db 实例

【\$orderby】

条件数组的格式：

```
// 某个字段正序 - ORDER BY create_date ASC :
array('create_date'=>1)

// 某个字段倒序 - ORDER BY create_date DESC :
array('create_date'=>-1)

// 组合：ORDER BY gid ASC, create_date DESC :
array('gid'=>1, 'create_date'=>-1)
```

【用例】

```
<?php
```

```
$conf = include './conf.php';
```

```
include './xiunophp/xiunophp.php';
```

```
// 查找 g=1 的用户记录，返回所有符合的记录
```

```
$arrlist = db_find('user', array('gid'=>1));
```

```
print_r($arrlist);
```

```
// 查找 g=1 && uid > 1 && uid < 100 的用户记录，返回所有符合的记录，并且按照 uid 正序排序（1: 正序；  
-1: 倒序）
```

```
$arrlist = db_find('user', array('gid'=>1, 'uid'=>array('>'=>1, '<'=>100)), array('uid'=>1));
```

```
db_find()
```

```
print_r($arrlist);
```

```
?>
```



db_count()

db_count()

```
db_count($table, $cond = array(), $d = NULL)
```

【功能】

统计满足条件的行数，成功返回总行数，失败返回 FALSE。

【参数】

\$table：表名
\$cond：条件
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$n = db_count('user');

if($n === FALSE) {
    echo $errstr;
} else {
    echo "总行数为：$n";
}

?>
```

db_maxid()

db_maxid()

```
db_maxid($table, $field, $cond = array(), $d = NULL)
```

【功能】

查找表中某列最大值，成功返回最大值，失败返回 FALSE。

【参数】

\$table：表名
\$field：列名
\$cond：条件
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$maxuid = db_maxid('user', 'uid');

if($maxuid === FALSE) {
    echo $errstr;
} else {
    echo "最大 UID 为:$maxuid";
}

?>
```

db_connect()

db_connect()

```
db_connect($d = NULL)
```

【功能】

测试数据库服务器能否联通，成功返回 TRUE，失败返回 FALSE。

【参数】

\$d : \$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = db_connect();

if($r === FALSE) {
    echo $errstr;
} else {
    echo "成功";
}

?>
```

db_truncate()

db_truncate()

```
db_truncate($table, $d = NULL)
```

【功能】

清空表数据，成功返回 TRUE，失败返回 FALSE。

【参数】

\$table：表名
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = db_truncate('user');

if($n === FALSE) {
    echo $errstr;
} else {
    echo "表数据已经清空";
}

?>
```


db_sql_find()

db_sql_find()

```
db_sql_find($sql, $key = NULL, $d = NULL)
```

【功能】

通过 SQL 的方式查询满足条件的记录，成功返回一个关联数组（包含多条记录），失败返回 FALSE。

【参数】

\$sql: SQL 语句，这个和 DB 驱动挂钩，目前仅支持 MYSQL
\$key: 使用指定列的值作为返回数组的 key
\$d: \$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = db_sql_find("SELECT * FROM bbs_user LIMIT 10");

if($arrlist === FALSE) {
    echo $errstr;
} else {
    print_r($arrlist);
}

?>
```

db_sql_find_one()

db_sql_find_one()

```
db_sql_find_one($sql, $d = NULL)
```

【功能】

通过 SQL 的方式查询一行记录，成功返回一个关联数组（包含一条记录），失败返回 FALSE。

【参数】

\$sql：SQL 语句，这个和 DB 驱动挂钩，目前仅支持 MYSQL
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arr = db_sql_find_one("SELECT * FROM bbs_user WHERE uid=1");

if($arr === FALSE) {
    echo $errstr;
} else {
    print_r($arr);
}

?>
```

db_exec()

db_exec()

```
db_exec($sql, $d = NULL)
```

【功能】

执行一条 SQL，成功返回受影响的行数，失败返回 FALSE。

【参数】

\$sql：SQL 语句，这个和 DB 驱动挂钩，目前仅支持 MYSQL
\$d：\$db 实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = db_exec("ALTER TABLE bbs_uset DROP COLUMN username");

if($r === FALSE) {
    echo $errstr;
} else {
    echo "执行成功。";
}

?>
```

db_new()

db_new()

```
db_new($dbconf)
```

【功能】

查找表中某列最大值，成功返回最大值，失败返回 FALSE。
一般不需要调用此函数，除非需要多个 DB 连接。

【参数】

\$dbconf : DB 配置数据

【用例】

```
<?php

$dbconf = array (
    'type' => 'pdo_mysql',
    'pdo_mysql' => array (
        'master' => array (
            'host' => 'localhost',
            'user' => 'root',
            'password' => 'root',
            'name' => 'test',
            'tablepre' => 'bbs_',
            'charset' => 'utf8',
            'engine' => 'innodb',
        ),
        'slaves' => array (),
    ),
);
include './xiunophp/xiunophp.php';

$newdb = db_new($dbconf);
echo $newdb->version();

?>
```

db_close()

db_close()

```
db_close($db = NULL)
```

【功能】

关闭 DB 连接。

【参数】

\$db : DB 实例

【用例】

```
<?php

$dbconf = array (
    'type' => 'pdo_mysql',
    'pdo_mysql' => array (
        'master' => array (
            'host' => 'localhost',
            'user' => 'root',
            'password' => 'root',
            'name' => 'test',
            'tablepre' => 'bbs_',
            'charset' => 'utf8',
            'engine' => 'innodb',
        ),
        'slaves' => array (),
    ),
);
include './xiunophp/xiunophp.php';

$newdb = db_new($dbconf);
db_close($newdb);

?>
```

缓存函数

CACHE 配置

cache_set()

cache_get()

cache_delete()

cache_truncate()

cache_new()

CACHE 配置

DB 配置

在使用 DB 相关函数的时候，需要配置数据库相关信息。一般存放在文件中。

新建配置文件 conf.php：

```
<?php
return array (
    'cache' => array (
        'enable' => true,
        'type' => 'xcache',
        'memcached' => array (
            'host' => 'localhost',
            'port' => '11211',
            'cachepre' => 'bbs_',
        ),
        'redis' => array (
            'host' => 'localhost',
            'port' => '6379',
            'cachepre' => 'bbs_',
        ),
        'xcache' => array (
            'cachepre' => 'bbs_',
        ),
        'yac' => array (
            'cachepre' => 'bbs_',
        ),
        'apc' => array (
            'cachepre' => 'bbs_',
        ),
        'mysql' => array (
            'cachepre' => 'bbs_',
        ),
    ),
);
?>
```

如果用 MySQL 来存储数据，得自己建立表：

```
# 缓存表，用来保存临时数据。
DROP TABLE IF EXISTS bbs_cache;
CREATE TABLE bbs_cache (
    k char(32) NOT NULL default '',
    v mediumtext NOT NULL,
    expiry int(11) unsigned NOT NULL default '0',      # 过期时间
    PRIMARY KEY(k)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

include 配置文件：

```
<?php
```

```
$conf = include './conf.php';  
include './xiunophp/xiunophp.php';  
  
// 下面就可以使用 cache_xxx() 系列函数了，具体请参看文档。  
cache_set('key1', 'value1');  
cache_get('key1');  
  
?>
```


cache_set()

cache_set()

```
cache_set($k, $v, $life = 0, $c = NULL)
```

【功能】

设置一条值，TRUE，失败返回 FALSE。

【参数】

\$k：键
\$v：值
\$life：生命周期，0 表示永不过期
\$c：\$cache 实例，一般可以省略，使用配置文件指定的参数实例化的实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = cache_set('key1', 'value1');

if($r === FALSE) {
    echo $errstr;
} else {
    echo "设置成功";
}

?>
```

cache_get()

cache_get()

```
cache_get($k, $c = NULL)
```

【功能】

获取键值，成功返回设置的值，失败返回 FALSE。

【参数】

\$k：键

\$c：\$cache 实例，一般可以省略，使用配置文件指定的参数实例化的实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

cache_set('key1', 'value1');

$v = cache_get('key1');

echo $v;

?>
```

cache_delete()

cache_delete()

```
cache_delete($k, $c = NULL)
```

【功能】

删除一条键值，成功返回 TRUE，失败返回 FALSE。

【参数】

\$k：键

\$c：\$cache 实例，一般可以省略，使用配置文件指定的参数实例化的实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

cache_set('key1', 'value1');

$r = cache_delete('key1');

if($r === FALSE) {
    echo $errstr:
} else {
    echo "删除成功":
}

?>
```

cache_truncate()

cache_truncate()

```
cache_truncate($c = NULL)
```

【功能】

清空 Cache , 成功 TRUE , 失败返回 FALSE。

【参数】

`$c` : `$cache` 实例, 一般可以省略, 使用配置文件指定的参数实例化的实例

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$r = cache_truncate();

if($r === FALSE) {
    echo $errstr:
} else {
    echo "清空成功":
}

?>
```

cache_new()

cache_new()

```
cache_new($cacheconf)
```

【功能】

根据指定的配置参数，实例化一个 Cache 类。
一般不需要调用此函数，除非需要多个 Cache 实例。

【参数】

`$cacheconf` : Cache 配置数据

【用例】

```
<?php

$cacheconf = array (
    'enable' => true,
    'type' => 'xcache',
    'xcache' => array (
        'cachepre' => 'pre_',
    ),
);
include './xiunophp/xiunophp.php';

$newcache = cache_new($cacheconf);

$newcache->set('key1', 'value1');

?>
```

数组增强

array_value()
array_filter_empty()
array_addslashes()
array_stripslashes()
array_htmlspecialchars()
array_trim()
array_diff_value()
array_assoc_slice()
arrlist_multisort()
arrlist_cond_orderby()
arrlist_key_values()
arrlist_values()
arrlist_change_key()
arrlist_chunk()

array_value()

array_value()

```
array_value($arr, $key, $default = '')
```

【功能】

以无 Notice 的方式返回数组中指定的键对应的值。

【参数】

\$arr：数组
\$key：键名
\$default： 在不存在的时候，默认值

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
echo array_value($_SERVER, 'REMOTE_ADDR');  
  
?>
```

array_filter_empty()

array_filter_empty()

```
array_filter_empty($arr)
```

【功能】

过滤数组中值为空的元素。

【参数】

\$arr : 数组

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$arr = array(0, 1, 2, 3, '', FALSE);
$r = array_filter_empty($arr);
print_r($r);

/* 结果 :

array(1, 2, 3)

*/

?>
```


array_addslashes()

array_addslashes()

```
array_addslashes(&$var)
```

【功能】

对数组的每一个元素应用 addslashes() 函数。

array stripslashes()

array stripslashes()

```
array stripslashes(&$var)
```

【功能】

对数组的每一个元素应用 stripslashes() 函数。

array_htmlspecialchars()

array_htmlspecialchars()

```
array_htmlspecialchars(&$var)
```

【功能】

对数组的每一个元素应用 htmlspecialchars() 函数。

array_trim()

array_trim()

```
array_trim(&$var)
```

【功能】

对数组的每一个元素应用 trim() 函数。

array_diff_value()

array_diff_value()

```
array_diff_value($arr1, $arr2)
```

【功能】

比较数组的值，如果不相同则保留，以第一个数组为准。

与 array_diff() 不同在于，支持关联数组。

【参数】

\$arr1：数组1，基准数组
\$arr2：数组2，比较的数组

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$arr1 = array('id'=>1, 'name'=>'name1');
$arr2 = array('id'=>1, 'name'=>'name2');
$arr = array_diff_value($arr1, $arr2);

/* 结果：

array('name'=>'name1')

*/

?>
```

array_assoc_slice()

array_assoc_slice()

```
array_assoc_slice($arrlist, $start, $length = 0)
```

【功能】

array_slice() 的关联数组版本。

【参数】

\$arrlist: 二维数组
\$start: 开始位置
\$length: 取多少条

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4);

// 查找 a=1, 结果集按照 b 倒序, 取第一页数据, 每页 20 条
$arrlist = array_assoc_slice($arrlist, 0, 1);
print_r($arrlist);

/* 结果 :

array('a'=>1);

*/

?>
```

arrlist_multisort()

arrlist_multisort()

```
arrlist_multisort($arrlist, $col, $asc = TRUE)
```

【功能】

对二维的关联数组进行排序。

一般应用在对从数据库中取出来的二维数组进行排序。

【参数】

\$arrlist：二维关联数组

\$col：排序的列

\$asc：是否为升序，TRUE：升序；FALSE：倒序

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$data = array();
$data[] = array('volume' => 67, 'edition' => 2);
$data[] = array('volume' => 86, 'edition' => 1);
$data[] = array('volume' => 85, 'edition' => 6);
$data[] = array('volume' => 98, 'edition' => 2);
$data[] = array('volume' => 86, 'edition' => 6);
$data[] = array('volume' => 67, 'edition' => 7);

$r = arrlist_multisort($data, 'edition', TRUE);

print_r($r);

/* 结果：
Array
(
    [0] => Array
        (
            [volume] => 86
            [edition] => 1
        )

    [1] => Array
        (
            [volume] => 67
            [edition] => 2
        )

    [2] => Array
        (
            [volume] => 98
            [edition] => 2
        )
)
```

```
[3] => Array
(
    [volume] => 85
    [edition] => 6
)

[4] => Array
(
    [volume] => 86
    [edition] => 6
)

[5] => Array
(
    [volume] => 67
    [edition] => 7
)

)
*/

?>

?>
```


arrlist_cond_orderby()

arrlist_multisort()

```
arrlist_cond_orderby($arrlist, $cond = array(), $orderby = array(), $page = 1, $pagesize = 20)
```

【功能】

对二维关联数组进行查找，排序，筛选，只支持一种条件排序。

【参数】

\$arrlist: 二维数组
\$cond: 条件
\$orderby: 排序条件
\$page: 页码
\$pagesize: 每页数量

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array(
    array('a'=>1, 'b'=>1, 'c'=>1),
    array('a'=>2, 'b'=>2, 'c'=>2),
    array('a'=>3, 'b'=>3, 'c'=>3),
);

// 查找 a=1, 结果集按照 b 倒序, 取第一页数据, 每页 20 条
$arrlist = arrlist_cond_orderby($arrlist, array('a'=>1), array('b'=>-1), 1, 20);
print_r($arrlist);

?>
```

arrlist_key_values()

arrlist_key_values()

```
arrlist_key_values($arrlist, $key, $value = NULL)
```

【功能】

以指定的 \$key \$value , 从二维关联数组中整理一个一维关联数组。

【参数】

\$arrlist: 二维数组
\$key: 键名
\$value: 键值

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array(
    array('id'=>1, 'name'=>'Jack'),
    array('id'=>2, 'name'=>'Rose'),
);

$arrlist = arrlist_key_values($arrlist, 'name', 'id');
print_r($arrlist);

/* 结果 :

array(
    'Jack'=>1,
    'Rose'=>2,
);

*/

?>
```

arrlist_values()

arrlist_values()

```
arrlist_values($arrlist, $key)
```

【功能】

以指定的 \$key , 从二维关联数组中获取一个自然数字索引的数组。

【参数】

\$arrlist: 二维数组
\$key: 键名
\$value: 键值

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array(
    array('id'=>1, 'name'=>'Jack'),
    array('id'=>2, 'name'=>'Rose'),
);

$arr = arrlist_values($arrlist, 'name');
print_r($arr);

/* 结果 :

array('Jack', 'Rose');

*/

?>
```

arrlist_change_key()

arrlist_change_key()

```
arrlist_change_key(&$arrlist, $key, $pre = '')
```

【功能】

改变二维数组的 key。

【参数】

\$arrlist: 二维数组
\$key: 键名
\$pre: 改变后的前缀, 默认为空

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array(
    0 => array('uid'=>1, 'username'=>'username1'),
    1 => array('uid'=>2, 'username'=>'username2'),
    2 => array('uid'=>3, 'username'=>'username3'),
);
arrlist_change_key($arrlist, 'uid');

print_r($arrlist);

/* 结果 :

array (
    1 => array('uid'=>1, 'username'=>'username1'),
    2 => array('uid'=>2, 'username'=>'username2'),
    3 => array('uid'=>3, 'username'=>'username3'),
);

*/

?>
```

arrlist_chunk()

arrlist_chunk()

```
arrlist_chunk($arrlist, $key)
```

【功能】

以二维数组的某一列的值作为键，对元素进行归类，重新整理出一个数组。

【参数】

\$arrlist: 二维数组
\$key: 键名

【用例】

```
<?php

$conf = include './conf.php';
include './xiunophp/xiunophp.php';

$arrlist = array(array('a'=>1, 'b'=>2), array('a'=>3, 'b'=>4));
$r = arrlist_chunk($arrlist, 'b');
print_r($r);

/* 结果：

Array
(
    [2] => Array
        (
            [0] => Array
                (
                    [a] => 1
                    [b] => 2
                )
        )
    [4] => Array
        (
            [0] => Array
                (
                    [a] => 3
                    [b] => 4
                )
        )
)

*/

?>
```


杂项函数

xn_strlen()
xn_substr()
xn_urlencode()
xn_urldecode()
xn_json_encode()
xn_json_decode()
xn_encrypt()
xn_decrypt()
xn_message()
xn_error()
xn_log()
xn_txt_to_html()
xn_rand()
xn_is_writable()
humandate()
humannumber()
humansize()
param()
lang()
url()
pagination()
is_robot()
http_get()
http_post()
https_get()
https_post()
http_multi_get()
file_replace_var()
file_get_contents_try()
file_put_contents_try()
in_string()
file_ext()
file_pre()
file_name()
http_url_path()
glob_recursive()
rmdir_recursive()

`copy_recursive()`

`_GET() _POST() _COOKIE() ...`

xn_strlen()

xn_strlen()

```
xn_strlen($s)
```

【功能】

UTF-8 的编码方式求字符串长度。

函数原型：

```
function xn_strlen($s) {  
    return mb_strlen($s, 'UTF-8');  
}
```

【参数】

\$s：字符串

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
echo xn_strlen('字符');  
  
// 结果： 2  
  
?>
```

xn_substr()

xn_substr()

```
xn_substr($s, $start, $len)
```

【功能】

UTF-8 的编码方式截取字符串。

函数原型：

```
function xn_substr($s, $start, $len) {  
    return mb_substr($s, $start, $len, 'UTF-8');  
}
```

【参数】

\$s：字符串
\$start：开始位置
\$len：截取长度（可以为负数，表示从后往前数）

【用例】

```
<?php  
  
include './xiunophr/xiunophr.php';  
  
echo xn_substr('字符', 0, 1);  
  
// 结果： 字  
  
?>
```

xn_urlencode()

xn_urlencode()

```
xn_urlencode($s)
```

【功能】

对字符串进行编码 URL 字符串，结果可以用户 URL 传递，比较方便。

【函数原型】

```
function xn_urlencode($s) {  
    $s = str_replace('-', '_2d', $s);  
    $s = str_replace('.', '_2e', $s);  
    $s = str_replace('+', '_2b', $s);  
    $s = str_replace('=', '_3d', $s);  
    $s = urlencode($s);  
    $s = str_replace('%', '_', $s);  
    return $s;  
}
```

【参数】

\$s：字符串

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$r = xn_urlencode('中国');  
print_r($r);  
  
// 结果： _E4_B8_AD_E5_9B_BD  
  
?>
```

xn_urldecode()

xn_urlencode()

```
xn_urlencode($s)
```

【功能】

解码 xn_urlencode() 编码的字符串。

【函数原型】

```
function xn_urldecode($s) {  
    $s = str_replace('_', '%', $s);  
    $s = urldecode($s);  
    return $s;  
}
```

【参数】

\$s : 字符串

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$r = xn_urlencode('_E4_B8_AD_E5_9B_BD');  
echo $r;  
  
//结果： 中国  
  
?>
```

xn_json_encode()

xn_json_encode()

```
xn_json_encode($arg)
```

【功能】

对变量进行 JSON 编码，不对中文进行 UNICODE 转义。

【参数】

\$s：字符串

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$arr = array ('name'=>'中文');  
$s = xn_json_encode($arr);  
  
echo $s;  
  
// 结果： {"name": "中文"}  
  
?>
```

xn_json_decode()

xn_json_decode()

```
xn_json_decode($json)
```

【功能】

对 JSON 格式的字符串进行解码，返回数组（非对象）。

【参数】

\$json : JSON 格式的字符串

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$json = '{"name":"中文"}';
$r = xn_json_decode($json);
print_r($r);

/* 结果：

Array(
    'name'=>'中文',
)

*/

?>
```

xn_encrypt()

xn_encrypt()

```
xn_encrypt($txt, $key = '')
```

【功能】

对字符串的可逆加密函数。根据 \$key 随机生成，结果可以直接 URL 传递。

此函数有 C 语言实现并且有所增强的 PHP 扩展版本(xiuno.so)，比 PHP 速度快 200+ 倍，对安全性和速度要求较高的情况下可以使用，可以定制 key，将 key 存放于 .so（内存）中（key 明文存放于文件不够安全），可以绑定 cpuid，防止 so 被拷贝后运行，有需要的请与我联系：axiuno@gmail.com

【参数】

\$txt：需要加密的字符串

【用例】

```
<?php
include './xiunophp/xiunophp.php';

$s = "1\n21232f297a57a5a743894a0e4a801fc3\n127.0.0.1";

$s2 = xn_encrypt($s, 'KSLEKFJ2K2KK');

print_r($s2);

// 结果： nUg0N_2F1sxjgARMREdcpf0NmDLiEL79QiyD9rSKLqr8wk1bn7_2bdreFHInHGrdqU_2F

?>
```

xn_decrypt()

xn_decrypt()

```
xn_decrypt($txt, $key = '')
```

【功能】

解密由 xn_encrypt() 加密的字符串。

【参数】

\$txt：需要解密的字符串
\$key：混淆的字符串

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$s = "nUg0N_2F1sxjgARMREdcpfD0NmDLiEL79QiyD9rSKLqr8wk1bn7_2bdreFHInHGrdqU_2F";
$s2 = xn_decrypt($s, 'KSLEKFJ2K2KK');

print_r($s2);

// 结果： 1 21232f297a57a5a743894a0e4a801fc3 127.0.0.1

?>
```


xn_message()

xn_message()

```
xn_message($code, $message)
```

【功能】

根据请求的方式，输出不同格式的内容，并且终止会话。

如果为 AJAX 请求，则输出 json 格式数据：{code: \$code, message: \$message}。

如果为普通请求，则输出 \$message。

【原型】

```
function xn_message($code, $message) {  
    global $ajax;  
    echo $ajax ? xn_json_encode(array('code'=>$code, 'message'=>$message)) : $message;  
    exit;  
}
```

【参数】

\$code：代码
\$message：内容

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
xn_message(0, 'OK');  
  
// 结果： {code: 0, message: "OK"}  
  
?>
```

xn_error()

xn_error()

```
xn_error($no, $str, $return = FALSE)
```

【功能】

对全局变量 \$errno, \$errstr 进行设置，并且默认返回 FALSE。

【原型】

```
function xn_error($no, $str, $return = FALSE) {  
    global $errno, $errstr;  
    $errno = $no;  
    $errstr = $str;  
    return $return;  
}
```

【参数】

\$no：错误码
\$str：错误字符串
\$return：返回值

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
function check_email($email) {  
    if(empty($email)) {  
        return xn_error(-1, 'Email 为空');  
    }  
}  
  
my_func('') OR echo $errstr;  
  
?>
```

xn_log()

xn_log()

```
xn_log($s, $file = 'error')
```

【功能】

记录信息到日志。

默认存到 `$conf['log_path']`，一天一个目录。

【参数】

`$s`：记录到日志的信息
`$file`：记录的文件名

【用例】

```
<?php  
include './xiunophp/xiunophp.php';  
xn_log("the log");  
?>
```

xn_txt_to_html()

xn_txt_to_html()

```
xn_txt_to_html($s)
```

【功能】

将文本格式转换为 HTML 格式。

【参数】

\$s：文本数据

【用例】

```
<?php
include './xiunophp/xiunophp.php';

$s = 'TEXT 文本数据：\r\n 内容';
$s2 = xn_txt_to_html($s);

echo $s2;

// 结果：TEXT 文本数据：<br>&nbsp;内容

?>
```

xn_rand()

xn_rand()

```
xn_rand($n = 16)
```

【功能】

产生随机字符串。

【参数】

\$n：位数

【用例】

```
<?php
include './xiunophp/xiunophp.php';

$s = xn_rand(16);

echo $s;

// 结果：kd3igi2a7we9gkj3

?>
```

xn_is_writable()

xn_is_writable()

```
xn_is_writable($file)
```

【功能】

检测文件或目录是否可写，兼容 Windows。

【参数】

\$file：文件/目录路径

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$file = './log/';  
$r = xn_is_writable($file);  
  
// 结果： TRUE/FALSE  
  
?>
```

humandate()

humandate()

```
humandate($timestamp, $lan = array())
```

【功能】

友好的显示日期。

【参数】

\$timestamp : UNIX 时间戳
\$lan : 语言包

语言包默认值：：

```
$lan = array(  
    'month_ago'=>'月前',  
    'day_ago'=>'天前',  
    'hour_ago'=>'月前',  
    'minute_ago'=>'月前',  
    'second_ago'=>'月前',  
);
```

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$n = time() - 86401;  
echo humandate($n);  
  
// 结果： 1 天前  
  
?>
```

humannumber()

humannumber()

```
humannumber($num)
```

【功能】

友好的数字。

【参数】

\$num：数值

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo humannumber(20000);

// 结果：2万

?>
```


humansize()

humansize()

```
humansize($num)
```

【功能】

友好的尺寸。

【参数】

\$num：数值

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo humansize(20000);

// 结果：20K

?>
```

param()

param()

param()

```
param($key, $defval = '', $htmlspecialchars = TRUE, $addslashes = FALSE)
```

【功能】

获取 GET POST COOKIE REQUEST 参数

注意在 4.0 正式版，addslashes() 默认是关闭的。

【参数】

\$key：键名
\$defval：键名不存在的时候的默认值，注意类型会被强制转换，支持字符串（默认），数字，一位数组
\$htmlspecialchars：是否安全处理：htmlspecialchars()
\$addslashes：是否安全处理：addslashes()

【用例】

```
<?php

// http://domain.com/user-list-123.htm

include './xiunophp/xiunophp.php';

echo param(0);      // user
echo param(1);      // list
echo param(2, 0);   // 123

// 结果： 1 天前

?>
```

lang()

lang()

```
lang($key, $arr = array())
```

【功能】

从语言包中获取值。

【参数】

\$key：键名
\$arr：变量

【用例】

```
<?php

$lang = array(
    'name'=>'名字',
    'user_login_successfully'=>'用户 {username} 登陆成功！'
);

include './xiunophp/xiunophp.php';

echo lang('name');
// 结果：名字

echo lang('user_login_successfully', array('username'=>'张三'));
// 结果：用户 张三 登陆成功！

?>
```

url()

url()

url()

```
url($url, $extra = array())
```

【功能】

根据 `$conf['url_rewrite_on']` 指定的格式生成 URL。

【参数】

`$url` : URL
`$extra` : 附加参数

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo url('user-login');

/* 结果 :
   ?user-login.htm      ( 当 $conf['url_rewrite_on'] = 0 )
   user-login.htm      ( 当 $conf['url_rewrite_on'] = 1 )
   ?/user/login        ( 当 $conf['url_rewrite_on'] = 2 )
   /user/login         ( 当 $conf['url_rewrite_on'] = 3 )
*/

?>
```

pagination()

pagination()

```
pagination($url, $totalnum, $page, $pagesize = 20)
```

【功能】

翻页函数，生成翻页的 HTML 代码字符串。

【参数】

\$url：需要生成的 URL 模板，包含 {page}，会被替换。
\$totalnum：总记录数
\$page：页码
\$pagesize：每页条数

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo pagination("user-list-{page}.htm", 30, 1, $10);

/* 结果：
    <li class="page-item{active}"><a href="user-list-1.htm" class="page-link">1</a></li>
    >
    <li class="page-item{active}"><a href="user-list-1.htm" class="page-link">2</a></li>
    >
    <li class="page-item{active}"><a href="user-list-1.htm" class="page-link">3</a></li>
    >
    */
?>
```

is_robot()

is_robot()

is_robot()

```
is_robot()
```

【功能】

判断是否为机器人

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo is_robot();

// 结果：TRUE/FALSE

?>
```

http_get()

http_get()

```
http_get($url, $timeout = 5, $times = 3)
```

【功能】

通过一个 HTTP GET 获取内容。

【参数】

\$url：完整的网址
\$timeout：超时时间，单位为秒
\$times：重试次数

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
echo http_get("http://bbs.xiuno.com/");  
  
/* 结果：  
  
<!DOCTYPE html>...  
  
*/  
  
?>
```

http_post()

http_post()

```
http_post($url, $post = '', $cookie='', $timeout = 10, $times = 3)
```

【功能】

通过一个 HTTP POST 请求获取内容。

【参数】

\$url : 完整的网址
\$post : POST 内容, 格式 : username=Jack&password=123465
\$cookie : COOKIE 数据, 格式 : username=Jack&password=123456
\$timeout : 超时时间, 单位为秒
\$times : 重试次数

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
echo http_postt("http://bbs.xiuno.com/user-login.htm", "username=admin&password=123456"  
);  
  
/* 结果 :  
  
WEB 服务器返回的信息  
  
*/  
  
?>
```


https_get()

https_get()

```
https_get($url, $cookie = '', $timeout = 5)
```

【功能】

通过一个 HTTPS GET 获取内容。

【参数】

\$url：完整的网址
\$timeout：超时时间
\$times：重试次数

【用例】

```
<?php  
include './xiunophp/xiunophp.php';  
echo https_get("http://bbs.xiuno.com/");  
  
/* 结果：  
  
<!DOCTYPE html>...  
  
*/  
  
?>
```

https_post()

https_post()

```
https_post($url, $post = '', $cookie = '', $timeout=30)
```

【功能】

通过一个 HTTPS POST 请求获取内容。

【参数】

\$url : 完整的网址
\$timeout : 超时时间
\$post : POST 内容, 格式: username=Jack&password=123465
\$cookie : Cookie 数据, 格式: sid=123&token=123

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
echo http_post("http://bbs.xiuno.com/user-login.htm", "username=admin&password=123456",  
    30, "username=zhangsan&password=123456", "bbs_sid=123456");  
  
/* 结果 :  
  
WEB 服务器返回的信息  
  
*/  
  
?>
```

http_multi_get()

http_multi_get()

```
http_multi_get($urls)
```

【功能】

多线程的方式通过一个 HTTP GET 获取内容。
一般在命令行下执行，做抓取比较有用。

【参数】

\$urls：多个网址

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$urlarr = array(
    "http://bbs.xiuno.com/thread-1.htm",
    "http://bbs.xiuno.com/thread-2.htm",
    "http://bbs.xiuno.com/thread-3.htm",
);
echo http_multi_get($urlarr);

/* 结果：

array(
    0=>'<!DOCTYPE html>...',
    1=>'<!DOCTYPE html>...',
    2=>'<!DOCTYPE html>...',
);

*/

?>
```

file_replace_var()

file_replace_var()

```
file_replace_var($filepath, $replace = array(), $pretty = FALSE)
```

【功能】

替换文件内容中的变量内容，支持 PHP JSON 格式的文件。

【参数】

\$filepath：文件路径
\$replace：替换的内容
\$pretty：是否保持优美格式

【用例】

文件 1.conf

```
return array(  
    'key1'=>'value1',  
    'key2'=>'value2',  
);
```

文件 1.json

```
{  
    'key1': "value1",  
    'key2': "value2",  
}
```

```
<?php  
include './xiunophp/xiunophp.php';  
file_replace_var('1.php', array('key1'=>'aaa'));  
file_replace_var('1.json', array('key1'=>'aaa'));  
/*  
1.php 结果：  
return array(  
    'key1'=>'aaa',  
    'key2'=>'value2',  
);  
1.json 结果：  
{
```

file_replace_var()

```
'key1': "aaa",  
'key2': "value2",  
}  
*/  
?>
```

file_get_contents_try()

file_get_contents_try()

```
file_get_contents_try($file, $times = 3)
```

【功能】

获取文件的内容，在并发的环境中容易读取失败，会自动重试。

【参数】

\$file：文件路径
\$times：重试次数

【用例】

```
<?php  
include './xiunophp/xiunophp.php';  
$s = file_get_contents_try('1.txt');  
echo $s;  
?>
```

file_put_contents_try()

file_put_contents_try()

```
file_put_contents_try($file, $s, $times = 3)
```

【功能】

将谁写入文件，在并发的环境中会锁定文件，会自动重试。

【参数】

\$file：文件路径
\$s：数据
\$times：重试次数

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$r = file_putt_contents_try("data", '1.txt');  
  
echo $r;  
  
// 结果：TRUE/FALSE  
  
?>
```

in_string()

in_string()

```
in_string($s, $str)
```

【功能】

判断一个字符串是否在另外一个字符串中。

默认分割符为英文半角逗号： ,

【参数】

\$s：小字符串
\$str：大字符串

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$r = in_string('ab', 'ab,abc,abcd')  
  
// 结果：TRUE/FALSE  
  
?>
```


file_ext()

file_ext()

```
file_ext($filename, $max = 16)
```

【功能】

获取文件的后缀，不包含.，可以设置最大长度。

【参数】

\$filename：文件名
\$max：最大长度（为了安全）

【用例】

```
<?php  
include './xiunophp/xiunophp.php';  
echo file_ext("abc.jpg.php");  
  
// 结果：php  
?>
```

file_pre()

file_pre()

```
file_pre($filename, $max = 32)
```

【功能】

获取文件名的前面一部分，不包含.，可以设置最大长度。

【参数】

\$filename：文件名
\$max：最大长度（为了安全）

【用例】

```
<?php  
include './xiunophp/xiunophp.php';  
echo file_pre("abc.jpg.php");  
// 结果：abc.jpg  
?>
```

file_name()

file_name()

```
file_name($path)
```

【功能】

从路径中获取文件名

【参数】

\$path : 文件路径

【用例】

```
<?php
include './xiunophp/xiunophp.php';
echo file_name("http://bbs.xiuno.com/view/img/logo.png");
// 结果：logo.png
?>
```

http_url_path()

http_url_path()

```
http_url_path()
```

【功能】

获取当前的 URL 路径（不包含文件名和参数）

【用例】

```
<?php
// URL : http://bbs.xiuno.com/admin/forum-list.htm#123
include './xiunophp/xiunophp.php';
echo http_url_path();
// 结果 : http://bbs.xiuno.com/admin/
?>
```

glob_recursive()

glob_recursive()

```
glob_recursive($pattern, $flags = 0)
```

【功能】

glob() 的递归版本，支持无限级子目录遍历。

【参数】

\$pattern：匹配表达式，支持通配符，与 glob() 参数一致
\$flags：标志，与 glob() 参数一致

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
glob_recursive('./*');  
  
// 结果：  
array(  
    './index.php',  
    './admin'  
    './admin/index.php'  
    ...  
);  
  
?>
```

rmdir_recursive()

rmdir_recursive()

```
rmdir_recursive($dir, $keepdir = 0)
```

【功能】

rmdir() 的递归增强版本，支持无限级子目录删除。

rmdir() 只能删除空目录

【参数】

\$dir：目录名

\$keepdir：是否保留最上一级目录（留一个空壳，用来清空目录下文件）

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
$r = rmdir_recursive("./tmp/", 1);  
  
// 结果：TRUE/FALSE  
  
?>
```

copy_recursive()

copy_recursive()

```
copy_recursive($src, $dst)
```

【功能】

copy() 的递归增强版本，支持无限级子目录拷贝。

copy() 只能拷贝文件

【参数】

\$src：源目录
\$dst：目标目录

【用例】

```
<?php  
  
include './xiunophp/xiunophp.php';  
  
copy_recursive('./plugin/xn_ad/', './plugin/xn_ad_backup/');  
  
// 结果：TRUE/FALSE  
  
?>
```

_GET() _POST() _COOKIE() ...

_GET() _POST() _COOKIE() ...

_GET() _POST() _COOKIE() ...

【功能】

无 Notice 方式的获取超级全局变量中的键值。

【函数原型】

```
// 无 Notice 方式的获取超级全局变量中的 key
function _GET($k) { return isset($_GET[$k]) ? $_GET[$k] : NULL; }
function _POST($k) { return isset($_POST[$k]) ? $_POST[$k] : NULL; }
function _COOKIE($k) { return isset($_COOKIE[$k]) ? $_COOKIE[$k] : NULL; }
function _REQUEST($k) { return isset($_REQUEST[$k]) ? $_REQUEST[$k] : NULL; }
function _ENV($k) { return isset($_ENV[$k]) ? $_ENV[$k] : NULL; }
function _SERVER($k) { return isset($_SERVER[$k]) ? $_SERVER[$k] : NULL; }
function GLOBALS($k) { return isset($GLOBALS[$k]) ? $GLOBALS[$k] : NULL; }
function G($k) { return isset($GLOBALS[$k]) ? $GLOBALS[$k] : NULL; }
function _SESSION($k, $v = FALSE) {
    global $g_session;
    return isset($_SESSION[$k]) ? $_SESSION[$k] : (isset($g_session[$k]) ? $g_session[$k] : NULL);
}
```

【用例】

```
<?php

include './xiunophp/xiunophp.php';

$uid = _SESSION('uid');
$sid = _COOKIE('sid');

// 结果：没有 Notice

?>
```