

1. cal_from_jd

`cal_from_jd (int $jd , int $calendar) : array`

jd วันจูเลียนเป็นจำนวนเต็ม

calendar ปฏิทินที่จะแปลง

cal_from_jd() แปลงวัน Julian ที่กำหนดใน jd เป็นวันที่ของปฏิทินที่ระบุ ค่าปฏิทินที่รองรับคือ CAL_GREGORIAN, CAL_JULIAN, CAL_JEWISH และ CAL_FRENCH

ตัวอย่าง แปลงจากจำนวน Julian Day เป็นปฏิทินที่รองรับ

```
<?php
$today = unixtojd(mktime(0, 0, 0, 8, 16, 2003));
print_r(cal_from_jd($today, CAL_GREGORIAN));
?>
```

2. apc_add

`apc_add (string $key , mixed $var [, int $ttl = 0]) : bool`

`apc_add (array $values [, mixed $unused = NULL [, int $ttl = 0]]) : array`

key เก็บตัวแปรโดยใช้ชื่อนี้ keys เป็นแคชที่ไม่ซ้ำกันเพื่อพยายามที่จะใช้ apc_add() FALSE ในการจัดเก็บข้อมูลที่มีสำคัญที่มีอยู่แล้วจะไม่เขียนทับข้อมูลที่มีอยู่และจะกลับมาแทน (นี่เป็นข้อแตกต่างระหว่าง apc_add() และ [apc_store\(\)](#) เท่านั้น)

var ตัวแปรที่จะเก็บ

ttl ใช้เวลาอยู่ เก็บ var ในแคชเป็น ttl เวลาวินาที หลังจากที่ ttl ผ่านไปตัวแปรที่เก็บไว้จะถูกลบออกจากแคช (ในคำขอถัดไป) หากไม่มีการ ttl ระบุ (หรือถ้า ttl เป็น 0) ค่าจะยังคงอยู่จนกว่าจะถูกลบออกจากแคชด้วยตนเองหรือมิฉะนั้นจะไม่มีอยู่ในแคช (ลบล้างรีสตาร์ทเป็นต้น)

values ชื่อในคีย์ ตัวแปรในค่า

ตัวอย่าง apc_add ()

```
<?php
$bar = 'BAR';
apc_add('foo', $bar);
var_dump(apc_fetch('foo'));
echo "\n";
$bar = 'NEVER GETS SET';
apc_add('foo', $bar);
var_dump(apc_fetch('foo'));
echo "\n";
?>
```

3. blenc_encrypt

`blenc_encrypt (string $plaintext , string $encodedfile [, string $encryption_key]) : string`

plaintext รหัสที่มาสำหรับการเข้ารหัส ไม่จำเป็นต้องมีแท็ก PHP เปิด / ปิด

encodedfile ชื่อไฟล์ที่ BLENC จะบันทึกแหล่งที่เข้ารหัส

encryption_key กุญแจสำคัญที่ BLENC จะใช้เพื่อเข้ารหัสเนื้อหาธรรมดา หากไม่ได้รับ BLENC จะสร้างรหัสที่ถูกตัด

ตัวอย่าง `blenc_encrypt ()`

```
<?php
/* read the PHP source code */
$source_code = file_get_contents("my_source_to_protect.php");

/* create the encrypted version */
$redistributable_key = blenc_encrypt($source_code, "my_source_encoded.php"
);

/* read which is the key_file */
$key_file = ini_get('blenc.key_file');

/* save the redistributable key */
file_put_contents($key_file, $redistributable_key, FILE_APPEND);
?>
```

4. apd_continue

`apd_continue (int $debug_level) : bool`

debug_level จำนวนเต็มที่เกิดขึ้นจากการบวกค่าคงที่XXX_TRACEเข้าด้วยกัน

MEMORY_TRACEมันจะไม่แนะนำให้ใช้ มันช้ามากและดูเหมือนจะไม่ถูกต้อง ASSIGNMENT_TRACEยังไม่ได้ใช้งาน

หากต้องการเปิดการติดตามการทำงานทั้งหมด (TIMING, FUNCTIONS, ARGS SUMMARY (เช่น strace -c)) ให้ใช้ค่า 99

ตัวอย่าง `apd_continue`

```
<?php
apd_continue(0);
?>
```

5. rename_function

`rename_function (สตริง $original_name , สตริง $new_name) : bool`

เปลี่ยนชื่อ orig_name เป็น new_name ในตารางฟังก์ชันทั่วโลก มีประโยชน์สำหรับการแทนที่ฟังก์ชันในตัวชั่วคราว

original_name ชื่อฟังก์ชันเดิม

new_name ชื่อใหม่สำหรับoriginal_nameฟังก์ชัน

ตัวอย่าง rename_function

```
<?php
rename_function('mysql_connect', 'debug_mysql_connect' );
?>
```