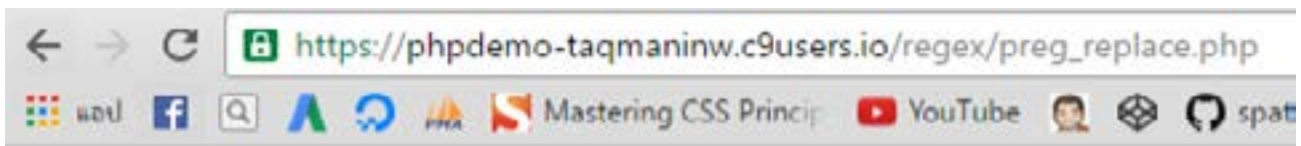


preg_replace ใช้งาน regular expression ในการค้นหาและแก้ไข

ตัวอย่าง

```
<?php
$copy_date = "Copyright 1999";
$copy_date = preg_replace("[0-9]+", "2000", $copy_date);
print $copy_date;
?>
```

ผลลัพธ์



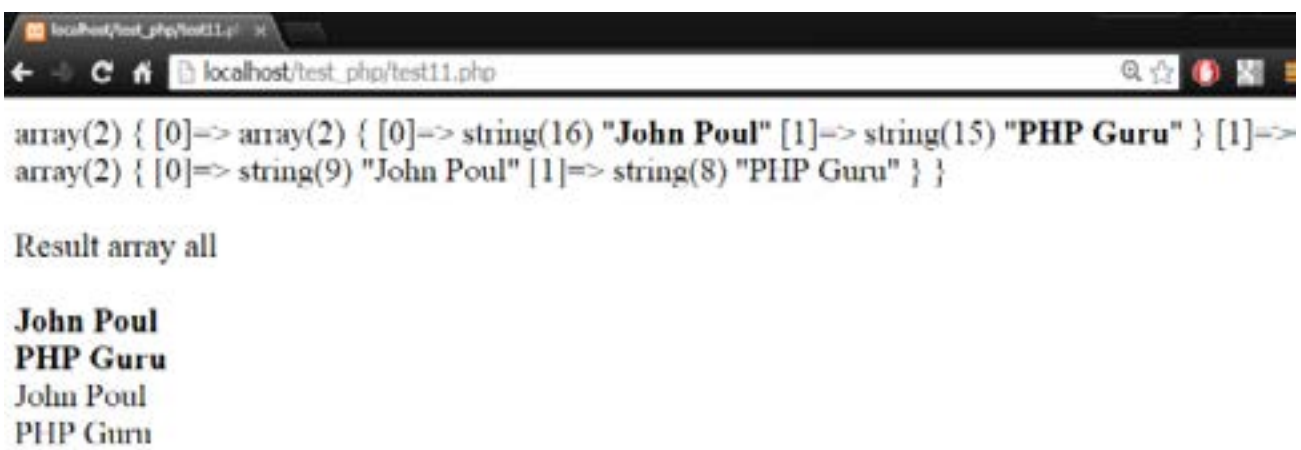
preg_match_all ใช้งาน regular expression ในการค้นหา โดย ค้นหาจนเจอที่ตรงหมดทุกตัว

ใช้งาน regular expression ในการค้นหา โดยค้นหาจนเจอที่ตรงหมดทุกตัว

ตัวอย่าง

```
<?php
$userinfo = "Name: <b>John Poul</b> <br> Title: <b>PHP Guru</b>";
preg_match_all ("/<b>(.*?)</b>/U", $userinfo, $matches);
var_dump($matches);
echo "<br><br> Result array all <br><br>";
echo $matches[0][0]."<br>";
echo $matches[0][1]."<br>";
echo $matches[1][0]."<br>";
echo $matches[1][1]."<br>";
?>
```

ผลลัพธ์



preg_split แบ่งข้อความออกโดยใช้ regular expression

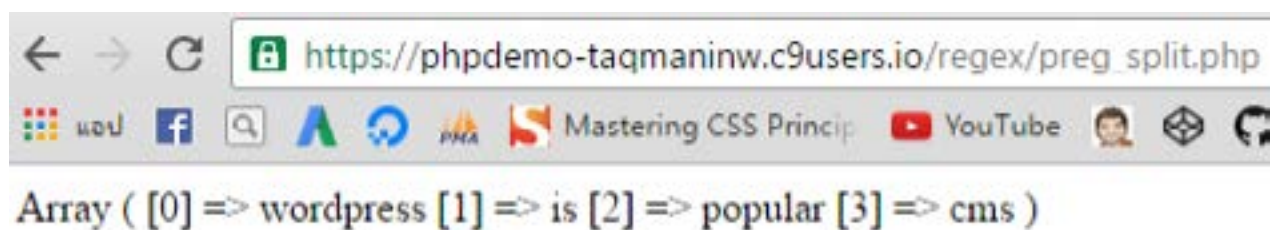
ใช้งาน regular expression ในการค้นหา โดยค้นหาจนเจอที่ตรงหมดทุกตัว

ตัวอย่าง

```
<?php
$keywords = preg_split("/[\s.]+/", "wordpress is popular cms");
print_r($keywords);
?>
```

ผลลัพธ์

ในตัวอย่างใช้การเว้นวรรค เป็นเงื่อนไขในการแบ่ง



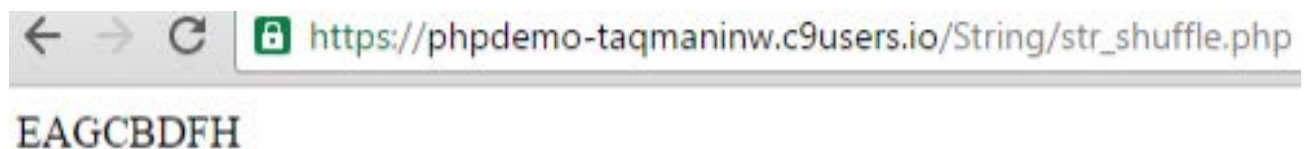
str_shuffle สลับตำแหน่งของตัวอักษรในคำ

ใช้สลับตำแหน่งแบบสุ่มในตัวอักษร

ตัวอย่าง

```
<?php
$str = 'ABCDEFGH';
$shuffled = str_shuffle($str);
echo $shuffled;
?>
```

ผลลัพธ์



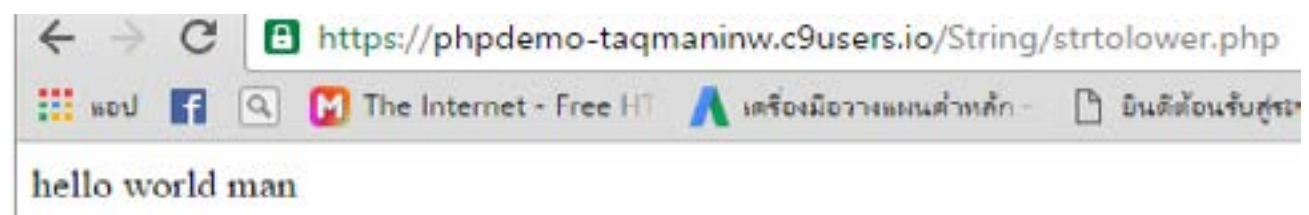
strtolower แปลงตัวพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก

ใช้สำหรับแปลงตัวพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก

ตัวอย่าง

```
<?php
$str = "Hello World Man";
$str = strtolower($str);
echo $str;
?>
```

ผลลัพธ์

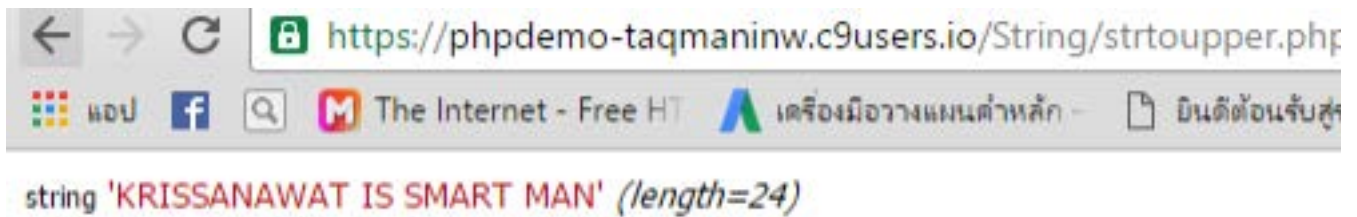


strtoupper ใช้สำหรับแปลงตัวพิมพ์เล็กให้เป็นตัวพิมพ์ใหญ่

ตัวอย่าง

```
<?php
$str = "Krissanawat Is Smart Man";
$str = strtoupper($str);
var_dump($str);
?>
```

ผลลัพธ์



str_repeat สร้างค่าที่กำหนดตามจำนวนที่กำหนด

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
echo str_repeat("สวัสดีครับ <br>", 10);
?>
```

ผลลัพธ์



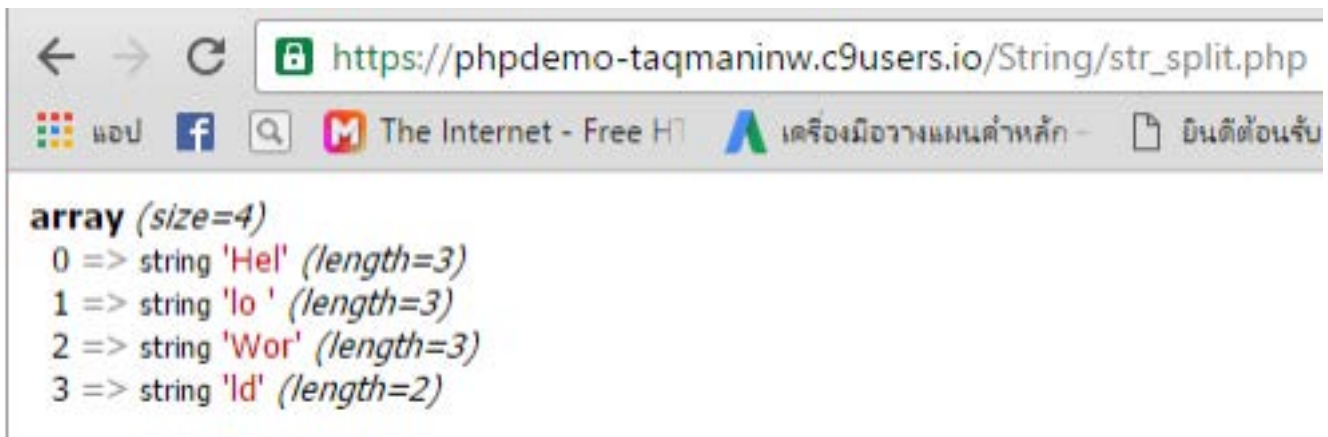
str_split แปลงค่าให้เป็นอาร์เรย์

ตัวอย่าง

```
<?php
$str = "Hello World";
$arr2 = str_split($str, 3);
var_dump($arr2);
?>
```

ผลลัพธ์

จะเห็นว่าตัวอักษรโดนแยกนำไปเก็บ ไว้ในคีย์ของอาเรย์โดยแบ่งจำนวนตัวอักษรที่บันทึกไว้แต่ละช่อง ตามที่กำหนด

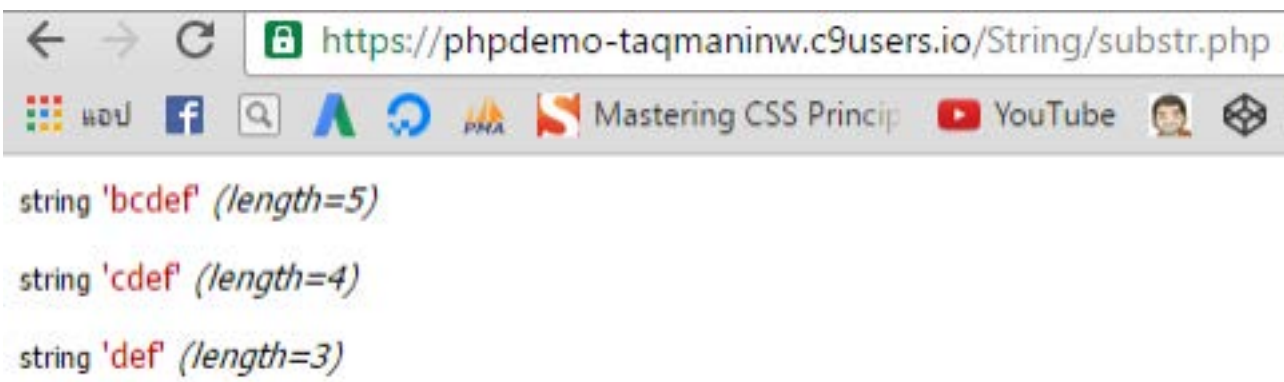


substr แยกคำในประโยคออกมาตามที่กำหนด

ตัวอย่าง

```
<?php
var_dump(substr("abcdef",1));
var_dump(substr("abcdef",2));
var_dump(substr("abcdef",3));
?>
```

ผลลัพธ์



wordwrap ตัดคำเพื่อขึ้นหน้าใหม่

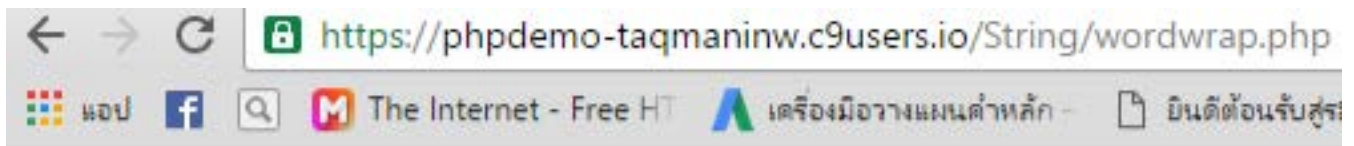
ใช้สลับตำแหน่งแบบสุ่มในตัวอักษร

ตัวอย่าง

```
<?php
$text = "this is a book";
$newtext = wordwrap($text, 8, "<br />", true);
echo "$newtext\n";
?>
```

ผลลัพธ์

จะเห็นว่าคำที่มีจำนวนอักขระเกิน 8 ตัวจะถูกแบ่งออก โดยเท็กที่ใช้ขึ้น บรรทัดใหม่



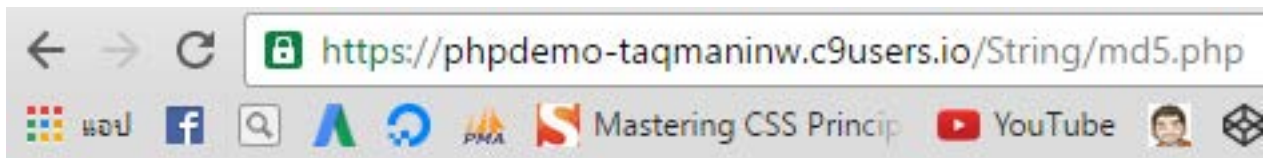
this is
a
bookkkkkk
kkkkkkkk
kkkkkkkk
kkkkkkkk
kk

md5 สร้างค่าแฮชแบบ md5

ตัวอย่าง

```
<?php  
$str = 'Hello World PHP';  
echo md5($str);  
?>
```

ผลลัพธ์



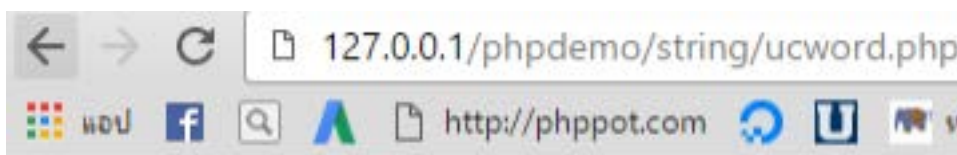
f979d325ce54eec225b345252c78673f

ucwords เปลี่ยนคำแรกในประโยคให้เป็นตัวใหญ่

ตัวอย่าง

```
<?php  
$foo = 'hello world!';  
echo ucwords($foo);  
?>
```

ผลลัพธ์



Hello World!

trim ลบช่องว่างในคำหรือประโยค

ตัวอย่าง

```
<?php
$text = "\t\tThese are a few words :) ... ";
$binary = "\x09Example string\x0A";
$hello = "Hello World";
$trimmed = trim($text, " \t.");
var_dump($trimmed);
$trimmed = trim($hello, "Hdle"); // ใส่ตัวที่ต้องการตรวจสอบเพิ่มเข้าไปด้วย
var_dump($trimmed);
$trimmed = trim($hello, 'Hdwr'); // ใส่ตัวที่ต้องการตรวจสอบเพิ่มเข้าไปด้วย
var_dump($trimmed);
?>
```

ผลลัพธ์



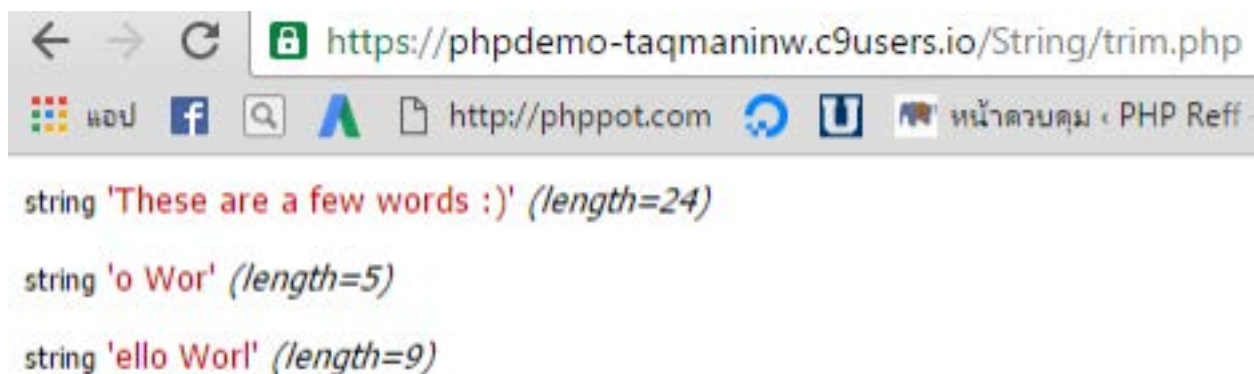
htmlspecialchars ใช้แสดงตัวอักษรที่ลักษณะเป็นโค้ด

ใช้สลับตำแหน่งแบบสุมในตัวอักษร

ตัวอย่าง

```
<?php
$new = htmlspecialchars("<a href='test'>Test</a>");
echo $new;
?>
```

ผลลัพธ์ โดยค่าเริ่มต้น ของฟังก์ชันจะใช้ลบเครื่องหมาย ดับเบิ้ลควอต



htmlspecialchars ใช้แสดงตัวอักขระที่ลักษณะเป็นโค้ดโดยไม่ต้องแปล

ตัวอย่าง

```
<?php
$str = "A 'quote' is <b>bold</b>";
echo $str;
echo "<br>";
echo htmlspecialchars($str);
?>
```

ผลลัพธ์

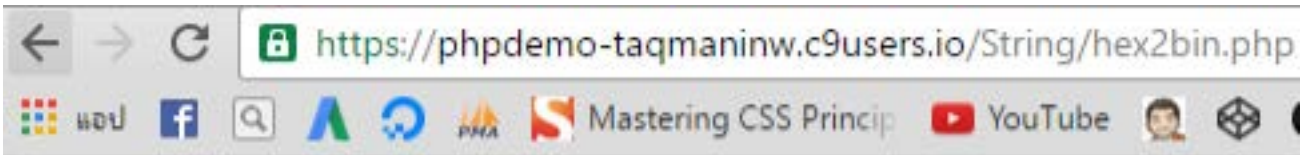
ก่อนใช้งานแท็ก จะถูกแปล แต่ หลังใช้ฟังก์ชัน จะไม่ถูกแปล

hex2bin แปลงเลขฐานสิบหกเป็นตัวอักษร

ตัวอย่าง

```
<?php
$hex = hex2bin("616473617361");
var_dump($hex);
?>
```

ผลลัพธ์

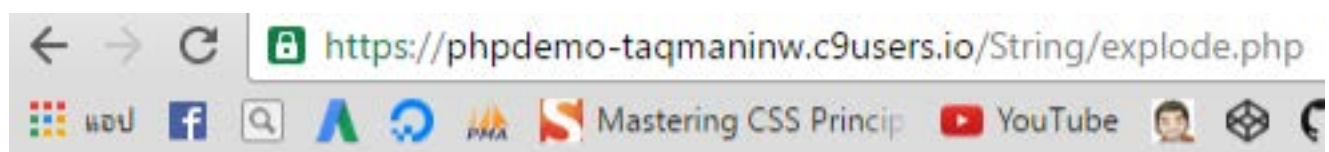


explode แยกตัวอักษรออกตามเงื่อนไขที่กำหนด

ตัวอย่าง

```
<?php
$input = "hello,World";
var_dump(explode(',', $input));
?>
```

ผลลัพธ์ ตัวอย่างนี้ใช้เครื่องหมายคอมมา เป็นเงื่อนไขในการแบ่ง

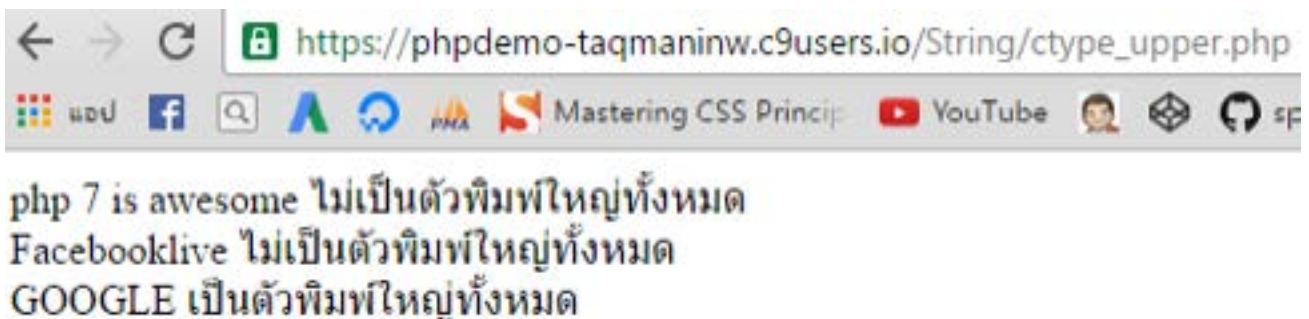


ctype_upper ตรวจสอบว่าประโยคนั้น มีเฉพาะตัวพิมพ์ใหญ่เท่านั้น หรือไม่

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$strings = array('php 7 is awesome', 'Facebooklive', 'GOOGLE');
foreach ($strings as $testcase) {
    if (ctype_upper($testcase)) {
        echo " $testcase เป็นตัวพิมพ์ใหญ่ทั้งหมด <br>";
    } else {
        echo " $testcase ไม่เป็นตัวพิมพ์ใหญ่ทั้งหมด<br>";
    }
}
?>
```

ผลลัพธ์



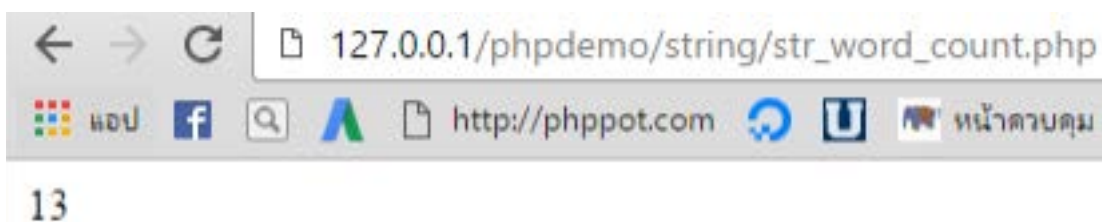
str_word_count นับจำนวนคำในประโยค

คือ ฟังก์ชันที่ใช้ นับจำนวนคำในประโยค

ตัวอย่าง

```
<?php
$string = 'This project contains a number of converters to and from different Blog
services';
echo str_word_count($string);
?>
```

ผลลัพธ์

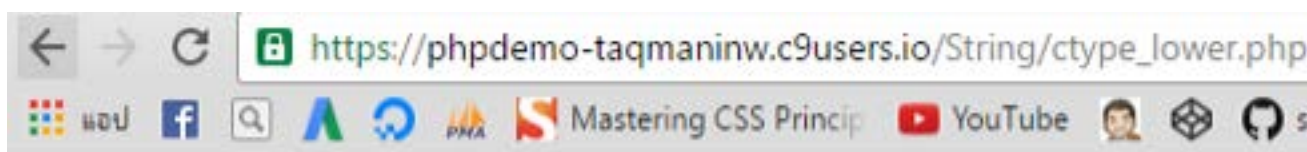


ctype_lower ตรวจสอบว่าประโยคนั้น มีเฉพาะตัวพิมพ์เล็กเท่านั้นหรือไม่

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$strings = array('php 7 is awesome', 'facebooklive', 'Alphabetic');
foreach ($strings as $testcase) {
    if (ctype_digit($testcase)) {
        echo " $testcase เป็นตัวพิมพ์เล็กทั้งหมด <br>";
    } else {
        echo " $testcase ไม่เป็นตัวพิมพ์เล็กทั้งหมด<br>";
    }
}
?>
```

ผลลัพธ์

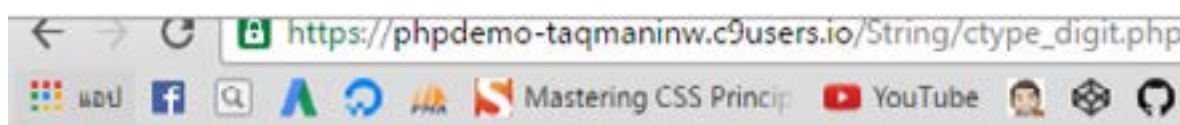


ctype_digit ตรวจสอบว่าประโยคนั้น มีเฉพาะตัวเลข เท่านั้น หรือไม่

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$strings = array('23423Ala', '12322', 'Alphabetic');
foreach ($strings as $testcase) {
    if (ctype_digit($testcase)) {
        echo " $testcase เป็นตัวเลขทั้งหมด <br>";
    } else {
        echo " $testcase ไม่เป็นตัวเลขทั้งหมด <br>";
    }
}
?>
```

ผลลัพธ์

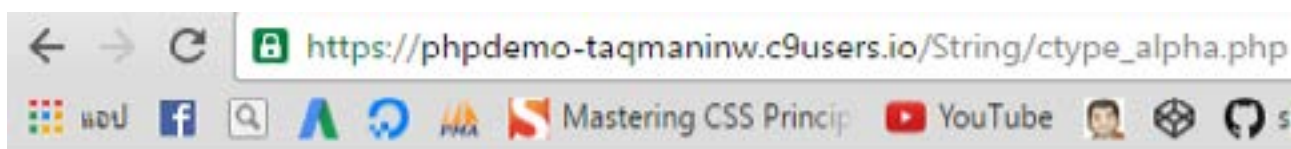


ctype_alpha ตรวจสอบว่าประโยคนั้น เป็นตัวอักษร เท่านั้น หรือไม่

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$strings = array('23423Ala', 'ALPHANUmeric', 'Alphabetic');
foreach ($strings as $testcase) {
    if (ctype_alpha($testcase)) {
        echo " $testcase อยู่ในตัวอักษรทั้งหมด <br>";
    } else {
        echo "$testcase ไม่อยู่ในตัวอักษรทั้งหมด <br>";
    }
}
?>
```

ผลลัพธ์



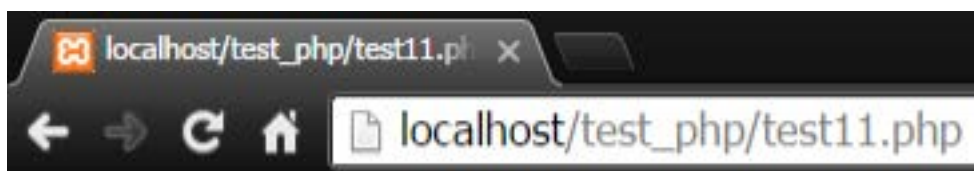
htmlspecialchars_decode ถอดรหัสตัวอักษรที่ถูกรหัสแปลงโดย htmlspecialchars

ถอดรหัสตัวอักษรที่ถูกรหัสแปลงโดย htmlspecialchars

ตัวอย่าง

```
<?php
$str = "This is some <b>bold</b> text.";
echo htmlspecialchars_decode($str);
?>
```

ผลลัพธ์



This is some **bold** text.

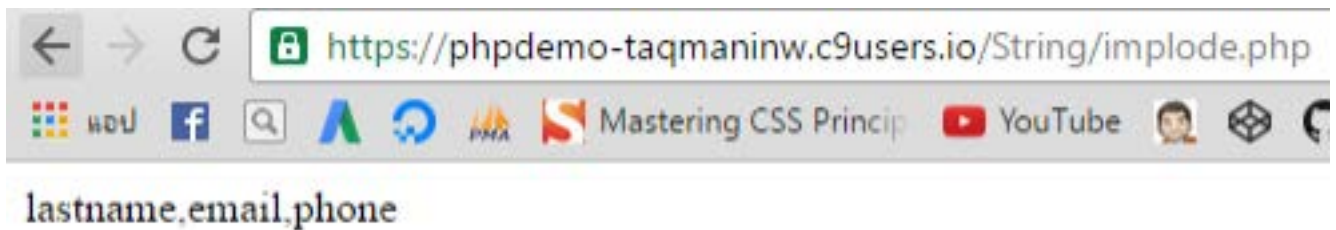
implode แปลงอาร์เรย์ให้กลายเป็นตัวอักษร

ตัวอย่าง

```
<?php
$array = array('lastname', 'email', 'phone');
$comma_separated = implode(",", $array);
echo $comma_separated;
?>
```

ผลลัพธ์

ในตัวอย่างนี้ใช้เครื่องหมายคอมมาในการคั่นแบ่ง

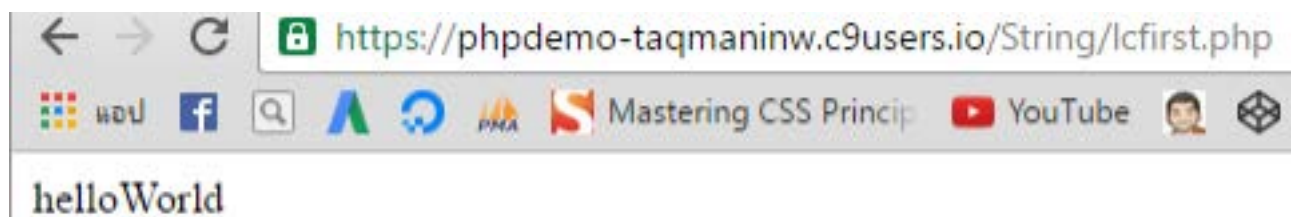


lcfirst แปลงให้ตัวอักษรตัวแรกของคำกลายเป็นตัวพิมพ์เล็ก

ตัวอย่าง

```
<?php
$string = 'HelloWorld';
echo lcfirst($string);
?>
```

ผลลัพธ์

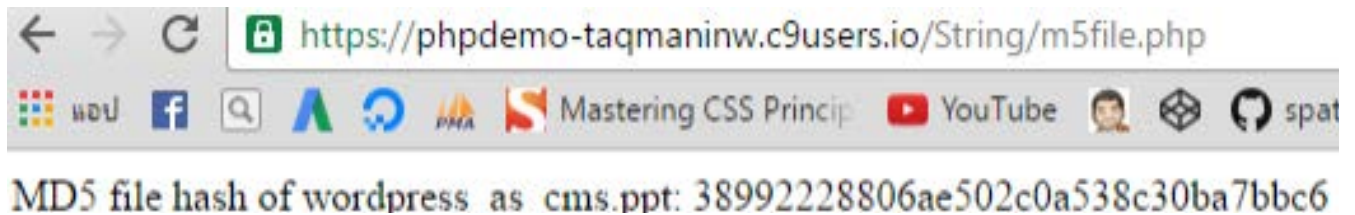


php md5_file สร้างค่าแฮชแบบ md5 โดยใช้ไฟล์

ตัวอย่าง

```
<?php
$file = 'wordpress_as_cms.ppt';
echo 'MD5 file hash of ' . $file . ' : ' . md5_file($file);
?>
```

ผลลัพธ์



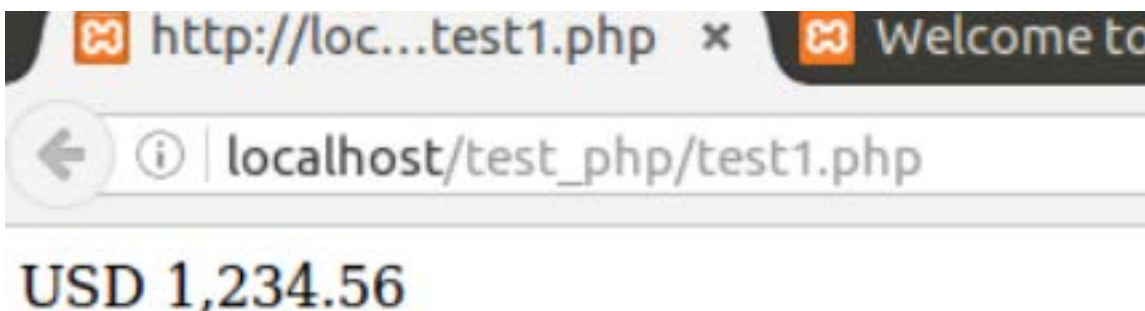
money_format ปรับแต่งให้ตัวเลขให้อยู่ในรูปแบบเงิน

ปรับแต่งให้ตัวเลขให้อยู่ในรูปแบบเงิน

ตัวอย่าง

```
<?php
$number = 1234.56;
setlocale(LC_MONETARY, 'en_US');
echo money_format('%i', $number) . "\n";
?>
```

ผลลัพธ์

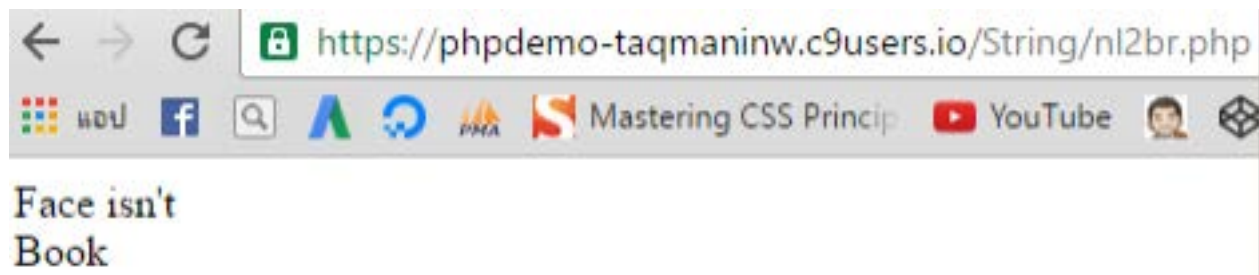


nl2br ช่วยเพิ่ม \n เป็น br แทนคำสั่ง ขึ้นบรรทัดใหม่เดิม

ตัวอย่าง

```
<?php
echo nl2br("Face isn't\n Book");
?>
```

ผลลัพธ์



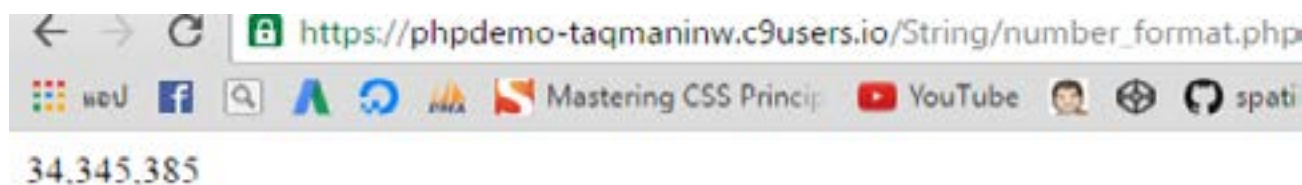
number_format จัดรูปแบบให้ตัวเลข เช่น ใส่ลูกน้ำ

ตัวอย่าง

```
<?php
echo number_format("34345385.39");
?>
```

ผลลัพธ์

สำหรับตัวอย่างเบื้องต้น จะเป็นการค้นแบบอังกฤษ ซึ่งใช้กันเป็นสากล



strcasecmp เปรียบเทียบข้อความหรือประโยค

ใช้การเปรียบเทียบ แบบไม่สนตัวพิมพ์ใหญ่หรือเล็ก หรือที่เรียกว่า case sensitive

ถ้าเทียบกันแล้วได้เป็น 0 คือ เท่ากัน

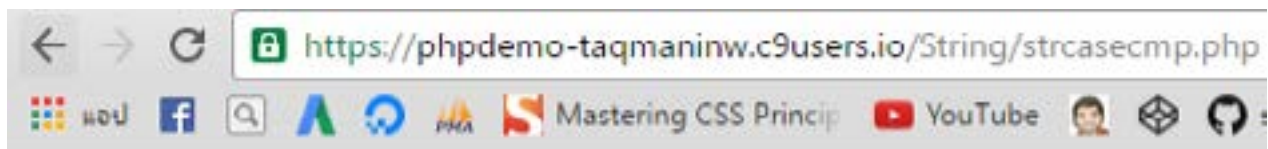
ถ้าน้อยกว่า 0 คือตัวแรกน้อยกว่าตัวที่ 2

ถ้ามากกว่า 0 คือตัวแรกมากกว่าตัวที่ 2

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$var1 = "Hello";
$var2 = "hello";
if (strcasecmp($var1, $var2) == 0) {
    echo '$var1 เหมือนกับ $var2 ในการเปรียบเทียบแบบ case sensitive';
}
?>
```


ผลลัพธ์



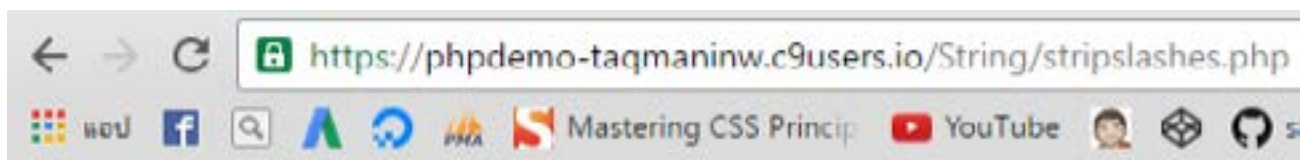
\$var1 เหมือนกับ \$var2 ในการเปรียบเทียบแบบ case sensitive

stripslashes ลบเครื่องหมายแบคสแลชออกจากประโยค

ตัวอย่าง

```
<?php
$str = "Is your name O\'reilly?";
echo stripslashes($str);
?>
```

ผลลัพธ์



Is your name O'reilly?

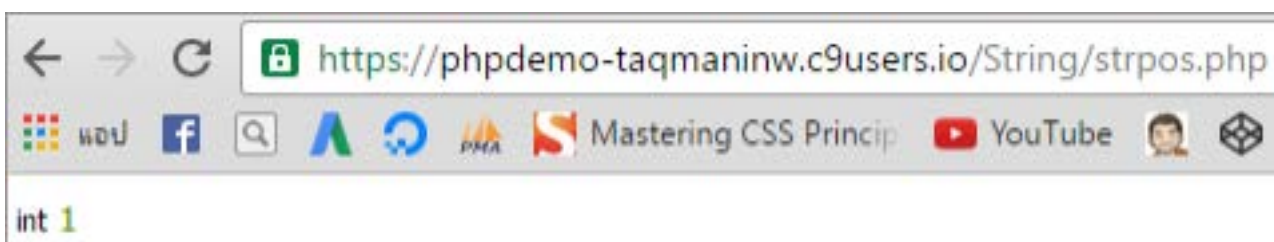
strpos หาตำแหน่งของตัวอักษร ตัวแรก

ตัวอย่าง

```
<?php
$string = 'abc';
$findme = 'b';
var_dump(strpos($string, $findme));
?>
```

ผลลัพธ์

ค่าที่ได้คือ ตำแหน่งของตัวอักษรที่อยู่ในคำหรือ ประโยค

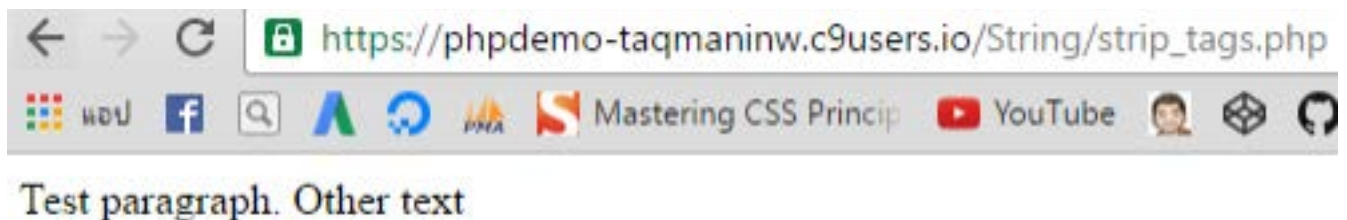


strip_tags ใช้ลบคำที่มีลักษณะเป็นโค้ด html หรือ php ออกจากประโยค

ตัวอย่าง

```
<?php
$text = '<p>Test paragraph.</p><!-- Comment --> <a href="#fragment">Other text</a>';
echo strip_tags($text);
?>
```

ผลลัพธ์

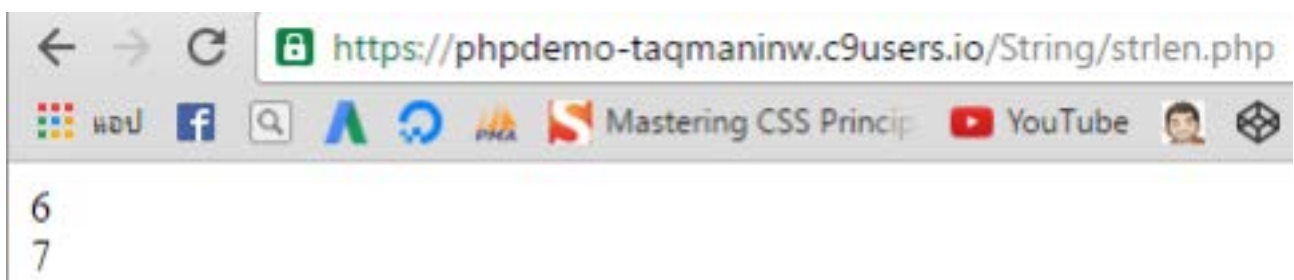


strlen นับจำนวนของคำรวมทั้งช่องว่างด้วย

ตัวอย่าง

```
<?php
$str = 'abcdef';
echo strlen($str); // 6
echo '<br>';
$str = ' ab cd ';
echo strlen($str); // 7
?>
```

ผลลัพธ์

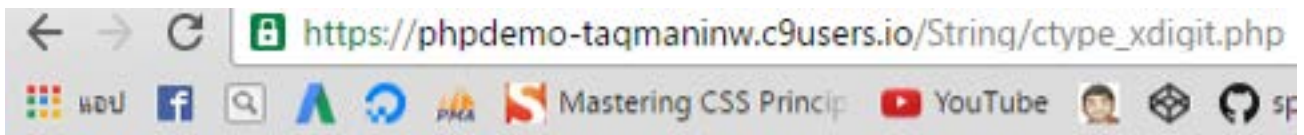


ctype_xdigit ตรวจสอบว่าประโยคนั้นมี เฉพาะเลขฐานสิบหกเท่านั้น หรือไม่

ตัวอย่าง

```
<?php
header("content-type: text/html; charset=UTF-8");
$strings = array('AB10BC99', 'AR1012', 'ab12bc99');
foreach ($strings as $testcase) {
    if (ctype_xdigit($testcase)) {
        echo " $testcase อยู่ในรูปแบบเลขฐาน 16<br>";
    } else {
        echo "$testcase ไม่อยู่ในรูปแบบเลขฐาน 16<br>";
    }
}
?>
```

ผลลัพธ์



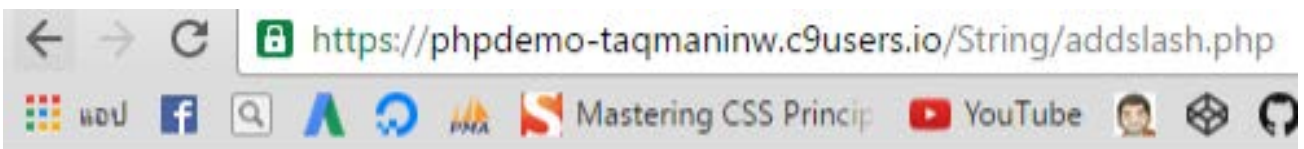
AB10BC99 อยู่ในรูปแบบเลขฐาน 16
AR1012 ไม่อยู่ในรูปแบบเลขฐาน 16
ab12bc99 อยู่ในรูปแบบเลขฐาน 16

addslashes ใส่เครื่องหมาย backslash ข้างหน้า predefined characters

ตัวอย่าง

```
<?php
$str = "Is your name 'Krissanawat'?"
echo addslashes($str);
?>
```

ผลลัพธ์



Is your name \'Krissanawat\'?

chunk_split ตัดตัวอักษรที่รวมเป็นประโยคออกเป็นส่วนๆ

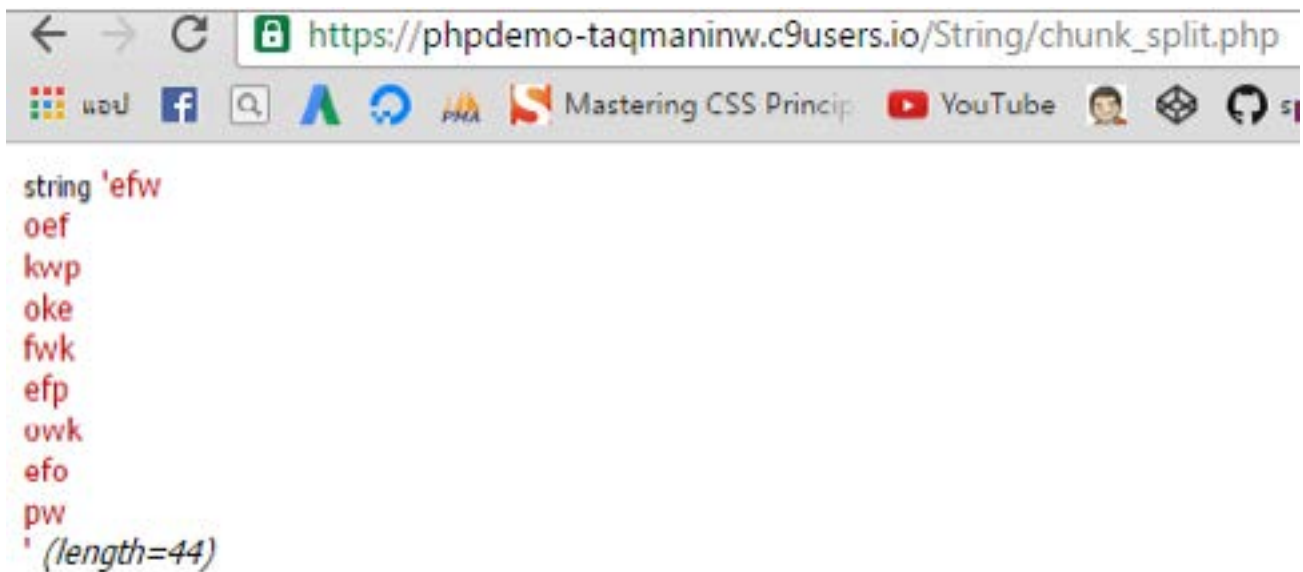
chunk_split คือ ฟังก์ชันที่ใช้ตัดตัวอักษรที่รวมเป็นประโยคออกเป็นส่วนๆ โดยพารามิเตอร์ที่จำเป็นต้องใช้อีก

1. ค่าที่ต้องการตัด
2. จำนวนของตัวอักษรต่อหนึ่งชุด

ตัวอย่าง

```
<?php
var_dump(chunk_split('efwoefkwpokefwkefpowkefopw',3));
?>
```

ผลลัพธ์



strstr ใช้หาสตริงจากจุดเริ่มต้นที่กำหนด

โครงสร้างของคำสั่ง

```
string strstr ( string $haystack , mixed $needle [ , bool $before_needle = false ] )
```

\$haystack คือ ข้อความสตริง

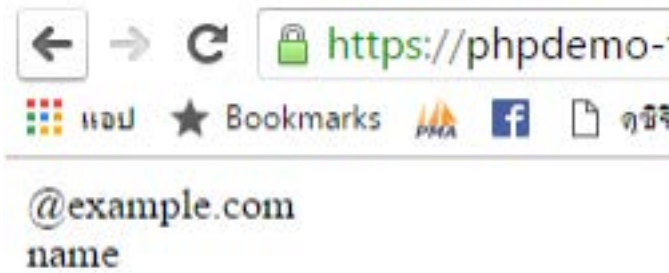
\$needle คือ จุดเริ่มต้นการหาสตริง

\$before_needle คือ การกำหนดว่าต้องการสตริงที่อยู่ก่อนหน้า \$needle หรือ หลังจาก \$needle

ตัวอย่าง

```
<?php
$email = 'name@example.com';
$domain = strstr($email, '@');
echo $domain, "<br>"; // prints @example.com
$user = strstr($email, '@', true); // As of PHP 5.3.0
echo $user; // prints name
?>
```

ผลลัพธ์

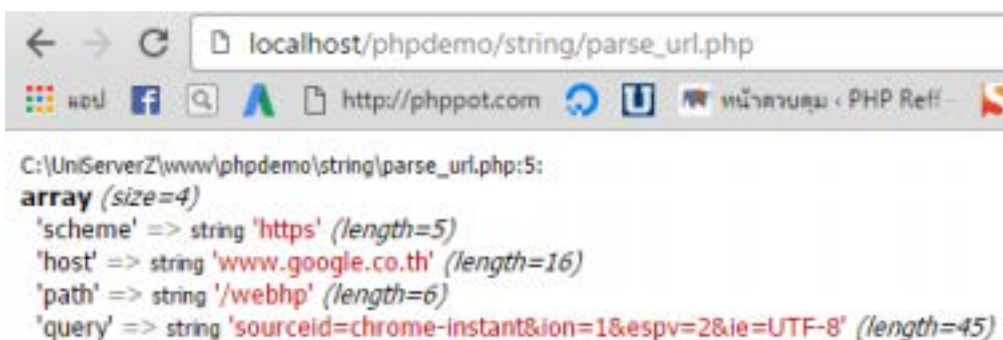


parse_url ตัดแบ่ง url ออกเป็นอาร์เรย์

ตัวอย่าง

```
<?php
$url = 'https://www.google.co.th/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8';
var_dump(parse_url($url));
?>
```

ผลลัพธ์

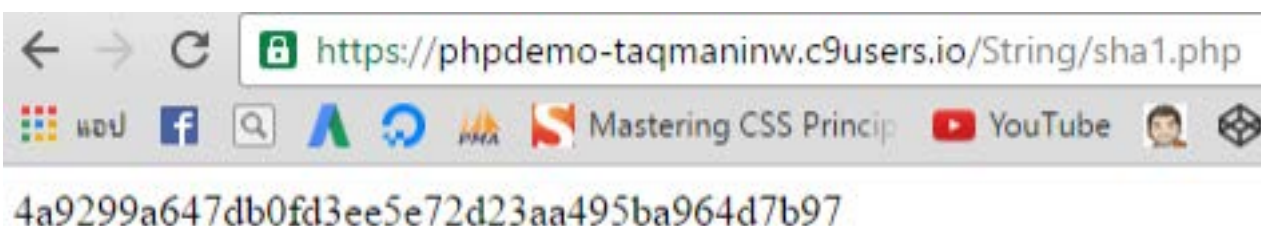


sha1 สร้างค่าแฮชจากอัลกอริทึม sha1

ตัวอย่าง

```
<?php
$string = 'PHP7 is Fast And High Performance';
echo sha1($string);
?>
```

ผลลัพธ์



sha1_file สร้างค่าแฮชจากอัลกอริทึม sha1 โดยใช้ไฟล์

ตัวอย่าง

```
<?php
$file = 'wordpress_as_cms.ppt';
echo 'SHA1 file hash of ' . $file . ' : ' . sha1_file($file);
?>
```

ผลลัพธ์

