الفصل الحادي عشر

توابع لم نتطرق لها

في هذا الفصل سوف نقوم بالتطرق إلى بعض التوابع المعرفة في PHP التي لم نتعرف عليها في الفصـول السابقة والتي قد نحتاج إلى استخدامها في كثير من التطبيقات، سنقدم بعض الشرح عن هذه التوابع مع أمثلة إن احتاج الأمر .

سوف نشرح هذه التوابع وفق الترتيب الأبجدي لها:

التابع abs : هذا التابع الرياضي يعطي القيمة المطلقة العددية للوسيط الممرر لها، كما أنه يعيد نوع المعطيات نفسه للوسيط الممرر ، الشكل العام للتابع هو:

```
abs (number)
```

أمثلة:

```
$abs = abs(-4.2);
$abs1 = abs(5);
$abs2 = abs(-5);
```

النتيجة تكون:

```
$abs = 4.2
$abs1 = 5
$abs2 = 5
```

الـتابع acos : هـذا التابع يعطينا الزاوية التي تكون تجيبها cos هو الوسيط الممرر لها، فمـثلاً الزاوية 1- = (π) cos <== π = (1-)acos وهذا التابع هو التابع العكسي للتابع cos الذي سنتعرف عليه لاحقاً.

```
$acos1 = acos (-1);
$acos1 = 3.14
```

التابع ( ) acosh : هو نفس التابع السابق ولكن يطبق على التوابع القطعية.

التابع ( ) addcslashes : الشكل العام لهذا التابع:

```
addcslashes (str, charlist);
```

يقـوم هذا التابع بإضافة خط مائل "\" قبل كل رمز من السلسلة str في حال هذا الرمز كان ضمن السلسلة charlist .

أمثلة:

```
$a =addcslashes('012','1..3');
$a1=addcslashes('aan','a..h');
$a2=addcslashes('aan','a..z');
```

النتيجة تكون كالتالي:

$a =0\1\2

الرقم 0 لم يسبق بالخط المائل لأنه غير موجود بالسلسلة 1..3

$a1=\a\an

الحرف n لم يسبق بالخط المائل لأنه غير موجود بالسلسلة a..z

$a2=\a\a\n

في المـثال الأخير جميع الأحرف الكبيرة والصغيرة سوف تسبق بالرمز \ حسب السلسلة a..z

التابع ( ) addslashes : يقوم هذا التابع بإضافة خط مائل "\" قبل نفس الرمز إذا كان وجوداً في السلسلة.

مثال:

```
$a=addslashes('\A\AS');
```

النتيجة:

$a=\\A\\AS

الـتابع array_change_key_case : هـذا التابع يحول فهارس المصفوفة الممررة له كوسيط إلى حالة الأحرف الصغيرة أو الأحرف الكبيرة حسب الوسيط الثاني الممرر له.

مثال:

```
$input_array =array("first" ==> 1, "Second"==>4);
array_change_key_case ($input_array,CASE_UPPER);
```

النتيجة تكون :

```
$input_array =array("FIRST"==> 1, "SECOND"==>4);
```

الـ تابع array_chunk : هذا التابع يحول المصفوفة الممررة له كوسيط إلى n مصفوفة ، حيث n هو وسيط ثاني :

مثال :

```
$input_array =array(“’a’,’b’,’c’,’d’,’e’);
print_r (array_chunk($input_array,2));
print_r (array_chunk($input_array,2,true));
```

النتيجة:

```
array  ( [0] ==>array ( [0] ==>a [1] ==> b)
         [1] ==>array ( [0] ==>c [1] ==> d)
         [2] ==>array ( [0] ==>e ))

array  ( [0] ==>array ( [0] ==>a [1] ==> b)
         [1] ==>array ( [2] ==>c [1] ==> d)
         [2] ==>array ( [4] ==>e ))
```

كما نلاحظ من النتيجة السابقة أن الوسيط الثالث وهو true يجعل فهارس كل مصفوفة تبدأ من حيث انتهت فهارس المصفوفة السابقة.

الـ تابع array_count_values : هذا التابع يحصي عدد مرات تكرار عناصر مصفوفة وهذا التابع يعيد لنا مصفوفة ، فهارسها هي قيم المصفوفة الأصلية وقيمها هو عدد تكرار هذه العناصر .

مثال:

```
$array = array (1,”hello’,1,”world”,”hello”);
print_r = (array_count_values ($array));
```

النتيجة:

```
Array ( [1] ==> 2
        [hello] ==> 2
        [world] ==> 1)
```

التابع array_diff : هذا التابع يأخذ عدد غير محدود من الوسطاء، والوسيط عبارة عن مصفوفة تحتوي على القيم الموجودة في الوسيط الأول وغير موجود في باقي الوسطاء.

مثال:

```
$array1 = array (“a” ==> “green”,”red”,”blue”,”red”);
```

```
$array2 = array (“b” ==> “green”,”yellow”,”red”);
$result = array_diff ($array1,$array2);
```

النتيجة ستكون مصفوفة تحوي على قيمة واحدة هي " blue " .

التابع array_fill : الشكل العام لهذا التابع :

```
$a = array_fill (n,m,string)
```

يقـوم هذا التابع بتوليد مصفوفة $a فيها m عنصر تبدأ فهارسها بالرقم n وجميع القيم عبارة عن السلسلة string .

مثال:

```
$a = array_fill (5,6,’banan’);
```

النتيجة مصفوفة $a تحوي القيم التالية:

```
Array ( [5] ==>banana [6] ==>banana [7] ==> banana
        [8] ==>banana [9] ==>banana [10] ==> banana )
```

التابع array_filter : الشكل العام لهذا التابع:

```
array_filter ($array,”function_name”)
```

يقوم هذا التابع بتطبيق التابع function_name على المصفوفة $array ويعيد مصفوفة $array بعـد تطبـيق التابع عليها (أي يعيد لنا عناصر المصفوفة التي ينطبق عليها التابع المعين).

مثال:

```
Function odd($var) {
return ($var% 2 = = 1);}
Function even ($var) {
return ($var% 2 = = 0);}
$array1= array(“a”==>1,”b”==>2,“c”==>3,“d” ==>4,“e” ==>5);
$array1= array(6,7,8,9,10,11,12);
echo “odd:\n”;
print_r (array_filter ($array1,”odd”));
echo “even:\n”;
print_r (array_filter ($array2,”even”));
```

النتيجة ستكون التالي:

```
odd:
Array([a] ==> 1   [c] ==> 3  [e] ==> 5)
```

even:
Array([0] ==> 6 [2] ==> 8 [4] ==> 10 [6] ==> 12)

الـتابع ( ) empty_flip : يقوم هذا التابع بعكس والتبديل بين القيم والفهارس بحيث يحمـل الفهـرس قيمة والقيمة فهرس ، وفي حال تصادف وجود قيمتين متساويتين فإنه يعتمد آخر قيمة ، وهذا التابع يعيد مصفوفة جديدة في حال نجاح التنفيذ، أو يعيد القيمة false في خال فشل تنفيذ التابع.

مثال :

```
$Irans = array(“a”==>1,”b”==>1,“c”==>2);
$Irans = array_flip ($Irans);
print_r ($Irans);
```

ستكون النتيجة كما يلي:

Array([1] ==> b [2] ==> c)

التابع array_key_name : الشكل العام لهذا التابع هو:

array_key_name (key, array_name)

يفحص هذا التابع عن وجود الفهرس key ضمن المصفوفة array_name ويعيد القيمة true في حال وجود هذا الفهرس أو يعيد القيمة false في حال عدم وجوده.

الـتابع array_keys : يعـيد هذا التابع مصفوفة جديدة قيمها هي فهارس المصفوفة الممـررة له كوسيط كما يعيد مصفوفة جديدة قيمها هي فهارس القيمة التي نبحث عن فهرسها والممررة للتابع كوسيط ثاني.

مثال:

```
$array = array (0==> 100, “color ==>”red”);
$array1 = array (”blue”, ”red”, “green” , ”blue”, ”blue”);
print_r (arraykeys($array));
print_r (arraykeys($array,”blue”));
```

النتيجة:

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

التابع array_map : يقوم هذا التابع بتطبيق تابع ما على عناصر المصفوفة المحددة ويعيد لنا مصفوفة جديدة .

مثال :

```
Function cube($n) {
        return $n*$n*$n;}
$a = array (1,2,3,4,5);
$b = array_map (“cube”,$a);
print_r ($b);
```

ستكون النتيجة كما يلي:

Array([0]==> 1 [3] ==> 8 [2] ==> 27 [3] ==> 64 [4] ==> 125)

ملاحظـة: التابع array_filter : يستثني عناصر المصفوفة التي لا تحقق التابع أما التابع array_map ينفذ التابع على كل عناصر المصفوفة.

التابع array_mery : يقبل هذا التابع أكثر من وسيطين من نوع مصفوفة ويقوم بدمج المصـفوفات جمـيعها في مصفوفة واحدة بحيث يضيف المصفوفة الثانية إلى نهاية المصفوفة الأولى ويضيف المصفوفة الثالثة إلى نهاية المصفوفة الثانية .

```
$array1 = array (1,2,3);
$array2 = array (4,5,6);
result =  array_mery ($array1,$array2);
```

النتيجة ستكون كما يلي:

Array([0]==>1 [1]==>2 [2]==>3 [3]==>4 [4]==>5 [5]==>6)

الـتابع array_mery_recursive : يقـوم هذا التابع بدمج أكثر من مصفوفة داخل مصـفوفة واحدة ، بحيث إذا التقى فهرسين متساويين ، يتم تشكيل مصفوفة داخلية ذات قيم عبارة عن مجموع القيم لهذا الفهرس من المصفوفتين.

مثال:

```
$ar1 = array(“color”==>Array(“favorit”==>”red” ),5);
$ar2= array(10,“color”==>Array(“favorit”==>”green” ),”blue”));
$result= array_mery_recursive ($ar1,$ar2);
```

النتيجة ستكون كما يلي:

```
Array([color]==>Array([favorit]==>Array
                                        ([0]==>red
                                         [1]==>green)
                                [0]==>blue)
```

[0]==>5
[1]==>10)

الـتابع array_pad : هذا التابع يضيف قيم جديدة لمصفوفة ، والشكل العام لهذا التابع على الشكل التالي :

array_pad (array_name, new_size,values)

الـتابع array_pad يجعل عدد قيم المصفوفة array_name يساوي إلى new_size بحيث تكون القيم المضافة هي values .

إذا كـان new_size رقم موجباً يتم إضافة القيم الجديدة إلى نهاية المصفوفة ، وإذا كان هذا الرقم سالباً فإنه يتم إضافة القيم الجديدة إلى بداية المصفوفة (قبل القيم الأصلية).

إذا كانـت القيمة المطلقة للرقم new_size أصغر أو يساوي حجم المصفوفة الأصلية ، عندها لا يتم إضافة قيم .

أمثلة :

```
$input = array(12,10,9);
$result = array_pad ($input,5,0);
```

النتيجة ستكون:

```
array (12,10,9,0,0)
$result = array_pad ($input,-7,0);
```

النتيجة ستكون:

```
array (0,0,0,0,12,10,9)
$result = array_pad ($input,2,0);
```

لا يتم إضافة القيم الجديدة

```
array(12,10,9)
```

الـتابع array_pop : هـذا الـتابع يقوم بحذف آخر عنصر من المصفوفة، وإذا كانت المصفوفة فارغة عندها التابع سوف يعيد القيمة Null .

مثال:

```
$stack= array(1,2,3,4);
$result = array_pop ($stack);
```

النتيجة ستكون:

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

التابع array_push : هذا التابع يضيف قيمة أو أكثر إلى مصفوفة.

مثال:

```
$ar= array(1,2);
 array_pop ($ar,3,4);
```

النتيجة ستكون:

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

الـتابع array_rand : هـذا الـتابع يقوم باختيار عنصر أو أكثر من مصفوفة حسب الوسيط الممرر لها ، والاختيار يكون بشكل عشوائي ، ويعيد النتيجة في مصفوفة.

مثال:

```
$input= array(1,2,3,4,5);
$rand_keys = array_rad($input,2);
print $input [$rand_key[0]];
print $input [$rand_key[1]];
```

النتيجة ستكون عرض رقمين عشوائيين يتغيران في كل عملية تحديث.

الـتابع array_reduce : هذا التابع يطبق تابع معرف بشكل متتالي على جميع عنصر المصـفوفة بحـيث يصـغر حجم المصفوفة إلى قيمة واحدة، الشكل العام لهذا التابع هو كمايلي:

```
array_reduce (array_name, function_name, initial);
```

الوسـيط الأخـير خياري إذا كان موجوداً ، فإنه يطبق على بداية العملية، أو يكون هو النتيجة إذا كانت المصفوفة فارغة.

مثال:

```
Function rsum($v,$w) {
  $v+=$w;
  return $v;   }
Function rmul($v,$w) {
  $v*=$w;
  return $v;   }
$a = array (1,2,3,4,5)
$x=array (   );
$b = array_reduce($a,"rsum");
```

```
$c = array_reduce($a,"rmul",0);
$d = array_reduce($x,"rsum",1);
```
النتـيجة ستكون أن قيمة المتحول $b=15 والمتحول $c=1*2*3*4*5*10=1200 والمتحول $d=1 .

التابع array_reverse : الشكل العام لهذا التابع كما يلي:

```
array_reduce (array_name,bool_reserve_key);
```

يقوم هذا التابع بتشكيل مصفوفة جديدة ، بحيث تكون عناصرها مرتبة بالعكس الوسيط، bool_reserve_key هو وسيط خياري فإذا كانت قيمته true فإن كل قيمة تحتفظ برقم فهرستها .

مثال:

```
$input = array("php",4,array("green","red"));
$result_array_reserve ($input);
$result_key =array_reserve ($input,true);
```

النتيجة : المصفوفة الأولى ستكون كما يلي:

```
Array ([0] ==> array ( [0] ==> green
                       [1] ==> red
       [1] ==> 4
       [2] ==> php )
```

المصفوفة الثانية ستكون كما يلي:

```
Array ([2] ==> array ( [0] ==> green
                       [3] ==> red
       [1] ==> 4
       [2] ==> php )
```

التابع array_shift : هذا التابع يقوم بحذف أول عنصر من المصفوفة .

مثال:

```
$stack = array (1,2,3,4);
$new_array = array_shift ($stack);
```

النتيجة ستكون:

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

التابع array_sum : يقوم هذا التابع بجمع قيم المصفوفة.

مثال:

```
$a= array (2,4,6,8);
echo “ sum(a) = “ array_sum($a);
```

النتيجة ستكون sum (a) = 20 .

التابع array_unshift : هذا التابع سوف يضيف مجموعة من القيم الممررة له كوسطاء إلى بداية المصفوفة المعنية .

مثال:

```
$queue = array (“orange”, “banana”);
array_unshift ($queue, “apple”,”raspberry”);
```

المصفوفة الناتجة ستكون على الشكل التالي:

Array([0]==>apple[1]==>raspberry[2]==>orange [3]==>banana)

التابع array_values : يقوم هذا التابع بإعادة مصفوفة جديدة تحوي على قيم المصفوفة الأصلية .

مثال:

```
$array = array(“size”==>”XL”,”color”==>”gold”);
print_r (array_value($array));
```

النتيجة:

Array ([0] ==> XL [1]==> gold)

التابع asin( ) : هذا التابع يعطينا عكس الجيب .

```
$a = asin (0.5);
echo $a;
```

النتيجة $a=

التابع asinh ( ) : هذا التابع يعطينا عكس الجيب القطعي .

التابع atan ( ) : هذا التابع يعطينا عكس الظل .

التابع atanh ( ) : هذا التابع يعطينا عكس الظل القطعي .

التابع basename : هذا التابع يعطينا اسم الملف بالنسبة لمسار معطى كوسيط.

مثال:

```
$path=”/home/http/html/index.php”;
```

$file = basename($path);//

النتيجة ستكون index.php

$file = basename($path,".php");//

النتيجة ستكون index

التابع bcadd : هذا التابع يقوم بجمع عددين الشكل العام للتابع كما يلي:

bcadd (left_op,right_op,scale)

الوسيط scale يحدد عدد الأعداد بعد الفاصلة العشرية (الدقة).

التابع bccomp : هذا التابع يقارن عددين مع بعضهما مع الأخذ بعين الاعتبار الوسيط الثالث scale لأخذ الدقة بعين الاعتبار والنتيجة ستكون 0 إذا كان العددين متساويين ، 1+ إذا كـان left_op أكـبر مـن right_op و1- إذا كـان left_op أصغر من right_op .

التابع bcdiv : يقوم هذا التابع بقسمة العدد left_op على right_op ، يعطي النتيجة مع الأخذ بعين الاعتبار الوسيط الثالث scale لتحديد الدقة .

bcdiv (left_op,right_op, scale);

التابع bcmod : يعطي هذا التابع باقي قسمة left_op على right_of .

bcmod (left_op,right_op);

الـتابع bcmul : يقوم هذا التابع بضرب left_op على right_op ويعطي النتيجة مع الأخذ بعين الاعتبار الدقة scale .

bcmul (left_op,right_op, scale);

الـتابع bcpow : هـذا الـتابع يضع الرقم x إلى القوة y مع الأخذ بعين الاعتبار الدقة scale لتحديد عدد الأعداد بعد الفاصلة العشرية.

الـتابع ( ) bcsqrt : هذا التابع يعطي الجذر التربيعي للوسيط الممرر له، مع الأخذ بعين الاعتبار الوسيط scale لتحديد الأعداد بعد الفاصلة العشرية.

الـتابع bcsub : هـذا الـتابع يطرح العدد right_op من left_op مع الأخذ بعين الاعتبار الوسيط scale لتحديدالدقة.

الـتابع bin2hex : يحول هذا التابع العدد الممرر له الوسيط والممثل بالطريقة الثنائية إلى القيمة الستة عشرية.

التابع bindec: يحول هذا التابع القيمة من ثنائية إلى عشرية.

الـتابع ceil : هـذا التابع يدور الرقم الذي يحوي فاصلة عشرية إلى أصغر رقم صحيح أكبر منه .

مثال :

```
echo ceil (4.3); //5
echo ceil (9.99); //10
```

التابع chr : يتقبل هذا التابع كوسيط رمز ASCII ويحوله إلى رمزه المكافئ.

التابع copy : يقوم بطبع الملفات

```
copy (source, dest)
```

التابع cos : يعطي هذا التابع تجيب الزاوية المعطاة cos(0)=1

التابع cosh : يعطي هذا التابع تجيب القطعي للزاوية المعطاة.

التابع count : هذا التابع يحصي عدد عناصر متحول في المصفوفة.

مثال:

```
$a[0] = 1: $a[1]=3 , $a[2]=5;
result = count ($a);    // result = 3
```

هـذا التابع قد عطي القيمة 0 للمتحولات التي لم يتم تعيينها بقيم ، أو يعيد هذه القيمة بالنسبة للمصفوفات.

التابع decbin : يحول القيمة المعطاة كوسيط من التمثيل العشري إلى التمثيل الثنائي.

الـتابع dechex : يحـول القيمة المعطاة كوسيط من التمثيل العشري إلى التمثيل الست عشري.

التابع decoct : يحول القيمة المعطاة كوسيط من التمثيل العشري إلى التمثيل الثماني.

التابع define : يعرف ثابت بقيمة :

```
define (“constant”, “hello”);
echo constant;    \\   hello
echo CONSTANT;  \\  CONSTANT
```

لهذا التابع وسيط ثالث خياري من نوع منطقي فإذا كان هذا الوسيط true هذا يعني أن التابع سوف لن يتحسس حالة الأحرف والتعليمة :

```
echo CONSTANT;
```

سوف تطبع النتيجة hello .

التابع defined : يقوم هذا التابع بفحص الوسيط الممرر له فإذا كان معرف كثابت منن قبل هذا التابع سوف يعيد القيمة true ، وإلا false .

التابع deg2rad : يحول هذا التابع العدد الممرر له والممثل بطريقة الدرجات إلى التمثيل بالراديان.

التابع die : هو اختصار للتابع exit الذي يطبع قيمة ويوقف تنفيذ البرنامج.

الـتابع disk_free_space : يمـرر لهذا التابع اسم فهرس، ويعطي هذا التابع المساحة الفارغة على القرص الصلب الحاوي على هذا الفهرس.

التابع empty: يفحص فيما إذا كان الوسيط الممرر له فارغ أم لا.

التابع end : يهيئ المؤشر الداخلي للمصفوفة إلى نهاية المصفوفة .

الـتابع eval : هذا التابع يطبع ما داخله من سلاسل، فإذا كانت هذه السلسلة أمر من أوامر PHP فإنه ينفذه مباشرةً.

التابع exp : مثال exp(2) في هذه الحالة يرفع القيمة e إلى القوة z أي e^z .

التابع Feof : هذا التابع يفحص نهاية ملف محدد معامله كوسيط، ويعيد هذا التابع القيمة true في حال وصل معامل الملف إلى نهاية الملف وإلا يعيد القيمة false .

التابع Fflush : هذا التابع يخرج المعلومات الموجودة في الذاكرة إلى الملف المحدد معامله كوسيط .

ملاحظة: يجب أن يكون الملف مهيأ عن طريق التابع Fopen .

الـتابع Fgate : يخرج هذا التابع سلسلة مؤلفة من حرف وذلك من ملف محدد معامله كوسيط .

الـتابع Fgets : هـذا التابع يخرج سطر من المعلومات ، وذلك من ملف محدد معامله كوسيط للتابع ، الشكل العام لهذا التابع :

Fgets (Fp,length);

يعيد هذا التابع سلسلة طولها length - 1 من الملف.

الـتابع File_exists : يفحص هذا التابع وجود ملف ما ، ويعيد القيمة true في حال وجد هذا الملف وإلا يعيد القيمة false، الشكل العام لهذا التابع :

File_exists (filename);

التابع Fileatime: يعيد هذا التابع زمن آخر ولوج للملف المحدد مساره كوسيط للتابع وفي حال الفشل يعيد القيمة false .

التابع Filectime: يعيد هذا التابع زمن آخر تغيير للملف المحدد مساره كوسيط للتابع وفي حال الفشل يعيد القيمة false .

الـتابع Fileowner : يعـيد هـذا التابع الرقم التعريفي ID لمالك الملف المحدد مساره كوسيط.

الـتابع Fileperms : يعـيد هـذا التابع السماحيات المعطاة على الملف المحدد مساره كوسيط.

التابع Filesize : يعيد هذا التابع حجم الملف المحدد مساره كوسيط.

التابع Filetype : يعيد هذا التابع نوع الملف المحدد مساره كوسيط.

الـتابع Floatual : يستخرج هذا التابع القيمة العددية من نوع float من متحول قد يحوي سلسلة نصية .

مثال:

```
$rar = 123.456 the;
$var1 = Floatual ($var);
echo $var1;
```

النتيجة 123.456

التابع Floor : هذا التابع يدور الرقم المعطى كوسيط إلى أكبر عدد صحيح أصغر منه .

مثال:

```
Floor (4.3) ==> 4
Floor (9.99) ==> 9
```

التابع Flush : هذا التابع يمحي محتويات الذاكرة buffer .

الـتابع Fstat : يعيد هذا التابع مصفوفة من المعلومات حول ملف محدد معامله كوسيط للتابع. المعلومات هي : عدد الارتباطات، مالك الملف، المجموعة المالكة للملف، حجمه، تاريخ آخر ولوج ، تاريخ آخر تعديل...

التابع getcwd : يحدد هذا التابع الفهرس الحالي الذي نعمل عليه، ويعيد سلسلة.

التابع getdate : يعيد مصفوفة قيمها هو التاريخ الحالي وعناصر المصفوفة هي:

الـثواني ، الدقائـق، الساعات ، رقم اليوم في الشهر ، رقم اليوم في الأسبوع ، الشهر، السنة، رقم اليوم في السنة، اسم اليوم، اسم الشهر.

الـتابع gettype : يحـدد نوع المتحول الممرر كوسيط، والأنواع Boolean منطقي ، integer ، double ، string ، مصفوفة ، أو فارغ null .

الـتابع hexdec : يحـول القـيمة المعطـاة من كوسيط من التمثيل الست عشري إلى العشري.

الـتابع imagecolortransparent : هذا التابع يحدد اللون المحدد للصورة على أساس أنه شفاف ، يأخذ معامل الصورة كوسيط.

التابع imagecreatfromgif : ينشأ صورة جديدة من ملف أو عنوان إنترنت ويأخذ مسار الصورة كوسيط.

التابع imagesx : هذا التابع يحدد عرض الصورة المحدد معاملها كوسيط.

التابع imagesy : هذا التابع يحدد طول الصورة المحدد معاملها كوسيط.

الـتابع in_array : يعيد القيمة true فيما إذا كان المتحول الممرر له كوسيط هو من نوع مصفوفة.

التابع is_bool : يعيد القيمة true فيما إذا كان المتحول الممرر له كوسيط هو من نوع منطقي.

التوابع التالية شبيهة بالتابعين السابقين:

is_nan, is_int, is_float, is_file, is_double, is_dir (الوسـيط الممرر يجب أن يكون ليس رقماً) .

is_readable, is_null (فحص الملفات القابلة للقراءة) . is_string ،
is_writable (يفحص إمكانية الكتابة على ملف محدد مساره كوسيط).
التابع isset : هذا التابع يفحص متحول فيما إذا كان مهيأ أم لا.
التابع log : يعيد اللوغاريتم العادي للعدد الممرر كوسيط.
التابع log10 : يعيد اللوغاريتم العشري للعدد الممرر كوسيط.
التابع max : يعيد هذا التابع أكبر رقم من مجموعة أرقام ممررة له كوسيط.
التابع min : يعيد هذا التابع أصغر رقم من مجموعة أرقام ممررة له كوسيط.
التابع mkdir : هذا التابع ينشئ فهرس.
الـتابع mysql_affected_rows : يعيد لنا هذا التابع عدد السجلات المتأثرة بتطبيق عبارة SQL عليها .
الـتابع mysql_close : هذا التابع يغلق وينهي الاتصال بقاعدة البيانات mysql وهو يكافـئ الـتابع odbc_close ويأخـذ هذا التابع وسيط هو معرف الاتصال بقاعدة المعطيات.
التابع mysql_connect : هذا التابع يقوم بإجراء الاتصال بقاعدة المعطيات mysql ، وهو يكافئ التابع odbc_connect يأخذ هذا التابع كوسيط أول اسم الملقم، ووسيط ثاني اسم المستخدم ، ووسيط ثالث كلمة سر قاعدة البيانات.
التابع mysql_db_create : ينشأ هذا التابع قاعدة بيانات جديدة من نوع mysql ويأخذ وسيط أول اسم القاعدة الجديدة.
الـتابع musql_data_seek : ينقل هذا التابع مؤشر النتيجة للقاعدة إلى السطر المحدد رقمه كوسيط ثاني للتابع، ويأخذ وسيط أول معرف الاتصال بقاعدة البيانات.
التابع mysql_drop_db : يحذف هذا التابع قاعدة بينات كاملة ، ويكون اسم قاعدة البيانات وسيط أول لهذا التابع ومعرف الاتصال بالقاعدة يكون وسيط ثاني.
التابع mysql_errno : يولد هذا التابع رقم يكافئ رسالة الخطأ الناتجة عن تطبيق عبارة SQL سابقة ويأخذ معرف الاتصال وسيط.

الـتابع mysql_error : يولد هذا التابع رسالة تكافئ رسالة الخطأ الناتجة عن تطبيق عبارة SQL ويأخذ معرف الاتصال كوسيط.

التابع mysql_fetch_array : يحول هذا التابع السجل الناتج عن تنفيذ عبارة SQL إلى مصفوفة وكل حقل من هذا السجل هو قيمة لهذه المصفوفة ، هذا التابع يكافئ التابع odbc_fetch_array .

الـتابع mysql_fetch_field : يسترجع هذا التابع عمود من سجل ويعيده كعنصر object .

الـتابع mysql_fetch_object : يسـترجع هذا التابع السجل المعاد من تنفيذ عبارة SQL كعنصر object وهذا التابع يكافئ odbc_fetch_object .

التابع mysql_field_len : يعيد هذا التابع طول السجل المعاد وهو يكافئ التابع odbc_field_len .

التابع mysql_field_seek : يحرك هذا التابع المؤشر داخل قاعدة البيانات إلى الإزاحة المحددة كوسيط ثاني ، يأخذ هذا التابع النتيجة عبارة SQL كوسيط .

الـتابع mysql_field_tabe : يحـدد هذا التابع اسم الجدول الذي يحوي حقل ما . النتيجة تكون وسيط أول وإزاحة الحقل تكون وسيط ثاني.

التابع mysql_field_type : يحدد هذا التابع نوع الحقل المحدد في نتيجة عبارة SQL، النتيجة تكون وسيط أول لهذا التابع وإزاحة الحقل تكون وسيط ثاني لهذا التابع.

الـتابع mysql_free_result : يحدد هذا التابع الذاكرة الحاوية على عبارة SQL ، معرف النتيجة يكون وسيط وحيد لهذا التابع.

التابع mysql_get_client_info : يعيد هذا التابع معلومات حول الزبون ، لا يحوي هذا التابع على وسطاء.

مثال:

```
Printf (“mysql client info:\n”, mysql_get_client_info( ) );
```

النتيجة:

```
Mysql client info: 2.23.39
```

التابع mysql_get_host_info : يعيد هذا التابع معلومات الملقم المضيف للقاعدة.
الـتابع mysql_get_proto_info : يعـيد هـذا التابع معلومات حول البروتوكول المطبق.
التابع mysql_get_server_info : يعيد هذا التابع معلومات حول الملقم
الـتابع mysql_insert_id : يعيد هذا التابع الرقم التعريفي لآخر عملية insert تمت، يأخذ هذا التابع وسيط معرف الاتصال.
التابع mysql_list_db : يعرف هذا التابع قائمة بأسماء كل قواعد البيانات الموجودة في mysql server ويأخذ هذا التابع معرف الاتصال كوسيط وحيد.
التابع mysql_list_table : يعرف هذا التابع يعرف هذا التابع أسماء الجداول المتاحة في قاعـدة ما ، هذا التابع يأخذ وسيط أول اسم القاعدة ومعرف الاتصال بالقاعدة وسيط ثاني.
الـتابع mysql_num_field: يحصـي هذا التابع عدد الحقول في النتيجة ، يأخذ هذا التابع معرف النتيجة كوسيط وحيد.
التابع mysql_num_rows: يحصي هذا التابع عدد السجلات في النتيجة ، يأخذ هذا التابع معرف النتيجة كوسيط وحيد.
الـتابع mysql_ping : يقوم هذا التابع بفحص الاتصال بالملقم، يعيد هذا التابع القيمة true في حال وجود الاتصال وإلا يعيد القيمة false .
الـتابع mysql_query : يرسـل هذا التابع عبارة SQL ، يأخذ هذا الوسيط عبارة SQL وسيط أول ، ومعرف الاتصال وسيط ثاني.
الـتابع mysql_select_db : يهيـئ هـذا التابع قاعدة معطيات للعمل معها ، اسم القاعدة يشكل وسيط أول ومعرف الاتصال يشكل وسيط ثاني.
التابع natsort : يجري هذا التابع ترتيب طبيعي على مصفوفة .

```
$array1 = array(“in12”,”in10”,”in2”,”in1”);
natsort ($array1);
```

النتيجة:

Array ([0] ==>in1 [1] ==>in2 [2] ==>in10 [3] ==>in12)

في حين تطبيق التابع التالي:

Sort ($array1);

يؤدي للنتيجة التالية:

Array ([0] ==>in1 [1] ==>in10 [2] ==>in12 [3] ==>in2)

التابع nl2br : هذا التابع يولد وسم HTML الذي يكسر السطر
 في بداية كل سطر .

التابع octdec : يحول هذا التابع العدد الممرر كوسيط من تمثيل ثماني إلى تمثيل عشري.

التابع odbc_close_all : يغلق هذا التابع كل القواعد التي تم الاتصال بـا.

الـتابع odbc_data_source : يعيد هذا التابع معلومات حول الاتصال الحالي المحدد معرفه كوسيط.

التابع odbc_error : يولد هذا التابع آخر شيفرة خطأ ، ويأخذ معرف الاتصال وسيط وحيد.

الـتابع odbc_errormsg : يولد هذا التابع آخر رسالة خطأ ، ويأخذ معرف الاتصال وسيط وحيد.

الـتابع odbc_exec : يحضـر وينفذ عبارة SQL ويأخذ معرف الاتصال وسيط أول وعبارة SQL وسيط ثاني .

الـتابع odbc_fetch_row : يمسـك هذا التابع سجل من البيانات ، معرف النتيجة يشكل الوسيط الأول ورقم السطر وسيط ثاني.

الـتابع odbc_field_num : يولد هذا التابع رقم العمود المطابق لاسم حقل . معرف النتيجة يشكل الوسيط الأول واسم الحقل وسيط ثاني.

التابع odbc_field_type : يولد هذا التابع نوع الحقل الذي يكون رقمه وسيط ثاني ومعرف النتيجة وسيط أول.

الـتابع odbc_num_field : يولد هذا التابع عدد الأعمدة في نتيجة ما حيث يكون معرف النتيجة وسيط أول.

الـتابع odbc_num_rows : يـولد هذا التابع عدد الأسطر في نتيجة ما حيث يكون معرف النتيجة وسيط أول.

التابع odbc_prepare : يجهز هذا التابع عبارة SQL للتنفيذ ، معرف الاتصال يكون وسيط أول وعبارة SQL وسيط ثاني.

الـتابع odbc_result : يعطي هذا التابع محتويات حقل محدد اسمه ، أو رقمه كوسيط ثاني ومعرف النتيجة يكون وسيط أول.

الـتابع odbc_result_all : يطـبع هـذا التابع نتيجة تطابق عبارة SQL في جدول HTML .

التابع ord : يعيد هذا التابع قيمة الـ ASCII رمز ممرر كوسيط لهذا التابع.

الـتابع pathinfo : يعـيد هذا التابع معلومات حول مسار. حيث المسار يكون وسيط جيد.

مثال:

```
$path=pathinfo (“/www/htdoc/index.htm”);
echo $path {“dirname”}      \\ ==>    www/htdoc
echo $path {“basename”}    \\ ==>   index.htm
echo $path {“extension”}    \\ ==>   htm
```

التابع phpinfo : يعطي هذا التابع معلومات عن PHP ومكتباتا المتوفرة على الملقم.

التابع phpversion : يولد هذا التابع النسخة الحالة للـ PHP.

التابع pi : يعطي هذا التابع القيمة π التي تساوي ٣,١٤ .

التابع pos : يعطي هذا التابع العنصر الحالي للمصفوفة.

التابع pow : الشكل العام للتابع pow (num1,num2) .

هذا العدد يرفع العدد num1 إلى القوة num2 .

التابع print : يطبع هذا التابع قيمة سلسلة ممررة كوسيط للتابع.

التابع print_r : يطبع معلومات متحول بشكل قابل للقراءة للشخص.

مثال:

```
$array (‘a’ ==> ‘apple’, ‘b’ ==> ‘banana’, ‘c’==> Array(‘x’,’y’,’z’);
```

print_r ($a);

النتيجة تكون:

Array([0] ==> apple [b] ==> banana
C==> Array ([0]==> x [1]==> y [z]==> z))

الـتابع rad2deg : يحول هذا التابع الرقم الممرر له كوسيط من قيمة راديان إلى القيمة المكافئة بالدرجات.

الـتابع rand : يولد هذا التابع قيمة عشوائية، إذا لم نحدد له بارامترات، فإنه يولد قيمة عشـوائية من الصفر ، وقيمة أخرى هو يحددها ، وإذا حددنا رقمين min و max فإنه يولد قيمة عشوائية بين هذين الرقمين.

التابع readfile : يخرج محتويات ملف محدد اسمه كوسيط للتابع.

الـتابع rename : يستعمل هذا التابع لتغيير اسم ملف محدد اسمه الأول كوسيط أول ، والاسم الجديد كوسيط ثاني.

التابع rmdir : يحذف هذا التابع فهرس محدد اسمه كوسيط.

الـتابع round : يدور هذا التابع الرقم العشري إلى أقرب رقم صحيح ، مع الأخذ بعين الاعتبار دقة الرقم المطلوب.

مثال:

Round (3.4) ==> 3
Round (3.5) ==> 4
Round (3.6) ==> 4
Round (3.6,0) ==> 4
Round (1.95583.2) ==> 1.96
Round (1241757,-3) ==> 1242000

التابع sin : يحسب قيمة الجيب لزاوية.

التابع sinh : يحسب قيمة الجيب القطعي لزاوية.

التابع sleep : ينفذ هذا التابع تأخير زمني لفترة زمنية محددة بالثواني ومررة كوسيط.

التابع sqrt : يحسب هذا التابع الجذر التربيعي لعدد مرر كوسيط للتابع.

الـتابعstrcmp : هـذا التابع يتوقع تمرير وسيطين له من نوع السلاسل، ويقارن هاتين السلسلتين، فإذا كانت السلسلة الأولى str1 أكبر من السلسلة الثانية str2 فإنه يعيد رقم أكـبرمن الصـفر، ويعيد رقم أصغر من الصفر في حالة كانت السلسلة الثانية أكبر من السلسلة الأولى، ويعيد الصفر في حالة التساوي ، وهذا التابع حساس لحالة الأحرف،كما أنه يعتبر الأرقام عبارة عن أحرف أي السلسلة /٢/ أكبر من السلسلة/١٢/.

الشكل العام لهذا التابع من الشكل التالي:

Strcmp (str1,str2);

نقصـد بالسلسـلة الأولى أكـبر من الثانية أي السلسلة الأولى تأتي قبل الثانية بالترتيب الأبجدي.

التابع strcasecmp: وهذا التابع شبيه بالتابع السابق عدا أنه حساس لحالة الأحرف.

الـتابع strnatcmp: هـذا الـتابع شبيه بالتابع strcmp،عدا أنه يأخذ بعين الإعتبار الأرقام،أي السلسلة/١٢/ سيعتبرها أكبر من السلسلة /٢/.

التابع strlen : يعيد هذا التابع طول سلسلة ممررة له كوسيط.

التابع strrev : يعكس عرض سلسلة ممررة له كوسيط.

Strrev (“hello”) ➔ olleh

التابع strpos : الشكل العامل للتابع هو:

str(string,char);

يقوم هذا التابع بالبحث عن أول وجود للرمز char داخل السلسلة string ، يعيد عدد يعبر عن موقع هذا الرمز ضمن السلسلة .

الـتابع strchr : هـذا الـتابع يطبق كما في التابع السابق ولكن البداية تكون من نهاية السلسلة String.

التابع strstr : هذا التابع يبحث عن سلسلة داخل سلسلة وهو شبيه بالتابع السابق ولكن عوضاً عن char يكون البحث عن سلسلة string.

الـتابع strtolower : هـذا التابع يحول السلسلة الممررة لهذا التابع من حالة الأحرف الكبيرة إلى حالة الأحرف الصغيرة.

الـتابع strtoupper : هـذا التابع يحول السلسلة الممررة لهذا التابع من حالة الأحرف الصغيرة إلى حالة الأحرف الكبيرة.

التابع strtr : الشكل العام لهذا التابع هو:

strtr (string,from,to)

هذا التابع يبحث عن السلسلة from ضمن السلسلة string ويبدلها بالسلسلة to .

التابع substr :يقتطع هذا التابع سلسلة من سلسلة أصلية ، الشكل العام لهذا التابع هو:

substr (string,start,length);

مثال:

substr (“abcdef”,1); ==> “bcdef”
substr (“abcdef”,1,3); ==> “bcd”
substr (“abcdef”,0,4); ==> “abcd”
substr (“abcdef”,0,8); ==> “abcdef”

$strin {0} ==> a
$strin {3} ==> d

المثالين الأخيرين طريقة ثانية لاختيار رمز واحد فقط من سلسلة.

التابع substr_count : هذا التابع يعيد عدد مرات تكرار سلسلة ضمن سلسلة .

مثال:

substr_count (“this is a test” , ”is”); ==> 2

التابع tan : يعيد قيمة الظل لزاوية .

التابع tanh : يعيد قيمة الظل القطعي لزاوية.

التابع time : يعيد هذا التابع قيمة الزمن بنظام unix .

التابع tmpfile : يولد هذا التابع ملف مؤقت.

التابع ucfirst : يحول هذا التابع الرمز الأول من السلسلة إلى حرف كبير.

التابع ucword : يحول هذا التابع الرمز الأول من كل كلمة ضمن سلسلة ما إلى حرف كبير.

التابع unlike : هذا التابع يحذف ملف اسمه محدد كوسيط للتابع.

التابع unset : يزيل تيئة متحول.

التابع usleep : ينفذ التابع sleep بحيث يكون الوسيط بالميكرو ثانية في حين نذكر أن التابع sleep ينفذ التأخير بالثانية.

الـتابع wordwrap : هـذا التابع يطبع سلسلة ممررة كوسيط أولي على سطرين بحيث يكون طول السطر الأول محدد بالوسيط الثاني.

مثال:

```
$text = the quick brown fox jumped over the lazy dog
$newtext = wordwrap ($text,20);
echo “$newtext”;
```

النتيجة:

```
The quick brown fox
jumped over the lazy dog
```