

PHP - Functii

Unul din motivele pentru care PHP are o functionalitate extrem de mare este disponibilitatea unui numar foarte mare de functii. Numarul acestora variaza in functie de instalarea PHP, dar in general, PHP pune la dispozitie cel putin 700 de functii variate, mergand pana la aproape 2000 de functii, daca sunt active toate extensiile oficiale. Din fericire, nu este necesar sa cunoastem toate aceste functii pentru a fi productivi in PHP. Intrucat functiile sunt organizate pe categorii, in general, este relativ usor sa gasim functiile care ne intereseaza. O lista organizata de categorii putem vedea de exemplu la adresa: <http://www.php.net/manual/en/funcref.php>.

De obicei, functiile furnizate de o extensie PHP sunt prefixate de numele extensiei. De exemplu, este usor de vazut ca functiile `mysql_fetch_array`, `mysql_connect`, `mysql_select_db` fac parte din aceeasi categorie.

Functii pentru lucrul cu array-uri

- `sort($arr)` - sorteaza un array in ordine crescatoare, dupa valori
- `rsort($arr)` - sorteaza un array in ordine descrescatoare, dupa valori
- `ksort($arr)` - sorteaza un array in ordine crescatoare, dupa chei
- `krsort($arr)` - sorteaza un array in ordine descrescatoare, dupa chei
- `in_array($element,$arr)` - verifica daca un element exista intr-un array
- `count()` - returneaza numarul de elemente dintr-un array

Functii pentru lucrul cu variabile si tipuri de date

- `is_array($var)` - verifica daca o variabila este de tip array
- `is_int($var)` - verifica daca o variabila este de tip numar intreg
- `is_string($var)` - verifica daca o variabila este de tip string
- `isset($var)` - verifica daca o variabila a fost setata
- `unset($var)` - elibereaza memoria folosita de o variabila
- `serialize($var)` - serializeaza o variabila PHP sub forma de string
- `unserialize($var)` - functie inversa serializarii

Functii pentru prelucrarea de text

chr(\$val) - returneaza un caracter cunoscandu-se valoarea ASCII

ord(\$char) - returneaza valoarea ASCII a unui caracter

explode(\$separator, \$string) - imparte un string in mai multe parti

```
$str = "AA,BB,CC,DD";  
$lista = explode(',', $str);  
// lista va fi array('AA','BB','CC','DD');
```

htmlspecialchars(\$str) - converteste toate caracterele din \$str care au semnificatie speciala in HTML (ex. >, <, &), in entitati ce pot fi afisate corect intr-un browser

```
echo htmlspecialchars("<body>"); // va afisa "&lt;body&gt;"
```

htmlspecialchars(\$str) - converteste din \$str doar caracterele &, <, > si optional, si ghilimelele

implode(\$str) - uneste toate elementele unui array intr-un singur string

ltrim(\$str), rtrim(\$str), trim(\$str) - elimina anumite caractere din stanga, dreapta sau ambele parti ale unui string. Caracterele eliminate sunt spatiile albe, tab-urile, caracterele newline

nl2br(\$str) - converteste dintr-un string toate caracterele de linie noua (\n) in codul HTML
, pentru afisare intr-un browser

str_replace(\$cauta, \$inlocuieste, \$str) - inlocuieste orice aparitie a stringului \$cauta cu \$inlocuieste, in stringul \$str, si returneaza rezultatul inlocuirii

strip_tags(\$str) - elimina complet tag-urile HTML si PHP dintr-un string

strlen(\$str) - returneaza dimensiunea stringului

strpos(\$str, \$cauta) - cauta prima aparitie a stringului \$cauta in \$str

strtolower(\$str) - converteste un string la litere mici

strtoupper(\$str) - converteste un string la litere mari

substr(\$str, \$start, [\$nr]) - returneaza o parte dintr-un string, incepand de la pozitia \$start (primul caracter e 0), si continuand cu un numar optional \$nr de caractere

OBS - daca se doreste extragerea unor caractere incepand de la sfarsitul stringului, se va folosi o valoare negativa pentru \$start

```
echo substr("abcdef", -2); // afiseaza "ef"
```

wordwrap(\$str, \$nr) - formateaza un string astfel incat sa fie \$nr caractere pe o linie

Functii pentru fisiere si directoare

In PHP, modul de lucru cu fisiere este aproape identic cu cel din C, existand un pointer de fisier care indica pozitia actuala la care se va scrie sau se va citi ceva, relativ la primul octet al fisierului.

fopen(\$file,\$mode) - deschide un fisier, in modul de lucru specificat in \$mode

O parte din specificatorii de mod sunt:

- r doar citire
- w doar scriere (daca fisierul nu exista il va crea)
- a doar scriere la sfarsitul fisierului

fread(\$resursa,\$n) - citeste pana la \$n octeti din fisierul \$resursa

filesize(\$filename) - returneaza dimensiunea in octeti a fisierului

feof() - returneaza TRUE daca pointerul curent este la sfarsitul fisierului, si FALSE in caz contrar

Exemplu pentru fopen, fread si feof:

```
<?php
    $handle = fopen("file.txt", "r");
    $contents = '';
    while (!feof($handle))
    {
        $contents .= fread($handle, 8192);
    }
    fclose($handle);
    echo "continutul fisierului este: $contents";
?>
```

In PHP, este posibil ca fisierul sa fie extern, de exemplu o pagina web :

```
<?php
    $handle = fopen("http://www.google.ro", "r");
    $contents = '';
    while (!feof($handle)) {
        $contents .= fread($handle, 8192);
    }
    fclose($handle);
    echo "continutul fisierului este: $contents";
?>
```

fwrite() - scrie un string intr-un fisier

```
<?php
    $fp = fopen('data.txt', 'w');
    fwrite($fp, 'aaaaa');
    fwrite($fp, '11111');
    fclose($fp);
    // continutul 'data.txt' este acum 'aaaaa11111'
?>
```

`file_get_contents($fisier)` - functie simpla pentru citirea continutului unui fisier
`file_put_contents($fisier,$str)` – functie simpla pentru scrierea unui string intr-un fisier
`is_dir($f)` - returneaza TRUE daca fisierul \$f este un director, si FALSE in caz contrar
`is_file($f)` - returneaza TRUE daca fisierul \$f este un fisier, si FALSE in caz contrar
`is_readable($f)` - returneaza TRUE daca fisierul/directorul \$f se poate citi/scana
`is_writable($f)` - returneaza TRUE daca fisierul/directorul \$f este accesibil pentru scriere
`mkdir($dir)` - creeaza un director nou
`unlink($f)` - sterge un fisier
`rmdir($d)` - sterge un director
`rename($nume_vechi,$nume_nou)` - redenumeste un fisier
`move_uploaded_file($fisier,$destinatie)` - muta un fisier care a fost uploadat

Functii pentru data

`date($format,[$unixtime])` - returneaza o data cu un format specific

```
echo date("d-m-Y") // afiseaza data curenta, in format zi-luna-an
```

OBS - specificatorii pentru format sunt descrisi detaliat in documentatia PHP pentru `date()`

OBS - UNIX TIME reprezinta timpul curent masurat în numar de secunde începând cu Epoca UNIX (1 ianuarie 1970 00:00:00 GMT). Intern, este reprezentat de un numar intreg pe 32 de biti.

`time()` - returneaza timpul curent, in format UNIX.

`mktime (lista de parametri)` - creeaza o data in formatul unix

```
echo date("M-d-Y",mktime (0,0,0,10,20,2008)); //afiseaza 20 oct 2008
```

`strtotime ()` - creeaza o data unix relativ la o data specificata(sau cea curenta)

```
echo date ("d-m-Y", strtotime ("+1 week")); //data care va fi peste o saptaman  
echo date ("d-m-Y", strtotime ("-1 week")); // data cu o saptamana in urma
```

Functii diverse

`md5()` - creeaza un hash in format md5

`sha1()` - creeaza un hash in format sha1

`base64encode($str)` - codeaza un string in formatul MIME base64

`base64_decode($str)` - decodeaza un string din formatul MIME base64 intr-un string normal

`eval($code)` - evalueaza un string sub forma de cod PHP

`rand()` - genereaza un numar intreg aleator, si poate primi ca parametru un interval dorit

`rand(10)` // returneaza un numar mai mare sau egal cu 10

`rand(20,30)` // returneaza un numar intre 20 si 30 (inclusiv)