PHP

Genera una password casuale, ma leggibile, a partire da una serie di parole definite

```
function random_readable_pwd($length=10){
    $words = 'dog,cat,sheep,sun,sky,red,ball,happy,ice,';
    $words .= 'green,blue,music,movies,radio,green,turbo,';
    $words .= 'mouse,computer,paper,water,fire,storm,chicken,';
    $words .= 'boot,freedom,white,nice,player,small,eyes,';
    $words .= 'path,kid,box,black,flower,ping,pong,smile,';
    $words .= 'coffee,colors,rainbow,plus,king,tv,ring';

   $words = explode(',', $words);
    if (count($words) == 0){ die('Wordlist is empty!'); }

    $pwd = '';
    while (strlen($pwd) < $length){
        $r = mt_rand(0, count($words)-1);
        $pwd .= $words[$r];
    }
     $num = mt_rand(1, 99);
    if ($length > 2){
        $pwd = substr($pwd,0,$length-strlen($num)).$num;
    } else {
        $pwd = substr($pwd, 0, $length);
    }
     return $pwd;
 }

// random_readable_pwd(10) => ritorna qualcosa di simile a: pingwater6, radiohap28, sunwhite84, happykid44,
```

Calcola la dimensione di un database MySQL

```
function CalcFullDatabaseSize($database, $db) {

    $tables = mysql_list_tables($database, $db);
    if (!$tables) { return -1; }

    $table_count = mysql_num_rows($tables);
    $size = 0;

    for ($i=0; $i < $table_count; $i++) {
        $tname = mysql_tablename($tables, $i);
        $r = mysql_query("SHOW TABLE STATUS FROM ".$database." LIKE '".$tname."'");
        $data = mysql_fetch_array($r);
        $size += ($data['Index_length'] + $data['Data_length']);
    };

    $units = array(' B', ' KB', ' MB', ' GB', ' TB');
    for ($i = 0; $size > 1024; $i++) { $size /= 1024; }
    return round($size, 2).$units[$i];
}

// Esempio d'uso
// apre la connessione mysql:
$handle = mysql_connect('localhost', 'user', 'password');

if (!$handle) { die('Connection failed!'); }

// recupera la dimensione di tutte le tabelle:
print CalcFullDatabaseSize('customer1234', $handle);
// --> ritorna il valore trovato, qualcosa come: 352.2 KB

// chiude la connessione:
mysql_close($handle);
```

Quantifica la somiglianza fra due parole

```
$word2compare = "stupid";

$words = array(
    'stupid',
    'stu and pid',
    'hello',
    'foobar',
    'stpid',
    'upid',
    'stuuupid',
    'sstuuupiiid',
);

while(list($id, $str) = each($words)){
    similar_text($str, $word2compare, $percent);
    print "Comparing '$word2compare' with '$str': ";
    print round($percent) . "%\n";
}
```

Contare i secondi impiegati da un processo

```
// mt_get: ritorna il microtime corrente
function mt_get(){
    global $mt_time;
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

// mt_start: avvia il contatore microtime
function mt_start(){
    global $mt_time; $mt_time = mt_get();
}

// mt_end: calcola il tempo trascorso
function mt_end($len=4){
    global $mt_time;
    $time_end = mt_get();
    return round($time_end - $mt_time, $len);
}
```

Genera una stringa casuale

```
function RandomString($len){
    $randstr = '';
    srand((double)microtime()*1000000);
    for($i=0;$i<$len;$i++){
        $n = rand(48,120);
```

```
      while (($n >= 58 && $n <= 64) || ($n >= 91 && $n <= 96)){
         $n = rand(48,120);
      }
      $randstr .= chr($n);
   }
   return $randstr;
}
```

Ritorna un valore "umano" per la dimensione di un file

```
function HumanReadableFilesize($size) {


   $mod = 1024;

   $units = explode(' ','B KB MB GB TB PB');
   for ($i = 0; $size > $mod; $i++) {
      $size /= $mod;
   }

   return round($size, 2) . ' ' . $units[$i];
}
```

Cancella i duplicati in un array

```
function remove_duplicated_values($arr){
   $_a = array();
   while(list($key,$val) = each($arr)){
      $_a[$val] = 1;
   }
   return array_keys($_a);
}
```

Effettuare il trim di un array in maniera ricorsiva

```
function TrimArray($Input){

   if (!is_array($Input))
      return trim($Input);

   return array_map('TrimArray', $Input);
}
```

Genera un colore in modo casuale

```
function random_color(){
   mt_srand((double)microtime()*1000000);
   $c = '';
   while(strlen($c)<6){
      $c .= sprintf("%02X", mt_rand(0, 255));
   }
   return $c;
}
```

Permettere il download di un file

```
$local_file = 'test.zip';
$download_file = 'your-download-name.zip';

if(file_exists($local_file) && is_file($local_file)) {
   // send headers
   header('Cache-control: private');
   header('Content-Type: application/octet-stream');
   header('Content-Length: '.filesize($local_file));
   header('Content-Disposition: filename='.$download_file);

   // flush content
   flush();

   // apre lo stream
   $file = fopen($local_file, "rb");
```

```
    // invia il file al browser
    print fread ($file, filesize($local_file));

    // chiude lo stream
    fclose($file);}
else {
    die('Error: The file '.$local_file.' does not exist!');
}
```

Converte un porzione di testo in link

```
function text2links($str='') {

    if($str=='' or !preg_match('/(http|www\.|@)/i', $str)) { return $str; }

    $lines = explode("\n", $str); $new_text = '';
    while (list($k,$l) = each($lines)) {
        $l = preg_replace("/([ \t]|^)www\./i", "\\1http://www.", $l);
        $l = preg_replace("/([ \t]|^)ftp\./i", "\\1ftp://ftp.", $l);

        $l = preg_replace("/(http:\/\/[^ )\r\n!]+)/i",
            "<a href=\"\\1\">\\1</a>", $l);

        $l = preg_replace("/(https:\/\/[^ )\r\n!]+)/i",
            "<a href=\"\\1\">\\1</a>", $l);

        $l = preg_replace("/(ftp:\/\/[^ )\r\n!]+)/i",
            "<a href=\"\\1\">\\1</a>", $l);

        $l = preg_replace(
            "/([-a-z0-9_]+(\.[_a-z0-9-]+)*@([a-z0-9-]+(\.[a-z0-9-]+)+))/i",
            "<a href=\"mailto:\\1\">\\1</a>", $l);

        $new_text .= $l."\n";
    }

    return $new_text;
}
```

Mostra i contenuti di una directory ordinandoli per data

```
function listdir_by_date($path){
    $dir = opendir($path);
    $list = array();
    while($file = readdir($dir)){
        if ($file != '.' and $file != '..'){
            // add the filename, to be sure not to
            // overwrite a array key
            $ctime = filectime($data_path . $file) . ',' . $file;
            $list[$ctime] = $file;
        }
    }
    closedir($dir);
    krsort($list);
    return $list;
}
```