

# APLIKACJE INTERNETOWE PHP



## TEMAT 3-05: Funkcje systemu plików.

Autor dokumentu: Wojciech Galiński

środa, 7 września 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>, <http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

## Zagadnienia obowiązkowe

1. **Funkcje biblioteczne** – język PHP dysponuje bardzo rozbudowaną biblioteką funkcji. Żaden inny język programowania nie ma tak bogatej biblioteki. Dokładne opisy funkcji wraz z przykładami znajdują się w podręczniku PHP (patrz: adres na końcu każdej kategorii).

W następnych punktach opisane zostały najważniejsze kategorie.

2. **Funkcje związane z katalogami** – wyróżniamy (więcej: <http://php.net/manual/pl/ref.dir.php>):

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>getcwd</b>	Zwraca adres bieżącego katalogu	<code>echo getcwd();</code>	<b>opendir, closedir, readdir, rewinddir</b>	Obsługa katalogu za pomocą wskaźnika katalogu ( <b>otwarcie / zamknięcie / czytanie / przewinięcie na początek katalogu</b> )	<pre>if (\$h=opendir('katalog')) { echo "Uchwył:\$h\n";   "Pliki:\n";   while ((\$f=readdir(\$h))     !==false)echo "\$f\n";   rewinddir(\$h);closedir(\$h); }</pre>
<b>chdir</b>	Zmienia bieżący katalog.	<pre>echo getcwd(); chdir('images'); echo getcwd();</pre>	<b>dir</b>	Obiektowy odpowiednik funkcji: opendir, readdir, rewinddir, closedir.	<pre>\$d = dir("/tmp"); echo 'Uchwył: '.\$d-&gt;handle. '&lt;br /&gt;Ścieżka: ' . \$d-&gt;path.'&lt;br /&gt;'; while ( (\$entry=\$d-&gt;read())   !==false )   echo \$entry.'&lt;br /&gt;'; \$d-&gt;close();</pre>
<b>scandir</b>	Wypisuje wszystkie pliki i katalogi z określonej lokalizacji.	<pre>\$dir = '/tmp'; print_r(scandir(\$dir)); print_r(scandir(\$dir, 1));</pre>			

3. **Funkcje systemu plików** – na niektórych serwerach, dostęp do tych funkcji jest ograniczony lub wręcz zablokowany ze względów bezpieczeństwa (więcej: <http://php.net/manual/pl/ref.filesystem.php>).

→ **tworzenie i usuwanie katalogów, plików oraz dowiązań** – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>mkdir, rmdir</b>	<b>Tworzy / usuwa</b> katalog.	<pre>\$d = 'katalog'; if (!file_exists(\$d))   \$b=mkdir(\$d); else \$b = rmdir(\$d); if (\$b) echo 'Sukces!';</pre>	<b>copy</b>	Kopiuje plik.	<pre>\$b=copy('p.txt','p2.txt'); if (!\$b) echo 'Błąd podczas ' . 'kopiowania pliku!';</pre>
<b>link, symlink</b> (tylko lokalnie, tylko Linux)	Tworzy dowiązanie <b>twarde / symboliczne</b> .	<pre>\$fn = 'plik.txt'; \$l = 'link'; \$sl = 'symlink'; \$b1 = link(\$fn, \$l); \$b2 = symlink(\$fn, \$l); var_dump(\$b1, \$b2);</pre>	<b>unlink</b>	Usuwa plik.	<pre>\$fn='skasuj_to.html'; if (unlink(\$fn))   echo 'Plik został usunięty.';</pre>
			<b>rename</b>	Zmienia nazwę pliku lub katalogu.	<pre>\$f1='nazwa_pliku.txt'; \$f2='nowa_nazwa.txt'; if (rename(\$f1, \$f2))   echo 'Zmiana nazwy pliku: OK';</pre>

→ **obsługa plików tymczasowych** – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>tempnam</b>	Generuje unikalną nazwę dla pliku tymczasowego.	<pre>\$s = 'Prefiks_'; echo tempnam('/tmp', \$s);</pre>	<b>tmpfile</b>	Tworzy plik tymczasowy.	<pre>\$ftmp = tmpfile(); fwrite(\$ftmp,'tekst'); fseek(\$ftmp, 0); echo fread(\$ftmp, 1024); // Zamyka i kasuje plik fclose(\$ftmp);</pre>

## ➔ funkcje wykorzystujące wyrażenia regularne – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>fnmatch</b>	Dopasowanie tekstu do wyrażenia regularnego.	<pre>\$kolor = 'lightgray'; \$wyr="*gr[ae]y"; if (fnmatch(\$wyr,\$kolor))     echo 'Oto jeden z ' .         'kolorów "gray"';</pre>	<b>glob</b>	Dopasowywanie nazw plików do wyrażenia regularnego.	<pre>foreach     (glob("*.php") as \$fn)     echo "\$fn (rozmiar: " .         filesize(\$fn).')&lt;br /&gt;';</pre>

## ➔ funkcje informacyjne związane z systemem plików – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>disk_total_space, disk_free_space, diskfreespace</b>	Zwraca ilość całkowitego / wolnego miejsca w systemie plików.	<pre>echo     disk_total_space('.').     ' '.     disk_free_space('.');</pre>	<b>dirname, basename, realpath, path_info</b>	Zwraca informacje o ścieżce do pliku.	<pre>\$f='/tmp/plik'; echo dirname(\$f).'     '.basename(\$f); \$f='..'; echo ' '.realpath(\$f); \$tab=pathinfo('.'); print_r(\$tab);</pre>
<b>file_exists</b>	Sprawdza czy plik lub katalog istnieje.	<pre>\$fn='plik.html'; \$b=file_exists(\$fn); if (\$b) file(\$f);</pre>	<b>filetime, filemtime, filectime</b>	Pobiera ostatni czas dostępu do / modyfikacji / i-węzłowy zmiany do pliku	<pre>\$fn='..';     filetime(\$fn).' '.     filemtime(\$fn).' '.     filectime(\$fn).' '.     fileperms(\$fn).' '.     fileowner(\$fn).' '.     filegroup(\$fn);</pre>
<b>is_dir, is_executable, is_file, is_link, is_readable, is_writable</b>	Mówi czy zbiór danych jest katalogiem / plikiem wykonywalnym / zwykłym plikiem / dowiązaniem symbolicznym / możliwy do odczytania / zapisywalny.	<pre>\$fn = \$_SERVER     ['script_filename']; var_dump(is_dir(\$fn),     is_executable(\$fn),     is_file(\$fn),     is_link(\$fn),     is_readable(\$fn),     is_writable(\$fn));</pre>	<b>fileperms, fileowner, filegroup</b>	Pobiera prawa dostępu / właściciela / grupę pliku.	
			<b>fileinode, filesize, filetype</b>	Pobiera i-węzeł / rozmiar / typ pliku.	<pre>\$fn = '..'; print_r(stat(\$fn)); echo fileinode(\$fn).' '.     filesize(\$fn).' '.     filetype(\$fn); \$fn = '/vmlinuz'; print_r(lstat(\$fn)); echo readlink(\$fn);</pre>
			<b>stat, fstat, lstat, readlink</b>	Pobiera informacje o pliku / dowiązaniu symbolicznym / celu dowiązania symbolicznego.	

## ➔ Zarządzanie prawami dostępu do elementów systemu plików – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>chmod, chown, lchown, chgrp, lchgrp</b>	Zmienia prawa dostępu / właściciela / grupę dla pliku / linku symbolicznego.	<pre>\$fn='plik.php'; chmod(\$fn, 755); chmod(\$fn, 'u=rwx,go=rx'); chown(\$fn, 'root');</pre>	<b>touch</b>	Zmienia czas modyfikacji pliku.	
			<b>umask</b>	Ustawia maskę praw dostępu dla tworzonych plików.	<pre>\$old = umask(0); touch('/tmp/plik'); umask(\$old);</pre>

## ➔ odczytywanie i zapisywanie danych w plikach – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>file_put_contents, file_get_contents, file, readfile</b>	Zapisuje zawartość zmiennej tekstowej do pliku. Czyta całą zawartość pliku do zmiennej tekstowej / do tablicy / na ekran.	<pre>\$fn = 'plik.txt'; \$tekst = 'ZSiPO'; \$t=file_put_contents     (\$fn, \$tekst); echo     file_get_contents(\$fn); print_r(file(\$fn)); readfile(\$fn);</pre>	<b>fputcsv, fgetcsv</b>	Wysyła do / Pobiera z pliku wiersz danych w formacie CSV.	<pre>\$t=array(1,2,'a','"b"'); \$f=fopen('a.csv','w'); fputcsv(\$f, \$t); fclose(\$f); readfile('a.csv');</pre>
			<b>parse_ini_string, parse_ini_file</b>	Odczytuje dane w formacie INI z pliku/tekstu.	<pre>\$s = "a=b&amp;nc=d"; \$t=parse_ini_string(\$s); print_r(\$t);</pre>

## Zadania

1. Pobierz następujące informacje o uruchamianym skrypcie: identyfikator właściciela i grupy pliku, rozmiar, typ pliku, numer i-węzła. Pobierz jego zawartość raz do tablicy, a raz jako tekst i wyświetl w przeglądarce.
2. Utwórz katalog i zapisz do niego 100 plików (nazwy numerowane, np. plik1, plik2, ...). W każdym z tych plików ma być 100 kolejnych liczb pierwszych (razem 10000 liczb pierwszych). Następnie odczytaj te pliki i wyświetl zawartość w przeglądarce. Poinformuj, który tekst należy do jakiego pliku.
3. Zapisz o pliku daty wszystkich niedziel w bieżącym roku rozdzielając je spacją. Następnie odczytaj je zapisz do tablicy i wyświetl tę tablicę.
4. Zapisz informacje o wybranych książkach do pliku konfiguracyjnego typu „ini”.
5. Zapisz listę 5 kolorów do pliku typu „csv”.