

# APLIKACJE INTERNETOWE PHP



## TEMAT 3-02: Funkcje obsługi liczb i zmiennych.

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: <http://php.net/manual/pl/funcref.php>).

W następnych punktach opisane zostały najważniejsze kategorie wraz z odsyłaczami do pełnej pomocy na temat opisanych funkcji.

2. **Funkcje matematyczne** – (więcej: <http://php.net/manual/pl/ref.math.php>).

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>M_E, M_SQRT2, M_SQRT3, M_SQRT1_2, NAN, INF</b>	Stałe matematyczne: Pi, stała Eulera, $\sqrt{2}$ , $\sqrt{3}$ , $\sqrt[3]{2}$ , nie liczba (Not a Number), nieskończoność (Infinity).	echo 2*M_E; echo M_SQRT2; echo M_SQRT3; echo M_SQRT1_2; echo NAN. ' '. INF;	<b>exp, log, log10</b>	Funkcje logarytmiczne: $e^x$ , $\ln x$ , $\log_b x$ , $\log_{10} x$ .	echo exp(1); echo log(pow(M_E, 2)); echo log(8, 2); echo log10(1000);
<b>pi, M_PI</b>	Wartość Pi	echo pi. ' '. M_PI;	<b>mod, fmod</b>	Reszta z dzielenia (modulo) dla liczb całkowitych i zmiennoprzecinkowych	echo mod(5, 3); echo mod(6, 3); echo mod(7, 3); echo fmod(7.5, 3.3);
<b>abs</b>	Wartość bezwzględna (moduł z liczby)	echo abs(-5);	<b>lcg_value</b>	Losuje liczbę zmiennoprzecinkową z zakresu [0,1).	echo lcg_value();
<b>bindec, decbin, oct, dec, decoct, hexdec, dechex, base_convert</b>	Konwersja pomiędzy liczbami dziesiętnymi, a: binarnymi / ósemkowym / szesnastkowymi / podstawa liczbowa z zakresu 2-36.	echo bindec('110'); echo decbin(11); echo octdec('110'); echo decoct(11); echo hexdec('110'); echo dechex(11); echo base_convert('7', 8, 2);	<b>srand, rand, getrandmax, mt_srand, mt_rand, mt_getrandmax</b>	Generatory liczb pseudolosowych (funkcje „mt_” reprezentują lepszy generator liczb pseudolosowych).	echo getrandmax(). ' '; echo rand(). '  '; echo rand(-5, 5). ' ' ; srand(20); echo rand(1, 10);
<b>ceil, floor, round</b>	Zaokrąglenie liczby w górę / w dół / według zasad matematyki.	echo ceil(2.1); echo ceil(-2.1); echo floor(2.9); echo floor(-2.9); echo round(2.5); echo round(-2.5);	<b>is_finite, is_infinite, is_nan</b>	Sprawdza, czy liczba jest/nie jest liczbą z zakresu obsługiwanego w PHP; sprawdza, czy jest to wartość liczbowa.	echo is_finite(1E308); echo is_finite(1E309); echo is_infinite(1E308); echo is_infinite(1E309); var_dump(is_nan('0')); \$nan = is_nan(sqrt(-1)); var_dump(\$nan);
<b>sin, cos, tan, deg2rad</b>	Funkcje trygonometryczne (parametr to kąt w radianach).	echo sin(deg2rad(30)); echo cos(deg2rad(90)); echo tan(deg2rad(45));	<b>min, max</b>	Wartość minimalna/maksymalna z 2 podanych wartości.	echo min(4, 5); echo max(4, 5, 6); echo max(-4, 5, 6, 7);
<b>asin, acos, atan, rad2deg</b>	Funkcje odwrotne do trygonometrycznych (zwraca kąt w radianach)	echo ra2deg(asin(0.5)); echo ra2deg(acos(0.5)); echo ra2deg(atan(1.0));	<b>pow, sqrt</b>	Potęgowanie i pierwiastek kwadratowy	echo pow(3, 4); echo sqrt(289);
<b>number_format</b>	Formatuje liczby jako tekst.	echo number_format(1221.5, 2, ',', ' ');			

3. **Funkcje daty i czasu** – (więcej: <http://php.net/manual/pl/ref.datetime.php>).

Funkcje	Opis	Przykłady
<b>date_diff</b>	Oblicza różnicę pomiędzy dwoma datami.	\$dni = date_diff(\$sty, \$lip);
<b>date</b>	Pobiera i formatuje datę i czas. Patrz: <a href="http://php.net/manual/pl/function.date.php">http://php.net/manual/pl/function.date.php</a> .	echo date('Y-m-d H:i:s'); \$t = mktime(0,0,9,9,11,1); echo date('Y-m-d', \$t);
<b>getdate, localtime</b>	Pobiera datę i czas i przechowuje ją w tablicy.	print_r(getdate()); print_r(localtime(time(), true));
<b>time, microtime</b>	Pobiera datę i czas i przechowuje ją jako liczbę sekund i mikrosekund od początku epoki UNIX.	echo time(). ' '. microtime();
<b>strtotime</b>	Zamienia angielskojęzyczne nazwy czasu na znaczniki czasowe (np. now, +1 day, ...).	echo strtotime('now'). ' '. strtotime('last Monday'). ' '. strtotime('+1 week'). ' '. strtotime('-2 hours');
<b>mktime</b>	Oblicza znacznik czasu UNIX dla podanej daty i czasu.	echo date("j.m.Y", mktime (0,0,0,1,0,2012)). ' ' . date("j.m.Y", mktime (0,0,0,1,1,2012));
Funkcje	Opis	Przykłady
<b>checkdate</b>	Sprawdza prawdziwość podanej daty.	\$b1 = checkdate(12, 31, 2012); \$b2 = checkdate(2, 29, 2013); var_dump(\$b1, \$b2);
<b>date_format</b>	Formatuje datę i czas.	echo date_format(\$sty, 'Y-m-d H:i:s'). ' ' . \$dni->format('%R%a');

<b>date_parse</b>	Zamienia tekst sformatowany jako data i czas do tablicy ze składowymi daty i czasu.	<code>print_r(date_parse("2013-06-28 12:00:00.75"));</code>
<b>date_default_timezone_set</b>	Ustawia strefę czasową dla skryptu PHP.	<code>date_default_timezone_set('Europe/Warsaw');</code>
<b>date_create</b>	Tworzy zmienną typu data i czas.	<code>\$sty = date_create('2013-01-01');</code> <code>\$lip = date_create('2013-08-01');</code>
<b>date_offset_get</b>	Zwraca różnicę sekund, pomiędzy strefą czasową UTC, a bieżącą strefą czasową.	<code>echo date_offset_get(\$sty). ' '. date_offset_get(\$lip);</code>
<b>date_time_set</b>	Ustawia czas do zmiennej daty i czasu.	<code>\$czas = new DateTime(); date_time_set(\$czas, 12, 0, 0);</code> <code>echo \$czas-&gt;format('H:i:s');</code>
<b>date_timestamp_get, date_timestamp_set</b>	Pobiera/Ustawia czas jako liczbę sekund od 1 stycznia 1970 (początek epoki UNIX).	<code>\$czas = new DateTime(); date_timestamp_set(\$czas, 155);</code> <code>echo date_timestamp_get(\$czas);</code>
<b>date_sunrise, date_sunset</b>	Informuje o czasie wschodu i zachodu słońca w dowolnym miejscu na Ziemi (przedstawiony przykład dotyczy Nysy).	<code>echo date_sunrise(time(), SUNFUNCS_RET_STRING, 50.46, 17.33, 90, 1). ' '.</code> <code>date_sunset(time(), SUNFUNCS_RET_STRING, 50.46, 17.33, 90, 1);</code>

#### 4. Funkcje obsługi zmiennych – (więcej: <http://php.net/manual/pl/ref.var.php>).

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
<b>isset</b>	Sprawdza, czy zmienna istnieje i nie jest NULL.	<code>\$a = 50; \$b = NULL;</code> <code>var_dump(isset(\$a));</code> <code>var_dump(isset(\$b));</code> <code>var_dump(isset(\$c));</code>	<b>intval;</b> <b>floatval;</b> <b>doubleval;</b> <b>strval</b>	Konwertuje wartość do liczby całkowitej / zmiennoprzecinkowej / tekstu.	<code>\$txt='9.87654321098765a';</code> <code>echo strval(\$txt). ' '.</code> <code>intval(\$txt). ' '.</code> <code>floatval(\$txt). ' '.</code> <code>doubleval(\$txt);</code>
<b>is_bool;</b> <b>is_int;</b> <b>is_integer;</b> <b>is_long;</b> <b>is_float;</b> <b>is_double;</b> <b>is_real;</b> <b>is_numeric</b>	Sprawdza, czy wartość zmiennej jest wartością logiczną/ liczbą całkowitą/liczbą zmiennoprzecinkową/ wartością liczbową (np. w postaci tekstu).	<code>\$b1 = is_int('0');</code> <code>\$b2 = is_bool(\$b1);</code> <code>var_dump(\$b1, \$b2);</code> <code>\$b3 = is_float(2.0);</code> <code>var_dump(\$b3);</code> <code>\$b4=is_numeric('0');</code> <code>\$b5=is_numeric('0x');</code> <code>var_dump(\$b4, \$b5);</code>	<b>empty</b>	Sprawdza, czy zmienna ma jedną z wartości: "", 0, 0.0, "0", NULL, FALSE, array, \$var; (niezainicjalizowana zmienna)	<code>\$a=0; \$b=1;</code> <code>\$b1 = empty(\$a);</code> <code>\$b2 = empty(\$b);</code>  <code>\$c; \$b3 = empty(\$c);</code> <code>var_dump(\$a, \$b, \$c);</code>
<b>is_string.</b> <b>is_array,</b> <b>is_null</b>	Sprawdza, czy wartość zmiennej jest tekstem / tablicą / pusta.	<code>\$b6=is_string(0x2);</code> <code>\$b7=is_string('0x2');</code> <code>var_dump(\$b6, \$b7);</code> <code>\$b8=is_array(array());</code> <code>\$b9=is_array('1,3');</code> <code>\$a; \$nan=is_null(NAN);</code> <code>var_dump(is_null(\$a));</code> <code>var_dump(\$nan);</code>	<b>is_scalar</b>	Sprawdza, wartość jest jednym z następujących typów: integer, float, string, boolean.	<code>\$b1=is_scalar(5);</code> <code>\$b2=is_scalar(5.0);</code> <code>\$b3=is_scalar('5');</code> <code>var_dump(\$b1,\$b2,\$b3);</code> <code>\$bb4=is_scalar(5==0);</code> <code>\$bb5=is_scalar(NAN);</code> <code>\$t=array(1, 2, 3);</code> <code>\$bb6=is_scalar(\$t);</code> <code>var_dump(\$b4,\$b5,\$b6);</code>
<b>is_callable</b>	Sprawdza, czy wartość zmiennej jest nazwą funkcji lub metody obiektu.	<code>function f() {}</code> <code>\$b = is_callable('f');</code> <code>var_dump(\$b);</code>	<b>is_object</b>	Sprawdza, czy wartość zmiennej jest obiektem.	<code>\$czas = New Datetime();</code> <code>\$b = is_object(\$czas);</code> <code>var_dump(\$b);</code>
<b>print_r,</b> <b>var_export,</b> <b>var_dump</b>	Wyświetla informacje o zmiennej	<code>\$tab = array(1,2, 'ab'=&gt;3);</code> <code>print_r(\$tab);</code> <code>var_export(\$tab);</code> <code>var_dump(\$tab);</code>	<b>is_resource</b>	Sprawdza, czy wartość zmiennej jest zasobem.	<code>\$my = mysql_connect();</code> <code>\$b = is_resource(\$my);</code> <code>var_dump(\$b);</code>
<b>unset</b>	Usuwa zmienną z pamięci.	<code>\$a=3; \$b='?'; \$c=0.0;</code> <code>echo \$a;</code> <code>unset(\$a); echo \$a;</code>	<b>get_defined_vars</b> Zwraca tablicę z listą wszystkich zmiennych zdefiniowanych w skrypcie.		<code>\$a; \$i = 6; \$f = 0.0;</code> <code>\$s = 'tekst';</code> <code>\$t = get_defined_vars();</code> <code>print_r(\$t);</code>
<b>gettype,</b> <b>settype</b>	Pobiera/Ustawia typ zmiennej.	<code>\$a=5;echo gettype(\$a).' ';</code> <code>settype(\$a, 'double');</code> <code>echo gettype(\$a);</code>	<b>get_resource_type</b> Zwraca informacje o typie zasobu w zmiennej.		<code>\$my = mysql_connect();</code> <code>\$s=get_resource_type(\$my);</code> <code>echo \$s;</code>

## Zadania

1. Zapoznaj się z wymienionymi funkcjami i wypróbuj dołączone przykłady. Jeżeli czegoś nie rozumiesz, zajrzyj pod podane adresy.
2. Sprawdź, o której wschodzi i zachodzi słońce w miejscowości, w której mieszkasz.
3. Wyświetl bieżącą datę i czas, np. wtorek, 25 grudnia 2012 (358 dzień roku, rok przestępny).
4. Utwórz funkcje obliczające NWD i NWW dla dowolnych 2 liczb.
5. Sprawdź, dla jakich kątów sinus oraz cosinus równy jest 0.5.
6. Sprawdź, ile dni już żyjesz, a następnie wyświetl zmienne użyte do tego zadania.
7. Sprawdź, ile potrzeba losowań z przedziału [0, 1), żeby otrzymać wartość większą od „0.999”.
8. Wykonaj symulację 1000 losowań LOTTO i do każdego losowania wylosuj własne liczby. Ile pieniędzy to kosztowało i jaki jest zysk albo strata?