

Lekcja 4 – Funkcje matematyczne oraz daty

Wyświetlanie daty i czasu

PHP dysponuje wieloma gotowymi funkcjami umożliwiającymi wyświetlanie daty i czasu. Podstawową funkcją jest **date**. Jej wywołanie może mieć postać:

date (format[, znacznik_czasu])

Zwraca ona ciąg znaków we wskazanym formacie. Parametr format określa jakie dane nas interesują, natomiast znacznik_czasu to argument opcjonalny znacznik czasu Unixa (timestamp).

Tabela wartości dla parametru format funkcji date().

| Znak | Opis |
|------|---|
| a | "am" lub "pm" |
| A | "AM" lub "PM" |
| d | dwucyfrowy (z poprzedzającym zerem) dzień miesiąca, np. 02, 17, 31 |
| D | trzyliterowy dzień tygodnia, np. "Mon", "Fri" |
| F | nazwa miesiąca, np. "March", "July" |
| h | dwucyfrowa (z poprzedzającym zerem) godzina w formacie 12-godzinnym, wartości od 01 do 12 |
| H | dwucyfrowa (z poprzedzającym zerem) godzina w formacie 24-godzinnym, wartości od 00 do 23 |
| g | godzina w formacie 12-godzinnym, wartości od 1 do 12 |
| G | godzina w formacie 24-godzinnym, wartości od 0 do 23 |
| i | dwucyfrowa (z poprzedzającym zerem) wartość określająca minuty, wartość od 00 do 59 |
| j | dzień miesiąca bez poprzedzającego zera, wartości od 1 do 31 |
| l | pełna nazwa dnia tygodnia, np. "Monday", "Friday" |
| L | wartość boolowska dla określenia roku przestępnego, jeśli rok przestępny to wartość "1", zaś jeśli nie to "0" |
| m | dwucyfrowa wartość określająca miesiąc, wartości od 01 do 12 |
| n | miesiąc bez poprzedzającego zera, wartość od 1 do 12 |
| M | trzyliterowy miesiąc, np. "Feb", "Jul" |
| s | dwucyfrowa wartość określająca sekundy, wartości od 00 do 59 |
| S | angielski przyrostek dla liczebników (...), np. "th", "nd" |
| t | ilość dni w danym miesiącu, wartości od 28 do 31 |
| U | ilość sekund od 1 stycznia 1970r |
| w | wartość numeryczna określająca dzień tygodnia, np. 0 (niedziela), 6 (sobota) |
| Y | czterocyfrowa wartość określająca rok, np. 1997, 2001 |
| y | dwucyfrowa wartość określająca rok, np. 99, 01 |
| z | dzień roku, wartości od 0 do 365 |

| | |
|---|--|
| W | Numer tygodnia w roku zgodnie ze standardem ISO-8601 |
| w | Dzień tygodnia w postaci numerycznej |

Przykład:

```
echo (date("l dS of F Y h:i:s A"), time());
// pokaże na ekranie tekst w stylu:
// Wednesday 12th of November 2001 06:26:14 PM
```

```
<?php
echo ("date('l\"Y-m-dl') = " . date("Y-m-d") . "<br>");
echo ("date('l\"d-m-Yl') = " . date("d-m-Y") . "<br>");
echo ("date('l\"j, M Yl') = " . date("j, M Y") . "<br>");
echo ("date('l\"jS, M Yl') = " . date("jS, M Y") . "<br>");
echo ("date('l\"G:i:sl') = " . date("G:i:s") . "<br>");
echo ("date('l\"h:i:sal') = " . date("h:i:sa") . "<br>");
echo ("date('l\"Y-m-d G:i:sl') = " . date("Y-m-d G:i:s") . "<br>");
?>
```

Drugą funkcją pozwalającą na pobranie informacji dotyczących daty i czasu jest **getdate**. Jek wywołanie ma postać:

```
getdate([znacznik_czasu])
```

Również w tym przypadku znacznik_czasu jest parametrem opcjonalnym. Wynikiem nie jest tym razem ciąg znaków, ale tablica asocjacyjna zawierająca pobrane dane. Poniższy przykład obrazuje w jaki sposób wykorzystać getdate do wyświetlania informacji o aktualnej dacie

```
<?php
$data = getdate();
$dzien = $data["mday"];
$miesiac = $data["mon"];
$rok = $data["year"];
if($dzien < 10) $dzien = "0" . $dzien;
if($miesiac < 10) $miesiac = "0" . $miesiac;
echo("Dziś jest $dzien-$miesiac-$rok r.");
?>
```

Jeśli chcemy uzyskać polskie określenia musimy sami zadbać o napisanie odpowiedniej procedury. Poniższy przykład ilustruje takie rozwiązanie:

```
<?php
$data = getdate();
$dzien = $data["mday"];
$miesiac = $data["mon"];
$rok = $data["year"];
```

```
switch($miesiac){
  case 1 : $miesiac = "stycznia"; break;
  case 2 : $miesiac = "lutego"; break;
  case 3 : $miesiac = "marca"; break;
  case 4 : $miesiac = "kwietnia"; break;
  case 5 : $miesiac = "maja"; break;
  case 6 : $miesiac = "czerwca"; break;
  case 7 : $miesiac = "lipca"; break;
  case 8 : $miesiac = "sierpnia"; break;
  case 9 : $miesiac = "wrzeźnia"; break;
  case 10 : $miesiac = "pałdziernika"; break;
  case 11 : $miesiac = "listopada"; break;
  case 12 : $miesiac = "grudnia"; break;
}

echo("Dziś jest $dzien $miesiac $rok r.");
?>
```

W podobny sposób możemy uzyskać polską nazwę dnia tygodnia. Tym razem skorzystamy z funkcji date (można także korzystać z funkcji getdate).

```
<?php
$dzienTygodnia = date("w");

switch($dzienTygodnia){
  case 0 : $dzienTygodnia = "niedziela"; break;
  case 1 : $dzienTygodnia = "poniedziałek"; break;
  case 2 : $dzienTygodnia = "wtorek"; break;
  case 3 : $dzienTygodnia = "środa"; break;
  case 4 : $dzienTygodnia = "czwartek"; break;
  case 5 : $dzienTygodnia = "piątek"; break;
  case 6 : $dzienTygodnia = "sobota"; break;
}

echo("Dziś jest $dzienTygodnia.");
?>
```

Do uzyskiwania danych opisujących datę i czas w językach narodowych można również wykorzystać funkcję strftime, która zwraca ciąg znaków sformatowany zgodnie z szablonem przekazanym jej jako argument. Jej wywołanie ma postać:

strftime(format[, timestamp])

Znaczniiki formatujące dla funkcji strftime

Poniższe symbole są rozpoznawane w szablonie formatującym:

- %a - skrótowa nazwa dnia tygodnia zgodnie z lokalizacją
- %A - pełna nazwa dnia tygodnia zgodnie z lokalizacją
- %b - skrótowa nazwa miesiąca zgodnie z lokalizacją
- %B - pełna nazwa miesiąca zgodnie z lokalizacją
- %c - preferowana reprezentacja daty i czasu zgodnie z lokalizacją
- %C - numer wieku (rok podzielony przez 100 i skrócony do liczby całkowitej, przedział od 00 do 99)
- %d - dzień miesiąca jako liczba dziesiętna (przedział od 01 do 31)
- %D - to samo co %m/%d/%y
- %e - dzień miesiąca jako liczba dziesiętna, przy czym pojedyncza cyfra poprzedzona jest spacją (przedział od " 1" do "31")
- %g - tak jak %G, ale bez uwzględnienia wieku
- %G - rok w zapisie czterocyfrowym, powiązany z numerem tygodnia wg ISO. Symbol ten ma ten sam format i wartość jak %Y, z tym wyjątkiem, że jeśli numer tygodnia wg ISO należy do poprzedniego lub następnego roku, to poprzedni lub następny rok jest zwracany przez ten symbol.
- %h - tak jak %b
- %H - godzina jako liczba dziesiętna w systemie 24-godzinnym (przedział od 00 do 23)
- %I - godzina jako liczba dziesiętna w systemie 12-godzinnym (przedział od 01 do 12)
- %j - dzień roku jako liczba dziesiętna (przedział od 001 do 366)
- %m - miesiąc jako liczba dziesiętna (przedział od 01 do 12)
- %M - minuty jako liczba dziesiętna
- %n - znak nowej linii
- %p - albo "am" lub "pm" zgodnie z podanym czasem, albo łańcuchy znaków odpowiadające lokalizacji
- %r - czas w notacji a.m. lub p.m.
- %R - czas w notacji 24-godzinnej
- %S - sekundy jako liczba dziesiętna
- %t - znak tabulacji
- %T - aktualny czas, odpowiednik %H:%M:%S
- %u - numer dnia tygodnia jako liczba dziesiętna [1,7], gdzie 1 oznacza poniedziałek

Ostrzeżenie

Sun Solaris podaje niedzielę jako 1, pomimo że ISO 9889:1999 (aktualny standard języka C) jasno określa, że powinien to być poniedziałek.

- %U - numer tygodnia aktualnego roku jako liczba dziesiętna, począwszy od pierwszej niedzieli jako pierwszego dnia pierwszego tygodnia
- %V - numer tygodnia aktualnego roku wg ISO 8601:1988 jako liczba dziesiętna, przedział od 01 do 53, gdzie tydzień 1 jest pierwszym tygodniem, którym ma co najmniej 4 dni w aktualnym roku, przy czym pierwszym dniem tygodnia jest poniedziałek. (Przy użyciu %G lub %g otrzymuje się rok, który odpowiada numerowi tygodnia dla podanego znacznika czasu).

- %W - numer tygodnia aktualnego roku jako liczba dziesiętna, począwszy od pierwszego poniedziałku, jako pierwszego dnia pierwszego tygodnia
- %w - dzień tygodnia jako liczba dziesiętna, począwszy od niedzieli - numer 0
- %x - preferowana reprezentacja daty, zgodnie z lokalizacją, bez czasu
- %X - preferowana reprezentacja czasu, zgodnie z lokalizacją, bez daty
- %y - rok jako liczba dziesiętna, bez uwzględnienia wieku (przedział od 00 do 99)
- %Y - rok jako liczba dziesiętna, z wiekiem włącznie
- %Z - strefa czasowa, nazwa lub skrót
- %% - znak "%"

Notatka: Nie wszystkie symbole konwersji mogą być obsługiwane przez twoją bibliotekę C, co oznacza, że nie będą obsługiwane przez PHP-owską funkcję **strftime()**. Oznacza to, że np. %e, %T i %D (oraz być może inne) nie będą funkcjonować w Windows.

Dane zwracane przez strftime, a więc nazwy miesięcy itp. będą zgodne z ustawieniami lokalnymi (narodowymi), które mogą być zmieniane za pomocą funkcji **setlocale**.

Wywołanie setlocale ma postać:

Setlocale(kategoria, lokale)

Parametr *kategoria* wskazuje jakie informacje mają być pobierane z uwzględnieniem ustawień narodowych, natomiast *locale* określa, który zestaw narodowy ma być użyty.

Wartości pierwszego parametru funkcji setlocale

- LC_ALL Wpływa na wszystkie parametry
- LC_COLLATE Wpływa na porównywanie ciągów znaków
- LC_CTYPE Wpływa na klasyfikację i konwersję znaków
- LC_MONETARY Wpływa na postać ciągów numerycznych i walutowych
- LC_NUMERIC Wpływa na postać separatora dziesiętnego.
- LC_TIME Wpływa na postać daty i czasu

Zachowanie strftime zależy również od tego, na jakim systemie operacyjnym działa PHP

```
<?php
setlocale(LC_ALL, 'pl_PL');
//w wersji dla Windows należy użyć:
//setlocale(LC_ALL, 'plk');
echo strftime("Bieżąca strefa czasowa: %Z<br>");
echo strftime("Data: %d-%m-%Y<br>");
echo strftime("Czas: %H:%M:%S<br>");
echo strftime("Mamy %U tydzień i %j dzień roku.<br>");
echo strftime("Dziś jest %A, %d %B %Y r.<br>");
?>
```

mktime

`mktime` -- Oblicza uniksowy znacznik czasu dla podanej daty

Opis

`int mktime` (`int` godzina, `int` minuta, `int` sekunda, `int` miesiąc, `int` dzień, `int` rok [, `int` letni/zimowy])

Uwaga: Proszę zwrócić uwagę na dziwną kolejność argumentów, zupełnie odmienną od spotykanej w standardowym wywołaniu uniksowym `mktime()`; która w dodatku nie jest praktyczna przy opuszczaniu argumentów od prawej do lewej (patrz niżej). Częstym błędem w skryptach są pomyłki w kolejności tych argumentów.

Funkcja zwraca uniksowy znacznik czasu odpowiadający podanym argumentom. Znacznik czasu jest liczbą całkowitą długą (`long integer`) zawierającą liczbę sekund dzielącą uniksową Epokę (1 stycznia 1970) od podanego w argumentach czasu.

Argumenty mogą być opuszczane w kolejności od prawej do lewej. Za każdy pominięty argument będzie wówczas wstawiona aktualna wartość, zgodnie z lokalnym czasem/datą.

Argument *letni/zimowy* może być ustawiony na 1, jeśli to czas zimowy, lub 0 jeśli letni, lub -1 (domyślnie) jeśli niewiadomo, czy letni czy zimowy. Jeśli niewiadomo, PHP spróbuje ustalić to samodzielnie. Może to powodować nieoczekiwane (ale na pewno poprawne) wyniki.

Funkcja `mktime()` przydaje się przy wykonywaniu arytmetyki dat i walidacji, gdyż automatycznie policzy właściwą wartość dla danych spoza przedziałów.

Parametr rok może być liczbą dwu lub czterocyfrową, wartości z przedziału 0-69 będą mapowane do 2000-2069, a z przedziału 70-99 do 1970-1999 (w systemach, w których `time_t` jest 32-bitową liczbą całkowitą ze znakiem, co jest obecnie najpopularniejszym rozwiązaniem, poprawny zakres argumentu *rok* zawiera się pomiędzy 1902 a 2037).

Windows: Żadna znana wersja systemu Windows nie obsługuje ujemnych znaczników czasu. Z tego powodu zakres poprawnych dat zawiera się pomiędzy rokiem 1970 a 2038.

Ostatni dzień dowolnego miesiąca może być wyrażony jako zerowy dzień następnego miesiąca, ale nie jako -1 dzień. Obydwa poniższe przykłady wyświetlą "Ostatni dzień lutego 2000 to: 29".

Data z rokiem, miesiącem i dniem równym zero jest niepoprawna (w przeciwnym razie oznaczałoby to 30.11.1999, co mogłoby powodować dziwne rezultaty).

Poniższy przykład ilustruje działanie funkcji `mktime` – obliczając liczbę dni pomiędzy dwiema datami:

```
<?php
$r1 = 2008;
$m1 = 5;
```

```
$d1 = 1;

$r2 = 2008;
$m2 = 9;
$d2 = 1;

$time1 = mktime(0, 0, 0, $m1, $d1, $r1);
$time2 = mktime(0, 0, 0, $m2, $d2, $r2);
$time = abs(ceil(($time1 - $time2) / 86400));

echo("Pomiędzy $d1-$m1-$r1, a $d2-$m2-$r2 mamy $time dni.");

?>
```

Pozostałe funkcje

PHP zawiera także inne funkcje czasu i daty, które mogą być wykorzystane przy uatrakcyjnianiu witryny WWW np.:

- checkdate
- gmtime
- localtime
- time

Poniżej skrypt obliczający czas generowania strony

```
<?php
$time1 = microtime(true);

//Przykładowa treść skryptu
for($i = 0; $i < 10000; $i++)
    for($j = 0; $j < 1000; $j++);

$time2 = microtime(true);
$time = round($time2 - $time1, 4);

echo "Strona została wygenerowana w czasie $time sekund.";

?>
```

Kolejny przykład porównuje funkcję date oraz gmtime

```
<?php
echo("Wywołania funkcji date:<br>");
echo("date('G:i') = " . date("G:i") . "<br>");
echo("date('r') = " . date("r") . "<br>");
echo("date('c') = " . date("c") . "<br>");
```

```
echo("date('Tl') = " . date("T") . "<br>");
echo("date('Zl') = " . date("Z") . "<br>");

echo("<br>Wywołania funkcji gmdate:<br>");
echo("gmdate('G:i') = " . gmdate("G:i") . "<br>");
echo("gmdate('r') = " . gmdate("r") . "<br>");
echo("gmdate('c') = " . gmdate("c") . "<br>");
echo("gmdate('Tl') = " . gmdate("T") . "<br>");
echo("gmdate('Zl') = " . gmdate("Z") . "<br>");
?>
```

Zadania

1. Korzystając z funkcji date, napisz skrypt wyświetlający następujące dane: nazwa dnia tygodnia (w języku polskim), pełna data (rok w postaci czterocyfrowej) czas w formacie dwunastogodzinnym z oznaczeniem przed południem/po południu oraz różnicę w stosunku do czasu GMT.
2. Napisz skrypt, który będzie odliczał liczbę dni do końca bieżącego roku.
3. Napisz skrypt, który pozwoli na stwierdzenie jakim dniem tygodnia był dowolnie wybrany dzień.
4. Napisz skrypt obliczający czas generowania strony, w którym wartości sekund i mikrosekund są wyświetlone oddzielnie
5. Napisz skrypt, który umożliwi wczytywanie różnych wersji strony zależności od pory dnia (np. dwie różne wersje: nocna i dzienna). Wykorzystaj funkcję date oraz instrukcję include.