

# Obsługa daty i czasu

## 1. Wyświetlanie daty i czasu

Podstawową funkcją pozwalającą na uzyskanie informacji o dacie jest **date**. Jej wywołanie ma 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 zawierający znacznik czasu Uniksa i określający interesującą nas datę.

Znaczniki formatujące dla funkcji **date**:

d - Dzień miesiąca, 2 cyfry z wiodącymi zerami

D - Tekstowy opis angielskiej nazwy dnia, trzy litery

j - Dzień miesiąca bez zer wiodących

l (mała litera 'L') - Pełen angielski opis dnia tygodnia

N - Liczbowa forma dnia tygodnia, zgodna z normą ISO-8601 (dodana w PHP 5.1.0)

S - Angielski przyrostek porządkowy dla dnia miesiąca, 2 litery

w - Liczbowa forma dnia tygodnia

z - Dzień roku (Zaczynając od 0)

W - Numer tygodnia w roku, zgodny z normą ISO-8601, Tygodnie rozpoczynają Poniedziałki (dostępne od PHP 4.1.0)

F - Pełen angielski opis, dnia miesiąca, taki jak January czy March

m - Liczbowa forma miesiąca, z zerami wiodącymi

M - Krótki, angielski opis miesiąca, trzy litery

n - Liczbowa forma miesiąca, bez zer wiodących

t - Ilość dni w danym miesiącu

L - Informacja o tym, czy rok jest przestępnym

o - Numer roku, zgodny z normą ISO-8601. Zwraca to taką samą wartość jak Y, z takim wyjątkiem, że numer tygodnia ISO (W) należy do poprzedniego lub następnego roku, niż rok użyty w tym miejscu. (dodane w PHP 5.1.0)

Y - Pełna liczbowa forma roku, 4 cyfry

y - Dwie cyfry reprezentujące rok

a - Pora dnia - dwie małe litery (przed/po południu) (ang. Ante/Post meridiem)

A - Pora dnia - dwie duże litery (przed/po południu) (ang. Ante/Post meridiem)

g - Godzina, w formacie 12-godzinnym, bez zer wiodących

G - Godzina, w formacie 24-godzinnym, bez zer wiodących

h - Godzina, w formacie 12-godzinnym, z zerami wiodącymi

H - Godzina, w formacie 24-godzinnym, z zerami wiodącymi

i - Minuty z zerami wiodącymi

s - Sekundy, z zerami wiodącymi

e - Identyfikator strefy czasowej (dodano w PHP 5.1.0)

I (duże i) - Informacja o tym, czy czas jest letni

O - Różnica z czasem Greenwich (GMT) w godzinach

P - Różnica z czasem Greenwich (GMT) z dwukropkiem pomiędzy godzinami i minutami (dodano w PHP 5.1.3)

T - Skrót dla strefy czasowej

Z - Różnica dla strefy czasowej w sekundach. Wyrównanie to jest zawsze ujemne dla stref położonych na zachód od południka 0, oraz dodatnie dla tych leżących na wschód od niego.

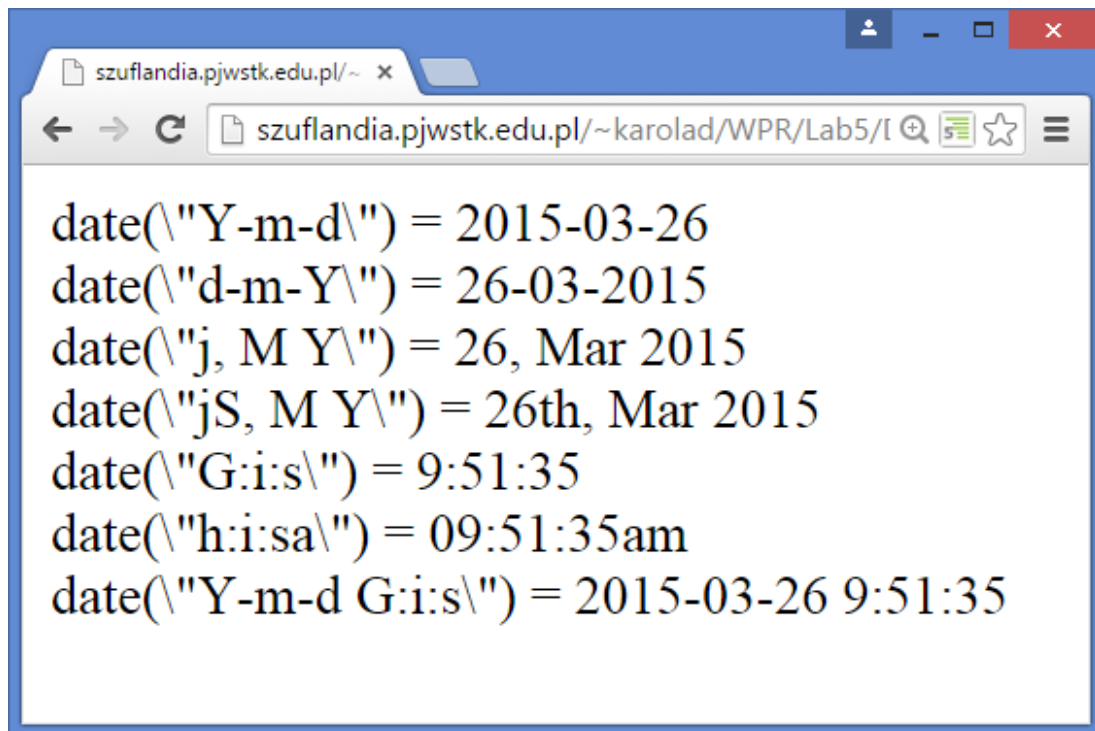
c - Data w standardzie ISO 8601 (dodana w PHP 5)

r - Data sformatowana zgodnie z RFC 2822

U - Sekundy liczone od ery UNIX-a (1 stycznia 1970 00:00:00 czasu Greenwich - GMT)

Przykładowe wywołanie funkcji **date**

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



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

**getdate([znacznik\_czasu])**

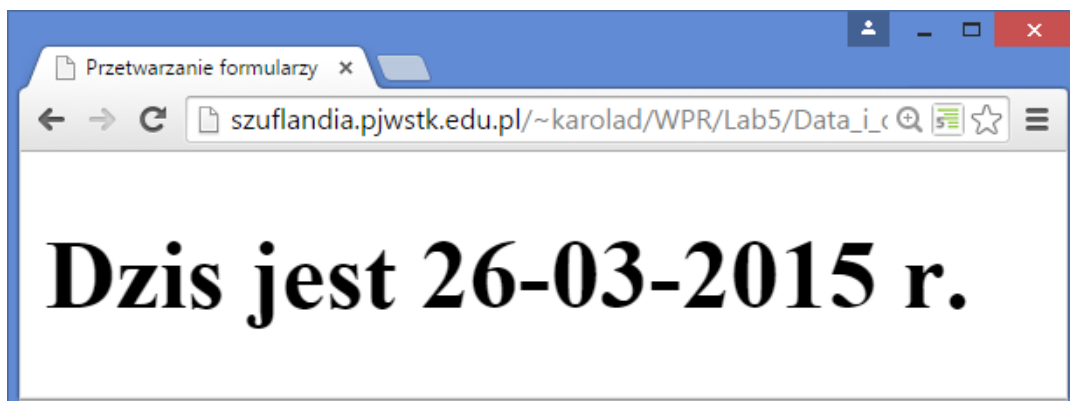
również w tym przypadku parametr **znacznik\_czasu** jest opcjonalny, a jego użycie ma takie samo znaczenie jak w przypadku **date**. Wynikiem działania **getdate** nie jest jednak ciąg znaków, ale tablica asocjacyjna zawierająca pobrane dane. Indeksy tej tablicy wraz z ich znaczeniami i przykładowymi wartościami:

### Klucze zwróconej tablicy asocjacyjnej

| Klucz     | Opis   | Przykłady zwracanych wartości                               |
|-----------|--|---|
| "seconds" | Ilość sekund   | 0 do 59   |
| "minutes" | Liczba minut   | 0 do 59   |
| "hours"   | Ilość godzin   | 0 do 23   |
| "mday"    | Liczba będąca dniem miesiąca   | 1 do 31   |
| "wday"    | Dzień tygodnia w postaci cyfry   | 0 (dla Niedzieli) aż do 6 (dla Soboty)                      |
| "mon"     | Miesiąc w postaci liczby   | 1 aż do 12  |
| "year"    | Pełen rok w postaci liczby, 4 cyfry  | Przykłady: 1999 lub 2003                                    |
| "yday"    | Dzień danego roku, w postaci liczby  | 0 aż do 365   |
| "weekday" | Pełna nazwa dnia tygodnia,   | w języku angielskim <i>Sunday</i> aż do <i>Saturday</i>     |
| "month"   | Pełna nazwa miesiąca, w języku angielskim, jak np. January lub March   | <i>January</i> aż do <i>December</i>                        |
| 0         | Sekundy, które upłynęły od Ery Uniksa, podobnie jak wartość zwracana przez <a href="#">time()</a> i użyta przez funkcję <a href="#">date()</a> . | Zależnie od systemu, zazwyczaj 2147483648 aż do 2147483647. |

### Wykorzystanie funkcji `getdate`

```
<?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.";
?>
```



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

```
strftime(format[ , timestamp])
```

Argument format to ciąg znaków, który może zawierać znaczniki przedstawione poniżej. Parametr timestamp jest opcjonalny. Pozwala on na uzyskanie ciągu znaków, który odpowiada konkretnej dacie. Jeśli zostanie pominięty, będzie użyty bieżący czas lokalny.

Znaczniki formatujące dla funkcji strftime

- %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-godzinny (przedział od 00 do 23)
- %I - godzina jako liczba dziesiętna w systemie 12-godzinny (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
- %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 "%"

Dane zwracane przez *strftime*, będą zgodne z bieżącymi ustawieniami lokalnymi (narodowymi), które mogą być zmieniane za pomocą funkcji *setlocale*. Wywołanie *setlocale* ma postać:

*setlocale(kategoria, locale)*

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

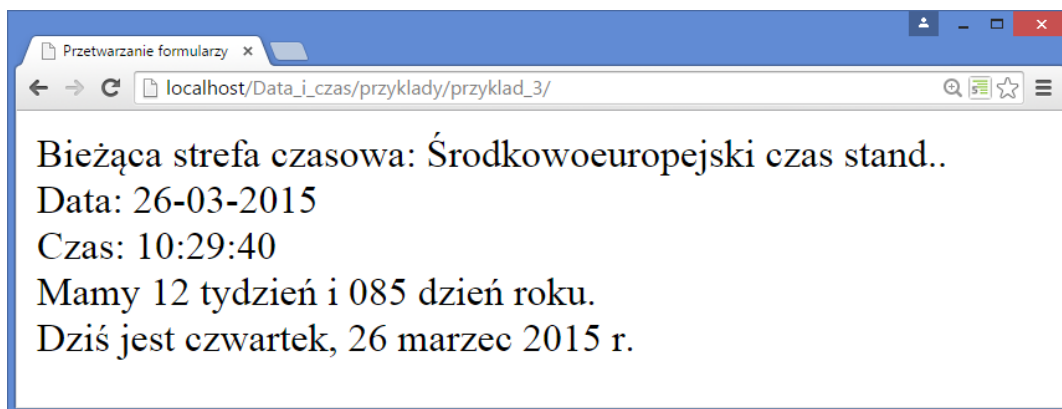
Możliwe wartości parametru *kategoria*:

- LC\_ALL dla wszystkich poniżej
- LC\_COLLATE ciąg dla porównania
- LC\_CTYPE na charakter i klasyfikację konwersji
- LC\_MONETARY na postać ciągów numerycznych i walutowych
- LC\_NUMERIC dla separatora dziesiętnego
- LC\_TIME do formatowania daty i czasu z strftime ()

Argument *locale* jest wyrażany w różny sposób w różnych systemach operacyjnych. W systemach uniksowych dla ustawień polskich należy użyć ciągu *pl\_PL*. W systemach rodziny Windows należy użyć *plk* lub *polish*.

Wykorzystanie funkcji setlocale

```
<?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 />");
?>
```



## 2. Tworzenie znacznika czasu

Do utworzenia znacznika czasu, który może być wykorzystywany jako drugi parametr funkcji opisywanych wcześniej, można wykorzystać funkcję mktime. W pełnej wersji przyjmuje ona aż siedem argumentów, chociaż w rzeczywistości wszystkie są opcjonalne. Schemat wywołania jest następujący:

***mktime(godzina, minuta, sekunda, miesiąc, dzień, rok)***

Parametry można pomijać wyłącznie od strony prawej do lewej. Brakujące parametry są uzupełniane przez czas lokalny. Jeśli więc interesuje nas znacznik czasu odpowiadający godzinie 13:24:52 w aktualnym dniu, zapisujemy wywołanie:

***mktime(13, 24, 52)***

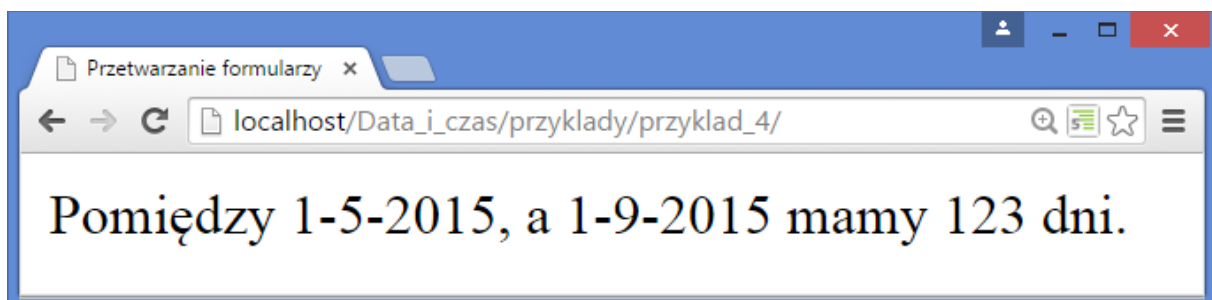
Jak w praktyce można wykorzystać funkcję ***mktime***, pokazuje poniższy skrypt. Pozwala on na obliczenie liczby dni występujących pomiędzy dwiema datami.

```
<?php
$r1 = 2014;
$m1 = 5;
$d1 = 1;

$r2 = 2014;
$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.";
?>
```



Znacznik czasu można również uzyskać poprzez konwersję ciągu znaków opisujących datę i czas. Służy do tego funkcja ***strtotime***, której wywołanie jest następujące:

***strtotime(format[, timestamp]);***

Argument ***format*** to opis daty (ciąg znaków), dla której chcemy uzyskać znacznik czasu. Natomiast ***timestamp*** to opcjonalny znacznik czasu, który zostanie wykorzystany przy obliczaniu wyniku.

Kilka przykładów:

Jeśli chcemy uzyskać aktualny znacznik czasu napiszemy:

***strtotime(„now”);***

Jeśli interesuje nas znacznik czasu dla 18 marca 1976 roku napiszemy:

***strtotime(„18 march 1976”);***



Jeśli potrzebujemy znacznika czasu sprzed 12 dni, wpiszemy:

***strtotime(„-12 day”);***

Jeśli potrzebujemy znacznika czasu który wystąpi za 2 dni 4 godziny 15 minut i 12 sekund od aktualnej godziny, wpiszemy:

***strtotime(„+2 day 4 hours 15 minutes 12 seconds”);***

Jeśli chcemy uzyskać znacznik czasu dla następnego piątku, wtedy wpiszemy:

***strtotime(„next friday”);***

Ostatnią funkcją, którą zajmiemy się w tej sekcji, to ***microtime***. Ona również zwraca aktualny znacznik czasu, z tym że zawierający dodatkowe dane odnośnie liczby mikrosekund. Ogólne wywołanie ma postać:

***microtime(tryb)***

Tryb jest tu opcjonalnym parametrem typu ***boolean***. W przypadku gdy jest obecny i równy ***true***, zwracana jest wartość rzeczywista. W pozostałych przypadkach zwracany jest ciąg w postaci ***msec sec***, gdzie ***msec*** oznacza liczbę mikrosekund, a ***sec*** liczbę sekund, które upłynęły od początku epoki Uniksa. Takie zachowanie pozwala np. obliczyć czas generowania strony.

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.";

?>
```



### 3. Pozostałe funkcje

Oprócz wspomnianych wyżej istnieje jeszcze wiele innych funkcji służących do obsługi daty i czasu. Nie potrzeby, aby je wszystkie omawiać – w tej części lekcji opisanych zostanie jedynie kilka. Będą to:

- ***checkdate***,
- ***gmdate***,
- ***localtime***,
- ***time***

Funkcja ***checkdate***

Zadaniem funkcji jest sprawdzenie, czy dane przekazane jej w postaci argumentów tworzą poprawną datę. Funkcja zwraca wartość **true**, jeżeli podany dzień miesiąc rok tworzą poprawną datę. Przykład wywołania (dla 25 maja 2014 roku):

***checkdate(5, 25, 2014)***

Funkcja ***gmdate***

Funkcja zwraca ciąg znaków sformatowany zgodnie z szablonem podanym jako parametr format. Jej wywołanie ma postać:

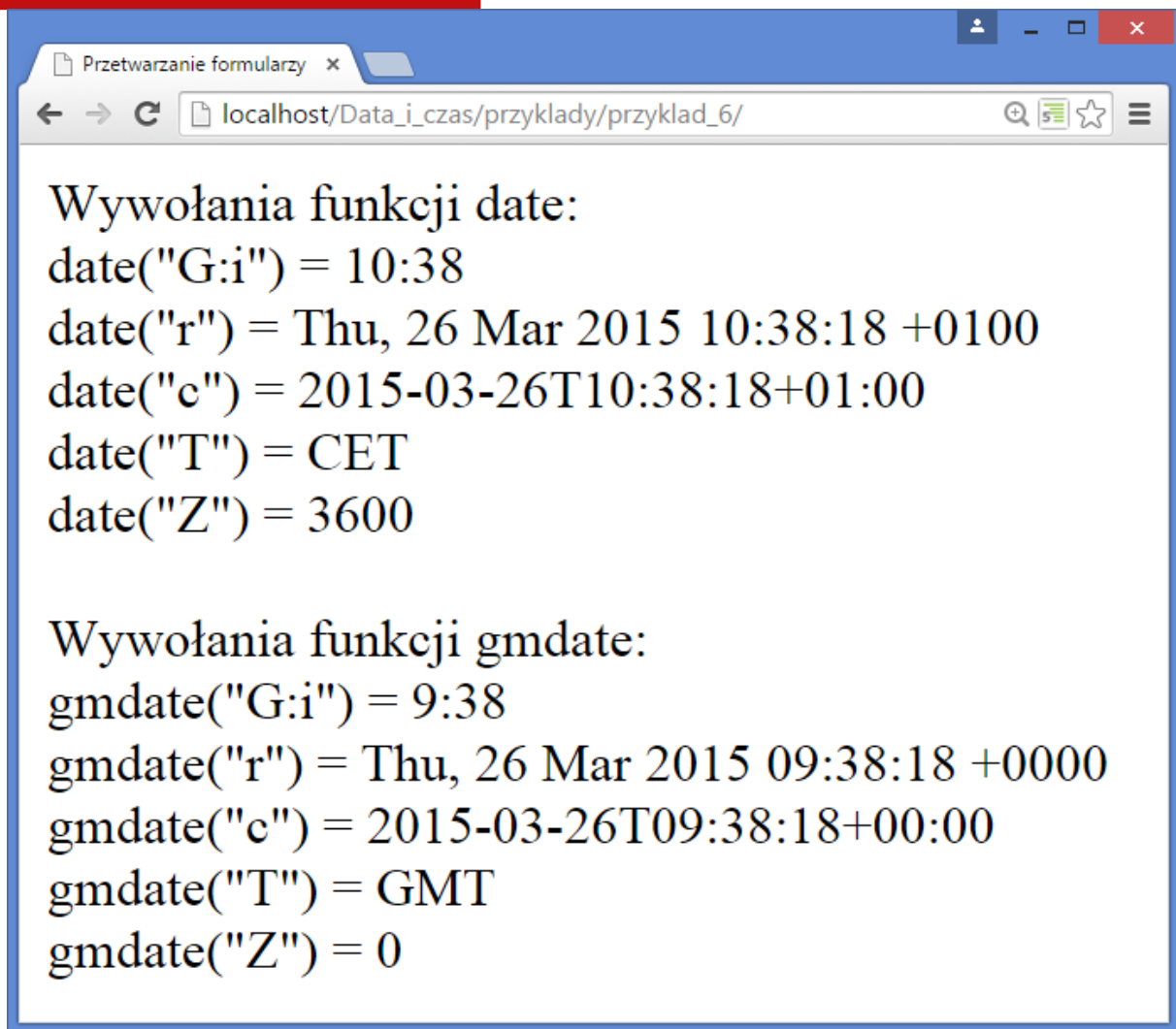
***gmdate(format[ , timestamp])***

W odróżnieniu od omawianej wcześniej ***date***, w tym przypadku zwracany jest czas GMT a nie lokalny.

Porównanie funkcji ***date*** i ***gmdate***

```
<?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(\"T\") = " . date("T") . "<br />";
echo "date(\"Z\") = " . 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(\"T\") = " . gmdate("T") . "<br />";
echo "gmdate(\"Z\") = " . gmdate("Z") . "<br />";
?>
```



The screenshot shows a web browser window with the address bar displaying 'localhost/Data\_i\_czas/przyklady/przyklad\_6/'. The page content is as follows:

```
Wywołania funkcji date:  
date("G:i") = 10:38  
date("r") = Thu, 26 Mar 2015 10:38:18 +0100  
date("c") = 2015-03-26T10:38:18+01:00  
date("T") = CET  
date("Z") = 3600  
  
Wywołania funkcji gmdate:  
gmdate("G:i") = 9:38  
gmdate("r") = Thu, 26 Mar 2015 09:38:18 +0000  
gmdate("c") = 2015-03-26T09:38:18+00:00  
gmdate("T") = GMT  
gmdate("Z") = 0
```

Funkcja localtime

Funkcja zwraca tablicę, której poszczególne elementy zawierają dane odnośnie daty i czasu.

**localtime** ([ int \$znacznik\_czasu [, bool \$asocjacyjna ] ] )

Pierwszym argumentem funkcji jest znacznik czasu. Jeśli żaden znacznik czasu nie zostanie podany, funkcja wykorzystuje aktualny czas, taki jak zwrócony przez time().

Drugi argument, *asocjacyjna*, jeśli jest ustawiony na FALSE, lub nie jest podany, powoduje zwrócenie zwykłej, indeksowanej liczbowo tablicy. Jeśli zaś jest ustawiony na TRUE, to funkcja zwróci tablicę asocjacyjną zawierającą wszystkie rozmaite elementy struktury zwracanej przez wywołanie funkcji. Nazwy kluczy tablicy są następujące:

"tm\_sec" - sekundy

"tm\_min" - minuty

"tm\_hour" - godziny

"tm\_mday" - dzień miesiąca

"tm\_mon" - miesiąc roku, 0 to styczeń

"tm\_year" - ilość lat od 1900

"tm\_wday" - dzień tygodnia

"tm\_yday" - dzień roku

"tm\_isdst" - czy aktualnie jest czas zimowy

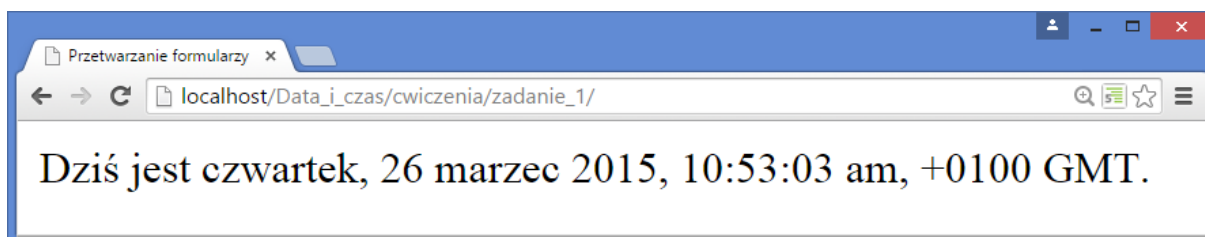
Funkcja **time()**

Funkcja zwraca znacznik czasu Uniksa określający aktualną datę i czas.

## Zadania

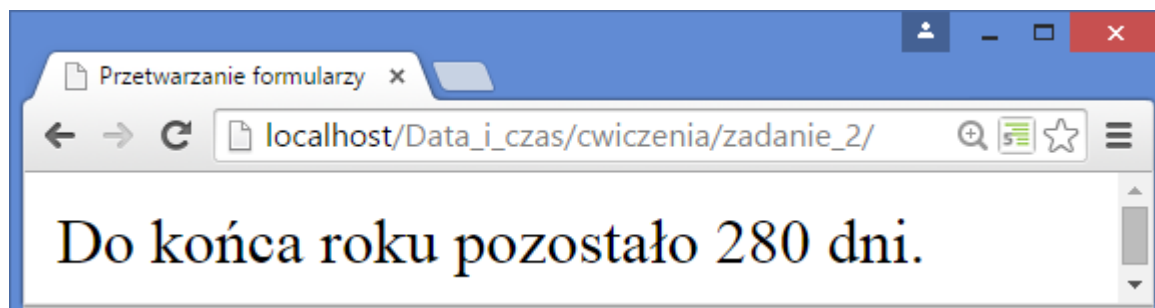
### Zadanie 1.

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.



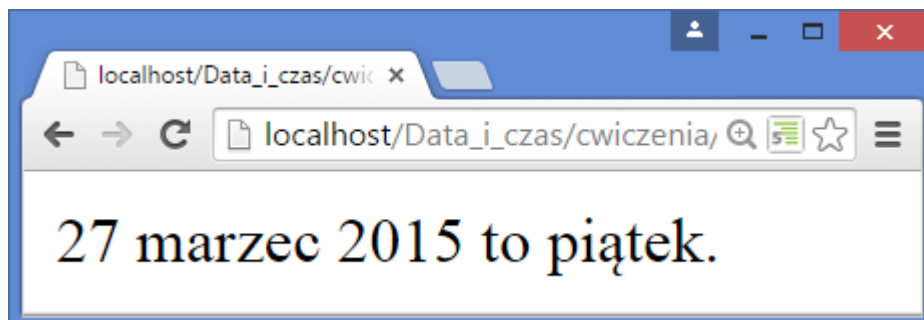
### Zadanie 2.

Napisz skrypt, który będzie wyświetlał liczbę dni pozostałych do końca bieżącego roku.



**Zadanie 3.**

Napisz skrypt, który pozwoli na stwierdzenie, jakim dniem tygodnia był dowolnie wybrany dzień.



**Zadanie 4.**

Napisz skrypt, który umożliwi wczytywanie różnych wersji strony w zależności od pory dnia. Wykorzystaj funkcje **date** oraz instrukcje **include**.

