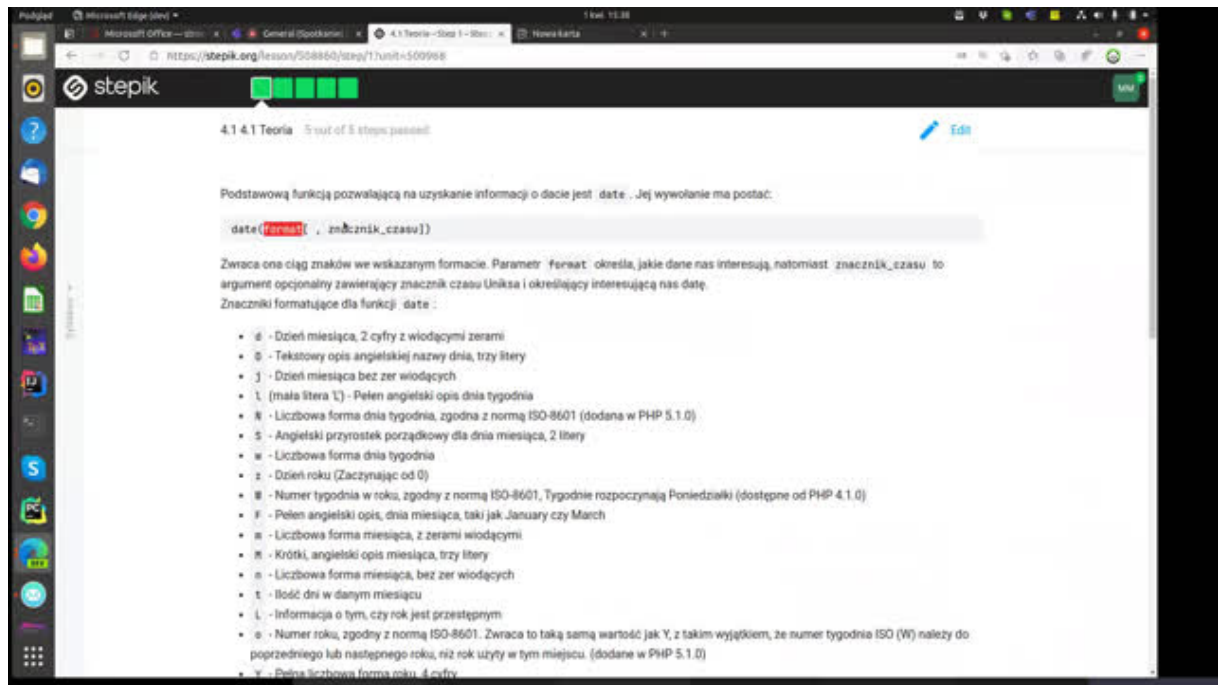


10.1 10.1 Teoria

Step 1



To watch this video please visit <https://stepik.org/lesson/508860/step/1>

Step 2

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-godzinny, bez zer wiodących
- `G` - Godzina, w formacie 24-godzinny, bez zer wiodących
- `h` - Godzina, w formacie 12-godzinny, z zerami wiodącymi

- **H** - Godzina, w formacie 24-godzinny, 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)

Przykład użycia 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 \>';
?>
```

Step 3

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:

Klucz	Opis	Przykłady zwracanych wartości
"seconds"	Ilość sekund	0-59
"minutes"	Ilość minut	0-59
"hours"	Ilość godzin	0-23
"mday"	Liczba będąca dniem miesiąca	1-31
"wday"	Dzień tygodnia w postaci cyfry	0 (dla niedzieli) aż do 6 (sobota)
"mon"	Miesiąc w postaci liczby	1-12
"year"	Pełny rok, w postaci liczby, 4 cyfry	1999, 2021
"yday"	Dzień danego roku, w postaci liczby	0-365
"weekday"	Pełna nazwa dni tygodnia	Sunday
"month"	Pełna nazwa miesiąca	January
0	Sekundy, które upłynęły od Ery Uniksa	Zależnie od systemu

Wykorzystanie funkcji `getdate()` :

```
<?php
    $data = getdate();
    $dzien = $data["mday"];
    $miesiac = $data["mon"];
    $rok = $data["year"];
    $dzien_roku = $data["yday"];
    $nazwa_miesiac = $data["month"];
    $nazwa_dzien = $data["weekday"];
    if ($dzien < 10)
        $dzien = "0" . $dzien;
    if ($miesiac < 10)
        $miesiac = "0" . $miesiac;
    echo "Dziś jest $dzien-$miesiac-$rok ";
    echo "Dziś jest $dzien_roku dzień roku ";
    echo "Dziś jest $nazwa_dzien miesiąca $nazwa_miesiac ";
?>
```

Step 4

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` - skrócowa nazwa dnia tygodnia zgodnie z lokalizacją
- `%A` - pełna nazwa dnia tygodnia zgodnie z lokalizacją
- `%b` - skrócowa 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.UTF-8');
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/>");
?>
```

Step 5

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 = 2018;
$m1 = 5;
$d1 = 1;
$r2 = 2018;
$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. Poniższy kod przedstawia kilka przykładów użycia funkcji `strtotime`.

```
<?php
    echo strtotime("now") . "<br/>";
    echo strtotime("18 march 1976") . "<br/>";
    echo strtotime("-12 day") . "<br/>";
    echo strtotime("+2 day 4 hours 15 minutes 12 seconds") . "<br/>";
    echo strtotime("next friday") . "<br/>";
?>
```

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.

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

Step 6

Oprócz wspomnianych wyżej istnieje jeszcze wiele innych funkcji służących do obsługi daty i czasu. Należą do nich między innymi:

- `checkdate` - Zadaniem tej 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.

```
<?php
    echo checkdate(5,25,2014);
    echo checkdate(2,29,2021);
?>
```

- `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.

```
<?php
echo "Wywołanie 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 "Wywołanie 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/>";
?>
```

- `localtime` - Funkcja zwraca tablicę, której poszczególne elementy zawierają dane odnośnie daty i czasu. Jej wywołanie jest następujące:

```
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:

Nazwa klucza	Opis klucza
<code>tm_sec</code>	sekundy
<code>tm_min</code>	minuty

Nazwa klucza	Opis klucza
tm_hour	godziny
tm_mday	dzień miesiąca
tm_mon	miesiąc roku, gdzie 0 to styczeń
tm_year	ilość lat od 1900
tm_wday	dzień tygodnia
tm_yday	dzień roku
tm_isdst	czy aktualnie czas zimowy

```
<?php
    $time = localtime(time(),True);
    $wday = $time["tm_wday"];
    $yday = $time["tm_yday"];
    $month = $time["tm_mon"] + 1;
    echo "Dziś jest $wday dzień tygodnia. Jest to $yday dzień w roku. Obecnie mamy $month miesiąc.";
?>
```