

APLIKACJE INTERNETOWE PHP



TEMAT 5-13: Interfejs narodowy i wielojęzyczny.

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. **Lokalizacje narodowe** – celem twórców języka PHP było ustawienie narodowej (w tym polskiej) lokalizacji (sposobu zmiany wielkości liter narodowych, wyświetlanie daty w języku narodowym, wyświetlanie narodowej waluty) za pomocą funkcji „setlocale”.

PRZYKŁAD: (polska lokalizacja)

```
setlocale(LC_ALL, 'pl_PL.utf8', 'pl_PL.UTF-8');  
$kwota = 1234.56; setlocale(LC_MONETARY, 'pl_PL.UTF-8');  
echo money_format('%i', $kwota). ' (~'.money_format('%^0n', $kwota). '');
```

Niestety (poza walutą), kodowanie UTF-8 nie działa prawidłowo i trzeba wprowadzić poprawki do kodu, co zostało opisane w kolejnych punktach niniejszego tematu.

2. **Zmiana wielkości liter (w tym polskich)** – do zmiany wielkości liter służą funkcje „strtolower” oraz „strtoupper”. Niestety nie obsługują one obecnie polskich znaków z kodowaniem UTF-8.

PRZYKŁAD:

```
// Poniższe funkcje zmieniają wielkość znaków łacińskich – polskie znaki są ignorowane  
$tekst='Żółć gęślą jaźń'; echo strtoupper($tekst). ' '. strtolower($tekst). '<br />';
```

Aby uwzględnić polskie znaki podczas zmiany wielkości liter zapisanych z zastosowaniem kodowania UTF-8, należy użyć poniższych funkcji:

- ➔ **mb_strtoupper(\$tekst, 'UTF-8')** – zmienia małe litery na duże;
- ➔ **mb_strtolower(\$tekst, 'UTF-8')** – zmienia duże litery na małe;
- ➔ **mb_convert_case(\$tekst, \$metoda_konwersji, 'UTF-8')** – zamienia litery w zależności od następujących metod konwersji:
 - ✓ **MB_CASE_UPPER**, **MB_CASE_LOWER** – zmienia małe/duże litery na duże/małe;
 - ✓ **MB_CASE_TITLE** – zmienia pierwsze litery w wyrazach na duże.

3. **Data po polsku** – po wybraniu polskiej lokalizacji, nazwy miesiąca nie są wyświetlane gramatycznie. Z pomocą przychodzi poniższa funkcja:

```
function localstrftime($format, $timestamp = null)  
{ if(is_null($timestamp)) $timestamp = time(); // domyślnie używamy bieżącego czasu  
  if(strpos($format, '%F') !== false) // poprawione nazwy miesięcy - kod: %F  
  { $m = date('m', $timestamp); // pobranie numeru miesiąca  
    switch($m)  
    { case 1: $m = 'stycznia'; break; case 2: $m = 'lutego'; break;  
      case 3: $m = 'marca'; break; case 4: $m = 'kwietnia'; break;  
      case 5: $m = 'maja'; break; case 6: $m = 'czerwca'; break;  
      case 7: $m = 'lipca'; break; case 8: $m = 'sierpnia'; break;  
      case 9: $m = 'września'; break; case 10: $m = 'października'; break;  
      case 11: $m = 'listopada'; break; case 12: $m = 'grudnia'; break; }  
    $format = str_replace('%F', $m, $format); }  
  return strftime($format, $timestamp); // wywołanie oryginalnej funkcji formatującej  
} // patrz też: http://pl.wikibooks.org/wiki/PHP/Internacjonalizacja
```

Teraz można prawidłowo wyświetlać datę po polsku, np.

```
setlocale(LC_ALL, 'pl_PL.utf8', 'pl_PL.UTF-8');  
echo localstrftime('%A, %e %F %Y'). ' r.<br />';
```

4. **Interfejs wielojęzyczny** – to interfejs uwzględniający różne wersje językowe tekstów. Teksty można podzielić na grupy, co przyspieszy wczytywanie stron z tłumaczonym tekstem. Stosując interfejs wielojęzyczny należy uwzględnić fakt, że długość tego samego tekstu w różnych językach bywa zwykle inna. Oto przykład dynamicznej internacjonalizacji tekstów zapisanych w plikach „.ini”. Zastosowano tu jedną grupę o nazwie „global”, ale można użyć wielu grup.

Zawartość pliku „./locale/langs.php”:

```
<?php
$_SESSION['katalog_locale'] = './locale';    $_SESSION['domyslne_tlumaczenie'] = 'global';
$_SESSION['domyslny_jezyk'] = 'pl';          $teksty = NULL;

// Wybranie pliku z tłumaczeniami
function wybierz_tlumaczenie($tlumaczenie = null)
{
    $_SESSION['tlumaczenie'] = ( is_null($tlumaczenie)?
                                $_SESSION['domyslne_tlumaczenie']: $tlumaczenie );
    $plik_ini = $_SESSION['katalog_locale']. '/'. $_SESSION['tlumaczenie'].'.ini';
    global $teksty;    $teksty = parse_ini_file($plik_ini, true);
}
// Wybranie języka
function wybierz_jezyk($param_jezyk = 'jezyk')
{
    $_SESSION['jezyk'] = (isset($_GET[$param_jezyk])?
                          $_GET[$param_jezyk]: $_SESSION['domyslny_jezyk']);
}
// Funkcja realizująca tłumaczenie tekstów
function __($tekst)
{
    global $teksty;
    if (!isset($_SESSION['tlumaczenie'])) wybierz_tlumaczenie();
    if (!isset($_SESSION['jezyk'])) wybierz_jezyk();
    if (is_null($teksty)) wybierz_tlumaczenie($_SESSION['tlumaczenie']);
    return ( isset($_SESSION['jezyk']) && isset($teksty[$_SESSION['jezyk']][$tekst])?
            $teksty[$_SESSION['jezyk']][$tekst]: $tekst );
}
?>
```

Zawartość pliku „./index.php”:

```
<?php
    session_start();
    require('./locale/langs.php');
?>
<!DOCTYPE html>
<html>
<head><meta charset="utf-8" /></head>
<body>
    <ul><?php wybierz_jezyk(); ?>
        <li><?php echo __('Strona główna'); ?></li>
        <li><?php echo __('O witrynie'); ?></li>
        <li><?php echo __('Artykuły'); ?></li>
    </ul>
</body>
</html>
```

Zawartość pliku „./locale/global.ini”:

```
[en]
Strona główna=Main page
O witrynie=About
Artykuły=Articles

[de]
Strona główna=Hauptseite
O witrynie=About
Artykuły=Artikel
```

5. **Automatyczna detekcja języka** – w tym celu można wykorzystać zmienną „\$_SERVER['HTTP_ACCEPT_LANGUAGE']”, która informuje o obsługiwanych językach przeglądarki (pierwszy z nich to bieżący język przeglądarki), np.

```
$jezyki = explode(';', $_SERVER['HTTP_ACCEPT_LANGUAGE']);
$biezacy_jezyk = explode(',', $jezyki[0]);    echo $biezacy_jezyk[0];
```

Każda witryna powinna umożliwiać także ręczną zmianę języka (wybrany język można zapisać w ciasteczkach). Użytkownik może wtedy wybrać dowolny język i zostanie to zapamiętane.

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Pobierz z formularza datę i wyświetl ją z polskimi nazwami miesięcy oraz dni tygodnia (dużymi literami).
3. W przykładzie z punktu „Interfejs wielojęzyczny” wyświetl menu w języku angielskim, niemieckim i jeszcze jednym, dla którego dodasz obsługę w pliku w formacie „.ini”.