

APLIKACJE INTERNETOWE PHP



TEMAT 5-10: Obsługa nagłówków HTTP i ciastek (cookie).

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. **Nagłówek HTTP** – początkowy fragment odpowiedzi na żądanie HTTP (musi być wysyłany przed właściwą treścią, np. przed dokumentem HTML). Zawiera m. in. typ MIME danych (np. text/html), kodowanie znaków (np. UTF-8), itd.
2. **Wysyłanie własnych nagłówków w języku PHP** – służy do tego funkcja „header”. W poniższym przykładzie jest ona wykorzystana do wysyłania dokumentu PDF w taki sposób, żeby internauta nie znał jego położenia na dysku. W razie nie znalezienia pliku następuje automatyczne przekierowanie do innej strony internetowej.

PRZYKŁAD:

```
<?php
    $nazwa_pliku = 'plik.pdf';
    if (!is_file($nazwa_pliku)) header('Location: http://www.wp.pl');
    else
    {
        header('Content-type: application/pdf');
        header('Content-Disposition: attachment; filename='. $nazwa_pliku);
        readfile($nazwa_pliku);
    }
?>
```

3. **Ciastka (cookies)** – małe kawałki informacji przesyłane w nagłówku HTTP i zapisywane przez przeglądarkę internetową na dysku lokalnym komputera klienckiego. Ze względów bezpieczeństwa, ciastko powinno być odczytywane TYLKO przez aplikację, która je utworzyła. Za pomocą ciastek nie należy przysyłać poufnych danych, np. loginy i hasła, ponieważ nie zapewniają dostatecznego bezpieczeństwa. Loginy i hasła przesyłamy za pomocą sesji.
4. **Obsługa ciastek w języku PHP** – służy do tego funkcja:

`setcookie($nazwa_ciastka, $wartosc_ciastka, $czas_wygasniecia_ciastka);`

Ciasteczka można podglądać i usuwać bezpośrednio w przeglądarce internetowej (np. w przeglądarce Firefox funkcja ta dostępna jest w menu: „Narzędzia | Opcje | Prywatność”).

PRZYKŁADY:

```
// ciasteczko ważne 1 rok
setcookie('preferowany język', 'pl', time() + 365 * 24 * 60 * 60);
// Ciasteczko ważne do końca sesji (wartość <= 0)
setcookie('preferowany język', 'pl', 0);
// Usunięcie ciastka (ustawienie daty ważności z przeszłości – wartość > 0)
setcookie('preferowany język', 'pl', 1); // data ważności: 1970-01-01 00:01
```

Funkcja „setcookie” musi zostać użyta przed wysłaniem nagłówków HTTP, tzn. przed wysłaniem jakiegokolwiek kodu HTML, chyba, że zastosuje się buforowanie wyjścia.

PRZYKŁAD:

```
// Poniższe ciasteczko jest ważne min. 2 lata
setcookie('ostatnie_odwiedziny', time(), time() + (2*365 + 1) * 24 * 60 * 60);
if (isset($_COOKIE['ostatnie_odwiedziny']))
    echo 'Witamy. Ostatnie odwiedziny: ' .
        date('Y-m-d H:i:s', $_COOKIE['ostatnie_odwiedziny']);
else echo 'Witamy po raz pierwszy.'; // ta przeglądarka na tym koncie
```

5. **Buforowanie wyjścia** – przechwytywanie treści wysyłanych za pomocą takich funkcji, jak: echo, print, itp. Treści te zapisywane są w buforze i wysyłane są w momencie wywołania specjalnej funkcji PHP. Buforowanie wyjścia inicjalizowane jest funkcją „ob_start()”.

Oto funkcje obsługi buforowania wyjścia:

- ➔ **„ob_start()” i „ob_end_flush()”** – po wywołaniu funkcji „ob_end_flush()” treści przesyłane są razem z nagłówkami HTTP;

PRZYKŁAD:

```
ob_start();
if (isset($_COOKIE['ostatnie_odwiedziny']))
    echo 'Witamy. Ostatnie odwiedziny: ' .
        date('Y-m-d H:i:s', $_COOKIE['ostatnie_odwiedziny']);
else echo 'Witamy po raz pierwszy.';
// Poniższe ciasteczko jest ważne min. 3 miesiące
setcookie('ostatnie_odwiedziny', time(), time() + 92 * 24 * 60 * 60);
ob_end_flush();
```

- ➔ **„ob_start()” i „ob_get_clean()”** – po wywołaniu funkcji „ob_get_clean()” treści zapisywane są do zmiennej (można to zastosować np. do cenzurowania treści);

PRZYKŁAD:

```
ob_start();
if (isset($_COOKIE['ostatnie_odwiedziny']))
    echo 'Witamy. Ostatnie odwiedziny: ' .
        date('Y-m-d H:i:s', $_COOKIE['ostatnie_odwiedziny']);
else echo 'Witamy po raz pierwszy.';
// Poniższe ciasteczko jest ważne min. 12 godzin
setcookie('ostatnie_odwiedziny', time(), time() + 12 * 60 * 60);
$html = ob_get_clean();
$html = str_replace(substr($html, strpos($html, '-')-4, 10), '*****', $html);

echo $html;
```

- ➔ **„ob_start('ob_gzhandler')” i „ob_implicit_flush(0)”** – w tym wypadku treści przesyłane są w postaci skompresowanej (kompresja „w locie”) za pomocą biblioteki „Gzip”.

PRZYKŁAD:

```
ob_start('ob_gzhandler');
echo 'Witamy.';
ob_implicit_flush(0);
```

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Utwórz skrypt wyświetlający 3 wybrane dokumenty PDF (ukryj ich położenie na dysku poprzez umieszczenie w podkatalogu, którego nazwy internauta nie będzie mógł odczytać).
3. Utwórz możliwie najprostszy skrypt informujący internautę o tym, który raz uruchomił skrypt (informację tą przechowujemy w ciasteczku).
4. Zmień poprzedni skrypt tak, żeby ciasteczka zapisywane było po wysłaniu do przeglądarki informacji z poprzedniego zadania.