Sesje, ciasteczka, wyjątki

Nie sposób wyobrazić sobie bez nich takich podstawowych zastosowań, jak logowanie użytkowników czy funkcjonowanie koszyka na zakupy. Oprócz tego dowiesz się, czym są wyjątki, jak je stosować oraz nauczysz się obsługiwać pliki wysłane na serwer.

Ciasteczka w PHP

Czym są ciasteczka (z ang. cookies)? To niewielki fragment informacji, który aplikacja internetowa może zapisać na komputerze użytkownika. Gdy przeglądarka łączy się ze stroną, w pierwszej kolejności szuka lokalnych cookies, odpowiednich dla danej witryny. Jeżeli znajdzie – zostaną one przekazane serwerowi.

```
<?php

// utworzenie ciasteczka
setcookie($nazwa, $wartosc, $koniec, $sciezka, $domena,$bezpieczne);
?>
```

Setcookie() może przyjąć 6 argumentów. Jednak wymagane jest podanie tylko pierwszego, którym jest nazwa ciasteczka. $wartosc to wartość przypisana do ciastka, $koniec wyraża datę wygaśnięcia, $sciezka i $domena mogą być stosowane do określenia adresów, dla których cookie jest ważne. Ostatni argument, $bezpieczne, oznacza, że cookie nie będzie wysyłane przez zwykłe połączenie HTTP.

Zastosowanie cookies

W praktyce, najczęściej używamy trzech pierwszych argumentów. Wszelkie zapamiętywanie zalogowanego użytkownika, oddania głosu w ankiecie itp. są obsługiwane właśnie z wykorzystaniem cookies. Zasada jest prosta – jeśli cookie istnieje, to znaczy, że głos został oddany. Jeżeli nie, można oddać głos, po czym cookie jest tworzone.

```
<?php

// zapisanie oddania głosu jednorazowego
setcookie('oddano_glos', '1');

// w przypadku gdy głosować można raz dziennie
setcookie('oddano_glos', '1', time()+3600*24);

?>
```

Można łatwo zauważyć, że data wygaśnięcia podawana jest w postaci liczby sekund od 1 stycznia 1970 roku. Wygodnie jest zatem użyć funkcji time(), zwracającej liczbę sekund, które upłynęły do chwili obecnej, po czym dodać żądany czas (w naszym przypadku jest to jeden dzień, gdyż 24 godziny = 24 * 1 godzina = 24 * 3600 sekund).

Sprawdzanie obecności ciasteczka

Wiesz, jak ustawić ciasteczko, jego wartość oraz datę wygaśnięcia. Czas teraz na sprawdzanie, czy

ciastko istnieje. Każde ciasto jest przechowywane w tablicy globalnej $_COOKIES. Sprawdzimy, czy istnieje cookie o nazwie „aktywacja”:

```
<?php

if(isset($_COOKIE['aktywacja']))
   echo "Ciasteczko istnieje";
else
   echo "Brak ciastecza o nazwie aktywacja";
?>
```

Wyświetlenie wartości ciasteczka

Jeżeli cisteczko istnieje to możemy sprawdzić jego wartość

```
<?php
// zapisanie oddania głosu jednorazowego
setcookie('beleco', 'pomidor');

if(isset($_COOKIE['beleco']))
   echo $_COOKIE['beleco'];
else
   echo "Brak ciastecza o nazwie beleco";
?>
```

Sesje w PHP

Kontrola sesji udostępnia możliwość śledzenia poczynań odwiedzającego, podczas pojedynczej wizyty na naszej stronie. Innymi słowy, umożliwia rozpoznanie, czy użytkownik dopiero co wszedł na stronę, czy przechodzi z innej podstrony serwisu. Dodatkowo, możemy tworzyć i usuwać zmienne, których istnienie jest ściśle powiązane z pojedynczą sesją (wizytą). Ich żywot kończy się wraz z zamknięciem okna przeglądarki lub przejścia na inną stronę.

Przechowywanie zmiennych sesji

Domyślnie, zmienne sesyjne są przechowywane na komputerze użytkownika w postaci cookies (poznanych na poprzedniej lekcji). Jeżeli odwiedzający ma zablokowane zapisywanie ciasteczek lub w pliku konfiguracyjnym ustawimy inaczej, możliwe jest dodawanie informacji o sesji do adresu URL.

Rozpoczynanie sesji

Zanim możliwe będzie tworzenie zmiennych sesji, należy ją zainicjować. Można to zrobić na dwa sposoby. Pierwszym, najprostszym z nich, jest wywołanie funkcji:

```
<?php
// inicjacja sesji
session_start();

?>
```

Działa ona następująco… Jeżeli identyfikator sesji nie istnieje, zostanie on utworzony (przez co my otrzymamy możliwość tworzenia zmiennych sesji). Jeśli natomiast takowy identyfikator już

istnieje, funkcja pobierze wartości wszystkich zmiennych obecnej sesji.

Tworzenie zmiennych sesji

Po wywołaniu funkcji **start_session()**, PHP udostępnia nam tablicę superglobalną **$_SESSION**. Od tej pory możemy tworzyć tymczasowe zmienne, które będą przechowywane w tej właśnie tablicy. Jej wartości pobierane są przez funkcję start_session() za każdym razem, gdy odświeżamy stronę.

```
<?php

// inicjacja sesji
session_start();

// utworzenie zmiennej sesji
$_SESSION['zm_sesji'] = 1;

?>
```

Powyższy skrypt najpierw inicjuje obsługę sesji, po czym tworzy zmienną, o nazwie ‚zm_sesji'. Podczas kolejnych wyświetleń strony, w tablicy superglobalnej $_SESSION, będzie przechowywany rekord z indeksem ‚zm_sesji', posiadający wartość **1**. Gdy wejdziemy na inną podstronę i wywołamy kod:

```
<?php
session_start();

// użycie zmiennej
if(isset($_$_SESSION['zm_sesji']))
      echo $_SESSION['zm_sesji'];

?>
```

Wyświetli się wartość zmiennej, zapisanej w poprzednim skrypcie. Jeżeli natomiast najpierw odwiedzimy tę podstronę, $_SESSION[‚zm_sesji'] nie będzie posiadała żadnej wartości. Nie zapamięta zapisanej wartości z poprzedniej wizyty. Na tym właśnie polegają zmienne sesyjne – istnieją wyłącznie podczas trwania jedej wizyty. Z każdą następną zmienne należy utworzyć ponownie.

Zakańczanie sesji

Czasem zachodzi potrzeba, by zakończyć sesję manualnie, zanim użytkownik skończy „zwiedzać" naszą stronę. Wyobraźmy sobie, że sesja przechowuje dane o zalogowanym użytkowniku. W momencie wylogowania, należy tę sesję zamknąć, mimo iż dana osoba może wciąż przeglądać stronę jako „niezalogowana". Można to osiągnąć poprzez wywołanie funkcji usuwającej id sesji:

```
<?php

// zniszczenie sesji
session_destroy();

?>
```

Usuwanie zmiennych sesji

Zanim jednak usuniemy identyfikator sesji, co zniszczy ją zupełnie, powinniśmy usunąć wszystkie zmienne z tablicy $_SESSION. Zmienne sesyjne usuwamy za pomocą funckji:

```
<?php

// usuwanie zmiennej
unset($_SESSION['nazwa_zmiennej']);

?>
```

Jeśli zmiennych jest naprawdę dużo, można zastosować pewien trik, który usunie je wszystkie. Wystarczy wywołać:

```
<?php

// usuwanie wszystkich zmiennych z $_SESSION
$_SESSION = array();

?>
```

Na tym kończę omawianie sesji w PHP. Znasz już najważniejsze zagadnienia, w podsumowaniu poćwiczysz praktykę. Teraz czas na wyjątki.

Czym są wyjątki?

Wyjątki służą do wyłapywania błędów w wykonywaniu skryptu. Przykładowo, gdy chcemy otworzyć jakiś plik, który nie istnieje, skrypt się sypie. Wtedy do akcji wkraczają wyjątki. Zamiast zakończyć program, zwracany jest wyjątek. Dzięki temu możemy wyświetlić odpowiedni komunikat o błędzie użytkownikowi.

Wyrzucanie wyjątków

W przypadku, gdy wystąpi błąd i chcemy wyrzucić pewien wyjątek, posługujemy się poleceniem **throw**. Wyrzucać, jak napisałem powyżej, możemy obiekty klasy Exception. W praktyce wygląda to tak:

```
<?php

if(!wykonaj_funkcje())
        throw new Exception('Funkcja nie mogła być wykonana', 5);

?>
```

Powyższy skrypt wyrzuci wyjątek w wypadku, gdy funkcja „wykonaj_funkcje()” zwróci false. Wyjątek o kodzie ‚5’ i komunikacie ‚Funkcja nie mogła być wykonana’.

Łapanie wyjątków

Skrypt będzie wyłapywać wyjątki jedynie w bloku **try**. Dlatego wszystkie podejrzane funkcje, które chcemy obsłużyć wyjątkami, musimy umieścić w:

```
<?php

// bezpieczne funkcje
try
{
        // podejrzane funkcje
}

?>
```

Następnie, jeżeli któraś z podejrzanych funkcji wyrzuci wyjątek, trzeba go „złapać” i odpowiednio obsłużyć. Wyjątki łapiemy za pomocą instrukcji catch. Poniżej przykład w praktyce:

```
<?php

// bezpieczne funkcje
try
{
        // podejrzane funkcje
}
catch (Exception $e)
{
        echo 'Wystąpił wyjątek nr '.$e->getCode().', jego komunikat to:
'.$e->getMessage();
}
?>
```

$e->getCode() zawiera kod wyrzuconego wyjątku, a **$e->getMessage()**komunikat. Jeżeli zastosowalibyśmy wyjątek z początku tej lekcji, $e->getCode() miałoby wartość ‚5’, a $e->getMessage() to ‚Funkcja nie mogła być wykonana’. W ten sposób użytkownik będzie wiedział, dlaczego pojawia się błąd i będzie mógł powiadomić o tym administratora.

Ćwiczenia:

1. Napisz formularz który będzie służył do tworzenia ciasteczek na stronie o dowolnej nazwie dowolnej wartości.
2. **Napisz formularz któremu podamy nazwę ciasteczka a on wyświetli jego wartość (nie zapomnieć o użyciu isset).**
3. napisz formularz który będzie zliczał kliknięcia guzika na stronie z wykorzystaniem cookies (będzie się wyświetlała liczba dotychczasowych kliknięć na stronie).
4. **napisz formularz który będzie zliczał kliknięcia guzika na stronie z wykorzystaniem zmiennych sesyjnych (będzie się wyświetlała liczba dotychczasowych kliknięć na stronie).**
5. **Dodaj do 2 formularza obsługę wyjątków (używając trow())**
6. Napisz formularz logowania: podajemy login hasło jak są zgodne z wartością wpisaną w skrypcie to przenosi nas na stronę gdzie będzie psiało jesteś zalogowany jako ….. login ma być zapisany w zmiennej sesyjnej jak na tą stronę się wejdzie bez podania loginu i hasła to ma pisać NIEZALOGOWANY.

   *** dodaj do punktu 6 żeby po czasie 5 minut nas wylogowywało (wykorzystaj zamiast cookies sesje).