

Zajęcia 3 - PHP

Na komputerze musi być zainstalowany serwer WWW np. Apache, interpreter PHP i serwer baz danych MySQL

Proszę założyć swój własny folder na serwerze Apache w folderze **htdocs**.

1. Serwer wirtualny –Windows/Apache [opcjonalnie]

Wygodnie jest uruchomić sobie serwer wirtualny.

Aby zainstalować serwer wirtualny należy w pliku konfiguracyjnym Apache **httpd.conf** dopisać:

(to co na czerwono –nazwy własne)

```
NameVirtualHost 127.0.0.1
<VirtualHost 127.0.0.1:80>
    DocumentRoot "C:/Program Files/Apache Software
Foundation/Apache2.2/htdocs/CMS_RJ/"
    ServerName cms
    <directory "C:/Program Files/Apache Software
Foundation/Apache2.2/htdocs/CMS_RJ/">
        Options Indexes FollowSymLinks
        AllowOverride all
        allow from all
    </directory>
</VirtualHost>
```

Uwaga!

W Windows w katalogu **C:\WINDOWS\system32\drivers\etc** trzeba dopisać w pliku **hosts** linię (lub jak nie ma stworzyć – bez rozszerzenia) (gdzie **cms** to **nazwa waszego serwera**):

127.0.0.1 **cms**

Serwer Apache ma opcję **mod_rewrite**, która umożliwia "przepisanie" adresu strony wynikowej o określonej nazwie na inną nazwę. Np. zamiast **index.html**, **index.php?mod=test&strona=1** wystarczy napisać **test1.html** itd.

Szczególnie przydaje się to gdy używamy skryptów, CMS-ów napisanych w PHP lub innym języku dynamicznego generowania stron WWW.

Reguły ustawia się w pliku **.htaccess** za pomocą wyrażeń regularnych.

Przykładowy wpis do pliku **htaccess**:

```
Options FollowSymLinks
RewriteEngine On
RewriteRule ^index.html$ index.php [L]
```

Składnia

```
RewriteRule wzorzec adres_docelowy [flagi]
```

adres_docelowy - definiuje plik na serwerze, do którego będą przekazywane zapytania (w adresie docelowym można definiować zmienne)

flagi - dodatkowa dyrektywa pozwalająca określić w jaki sposób traktowana jest dana reguła

[L] - (last) jeżeli reguła dopasuje wzorec, to nie sprawdza już kolejnych reguł.

[NC] - (nocase) reguła nie zwraca uwagi na wielkość znaków.

[R=kod] - (redirect) powoduje przekierowanie adresu wraz z kodem stanu. Standardowo wysyłany jest kod 302 (Moved Temporarily).

[F] - (forbidden) zwraca kod stanu 403 i do adresu takiego nie ma dostępu.

[N] - (next) rozpoczyna ponownie przeszukiwanie reguł, lecz tym razem używa już jako wzorca przepisanego adresu przez wszystkie wywołane reguły.

Więcej np. http://www.webhost.pl/mod_rewrite-krotkie-adresy-url.html

Jednak, aby zadziałał moduł **mod_rewrite** należy sprawdzić czy w pliku **httpd.conf** jest odblokowana linia:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Uwaga: reguły zapisane w *.htaccess* możemy z powodzeniem zawrzeć w ramach dyrektywy `<Directory>` w głównym pliku ustawień Apache'a.

1.1. Ustawienia bardziej zaawansowane w pliku **htaccess** (CMS z książki):

- kompresja zawartości stron internetowych (*oszczędności czasu*) – odpowiedzialny za to jest moduł `mod_deflate` Apache

```
<IfModule mod_deflate.c>
# kompresja plików typu DEFLATE
SetOutputFilter DEFLATE
</IfModule>
```

- dodatkowo (*krótkie symboliczne nazwy*)

```
Options FollowSymLinks
RewriteEngine On
```

- dyrektywa

```
php_flag magic_quotes_gpc off
```

```
# dyrektywa magic_quotes_gpc powoduje automatyczne wykonanie operacji addslashes() na danych odebranych metodami GET, POST oraz pochodzących z ciasteczek
```

- oraz

```
RewriteRule ^index.html$ index.php [L]
RewriteRule ^([^.]{3}[^.]*)$ /index.php?page=$1 [QSA,L]
```

To zapisy to dotyczące przekierowań (*drugie za pomocą wyrażenia regularnego – przechwytyje żądania stron, których nazwy składają się z trzech lub więcej znaków i nie zawierają kropki, przekierowuje do index.php, QSA – dalsze przekazywanie łańcuchów zapytań*).

2. Zapoznanie się z konfiguracją serwera HTTP i interpretatorem skryptów PHP

2.1. Stwórz pierwsze skrypty PHP o treści:

cw1 .php

```
<?php
echo "Hello World";
?>
```

cw2 .php

```
<?php
    phpInfo();
?>
```

Zapisz je w swoim folderze na serwerze WWW i uruchom pliki z poziomu przeglądarki WWW. Przeanalizuj wyniki działania skryptu 2.

2.2. Stwórz dokument w języku HTML (stronę) i wstaw treść skryptu cw1.php w sekcji <body>. Zapisz ten dokument **z rozszerzeniem .php** i uruchom.

2.3. Stwórz dokument o nazwie cw3.html i zawartości:

```
<html>
<head>
<title>PHP - cwiczenie 3 </title>
</head>
<body>
    Witaj <p/>
    <form name="login_form" action="php_ini.php" method="get">
        <p>
            nazwa:<input type="text" name="nazwa" maxlength="20" size="20"/>
        </p>
        <p>
            hasło:
            <input type="password" name="haslo" maxlength="20" size="20"/>
        </p>
        <p>
            <input type="submit" name="zaloguj" value="zaloguj"/>
        </p>
    </form>
</body>
</html>
```

Uruchom go dwukrotnie, za drugim razem zmieniając metodę na „post” method="post".
Co się zmieniło?

3. PHP i baza MySQL

3.1. Bezpieczny kod PHP

Utwórz plik **cw4.php** z prostym formularzem:

```
<?php echo '<!doctype html>' ?>
<html lang="pl">
  <head>
    <meta charset="UTF-8">
    <title>Formularz</title>
  </head>
  <body>
<?php
if (isset($_GET['tekst'])) { // jest wpisana jakaś wartość w formularzu
    $tekst = $_GET['tekst'];
    print "Wpisana wartość to <b> $tekst </b><br/>";
    print '<a href="cw4.php"> Powrót do formularza </a>';
} else
{ // nie ma wpisanych danych, wyświetlasz formularz
    print '<form action=" cw4.php " method=get>';
    print '<input type="text" name="tekst">';
    print '<input type="submit" value="Wyśli">';
    print '</form>';
}
?>
</body>
</html>
```

Uruchom trzykrotnie plik w przeglądarce wpisując kolejno:

- jakiś *dowolny tekst* w pole edycyjne
- dokładnie to: `<h1> tekst </h1>`
- dokładnie to: `<script> alert(123); </script>`

Zinterpretuj wyniki.

Skopiuj poprzedni plik (**cw5.php**) i zapisz go z nową **cw5.php**.

W pliku cw5.php zastąp wpis:

```
$tekst = $_GET['tekst'];
```

wpisem:

```
$tekst = htmlspecialchars ($_GET['tekst']);
```

oraz popraw w dwóch miejscach **nazwę pliku** w tekście na aktualną **cw5.php**.

Możesz oczywiście użyć metody `post` zamiast `get`.

Ponownie uruchom go wpisując w polu edycyjnym dane jak poprzednio (a-c).

Zauważ różnice.

3.2. PHP i baza

Utwórz tabelę w bazie MySQL np. o nazwie **telefony** i czterech polach:

- nr – typ int, pole kluczowe (wybierz automatyczne nadawanie wartości klucz)
- imie – typ text
- nazwisko – typ tekst
- telefon – typ tekst

Wypełnij tabelę kilkoma rekordami (min. 3).

Stwórz nowy dokument w języku PHP odczytujący wszystkie rekordy z twojej tabeli i wypisujący ich zawartość w osobnych liniach na ekranie przeglądarki.

Wykorzystaj poniższy kod zmieniając **odpowiednie fragmenty** – dostosuj do własnych danych:

```
<?php

mysql_connect ("localhost","rj","rj") or die("Nie można się połączyć.
Komunikat: ".mysql_error()."; Numer błędu: ".mysql_errno());

$sql = mysql_select_db("moja_baza") or die("Nie można
pobrać".mysql_error()."; Numer błędu: ".mysql_errno());

$zapytanie = "SELECT * FROM telefony;";

$wykonaj = mysql_query($zapytanie) or die("Błąd w zapytaniu");

    while($row = mysql_fetch_row($wykonaj))
        { print_r($row);      echo "<br/>";
        }
mysql_close();
?>
```

Zadanie do wykonania

1. Załóż tabelę, która będzie zawierała pola odpowiadające **polom twojej ankiety**.
W przypadku wyboru wielokrotnego (*checkbox*) zastosuj osobne pola tabeli – *ułatwi ci to późniejszą analizę odpowiedzi*.
2. Napisz skrypt php, który będzie obsługiwał twoją ankietę, i zapisywał dane użytkownika w tabeli bazy. Zastosuj metodę POST:

<form action="twój_skrypt.php" method="post">.

Sprawdź też czy osoba np. o danym adresie e-mail już raz wysłała ankietę i blokuj wpis do bazy jeśli tak.

W skrypcie wykorzystaj poniższe informacje:

- zmienną `$_POST[]` do pobrania danych wysyłanych z formularza, np.

```
$email = $_POST["email"];
```

↑
zmienna

↑
nazwa pola

- zastosuj funkcje:

a) sposób tradycyjny – baza MySQL (lub mysqlli – jeśli obiektowo)

```
mysql_connect("host","uzytkownik","haslo");  
mysql_select_db("baza");  
mysql_query($zapytanie);  
mysql_close();
```

np. zapytanie:

INSERT INTO tabela VALUES(wartości); // by dopisać rekord tabeli.

b) lub skorzystaj z biblioteki obiektowej PDO

(w pliku konfiguracyjnym PHP: php.ini należy odblokować
extension=php_pdo_mysql.dll)

Przykład niżej – pobranie danych z tabeli

```
<?php  
  
try  
{  
    $pdo = new PDO('mysql:host=localhost; port=3307;dbname=test', 'root', '');  
    $pdo -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
    $stmt = $pdo -> query('SELECT id, nazwisko, imie FROM moja ORDER BY id');  
    echo '<ul>';  
    foreach($stmt as $row)  
    {  
        echo '<h1>'.$row['imie'].'</h1>';  
echo $row['nazwisko'].'<hr />';  
    }  
    $stmt -> closeCursor();  
    echo '</ul>';  
}  
catch(PDOException $e)  
{  
    echo 'Połączenie nie mogło zostać utworzone: ' . $e->getMessage();  
}  
?>
```

Uwaga!

Tworząc kod projektu witryny należy także wykorzystać mechanizm sesji; pomoc np.

<http://webmade.org/porady/sesje-php-system-logowania.php>