

Inlämningsuppgift 3

PHP, del 2

Uppgiften ingår i kursen ”Scriptspråk i webbmiljö”



Inlämningsuppgift 3 – PHP, del 2

I detta dokument finns anvisningar för inlämningsuppgift 3 på kursen ”Scriptspråk i webbmiljö”. Inlämningsuppgiften handlar om PHP.

Dagens uppgift består av två uppgifter. Uppgift 1 (A & B) handlar om funktioner i PHP, uppgift 2 (A, B & C) täcker PHP & MySQL.

Sista inlämningsdag för inlämningsuppgift 3 är den 11 oktober 2015.

Funktioner i PHP

Ibland behöver man bara några få rader av PHP-kod i sina filer och då är det kanske onödigt att dela upp koden i funktioner. Om det rör sig om mera kod kan det vara bättre att dela upp koden och definiera funktioner som var för sig gör en eller några få saker i taget. Att skriva funktioner är att föredra inte minst för att då blir enklare att återanvända kod.

Man kan definiera sina funktioner direkt i den PHP-fil där man behöver dem, t.ex.

```
<html>
<head><title></title></head>
<body>

<?php

function my_conc($s1, $s2){
    return $s1 . $s2;
}

function my_square($x){
    return $x * $x;
}

function farlig(&$s){
    $s .= " CHANGED!!!";
}

function ej_farlig($s){
    $s .= "CHANGED!!!";
}

$c = my_square(23);
print("Anropet <code>my_square(23)</code> ger resultat $c\n<p>");

$s = my_conc("Sven", "Svensson");
print("Anropet <code>my_conc('Sven','Svensson')</code> ger resultat $s\n<p>");

$s = "Detta är en sträng som ska skickas som parameter till
farlig(\$s)";
print("\$s = '$s'\n<br>");
farlig($s);
print("Efter anropet <code>farlig(\$s)</code>:\n<br>");
print("\$s = '$s' \n<p>");

$s1 = "Test string";
```

```
print("\$s1 = '$s1'\n<br>");
ej_farlig($s1);
print("Efter anropet <code>ej_farlig(\$s1)</code>:\n<br>");
print("\$s1 = '$s1'\n<br>");
ej_farlig(&$s1);
print("Men efter anropet <code>ej_farlig(&\$s1)</code>:\n<br>");
print("\$s1 = '$s1'\n<p>");

?>

</body>
</html>
```

Man kan även använda *include*-filer, d.v.s. definiera funktioner, variabler, konstanter etc. i en PHP-fil som då kan inkluderas i andra filer. Det kan vara en bra idé att använda include-filer för de funktioner som används ofta och i olika filer (t.ex. koppla upp sig mot en databas). Exemplet ovan kan med en *include*-fil se ut så här:

```
<html>
<head><title></title></head>
<body>

<?php

include 'functions.php';

$c = my_square(23);
print("Anropet <code>my_square(23)</code> ger resultat $c\n<p>");

$s = my_conc("Sven", "Svensson");
print("Anropet <code>my_conc('Sven','Svensson')</code> ger resultat
$s\n<p>");

$s = "Detta är en sträng som ska skickas som parameter till
farlig(\$s)";
print("\$s = '$s'\n<br>");
farlig($s);
print("Efter anropet <code>farlig(\$s)</code>:\n<br>");
print("\$s = '$s' \n<p>");

$s1 = "Test string";
print("\$s1 = '$s1'\n<br>");
ej_farlig($s1);
print("Efter anropet <code>ej_farlig(\$s1)</code>:\n<br>");
print("\$s1 = '$s1'\n<br>");
ej_farlig(&$s1);
print("Men efter anropet <code>ej_farlig(&\$s1)</code>:\n<br>");
print("\$s1 = '$s1'\n<p>");

?>

</body>
</html>
```

där *functions.php* innehåller följande kod:

```
<?php

function my_conc($s1, $s2){
    return $s1 . $s2;
}

function my_square($x){
    return $x * $x;
}

function farlig(&$s){
    $s .= " CHANGED!!!";
}

function ej_farlig($s){
    $s .= "CHANGED!!!";
}

?>
```

Uppgifter, del 1

Inlämningsuppgift, del 1, består av två deluppgifter där båda är obligatoriska.

Uppgift 1 (A)

Skriv en PHP-fil som definierar följande:

- En funktion *green(\$s)* som gör att *\$s* skrivs i grön färg. Du skall kunna använda t.ex.

```
green("Detta skrivs i grön färg");
```

för att få utskriften "**Detta skrivs ut i grön färg**". Tips: Använd ``

- En funktion *write_array(\$arr)* som tar emot arrayen *\$arr* som parameter och visar alla array-element i en tabell enligt

1	<i>element 1</i>
2	<i>element 2</i>
3	<i>element 3</i>
4	<i>element 4</i>
...	...
n	<i>element n</i>

T.ex.:

```
$arr = array("first", "second", "third", "fourth");  
write_array($arr);
```

skall resultera i följande tabell:

1	first
2	second
3	third
4	fourth

- En funktion *my_max(\$arr)* som tar emot en array av heltal och returnerar det största talet. T.ex.

```
$arr = array(1, 34, 2, 5, 46, 7, 8);  
$max = my_max($arr);  
print("Det största talet i arrayen är $max");
```

skall alltså ge utskriften "Det största talet i arrayen är 46".

Uppgift 1 (B)

Skriv en PHP-fil som inkluderar filen från uppgift 1 (A) och testar alla funktioner i den.

PHP & MySQL

I dokumentet ”Kom igång med MySQL” hittar du lite information som ska hjälpa dig att komma igång med MySQL (skapa databaser, tabeller, relationer, manipulera databaser etc.).

OBS! Om du inte har angett något speciellt när du skapade din databas via WampServer kommer ditt användarnamn att vara ”root” och ditt lösenord att vara blankt.

På <http://www.php.net/manual/en/ref.mysql.php> finns en översikt över alla MySQL-specifika PHP-funktioner. Här följer dock förklaring och exempel för några av de viktigaste:

- `mysql_connect('host', 'user', ['password'])`
används för att koppla upp mot en databasserver. Giltigt användarnamn måste anges, och eventuellt användarens lösenord, t.ex.

```
$conn = mysql_connect('localhost', 'root', '');
```

definierar kopplingen "\$conn" som kopplar upp mot "localhost" som användaren "root", identifierad m.h.a. lösenordet "".

- `mysql_select_db('databasname')`
motsvarar MySQL-kommandot `use databasname`, d.v.s väljer databasen som man skall arbeta mot. T.ex.

```
mysql_select_db('mindatabas');
```

väljer databasen "*mindatabas*" som användaren "*root*" har åtkomst till.

- `mysql_query($query)`
skickar SQL-frågan *\$query* till databasen. Funktionen returnerar ett resultatset om query är av typen SELECT, eller TRUE/FALSE om query baseras på INSERT/DELETE/UPDATE. T.ex.

```
$query = "SELECT * FROM test_table";  
$result = mysql_query($query);
```

returnerar i "*\$result*" ett set innehållande alla poster från tabellen "*test_table*" från databasen "*mindatabas*".

Exempel 2

```
$query = "INSERT INTO test_table VALUES('Johan', 'Johansson')";  
$result = mysql_query($query);
```

returnerar TRUE i "*\$result*" om allt var OK med INSERT, annars får "*\$result*" värdet FALSE.

- `mysql_free_result($result)`
nollställer resultatsetet "*\$result*", d.v.s. sätter den till NULL.
- `mysql_close($conn)`
stänger connection "*\$conn*".

Anta att användaren "*root*" med lösenord "" har i sin databas "*mindatabas*" en tabell som heter "*test*". Tabellen innehåller tre fält:

- **pers_id** som är av typen INT NOT NULL AUTO_INCREMENT
- **namn** som är av typen VARCHAR 15
- **e_namn** som är av typen VARCHAR 15

Följande PHP-kod lägger till tre poster i tabellen och skriver sedan ut alla poster.

```
<?php
```

```
$conn = mysql_connect("localhost", "root", "") or die("Could  
not connect");  
mysql_select_db("mindatabas") or die("Could not select database");
```

```
$q1 = "INSERT INTO test VALUES (NULL, 'Peter', 'Persson')";  
$q2 = "INSERT INTO test VALUES (NULL, 'Johan', 'Johansson')";  
$q3 = "INSERT INTO test VALUES (NULL, 'Sven', 'Svensson')";  
$result = mysql_query($q1) or die("Query failed");  
$result = mysql_query($q2) or die("Query failed");
```

```
$result = mysql_query($q3) or die("Query failed");

$q4 = "SELECT * FROM test";
$result = mysql_query($q4) or die("Query failed");

print "<table>\n";
while ($line = mysql_fetch_array($result)) {
    print "<tr>\n";
    $s1 = $line['pers_id'];
    $s2 = $line['namn'];
    $s3 = $line['e_namn'];
    print "<td>Id:</td><td>$s1</td>\n";
    print "<td>Namn:</td><td>$s2</td>\n";
    print "<td>Efternamn:</td><td>$s3</td>\n";
    print "</tr>\n";
}
print "</table>\n";

mysql_free_result($result);
mysql_close($conn);
?>
```

Observera att vi skickar NULL istället för något numeriskt värde för "pers_id". Detta p.g.a att "pers_id" är av typen AUTO_INCREMENT, vilket motsvaras av en *räknare*.

Uppgifter, del 2

Inlämningsuppgift, del 2, består av tre deluppgifter där samtliga är obligatoriska.

Uppgift 2 (A)

I din databas, skapa en tabell `records` med följande kolumner (fält):

1. `rec_id` INT NOT NULL AUTO_INCREMENT
2. `title` VARCHAR(50)
3. `author` VARCHAR (50)
4. `year` INT
5. `price` FLOAT(4,2)

Uppgift 2 (B)

Skapa ett PHP-ssript, *select_records.php*, som listar all information för alla skivor som finns lagrade i databasen (i tabellen `records`). Utskriftet skall vara snyggt formaterat.

Du ska även skapa HTML-filen, *select_records.html*. Från denna fil skall *select_records.php* anropas.

Uppgift 2 (C)

Skapa ytterligare ett PHP-script, *insert_records.php*, som från en HTML-fil tar emot information (title, author, year, price) om en ny skiva och lägger till skivan (en ny post alltså) i tabellen *records*. Om skivan redan är registrerad skall den naturligtvis inte registreras på nytt.

Även här ska du skapa en HTML-fil. Denna fil skall innehålla själva registreringsformuläret där titel, artist, utgivningsår och pris skall kunna anges.

Redovisning

Uppgifterna skall skickas till din lärare via ”Uppgifter” i Blackboard. För att skicka in uppgiften måste du klicka på ”Skicka”. Om du vill spara pågående arbete utan att skicka in kan du klicka på ”Spara som utkast”.

OBS! Alla uppgifter redovisas med att endast en enda (ZIP eller RAR)-fil skickas. Lägg alla filer och kataloger tillhörande din uppgift i en katalog, packa filen så att alla underkatalogstrukturer bibehålls, och namnge filen på ett lämpligt sätt (t.ex. Olof_Olsson_Inlupp1.zip). Alternativt skicka en länk till var jag kan titta på webbsidorna (i så fall behöver själva filerna ej skickas in).