

# PHP och MySQL

# Syfte

- Ge de praktiska kunskaper och färdigheter som krävs för att bygga webbaserade informationssystem med databaser, mer specifikt LAMP-lösningar (Linux, Apache, MySQL, PHP)

## **Mål: Efter föreläsningen ska ni kunna...**

- den viktigaste syntaxen i PHP.
- variabler och tilldelning, villkor, loopar, funktioner, modularisering
- hur databasteori kan appliceras praktiskt i ett informationssystem.
- Använda enkel mailhantering via webb. "Gör din egen hotmail"

# PHP - Inledning

- PHP är uppbyggt från grunden för att passa webben.
- Är ett så kallat "server pages-språk", dvs html-kod blandas med programkod.
  - + Stor spridning
  - + Snabbt
  - + Enkelt
  - Ganska ostrukturerat, om en funktion är användbar läggs den in.
  - Lite inkompatibelt mellan olika versioner

# PHP grunder

- För att kunna använda PHP på en webbserver krävs att webbservern är konfigurerad för detta
  - t.ex. på universitets-laborationswebbservers, men troligen inte spray, geocities etc.
- Filnamnen ska sluta på .php
- Programkod omsluts av processinstruktionen

```
<?php
    diverse programinstruktioner
?>
```

# PHP grunder (2)

- En websida kan bestå av flera block html-kod och php-kod

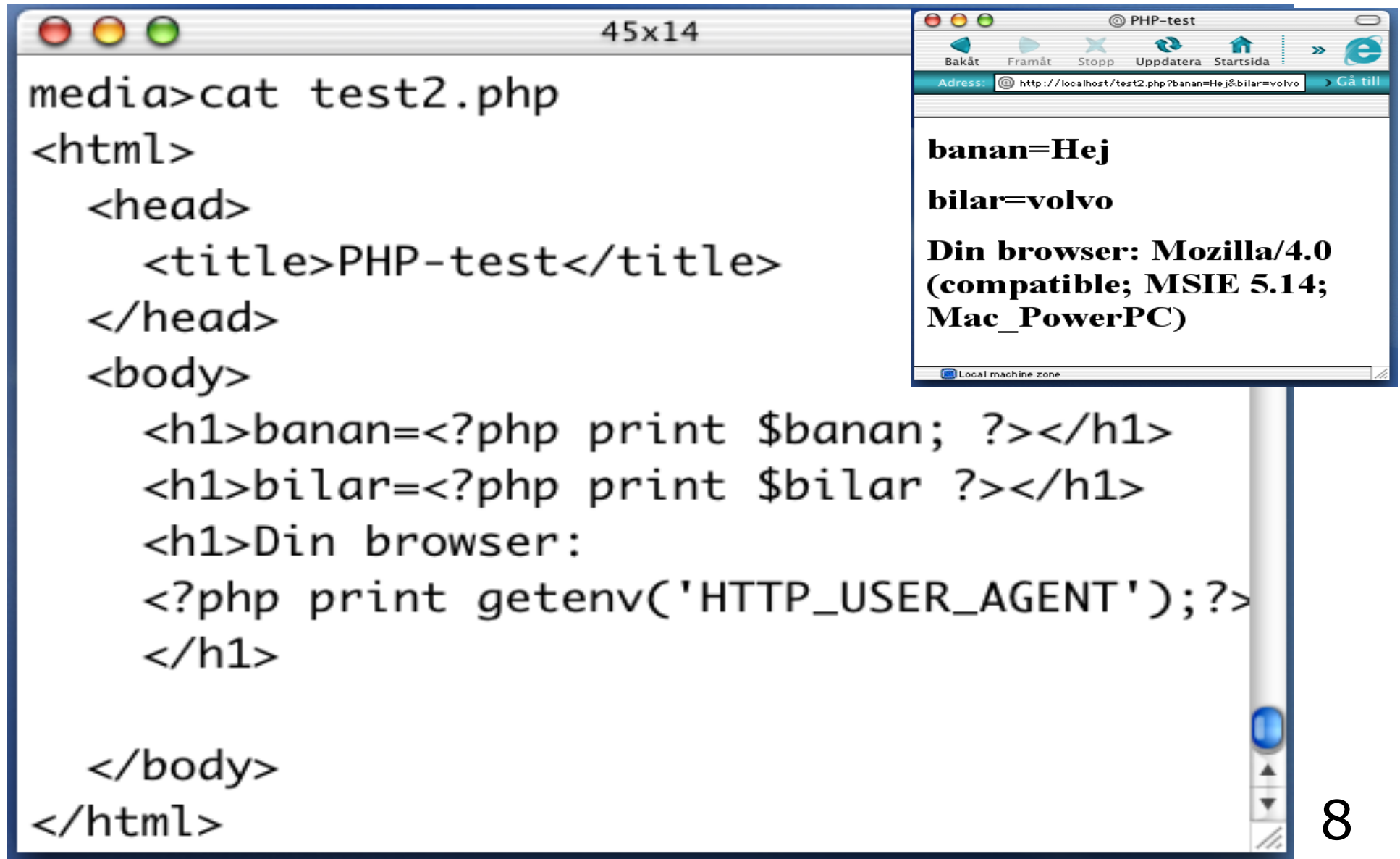
```
<html>
  <head><title>Test</title></head>
  <body>
    <?php
      $a=17;
      print $a;
    ?>
    <br/>
    <?php
      print strftime( "%Y-%m-%d",time() );
    ?>
  </body>
</html>
```

# Variabler

- PHP har ingen "strikt" datatypning. Variabler måste inte deklareraras innan de används.
- Variabler föregås av \$-tecken

```
<?php
    $banan = 17;
    $gurka = "Hej hopp ";
    $myFloat = 14.1;
    print $banan;
    print $gurka;
    print $myfloat;
?>
```

## Variabler från formulär fungerar som "vanliga" variabler



The image shows a terminal window on the left and a web browser on the right. The terminal window, titled '45x14', shows the command 'media>cat test2.php' and the contents of the file 'test2.php'. The file contains an HTML document with three 

# tags. The first two tags display the values of the query string variables 'banan' and 'bilar' using PHP's 'print' function. The third tag displays the value of the 'HTTP\_USER\_AGENT' environment variable using 'getenv()'. The web browser on the right, titled 'PHP-test', shows the rendered output of the PHP script. It displays 'banan=Hej', 'bilar=volvo', and 'Din browser: Mozilla/4.0 (compatible; MSIE 5.14; Mac\_PowerPC)'. The browser's address bar shows the URL 'http://localhost/test2.php?banan=Hej&bilar=volvo'.

```
media>cat test2.php
<html>
  <head>
    <title>PHP-test</title>
  </head>
  <body>
    <h1>banan=<?php print $banan; ?></h1>
    <h1>bilar=<?php print $bilar ?></h1>
    <h1>Din browser:
    <?php print getenv('HTTP_USER_AGENT');?>
    </h1>

  </body>
</html>
```

**banan=Hej**  
**bilar=volvo**  
**Din browser: Mozilla/4.0 (compatible; MSIE 5.14; Mac\_PowerPC)**



# På era webbservrar

- Ofta är webbservrar konfigurerade så att variabler inte importeras "direkt", utan man måste explicit importera dem enligt...

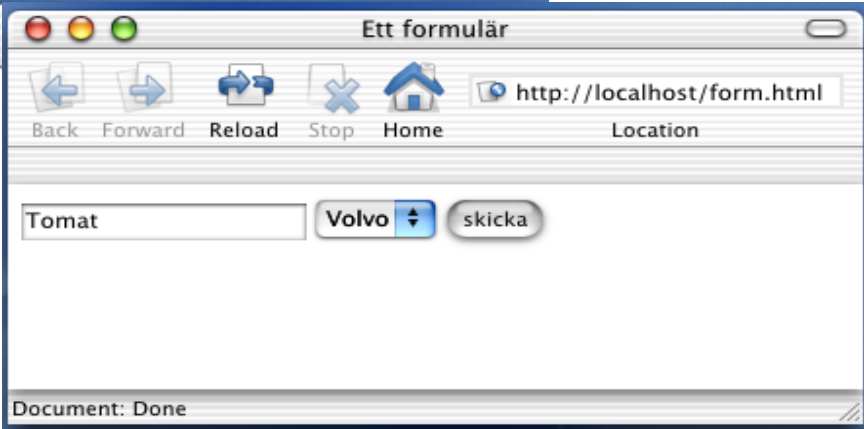
```
<?php import_request_variables("gP","rvar_"); ?>
```

- "gP" står för att man importerar både "GET" och "POST"-variabler
- "rvar\_" är ett godtyckligt (och frivilligt) prefix, vilket kan hjälpa till för att man inte ska blanda ihop externa och lokala variabler.
- Det gör att variabler från t.ex. formulär blir åtkomliga med prefixet rvar, t.ex.

```
<?php print $rvar_banan;?>
```

# ... och så här gjorde man ett formulär i html

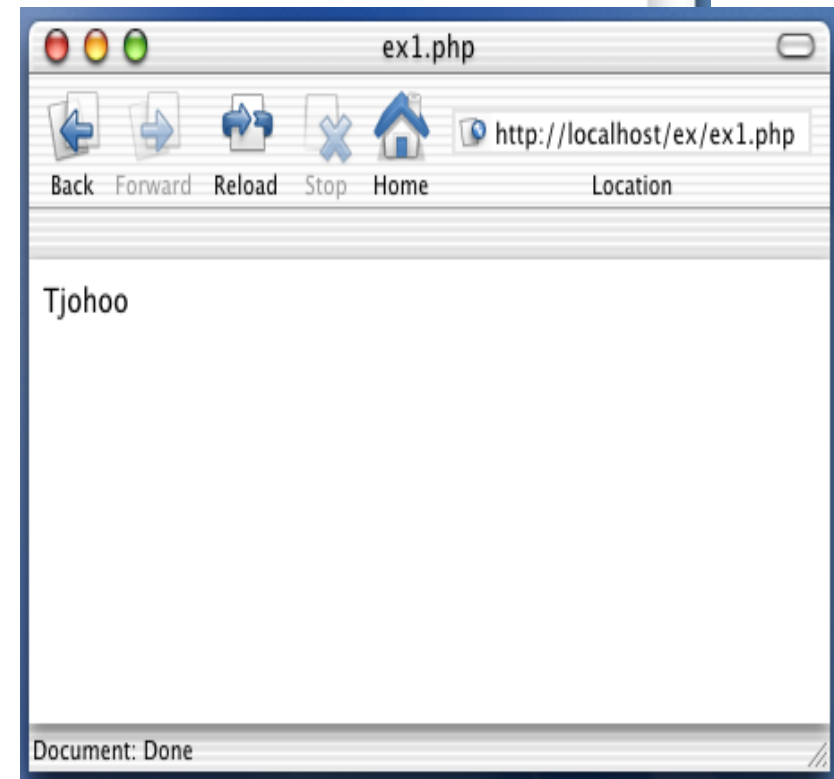
```
media>cat form.html
<html>
<head>
  <title>Ett formulär</title>
</head>
<body>
  <form action="car-form.php" method="GET">
    <input name="banan" value="Tomat"/>
    <select name="bilar">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
    </select>
    <input type="submit" value="skicka"/>
  </form>
</body>
</html>
media>
```



The screenshot shows a web browser window titled "Ett formulär" with the address bar displaying "http://localhost/form.html". The browser's navigation bar includes buttons for Back, Forward, Reload, Stop, and Home. The rendered HTML form is displayed below the address bar, featuring a text input field containing "Tomat", a dropdown menu with "Volvo" selected, and a "skicka" button. The status bar at the bottom of the browser window indicates "Document: Done".

# Villkor med if-else

```
/usr/bin/login (ttyp6)
media>more ex1.php
<html>
  <head><title>ex1.php</title></head>
  <body>
    <p>
      <?php
        $a=15;
        if ($a >17) {
          print "Hej hopp";
        } else {
          print "Tjohoo";
        }
      ?>
    </p>
  </body>
</html>
media>
```



# Block

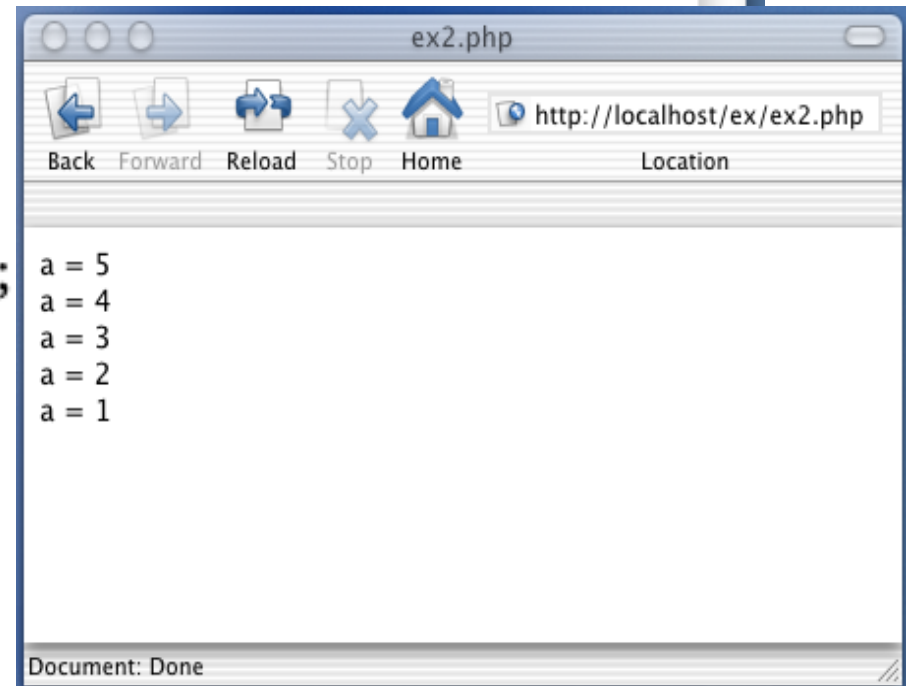
```
media>more ex1.php
<html>
  <head><title>ex1.php</title></head>
  <body>
    <p>
      <?php
        $a=15;
        if ($a >17) { ←
          print "Hej hopp";
        } else { ←
          print "Tjohoo";
        }
      ?>
    </p>
  </body>
</html>
media>
```

- För att knyta samman flera satser till ett block, motsvarande BEGIN och END i Pascal används "måsvingar" {}.
- Dessa fås på en mac genom att samtidigt trycka ner alt-shift-8 respektive alt-shift-9

# Loopar med while

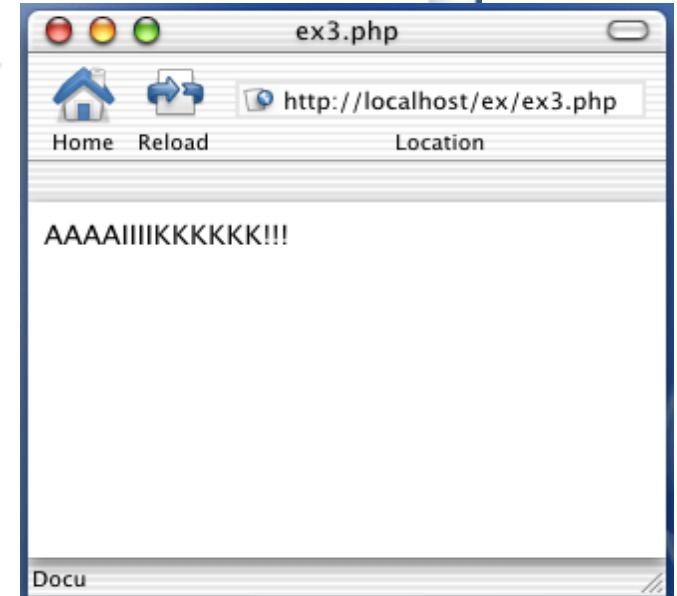
Obs! Vanlig html-kod genereras

```
media>cat ex2.php
<html>
  <head><title>ex2.php</title></head>
  <body>
    <p>
      <?php
        $a=5;
        while ($a > 0) {
          print "a = $a <br/>";
          $a=$a-1;
        }
      ?>
    </p>
  </body>
</html>
media>
```

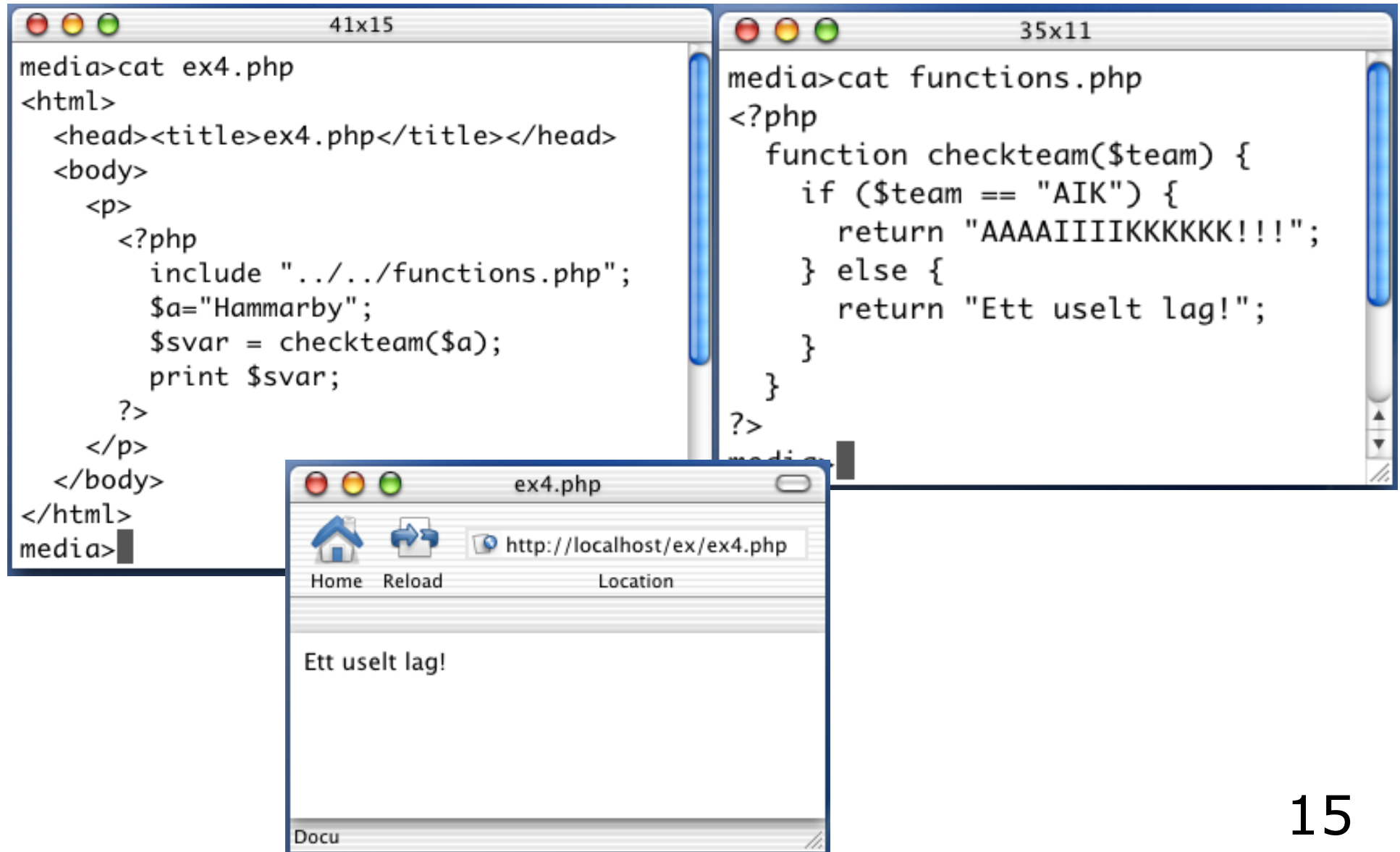


# Funktioner

```
media>more ex3.php
<html>
  <head><title>ex3.php</title></head>
  <body>
    <p>
      <?php
        function checkteam($team) {
          if ($team == "AIK") {
            return "AAAAIIIIKKKKKK!!!";
          } else {
            return "Ett uselt lag!";
          }
        }
        $a="AIK";
        $svar = checkteam($a);
        print $svar;
      ?>
    </p>
  </body>
</html>
media>
```



# Includes



The image shows a terminal window and a web browser window. The terminal window has two panes. The left pane shows the contents of `ex4.php`, which includes an HTML header, a body, and a PHP block that includes `../../functions.php`, sets `$a="Hammarby"`, calls `checkteam($a)`, and prints the result. The right pane shows the contents of `functions.php`, which defines a `checkteam` function that returns "AAAAIIIIKKKKKK!!!" for "AIK" and "Ett uselt lag!" for any other team. The web browser window, titled `ex4.php`, shows the output of the script: "Ett uselt lag!".

```
media>cat ex4.php
<html>
  <head><title>ex4.php</title></head>
  <body>
    <p>
      <?php
        include "../../functions.php";
        $a="Hammarby";
        $svar = checkteam($a);
        print $svar;
      ?>
    </p>
  </body>
</html>
media>
```

```
media>cat functions.php
<?php
  function checkteam($team) {
    if ($team == "AIK") {
      return "AAAAIIIIKKKKKK!!!";
    } else {
      return "Ett uselt lag!";
    }
  }
?>
```

ex4.php

Home Reload Location

http://localhost/ex/ex4.php

Ett uselt lag!

Docu

## Includes (2)

- Includes är lämpliga för att lägga funktioner som förekommer flera gånger.
- Det går även att lägga t.ex. variabler i "config-filer" som återkommer i flera filer, t.ex. IP-adresser till andra datorer, databaslösenord etc.
- Includefilerna kan, till skillnad från html och php-filer, ligga var som helst i filstrukturen vilket kan vara både bra och dåligt ur säkerhetssynvinkel  
(include "/etc/passwd")



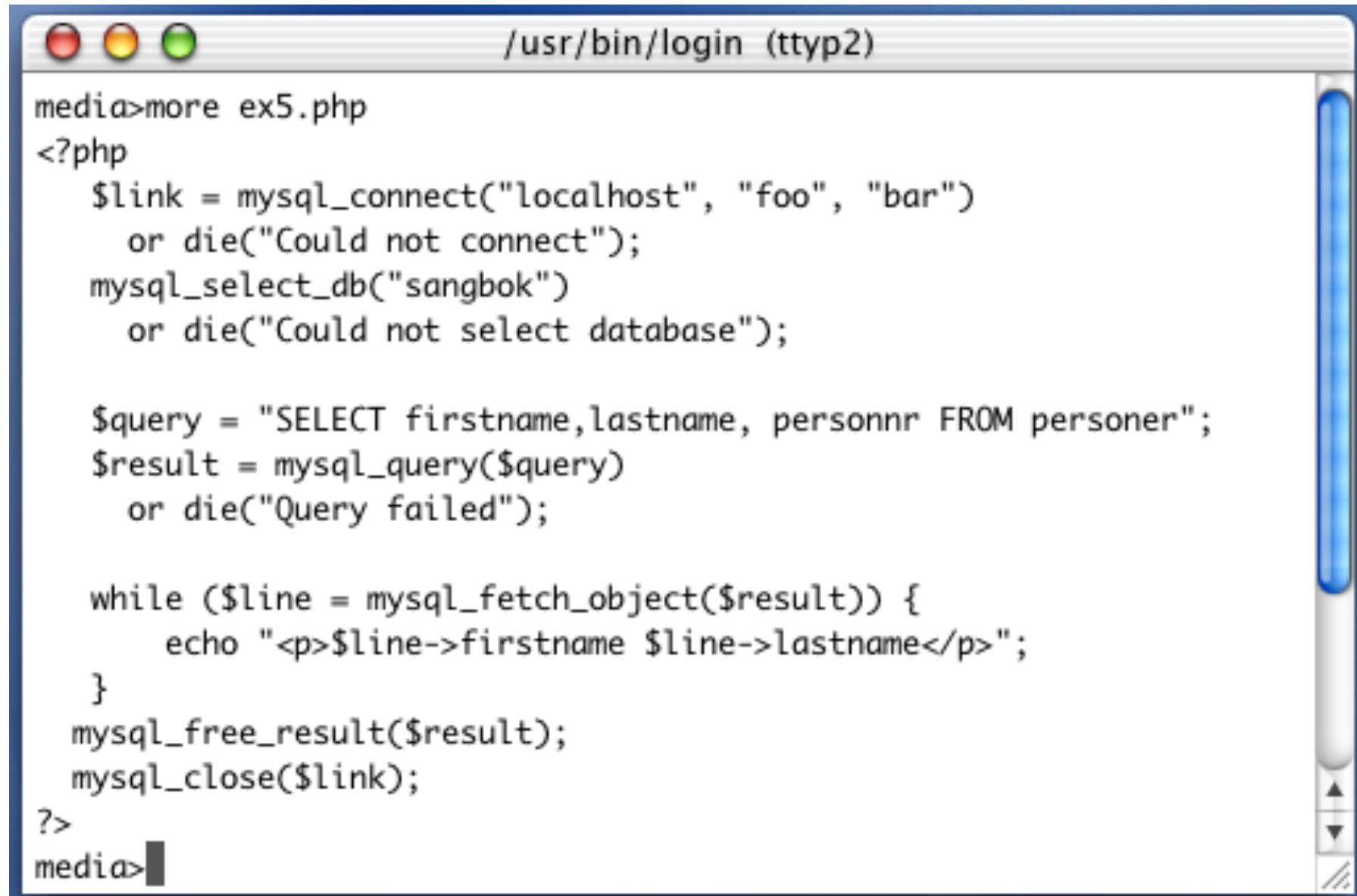
# PHP och Databaser - MySQL

- MySQL är en gratis (open source) databashanterare till Unix och Windows ([www.mysql.com](http://www.mysql.com)) utvecklat av ett svenskt företag
  - + Gratis
  - + Brett "community support"
  - + Mycket snabb
  - Klarar inte full SQL
  - Kan inte hantera rollbacks (dock finns vissa möjligheter i nyare versioner, men då påverkas prestanda negativt)

# Webbgränssnitt till MySQL

- Det finns ett utmärkt webbaserat gränssnitt, phpMyAdmin, som används i kursen.
- Används huvudsakligen för att skapa och modifiera tabeller, samt lägga in testdata i början.

# MySQL fr n php



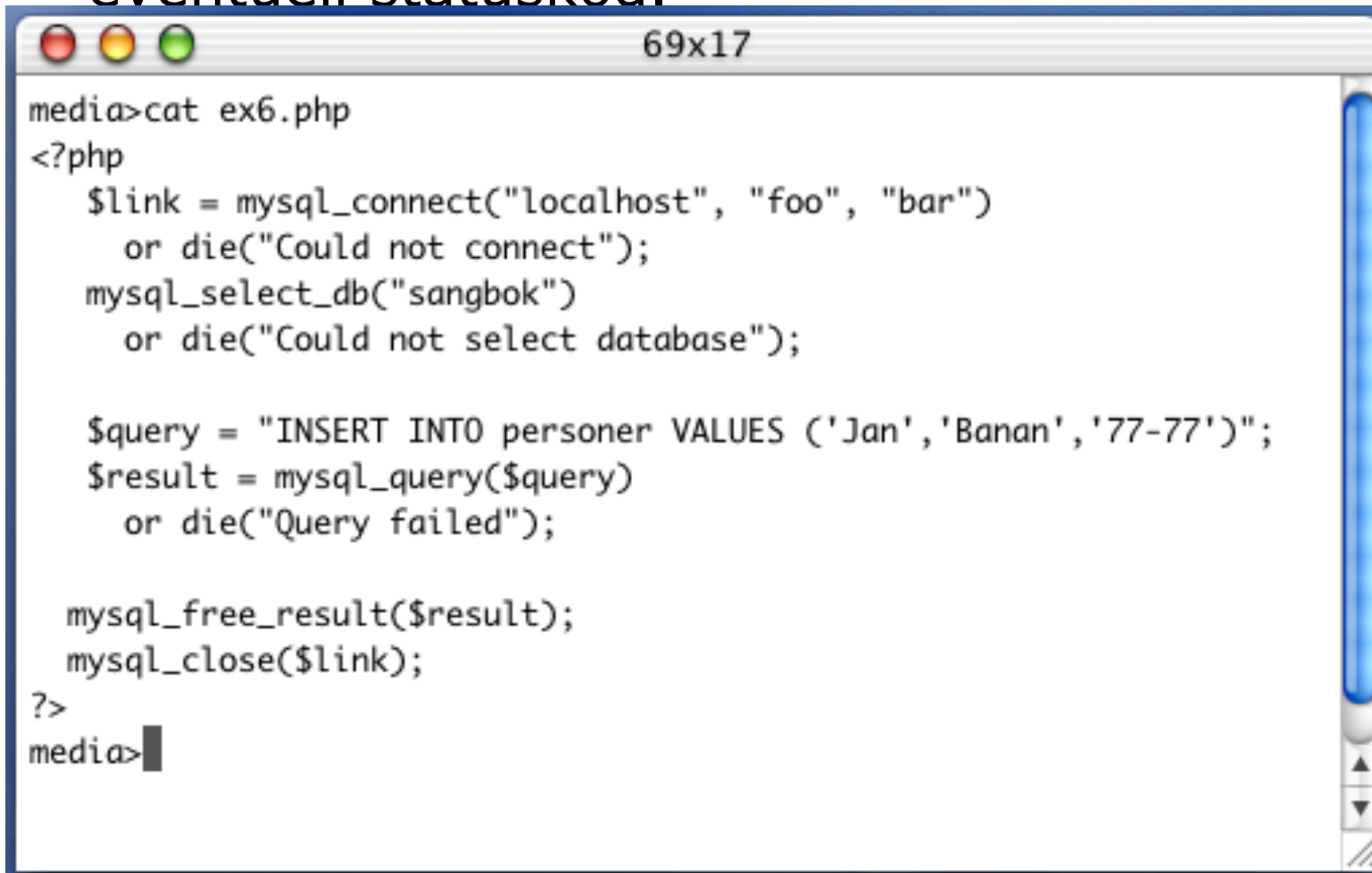
```
/usr/bin/login (ttyp2)
media>more ex5.php
<?php
    $link = mysql_connect("localhost", "foo", "bar")
        or die("Could not connect");
    mysql_select_db("sangbok")
        or die("Could not select database");

    $query = "SELECT firstname,lastname, personnr FROM personer";
    $result = mysql_query($query)
        or die("Query failed");

    while ($line = mysql_fetch_object($result)) {
        echo "<p>$line->firstname $line->lastname</p>";
    }
    mysql_free_result($result);
    mysql_close($link);
?>
media>
```

# INSERT och UPDATE

- Insert och update fungerar som select, framtaget att satsen inte returnerar något "result-set", endast en eventuell statuskod.



```
media>cat ex6.php
<?php
    $link = mysql_connect("localhost", "foo", "bar")
        or die("Could not connect");
    mysql_select_db("sangbok")
        or die("Could not select database");

    $query = "INSERT INTO personer VALUES ('Jan', 'Banan', '77-77')";
    $result = mysql_query($query)
        or die("Query failed");

    mysql_free_result($result);
    mysql_close($link);
?>
media>
```

# PHP + XSL

- Före själva innehållet skickas en header tillbaka till en klient som bl.a. Talar om vilken sorts innehåll det är fråga om. S.k. Mime-typ.
  - text/html, image/gif, audio/aiff, text/xml
- För att Mozilla eller IE ska förstå att den ska försöka göra XSLT-transformationer måste mime-typen vara text/xml. Denna måste i så fall sättas explicit.
- Det kan vara användbart att avgöra mime-typ beroende på vilken webbläsare som försöker komma åt sidan (ingen idé att försöka göra XSLT om webbläsaren ändå inte klarar det).
- Transformationerna kan även ske på servern, t.ex. via PHP

# PHP + mysql + XSL

```
<?php
    header("Content-type: text/xml");
    $UA = getenv('HTTP_USER_AGENT');
    if (ereg("Symbian", $UA)) {
        print '<?xml-stylesheet type="text/xsl" href="symb.xsl"?>';
    }
    $link = mysql_connect("localhost", "sangbok", "sangbok")
        or die("Could not connect");
    mysql_select_db("sangbok")
        or die("Could not select database");

    print "<songbook>";
    $query = "SELECT title, melody, text FROM song";
    $result = mysql_query($query)
        or die("Query failed");
    while ($line = mysql_fetch_object($result)) {
        print "<song>";
        print "<title>$line->title</title>";
        print "<melody>$line->melody</melody>";
        print "<text>$line->text</text>";
        print "</song>";
    }
    mysql_free_result($result);
    mysql_close($link);
    print "</songbook>";
```

# Sessioner i PHP

- I PHP finns inbyggt stöd för sessioner vilket gör hantering av variabler mellan sidor enkelt.

fil1.php

```
<?php
import_request_variables("gP", "rvar_");
session_start();
$_SESSION['IMEI'] = $rvar_IMEI; // store session data
print "<a href='fil2.php'>Fil 2</a>";
?>
```

fil2.php

```
<?php
session_start();
print "IMEI = " . $_SESSION['IMEI'];
?>
```

- För mer information, se: <http://se.php.net/manual/en/ref.session.php>

# Mail i php

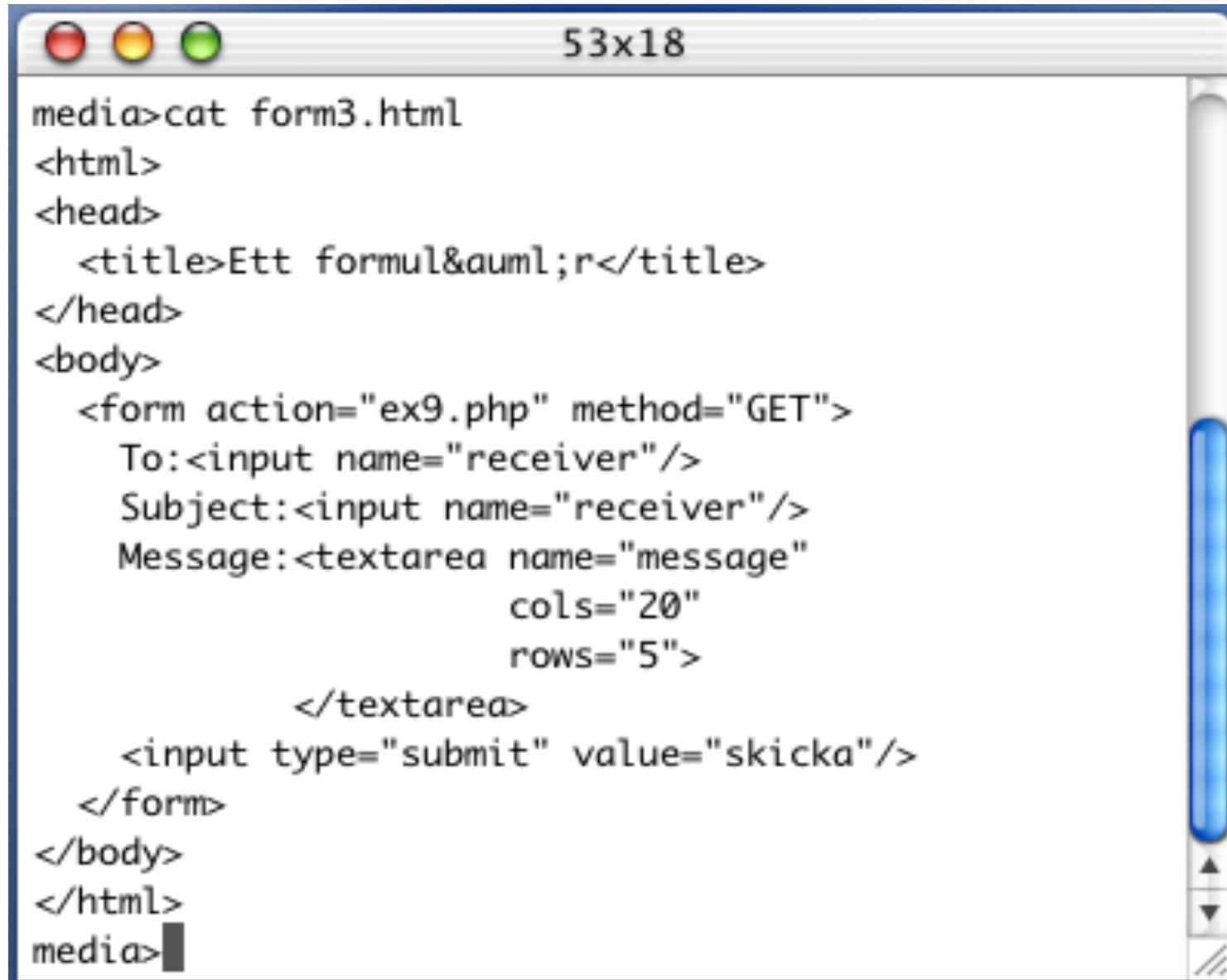
- Att skicka ett mail från php är enkelt (om webbservern är konfigurerad för det (vilket (nog) inte är fallet för er))

A screenshot of a terminal window with a title bar showing three colored buttons (red, yellow, green) and the text "52x14". The terminal content shows a command prompt "media>" followed by "cat ex8.php". The output is an HTML document with a PHP script. The PHP script sets a receiver email address, uses the mail() function to send an email with the subject "Mail skickat med PHP" and body "Hejhopp\nbody-delen av meddelandet", and then prints an HTML paragraph. The terminal ends with the prompt "media>" and a cursor.

```
media>cat ex8.php
<html>
  <head><title>Emailtest</title></head>
  <body>
<?php
  $receiver = "nobody@somewhere.com";
  mail("$receiver", "Mail skickat med PHP",
      "Hejhopp\n
body-delen av meddelandet");
  print "<p>Mail skickat till $receiver</p>";
?>
  </body>
</html>
media>
```



# Formulär för enkel webbmail

A terminal window with a title bar containing three colored buttons (red, yellow, green) and the text "53x18". The terminal displays the command "media>cat form3.html" and the output, which is an HTML document. The HTML document has a title "Ett formulär" and a form with fields for "To:", "Subject:", and "Message:", and a "skicka" button.

```
media>cat form3.html
<html>
<head>
  <title>Ett formulär</title>
</head>
<body>
  <form action="ex9.php" method="GET">
    To:<input name="receiver"/>
    Subject:<input name="receiver"/>
    Message:<textarea name="message"
              cols="20"
              rows="5">
              </textarea>
    <input type="submit" value="skicka"/>
  </form>
</body>
</html>
media>
```

# Sammanfattning

- Databashantering är enkel i PHP. Grundprincipen är:
  - Skapa en uppkoppling mot en databas
  - Exekvera ett statement
  - Om det var ett select-statement så iterera över en resultatmängd (dvs loopa över alla rader som returneras).
- Sessioner behövs för att kunna hålla ett tillstånd bland en mängd webbsidor. Ganska enkelt, använd gärna i projekten.

# Referenser

## PHP

- [www.zend.com](http://www.zend.com)
- [www.php.net](http://www.php.net)

## MySQL

- [www.mysql.com](http://www.mysql.com)
- Allt möjligt
  - [www.google.com](http://www.google.com)
  - [Groups.google.com](http://Groups.google.com)