



KTH
Kungliga Tekniska Högskolan

Webbaserade informationssystem med PHP och databaser

2D1522 Datorteknik och -kommunikation
2D2051 Datbasteknik och datorkommunikation

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

1



KTH
Kungliga Tekniska Högskolan

Dagens föreläsning

- Syfte
 - Ge de praktiska kunskaper och färdigheter som krävs för att bygga webbaserade informationssystem med databaser.
- Mål
 - Lära er den viktigaste syntaxen i PHP.
 - Variabler och tilldelning, villkor, loopar, funktioner, modularisering
 - Lära er hur databasteori kan appliceras praktiskt i ett informationssystem.
 - Lära er hur man kan göra sessioner, dvs hantera ett tillstånd såsom en shoppingvagn eller en inloggning.
 - URL-rewriting, Hidden fields, Cookies
 - Mailhantering via web. ”Gör din egen hotmail”

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

2



KTH
Kungliga Tekniska Högskolan

Dagens innehåll

- Hur skapa dynamiskt innehåll med PHP
- PHP-syntax
- Web och databaser
- Sessionshantering
- Mailhantering via web

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

3



KTH
Kungliga Tekniska Högskolan

Observationer gällande cgi-script

- I de flesta fall ska output ha mime-typen text/html
- I dynamiskt genererade html-sidor är ofta största delen av sidan ”statisk”, dvs statiska print-satser utan något dynamiskt innehåll.
- De flesta programmeringsspråk är utvecklade ”före” webben, och är inte specialanpassade just för web-bruk.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

4



Server-Pages-språk

- Snart efter webbens uppkomst utvecklades flera så kallade "server pages-språk".
- Specialanpassade för web-bruk
- Hanterar problemen på föregående sida på ett "bra sätt".
- Exempel: asp, jsp, php

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

5



PHP

- I denna kurs använder vi i år språket PHP som vunnit mycket i popularitet de senaste åren.
- Enkelt för nybörjare
- Stor spridning (5 miljoner servers för några år sedan)
- Snabbt

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

6



Grundtanken i server-pages-språk

- Är nästan som vanliga html-dokument, men på de platser man vill ha dynamiskt innehåll kan man placera programkod.
- Webservern noterar att sidan är en PHP-sida och låter en php-tolk förbehandla sidan, dvs exekvera programsnuttarna.
- Resultatet är ren html-kod (eller xml eller gif eller vad man vill) och tolkas som vanligt av webläsaren.
- Om inget annat anges sätts MIME-typen automatiskt till text/html

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

7



Enkelt exempel i PHP

```
media>cat test.php
<html>
<head>
<title>PHP-test</title>
</head>
<body>
<h1>
<?php print date("Y-m-d");?>
</h1>
</body>
</html>
media>
```

The browser window shows the output of the PHP code: 2002-05-02

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

8



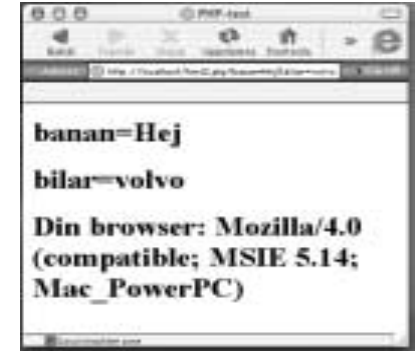
Formulär-variabler i PHP

- Det är mycket enkelt att komma åt variabler som skickats i en GET eller POST-request.
- De kan refereras genom `$_REQUEST["variabelnamn"]`, ibland även med `$variabelnamn` beroende på hur PHP installerats.
- Övriga variabler (t.ex. `HTTP_USER_AGENT`) kan hämtas med funktionen `GETENV`.



Variabelexempel i PHP

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



Dagens innehåll

- Hur skapa dynamiskt innehåll med PHP
- PHP-syntax
- Web och databaser
- Sessionshantering
- Mailhantering via web



Språket PHP

- PHP är uppbyggt från grunden för att passa webben.
- Är ett så kallat "server pages-språk", dvs html-kod blandas med programkod.
- Fördelar
 - Stor spridning
 - Snabbt
 - Enkelt
- Nackdelar
 - Ganska ostrukturerat, om en funktion är användbar läggs den in.
 - Lite okompatibelt mellan olika versioner (exempel på detta kommer längre fram)



PHP grunder

- För att kunna använda PHP på en webserver krävs att webservern är konfigurerad för detta
 - t.ex. dupond.nada.kth.se:8080 och www.nada.kth.se, 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><?php print date("Y-m-d");?></title>
  </head>
  <body>
    <?php
      $a=17;
      print $a;
    ?>
  </body>
</html>
```



Variabler

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

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



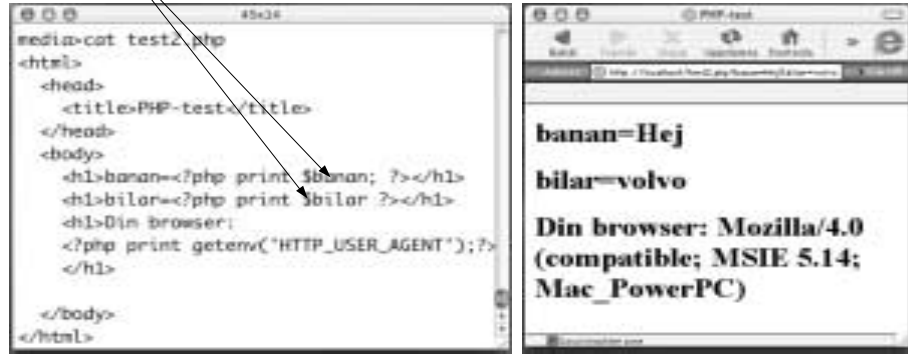
Om vi gör ett formulär i html ...





... så kommer vi åt variablerna från formuläret på olika sätt beroende på hur PHP lagts upp:

Som "vanlig" variabel (detta är en säkerhetsrisk och är vanligen avstängt).



Ett säkrare sätt att komma åt formulärvariabler

```
<html>
<head>
  <title>PHP-test</title>
</head>
<body>
  <h1>banan=<?php print $_REQUEST["banan"]; ?></h1>
  <h1>bilar=<?php print $_REQUEST["bilar"]; ?></h1>
  <h1>Din browser:
  <?php print getenv("HTTP_USER_AGENT"); ?>
  </h1>
</body>
</html>
```

Arrayen `$_REQUEST` användas för både GET-, POST- (och cookie-) variabler.



Formulär-variabler i PHP

- Det är mycket enkelt att komma åt variabler som skickats i en GET eller POST-request.
- De kan refereras genom `$_REQUEST["variabelnamn"]` ibland även med `$variabelnamn` beroende på hur PHP installerats.
- Övriga variabler (t.ex. `HTTP_USER_AGENT`) kan hämtas med funktionen `GETENV`.



Villkor med if-else





Block

```

media>more ex1.php
<html>
<head><title>ex1.php</title></head>
<body>
<p>
$0=15;
if ($0 > 10) {
print "Hej hej!";
} else {
print "Tjæhej";
}
</p>
</body>
</html>
media>

```

För att knyta samman flera satser till ett block används ”måsvingar” {} på samma sätt som i Java och C. Dessa fås vanligen genom att samtidigt trycka ner altgraph-7 respektive altgraph-0



Loopar med while

Obs! Vanlig html-kod som genereras

```

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

```



Funktioner

```

media>more ex3.php
<html>
<head><title>ex3.php</title></head>
<body>
<?php
function checkteam($team) {
if ($team == "AIK") {
return "AAAAIIIIKKKKKK!!!";
} else {
return "Ett uselt lag!";
}
}
$0="AIK";
$svar = checkteam($0);
print $svar;
?>
</p>
</body>
</html>
media>

```



Includes

```

media>cat ex4.php
<html>
<head><title>ex4.php</title></head>
<body>
<p>
<?php
include "../functions.php";
$0="Hammarby";
$svar = checkteam($0);
print $svar;
?>
</p>
</body>
</html>
media>

```





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”`)



Databaser och web

- PHP-syntax
- Web och databaser
- Sessionshantering
- Mailhantering via web



MySQL

- Gratis (open source) databashanterare till Unix och Windows (www.mysql.com) utvecklat sv svenskt företag
- Fördelar
 - Gratis
 - Brett ”community support”
 - Mycket snabb då man gör enkla saker



Informix

- I år använder vi Informix som ni redan använt i laboration 2.
- Fördelar
 - Klarar hela SQL 1
- Nackdelar
 - Klarar endast SQL 1



Informix använda dbaccess

```
nadall:~>ssh dupond
inge@dupond's password: nada-lösenordet
Last login: Fri May 02 2003 14:14:57 from
nadall.nada.kth.
Sun Microsystems Inc. SunOS 5.6
Generic August 1997
No mail.
[1] 3756
dupond>module add informix/2000
dupond>dbaccess
```



Använda dbaccess

```
DBACCESS: Query-language Connection Database Table Session Exit
Use SQL query language.
```

```
----- Press CTRL-W for Help -----
```

Flytta -> med piltangent

Välj genom att trycka på return

```
DBACCESS: Query-language Connection Database Table Session Exit
Select, Create, Info, Drop or Close a database.
```

```
----- Press CTRL-W for Help -----
```



Använda dbaccess forts

```
DATABASE: Select Create Info Drop cLose Exit
Select a database to work with.
```

```
----- Press CTRL-W for Help -----
```

```
SELECT DATABASE >>
```

```
Select a database with the Arrow Keys, or enter a name, then press
Return.
```

```
----- @course_2000 ----- Press CTRL-W for Help -----
```

```
_me99_enj@course_2000      aaaaarvmart@course_2000
aaadanielnovak@course_2000 aaaskivor@course_2000
aaaaarvidmartin@course_2000 ...
```

Flytta ner med



Dbaccess – välja databas

```
SELECT DATABASE >>
```

```
Select a database with the Arrow Keys, or enter a name, then press
Return.
```

```
----- @course_2000 ----- Press CTRL-W for Help -----
```

```
...
cd@course_2000          danielmartin@course_2000
cdb@course_2000        datkom03koklista@course_2000
cdhregel@course_2000  datkom03varuhuset@course_2000
danemarwiss@course_2000 datkom04koklista@course_2000
...
```

Välj



Dbaccess – vidare till SQL-editorn

```

DATABASE:  Select Create Info Drop cLose Exit
Return to the DBACCESS menu.

-- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

```

Välj

```

DBACCESS:  Query-language Connection Database Table Session Exit
Use SQL query language.

-- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

```

Välj



Dbaccess – SQL-editorn

```

SQL:  New Run Modify Use-editor Output Choose Save Info Drop Exit
Enter new SQL statements using SQL editor.

-- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

```

Exekvera
SQL-satser

Välj

```

SQL-editorn          Skriv SQL-satser här

NEW:  ESC    = Done editing      CTRL-A = Typeover/Insert    CTRL-R = Redraw
      CTRL-X = Delete character  CTRL-D = Delete rest of line

----- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

```



Informix – skapa en tabell

```

SQL:  New Run Modify Use-editor Output Choose Save Info Drop Exit
Run the current SQL statements.

----- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

create table tabort (hej integer, hopp varchar(30, 2));

Table created.

```

Tabellnamn

Kolumnnamn

Datatyp



Informix – sätta in värden i en tabell

```

SQL:  New Run Modify Use-editor Output Choose Save Info Drop Exit
Run the current SQL statements.

----- datkom03varuhuset@course_2000 ---- Press CTRL-W for Help -----

insert into tabort values (7, "Banan");

1 row(s) inserted.

```

Tabellnamn

Värde kolumn 1

Värde kolumn 2



Informix – selektera värden från en tabell

```
NEW:   ESC   = Done editing      CTRL-A = Typeover/Insert    CTRL-R = Redraw
       CTRL-X = Delete character  CTRL-D = Delete rest of line
```

```
----- datkom03varuhuset@course_2000 ----- Press CTRL-W for Help -----
select * from tabort;
```

Kolumnnamn

```
SQL:   New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.
```

```
----- datkom03varuhuset@course_2000 ----- Press CTRL-W for Help -----
```

```
    hej hopp
```

```
        7 Banan
```

```
1 row(s) retrieved.
```



Informix – uppdatera rader i en tabell

```
SQL:   New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.
```

```
----- datkom03varuhuset@course_2000 ----- Press CTRL-W for Help -----
```

```
update tabort set hopp="Tomat" where hej=7;
```

```
1 row(s) updated.
```



Informix – ta bort rader från en tabell

```
SQL:   New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.
```

```
----- datkom03varuhuset@course_2000 ----- Press CTRL-W for Help -----
```

```
delete from tabort where hej=7;
```

```
1 row(s) deleted.
```



Informix – radera en tabell

```
SQL:   New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.
```

```
----- datkom03varuhuset@course_2000 ----- Press CTRL-W for Help -----
```

```
drop table tabort;
```

```
Table dropped.
```



Informix från php

- Obs!!! Från och med nu skippar jag html-innehållet som ska vara före och efter php-blocken för att spara skärmutrymme!!!

```
<?php
$conn_id = ifx_connect ("sangbok@course_2000")
    or die("Kunde inte kontakta databasen sangbok");
$query = "SELECT firstname, lastname, personnr FROM personer"; ← SQL-sats
$result = ifx_query ($query, $conn_id) ← Exekvera SQL-sats
    or die("Kan inte hämta personer");
while (is_array($row = ifx_fetch_row($result, "NEXT"))) {
    print("<p>$row[firstname] $row[lastname]</p>\n");
}
ifx_free_result($result);
ifx_close($conn_id);
?>
```



INSERT och UPDATE

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

```
<?php
$conn_id = ifx_connect ("sangbok@course_2000")
    or die("Kunde inte kontakta databasen sangbok");
$query = "INSERT INTO personer VALUES ('Jan', 'Banan', '77-77')";
$result = ifx_query ($query, $conn_id)
    or die("Kan inte sätta in i personer");
ifx_close($conn_id);
?>
```



Sessionshantering

- PHP-syntax
- Web och databaser
- Sessionshantering
- Mailhantering via web



Vad är en session

- Webben är i princip tillståndslös. Varje hämtning av en sida är oberoende av andra hämtningar.
- Ofta vill man dock ha en "kontext", där resultat av tidigare sidladdningar kan påverka framtida.
- En session skapar just en sådan kontext



Hur sessioner?

- Grundprincipen är enkel:
 - Se till att skaffa en unik identifierare som ska identifiera sessionen, en s.k. sessionsid.
 - Se till att denna sessionsid på något sätt är tillgänglig i alla script som ska utgöra sessionen, dvs att den på något sätt skickas med från klienten vid varje uppkoppling.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

45



Hidden fields

- Den enklaste, och mest ursprungliga metoden är att lägga in denna sessionsid i ett dolt fält (hidden variable) som fylls i i varje html-formulär som utgör sessionen.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

46



Exempel hidden fields



2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

47



URL-rewriting

Det är inte nödvändigt att använda formulär överallt i en session. Man kan istället simulera formulär genom att i alla sidor som skickas till klienten, skriva om länkar så att de ser ut som de kom från ett formulär.

Exempel:

``

ersätts med

``

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

48



Nackdelar med hidden fields och URL-rewriting

- Normalt kombinerar man hidden fields och URL-rewriting.
- Formulär använder hidden fields och övriga länkar använder URL-rewriting.
- Nackdelen med denna teknik är att så fort man går till en extern site så tappas sessionen.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

49



Sessioner med Cookies

- Cookies är ett annat, bättre sätt att hantera sessioner .
- Varje server har rätt att lagra upp till 256 tecken på klientdatorn i en speciell cookiefile. Detta är dock kontroversiellt hos somliga användare.
- Det som lagras är ett antal <variabelnamn><variabelvärde>
- Oftast är det smartast att lagra endast en sessionsid på klienten, och sedan låta all övrig intressant information som servern kan vilja lagra ligga på servern i en databas där sessionsiden är "primary key" för övrig data.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

50



```
medico@cat: ~$ cat ex7.php
<?php
setcookie("email", "bjorn@kth.se");
?>
<html>
<head><title>ex7.php</title></head>
<body>
<p>
<a href="ex7.1.php">Till nästa sida</a>
</p>
</body>
</html>
medico@cat: ~$

medico@cat: ~$ cat ex7.1.php
<?php
if(isset($_COOKIE["email"])){
    echo $_COOKIE["email"];
}
?>
<html>
<head><title>ex7.1.php</title></head>
<body>
<p>
    din email är: bjorn@kth.se
</p>
</body>
</html>
medico@cat: ~$
```

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

51



Mer om cookies

- Namnkonflikter
 - Om det finns risk för namnkonflikter, dvs att man använder variabelnamn i cookies som även kanske används "lokalt" går det att komma åt cookies via `_REQUEST`.
- `echo "$_REQUEST[email]";`
- Säkerhet
 - En server kan endast komma åt cookies som servern själv har satt.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

52



Mer om cookies (2)

- Värdena på cookies för en viss webbplats skickas med i http-headern varje gång en request görs.
- Från servern:
 - `Content-type: text/html`
 - `Set-Cookie: foo=bar; path=/; expires Mon, 09-Dec-2002 13:46:00 GMT`
- Från klienten
 - `Content-type: text/html`
 - `Cookie: foo=bar`
- Om inget expire-date anges raderas cookien när webbläsaren avslutas.

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

53



Mail

- PHP-syntax
- Web och databaser
- Sessionshantering
- Mailhantering via web

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

54



Mail i php

- Att skicka ett mail från ett php-program är enkelt. I sin enklaste form behövs en mottagaradress, en rubrikerad och en meddelandekropp

```

medioweb:~$ cat test.php
<html>
<head>
<title>E-mailtest</title></head>
<body>
<?php
$receiver = "robod@somewhere.com";
mail("$receiver", "Mail skickat med PHP",
"hej hopp!");
body-delen av meddelandet");
print "<?phpMail skickat till $receiver</p>";
?>
</body>
</html>
medioweb:~$

```

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

55



skicka webmail

```

medioweb:~$ cat form.html
<html>
<head>
<title>E-mailtest</title></head>
<body>
<?php
$form action="test.php" method="GET";
To:<input name="receiver"/>
Subject:<input name="subject"/>
Message:<textarea name="message"
cols="20"
rows="3">
</textarea>
<input type="submit" value="skicka"/>
</form>
</body>
</html>
medioweb:~$

```

```

medioweb:~$ cat test.php
<html>
<head>
<title>E-mailtest</title></head>
<body>
<?php
$receiver = "robod@somewhere.com";
mail("$receiver", "$subject", "$body");
print "<?phpMail skickat till $receiver</p>";
?>
</body>
</html>
medioweb:~$

```

2006-04-20

© Björn Hedin, Inge Frick, NADA/KTH 2002-2006

56



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 websidor
- Hidden fields och URL-rewriting kan vara enkla ibland, men oftast är cookies bättre (alla användare accepterar dock inte cookies).
- Mailhantering är mycket enkel via webben (dock något mer komplicerad för att läsa mail).



Referenser

Referenser

Cookies

- <http://perl.about.com/library/weekly/aa051600d.htm>
- <http://www.cookiecentral.com/>

PHP

- www.zend.com
- www.php.net (observera att man kan söka efter funktioner i manualen)
- www.php.net/manual/sv/printwn/index.php (manualen)