

Lektion 4

DD2390
Internetprogrammering
6 hp

Innehåll

- PHP
- Databasinteraktion
- AJAX

PHP är

- Ett serverscriptspråk för att skapa dynamiska websidor, nuvarande version PHP 5.3.6, se www.php.net.
- Ett open-source projekt, PHP körs som en modul under apache, se www.apache.org.
- Finns även som plug-in till IIS (cgi-bin)
- Finns stöd för på CSC:s unix konton under public_html katalogen
- Mycket populärt, går snabbt att komma igång med, mindre komplicerat än Java EE och ASP.NET.

PHP-mode

- PHP-koden läggs inbäddad i en html-sida (med ändelse .php istället) och man alternerar mellan html/php-läge med speciella taggar för detta ändamål

```
<? sats; ?>
<?= uttryck ?>
<?php kod ?>
<script language="php">
```
- Variabelnamn är case-sensitive men inte funktionsnamn
- Språket liknar unix c-shellsript.
- Svagt typat

Några exempel

- Dagens datum:

```
<h1>Testprogram i php</h2>
Dagens datum: <? echo date("Y-m-d"); ?>
```
- Skriva ut ett slumptal:

```
<?
  srand(time());
  $random = (rand()%9);
  print("Slumptal mellan 0 och 9 är: $random");
?>
```

Syntax

- Utmatning

```
echo "hej", "hopp";           // echo kan ta multipla argument
print("hej");                // parentes ej nödvändig
printf("PI = %4.2f",22/7);    // formaterad sträng
```
- Variabler

```
$n = 0;
$namn = "Ada";
$orden = array("hej", "hopp");
```
- Slingor

```
while($n < 10){...}
do{...}while($n<10);
for($i = 0; $i < 10; $i++){...}
foreach ($orden as $element){
  print "$element";
}
```
- Villkor

```
if(strcmp($namn, "Ada")==0) echo "Matchar";
else echo "Matchar inte";
```

Syntax, forts.

- **Funktioner**
 - funktion `lika($var1, $var2){`
 - `if($var1===$var2) return True`
 - `else return False;`
 - `}`
- `$a = array ('test' => 1, 'hello' => NULL);`
Scope
 - Variabler i funktioner är lokala
 - Kan deklareras *global*
 - Även i arrayen `$GLOBALS`

Klasser / objekt

```
<?php
class Cart {
var $items=array();
function add_item ($artnr, $num) {
    $this->items[$artnr] += $num;
}
function remove_item ($artnr, $num) {
    if ($this->items[$artnr] > $num) {
        $this->items[$artnr] -= $num;
        return true;
    }
    else {
        return false;
    }
}
}
?>
```

Fördefinierade variabler

- Via fördefinierade variabler når man information om t ex klienten eller servern, samtliga har formatet `$_KATEGORI[VARIABELNAMN]`.
- Kategorier:
 - Server variables: `$_SERVER`
 - Environment variables: `$_ENV` (information om OS-miljön)
 - Request variabler: `$_REQUEST`, består av följande undergruppering
 - HTTP Cookies: `$_COOKIE`
 - HTTP GET variables: `$_GET`
 - HTTP POST variables: `$_POST`
 - HTTP File upload variables: `$_FILES`
 - Session variables: `$_SESSION`
 - Global variables: `$GLOBALS` (inget `“_”`)

Sessionsid

- Varje klient som skickar en HTTP-request tilldelas en sessionskaka genom headern `“Set-Cookie”` i serverns HTTP-response. Denna skickas sedan med varje gång klienten återkommer till sidan. Med denna metod `“trackar”` man en klient på en webserver.

```
<?
session_start();
if(empty($_SESSION['time']))$_SESSION['time']=time();
echo "Tidpunkt: " . $_SESSION['time'] . "<br>";
echo "Sessionskaka: " . session_id();
?>
```

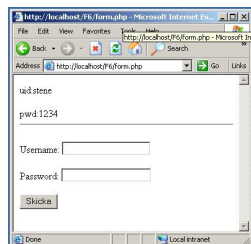
Formulär-exempel

```
<html>
<body>

<?php
if(!empty($_POST['uid']))
echo "<p>uid:<“ . $_POST['uid']";
if(!empty($_POST['pwd']))
echo "<p>pwd:<“ . $_POST['pwd']";

?>
<hr>
<form method=post>
<p><input type=text name=uid>
<p><input type=password name=pwd>
<p><input type=submit value="Skicka">
</form>

</body>
</html>
```



Mailexempel (klient + serverscript)

```
<?
mail($_REQUEST['email'], "Rubrik".$_REQUEST['meddelande']);
?>
<html><head><title>Skicka mail</title>

<script language="javascript">
function kontroll(form){
    if(form.email.value.indexOf("@")!=-1){
        alert("Du måste ange en epostadress!");
        return false;
    }
    return true;
}
</script></head><body>

<form onSubmit="return kontroll(this)">
<p><input type="text" name="email">
<p><input type="text" name="meddelande">
<p><input type="submit" value="Skicka">
</form>

</body></html>
```

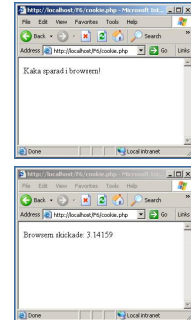
Cookies

- En kaka kan sättas genom
 - setcookie ("TestCookie", "1234", time()+3600, "/F4/", "localhost", 0);
 - 1. Namn (obligatorisk)
 - 2. Värde (obligatorisk)
 - 3. Expires (valfri)
 - 4. Path relativt den egna servern t ex <http://localhost/F6/cookie.php> (valfri)
 - 5. Host (valfri)
 - 6. Krypterad sättning av kakan (valfri)
- Kakor kan läsas genom
 - echo \$_COOKIE["TestCookie"];

Cookie-exempel

```
<?php
if(empty($_COOKIE["pi"])){
    setcookie ("pi","3.14159",time() +
    3600);
    echo "Kaka sparad i browsern!";
}
else
    echo "Browsern skickade: " .
    $_COOKIE["pi"];
?>

<html>
<body>
</body>
</html>
```



PostgreSQL

```
<html> <head> <title>Test</title> </head>
<body bgcolor="white">
<?
    $link = pg_connect("dbname=uid user=uid password=slask");
    $result = pg_exec($link, "select * from bostader");
    $numrows = pg_numrows($result);
    echo "<p>link = $link<br> result = $result<br> numrows = $numrows</p> ";
?>
<table border="1"> <tr> <th>Adress</th> </tr>
<? // Loop on rows in the result set.
for($i = 0; $i < $numrows; $i++) {
    echo "<tr>\n";
    $row = pg_fetch_array($result, $i);
    echo "<td>", $row["address"], "</td>";
}
    pg_close($link);
?>
</table> </body> </html>
```

AJAX

- AJAX ger en möjlighet att göra ett http-anrop för att uppdatera endast en del av en sida, detta sker med hjälp av ett XMLHttpRequest objekt.
- open() metod
- send() metod
- Möjliga värden för serverns svar på readystate
 - 0The request is not initialized
 - 1The request has been set up
 - 2The request has been sent
 - 3The request is in process
 - 4The request is complete

ajax.html

```
<html>
<head>
<script type="text/javascript" src="ajax.js"></script>
</head>
<body>
<form>
Välj en tabell:
<select name="table" onchange="showUser(this.value)">
<option value="bostader">bostader</option>
<option value="slask">slask</option>
</select>
</form>
<br>
<div id="answer"><b>Resultatet matas ut här.</b></div>
</body>
</html>
```

ajax.js

```
var xmlhttp;
function showUser(str)
{
    xmlhttp=new XMLHttpRequest();
    if (xmlhttp==null)
    {
        alert("Browser does not support HTTP Request");
        return;
    }
    var url="ajax.php";
    url=url+"&table="+str;
    xmlhttp.onreadystatechangeChanged;
    xmlhttp.open("GET",url,true); //asynkron
    xmlhttp.send(null); //ingen body i request
}
function stateChanged()
{
    if (xmlhttp.readyState==4) // klar
    {
        document.getElementById("answer").innerHTML=xmlhttp.responseText;
    }
}
```

ajax.php

```
<?php
$table=$_GET['table'];

$con = pg_connect("host=pgsql0 dbname=stene user=stene password=");
if (!$con)
{
    die('Could not connect: ' . pg_last_error());
}

$sql="SELECT * FROM " . $table;
$result = pg_exec($sql);

echo "<table border='1'>";

while($row = pg_fetch_array($result))
{
    echo "<tr>";
    for($i = 0; $i < pg_num_fields(); $i++){
        echo "<td>" . $row[$i] . "</td>";
    }
    echo "</tr>";
}
echo "</table>";

pg_close($con);
?>
```