

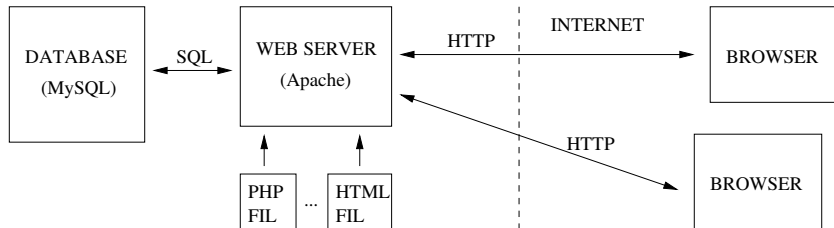
Scripting, Databaser og Systemarkitektur, E2007

Forelæsning

- ▶ Hvad har vi lært indtil nu?
- ▶ Cookies
- ▶ Sessions
- ▶ Spørgetime, hvad skal vi lave?
- ▶ Introduktion til øvelsen
- ▶ Har vi god tid laver vi en ad hoc applikation.

Hvad har vi lært indtil nu?

Oversigt:



En PHP-fil:

```
<body>
<?php
    $hello = "<b>Hello</b>";
    $world = "<i>world!</i>";
    echo $hello.$world;
?>
</body>
```

En anden PHP-fil:

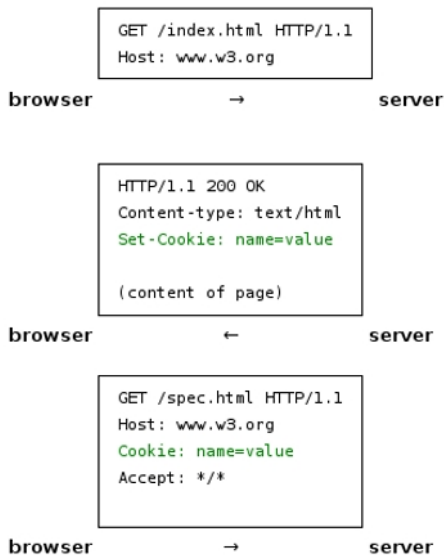
```
<body>
<?php
    echo "<b>Hello</b>";
    echo "<i>world!</i>";
?>
</body>
```

Hvad har vi lært indtil nu?

- ▶ Variabler, tal, strenge og arrays
- ▶ Beregninger
- ▶ if-sætninger og løkker
- ▶ Funktioner og kodegenbrug
- ▶ Indhentning af data fra brugere med forms
- ▶ Check af brugerindtastninger med regulære udtryk
- ▶ SQL (create, update, insert, delete, join, foreign key references....)
- ▶ Databasetilgang via PHP
- ▶ SQL forespørgsler via PHP
- ▶ Afsending af emails
- ▶ Datoer og beregning af tid

Hvad er Cookies

- ▶ En cookie er en tekst-streng som en web-server sender til en browser og som browseren returnerer uændret når browseren igen besøger det samme site. Se evt. [wp.netscape.com/...](http://wp.netscape.com/)



Billedet er taget fra Wikipedia

Hvad er Cookies

- ▶ Cookies er nyttige til håndtering af simpel form for tilstand — session tracking:
 - ▶ Brugerspecialisering (Customization)
 - ▶ Fokusering af banner ads — hvad var en bruger interesseret i sidst? (tracking)
 - ▶ Adgangskontrol — login mekanisme
- ▶ I de fleste browsere kan man se hvilke cookies der er aktive i filen `cookies.txt`:
 - ▶ **XP:** C:/Documents and Settings/"USERNAME"/Application Data/Mozilla/Firefox/
 - ▶ **OS X:** /Users/"USERNAME"/Library/Application Support/Firefox/Profiles/
 - ▶ **Linux:** /home/\$USER/.mozilla/firefox/

Cookies, sådan ser de ud

```
# Netscape HTTP Cookie File
# http://wp.netscape.com/newsref/std/cookie_spec.html
# This is a generated file! Do not edit.
linuxlab.dk FALSE / FALSE 1262307600 ad_browser_id 13
linuxlab.dk FALSE / FALSE 1262307600 last_visit 988628269
linuxlab.dk FALSE / FALSE 1262307600 second_to_last_visit 9886...
linuxlab.dk FALSE / FALSE 1262307600 ad_user_login 60,4C6F
```

Filformatet er som følger:

Domain : linuxlab.dk

? : FALSE

Path : /

Secure : FALSE

Expires : 1262307600

Name : add_browser_id

Value : 13

- ▶ Almindelige browsere understøtter kun omkring 20 cookies pr. site
- ▶ Almindelige browsere understøtter kun omkring 300 cookies totalt
- ▶ En cookie kan højest være 4 kilobytes stor
- ▶ Brugeren kan slå cookies fra
- ▶ Sikkerhedsproblemer med Cookies
 - ▶ “trekants-attack”
 - ▶ Cookies kan true privatlivet
 - ▶ Cookies kan stjæles

Se evt. http://en.wikipedia.org/wiki/HTTP_cookie

Sikkerhedsproblem med Cookies — et “trekants-attack”

Hvis et site benytter cookies til identification af en person kan personen blive udsat for et “trekants-attack”:

- ▶ En skurk sætter en webside op der omdirigerer (redirects) personer til Amazon’s bestillingsside for en bestemt bog (udenom alle forms)
- ▶ Herefter sender skurken en email til en “uskyldig person” som skurken ved køber ind på Amazon; skurken beder den “uskyldige person” om at trykke på et link
- ▶ Når den uskyldige person klikker på linket bliver der bestilt en bog uden at personen kan afværge købet — Amazon får tilsendt brugerens cookie og accepterer bestillingen!

Løsning?

- ▶ Tilføj “Godkend-side” til web-sitet

Der er en grund til at nogle personer vælger at slå cookies fra i deres browser:

- ▶ Søgemaskiner viser ads for hvad man søgte efter sidste gang
 - ▶ Problematisk hvis man bliver kigget over skulderen af arbejdsgiveren
 - ▶ eller af konen?
- ▶ Tilstandsproblemer:
 - ▶ Forskelle mellem tilstand implementeret med cookies og tilstand implementeret med formvariabler — “Back”-knappen
 - ▶ Problemer med cookies når flere browsere benyttes samtidig

- ▶ Cookies gemmes som tekst på brugerens maskine, så fysisk adgang eller en ubeskyttet computer kan lade skurken kopiere cookien.
- ▶ Sendes cookien ukrypteret, eksempelvis via åbne trådløse net, er det en forholdsvis simpel sag at opsnappe cookien når den sendes.

Løsning?

- ▶ Benyt en sikker forbindelse
- ▶ Lad **aldrig** cookien indeholde fortrolige informationer

Registrering af cookies

- ▶ For at registrere at en cookie skal sættes i browseren kan PHP-funktion `setcookie()` benyttes:
 - ▶ `setcookie(name, value, expire, path, domain, secure)`
- ▶ Kortere former er tilladte:
 - ▶ `setcookie(name, value)`
 - ▶ Eksempel: `setcookie("a", "5")`
- ▶ Cookies skal sættes/slettes/rettes før der sendes noget til browseren (med f.eks. `echo`)

Eksempel 1.

```
setcookie('language', 'Dansk', time()+3600, '/', '.itu.dk', 0);
```

Vores cookie "language" med værdien "Dansk",
udløber "time()+3600", dvs om 1 time
path er sat til "/" så den gælder for alle niveauer på sitet
domain er sat til .itu.dk

Dvs. webmail.itu.dk, sysadm.itu.dk, people.itu.dk, etc.
Secure sættes til false (true ville være '1')

Eksempel 2.

```
setcookie('language', 'Engelsk', time()+36000, '/mail/');
```

Vores cookie "language" med værdien "Engelsk", udløber "time()+36000", dvs om 10 timer path er sat til "/mail/" så den gælder fra mail mappen og dybere Eks. a.dk/mail, a.dk/mail/junk

```
vi lader domain og secure være tomme,  
  default domain: nuværende domæne (eks. a.dk)  
  default secure: false
```

Læsning af cookies i PHP

- ▶ For at læse en cookie sendt med en forespørgsel kan et PHP-script kigge i et cookie-array:
 - ▶ `$mycookie = $_COOKIE["mycookie"];`
 - ▶ `echo $_COOKIE["language"];`
- ▶ Vi kan tjekke om en cookie findes, ligesom vi tjekker om en variabel findes.

```
if(@$_COOKIE['mycookie']!= ''){  
    // Hvad er det nu @ gør?  
    // Hvorfor bruger vi den her?  
}
```

```
if( isset($_COOKIE['mycookie']) ) {  
    // Ny funktion, isset().  
    // Returner en boolsk værdi, angiver om en variabel findes  
}
```

Sletning af Cookies

- ▶ Cookies slettes ved at sætte en cookie med udløb i fortiden!

```
//vi skal benytte næsten samme parametre som da vi satte den  
//en cookie sat med:
```

```
setcookie('mycookie', 'sweet', time()+100, '/', '.a.dk', 1);
```

```
//skal fjernes med
```

```
setcookie('mycookie', '', time()-100, '/', '.a.dk', 1);
```

```
<?php
    $count = @$_COOKIE["count"];
    if ( $count == "" ) {
        $count = 0;
    }
    // vi sætter ikke "expire" på cookien, den forsvinder
    // når browseren lukkes
    setcookie("count", $count + 1);

    echo "<h2>CookieCount: $count</h2>
        <a href=\"cookiecount.php\">Up</a>";
?>
```

Bemærk:

- ▶ Der sendes ikke formvariabler med til scriptet i Up-linket!
- ▶ Hvordan virker scriptet når flere browsere er åbne på samme client?

Eksempel: Ulovlige handlinger

Eksempel 1:

```
<html>.....  
<?php  
    //vi sætter en cookie der udløber om en time  
    setcookie("language", "da", time()+3600);  
?>
```

Eksempel 2:

```
<?php  
    echo "Hej";  
    //vi fjerner cookien "language"  
    setcookie("language", "", time()-3600);  
?>
```

Hvorfor er de to eksempler "ulovlige"?

- ▶ Der kan ikke sættes en cookie efter der er udskrevet tekst til browseren

- ▶ Med sessions gemmes tilstand på webserveren og kun et *sessionid* gemmes som en cookie på clienten.
- ▶ PHP har god support for sessions:
 - ▶ Funktionen `session_start()` ser om en *sessionid* allerede er tilstede som en cookie på browseren. Ellers genereres et nyt *sessionid* som gemmes på browseren som en cookie.
 - ▶ Når en session er startet (med `start_session()`) kan data hentes *OG* gemmes fra et session-array:
 - ▶ `echo $_SESSION["myvar"]`
 - ▶ `$_SESSION["myvar"]='myval'`

- ▶ Funktionen `session_destroy()` kan benyttes til at ryde op efter en session, dvs, lager frigives og `sessionid` cookie slettes på klienten.
- ▶ For at "ødelægge" brugerens session med `session_destroy()` skal `session_start()` være kaldt!!!
- ▶ Funktionen `session_start()` skal ligesom `setcookie()` kaldes *før* der sendes noget til browseren

Potentielle problemer med sessions

- ▶ Cookie-problematik
- ▶ Det er ikke klart hvornår web-serveren kan antage at sessionlageret kan genbruges eller smides væk

Løsningen: sessionid kan sendes som URL parameter, se IPMA s. 317-318

Implementation af en simpel indkøbskurv ved brug af sessions.

```
<?php session_start();
if ( @$_REQUEST['submit'] == "Empty Basket" ) { // destroy basket
    session_destroy();                          // destroy session
    header("Location: basket.php"); exit;        // reload and exit
}
$kurv = @$_SESSION["kurv"];
if ( @$_REQUEST['new'] != "" ){
    $kurv[] = $_REQUEST['new']; // maybe add new stuff
}
$_SESSION["kurv"] = $kurv;
//forsættelse følger .....
```

Eksempel: Session Basket — basket.php, del 2

```
//del 2
echo "<html><title>Session Basket</title>
    <body><h2>Session Basket</h2>
    <form action=\"basket.php\">
    <ul>";
// loop igennem arrayet
for ( $i = 0 ; $i < count($kurv) ; $i++ ) {
    echo "<li>$kurv[$i]</li>";
}
echo "<li> <input type='text' name='new'>
    <input type='submit' value='Add'>
    <input type='submit' name='submit' value=\"Empty Basket\">
    </ul></form></body></html>";
?>
```

basket.php source

Eksamen ligger den 7. januar 2008

Hvornår skal vi mødes?

- ▶ Torsdag d. 13. december 2007
- ▶ Torsdag d. 3. januar 2008
- ▶ Fredag d. 4. januar 2008
- ▶ Lørdag d. 5. januar 2008

Hvad skal vi snakke om?

- ▶ Arrays, løkker, variable, html, SQL, Reg Expr ?????
- ▶ Hvem laver en afstemningsapplikation ?????
- ▶ Vi kan gennemgå et længere eksempel ?????

Opgavesæt 11 er en “åben øvelse”

Hvis du har en ide til en lille web-service har du nu mulighed for at bygge den!

Næste gang gennemgår vi et tidligere eksamenssæt, lav det evt. inden, så ved du hvad det drejer sig om og hvad du har svært ved

Så er det også lettere at stille spørgsmål