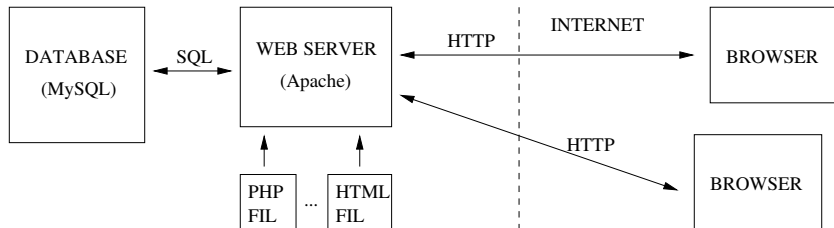


- ▶ Hvad har vi lært indtil nu?
- ▶ Datoer i PHP
 - ▶ Forskellige datoformater
 - ▶ Check af dato
- ▶ Repetition: fire steps til udvikling af web-sites
- ▶ Eksempel: Huskeliste

Hvad har vi lært indtil nu?

Oversigt:



En PHP-fil:

```
<?php
function makeBold($str){
    return '<b>'.$str.'</b>';
}

echo makeBold('hello');;
echo "<i>WORLD</i>";
?>
```

For at regne med datoer er det nødvendigt at kende til nogle forskellige formater:

- ▶ UNIX Epoch: antal sekunder siden januar 1970 — på en 32-bit maskine løber vi tør for sekunder i år 2038!
- ▶ Epoch bliver ofte kaldet et `timestamp`
- ▶ Forskellige streng-repræsentationer:
 - ▶ 2001-05-21 ← *ISO international standard*
 - ▶ 21/5-01
 - ▶ 21-05-2001
 - ▶ ...

- ▶ PHP funktionen `time` returnerer antal sekunder siden januar 1970 (Epoch)
- ▶ PHP funktionen `mktime` kan benyttes til at omregne en given dato og tidspunkt til Epoch-tid
- ▶ PHP funktionen `date` kan benyttes til at omregne en Epoch-tid til en lang række forskellige datoformater

Funktionerne `time()` og `mktime()`

- ▶ `time()` returnerer antallet af sekunder siden 1/1-1970

```
<?php
    $nextweek = time() + 60 * 60 * 24 * 7;
    // $nextweek er nu noget a la 1208872438
?>
```

- ▶ `mktime()` returnerer et "Unix timestamp" for en given dato
- ▶ `mktime($hour, $minute, $second, $month, $day, $year)`
parametrene kan undlades fra højre mod venstre.
- ▶ Undlader vi parametre benytter den serverens tid. (dvs NU)

```
<?php
    $lecturefinished = mktime(19, 0, 0, 4, 17, 2008);
    // $lecturefinished er nu 1208451600
    $seconds2go = $lecturefinished - time();
    // $seconds2go er nu ca 5400
?>
```

?? hvorfor regner vi med sekunder?

Funktionen `date()`

- ▶ Funktionen `date()` kan tage to parametre:
 - ▶ Måden (`$format`) datoen skal præsenteres på
 - ▶ og eventuelt et unix timestamp

Udelader vi et timestamp bruger den serverens tid (`nu`)

- ▶ `$format` er en streng der konverteres til forskellige elementer i en dato, eksempelvis:
 - ▶ "d" dagen, i formatet DD
 - ▶ "M" måneden, i formatet Jan, Feb etc
 - ▶ "Y" året, i formatet YYYY
- ▶ På <http://dk.php.net/date> kan alle mulighederne ses.
- ▶ På side 206 i lærebogen er en mindre oversigt.

Eksempler på datoformater med PHP funktionen date

`$time` er vores timestamp for dagen idag

```
$time=time();
```

1:	<code>date("Y-m-d", \$time)</code>	→	2008-04-17
2:	<code>date("j/n-y", \$time)</code>	→	17/4-08
3:	<code>date("l m-d-Y", \$time)</code>	→	Thursday 04-17-2008
4:	<code>date("F jS Y", \$time)</code>	→	April 17th 2008
6:	<code>date("Y-m-d", \$time + 60*60*24)</code>	→	----- ?
7:	<code>date("j/n-y", \$time + 2*60*60*24)</code>	→	----- ?

Hvis andet argument til funktionen `date` udelades returnerer funktionen dagens dato:

```
8: date("F jS Y") → April 17th 2008
```

Eksempel: Dato-beregner — datecheck.php

```
<form action='datecheck.php' method='post'>
  Enter a date in the form YYYY-MM-DD:<br>
  <input size='4' name='y'>-
  <input size='2' name='m'>-
  <input size='2' name='d'>
  <p><input type='submit' value='Check'></p> </form>
<?php
if($_REQUEST['y']!=''){
  $y = $_REQUEST['y']; $m = $_REQUEST['m']; $d = $_REQUEST['d'];
  if ( ! checkdate($m,$d,$y) ) {
    echo "The date is invalid!";
    exit;
  }
  $epoch = mktime(0,0,0,$m,$d,$y);
  echo "The Epoch time for the date $y-$m-$d is $epoch<P>";
  $ugedag = date("l", $epoch);
  echo "The day is a $ugedag";
} ?>
```

Vores fremgangsmåde:

Trin 1: Konstruktion af datamodel

- ▶ Hvilken information skal gemmes og hvordan skal den repræsenteres?

Trin 2: Udvikling af data-transaktioner

- ▶ Hvordan indsættes data i databasen?
- ▶ Hvordan udtrækkes data fra databasen?
- ▶ Fungerer vores model?

Trin 3: Konstruktion af web-forms og site-map

- ▶ Brugergrænsefladen er HTML-kode (forms)

Trin 4: Konstruktion af PHP-filer

- ▶ SQL (Structured Query Language) bruges til de egentlige data-transaktioner

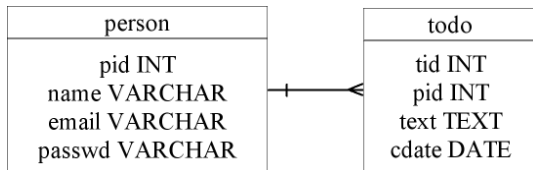
Bemærk: Jo mere tid der bruges på Trin 3, jo lettere bliver Trin 4!

Eksempel: Huskelisteservice

Ide: Konstruer en huskelisteservice hvor brugere kan oprette nye emner og slette afsluttede emner.

- ▶ Emner der overskrider en indtastet afslutningsdato skal vises med rødt, dagens emner viser vi med grønt
- ▶ Servicen skal virke for flere personer samtidigt og uafhængigt

Trin 1: Datamodel — E-R diagram



Bemærk: Gaflen angiver at der til hver person kan være knyttet flere emner (en-til-mange relation)

→ Eksempel

Omformning af E-R diagram til tabeller:

```
CREATE TABLE person (  
  pid INT(11) NOT NULL AUTO_INCREMENT,  -- person id  
  name VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  passwd VARCHAR(10) NOT NULL,  
  PRIMARY KEY (pid)  
) TYPE=InnoDB;  
  
CREATE TABLE todo (  
  tid INT(11) NOT NULL AUTO_INCREMENT,  -- todo id  
  pid INT(11) NOT NULL,                 -- person id  
  text VARCHAR(100) NOT NULL,  
  cdate DATE NOT NULL,                  -- completion date  
  PRIMARY KEY (tid),  
  INDEX(pid),  
  FOREIGN KEY (pid) REFERENCES person(pid)  
) TYPE=InnoDB;
```

Trin 2: Datatransaktioner

1. Indsættelse af person i person tabel:

```
insert into person (pid, name, email, passwd)
values (1, 'Jonas Holbech', 'holbech@itu.dk', 'hej');
```

2. Indsættelse af todo emne:

```
insert into todo (pid, text, cdate)
values (1, 'Husk forelæsning', '2007-11-15');
```

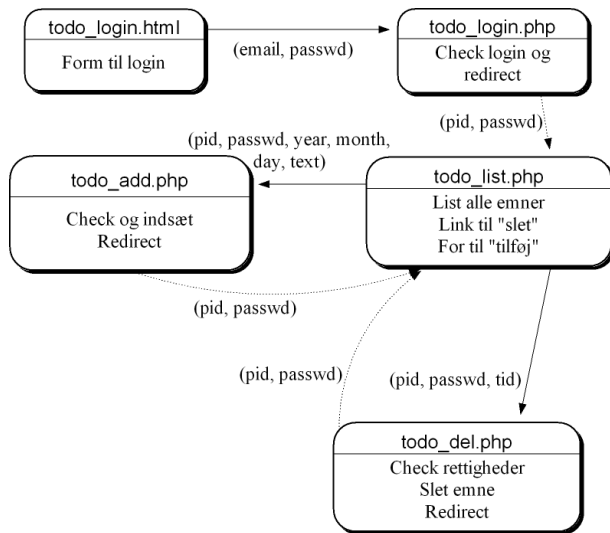
3. Sletning af todo emne:

```
delete from todo where tid = 1;
```

4. Listning af alle emner knyttet til en person:

```
select tid, text, cdate from todo
where pid = 1
order by cdate ASC;
```

Trin 3: Konstruktion af site-map



Plan for filerne:

- ▶ `todo_login.php`: Find person id på baggrund af email og password — redirect til `todo_list.php`:
- ▶ `todo_list.php`: Vis emner for den pågældende person
- ▶ `todo_del.php`: Slet et emne for en person
- ▶ `todo_add.php`: Tilføj et emne for en person
- ▶ `todo_lib.php`: Indeholder hjælpefunktioner
- ▶ `check_vars.php`: Indeholder hjælpefunktioner

Spørgsmål:

- ▶ Hvilke formvariable forventer filen `todo_login.php`?
- ▶ Hvilke formvariable forventer filen `todo_list.php`?
- ▶ Hvilke formvariable forventer filen `todo_del.php`?
- ▶ Hvilke formvariable forventer filen `todo_add.php`?

Trin 4: — Utility-filen `todo_lib.php`

```
<?php include("mydb2.php");           // Database
include("check_vars.php");           // Til at tjekke variable

function validate_passwd($pid,$passwd) {
    chk_heltal($pid);
    chk_passwd($passwd);
    $rows = mysql_query("select email, name from person
        where pid = $pid and passwd = '$passwd'");
    if ( $row = mysql_fetch_row($rows) ) {
        return "<a href=\"mailto:$row[0]\">$row[1]</a>";
    }
    error("Wrong password!");
} ?>
```

- ▶ Ved at inkludere filen `todo_lib.php` i andre filer bliver funktioner i `mydb2.php` og `check_vars.php` også tilgængelige
- ▶ Funktionen `validate_passwd` returnerer personens navn til inkludering i HTML kode, hvis personen er autentificeret

Processing af Login-form — filen todo_login.php:

```
<?php include("todo_lib.php");
    $passwd = $_REQUEST['passwd'];   $email = $_REQUEST['email'];
    chk_passwd($passwd);             chk_email($email);

    mydb_connect();
    $rows = mysql_query("select pid from person
                        where email = '$email'
                        and passwd = '$passwd'");
    if ( $row = mysql_fetch_row($rows) ) {
        header("Location: todo_list.php?pid=$row[0]&passwd=$passwd");
        exit;
    }
    echo "Forkert adgangskode eller ugyldig email"; ?>
```

Bemærk:

- ▶ Hvis password er i orden “hoppes” videre til filen todo_list.php
- ▶ Password og person-identitetsnummer sendes med som formvariabler. **Hvorfor?**

Scriptet fylder cirka 60 linier så det gennemgås i browseren:

`todo_list.php`

Bemærk:

- ▶ Vi autentificerer personen med funktionen `validate_passwd`
- ▶ Password og person-identitetsnummer sendes med videre til både `todo_del.php` og `to_add.php`
- ▶ Vi benytter dags-dato som forslag til en dato i datoindtastningsfeltet

Sletning af todo-emner — filen `todo_del.php`:

```
<?php
    include("todo_lib.php");
    $passwd = $_REQUEST['passwd'];
    $tid = $_REQUEST['tid'];
    $pid = $_REQUEST['pid'];
    chk_heltal($tid);

    mydb_connect();
    validate_passwd($pid,$passwd);
    mysql_query("delete from todo where tid=$tid");
    header("Location: todo_list.php?pid=$pid&passwd=$passwd");
?>
```

Bemærk:

- ▶ Når emnet er slettet “hoppes” videre til filen `todo_list.php`
- ▶ Formvariableerne `pid` og `passwd` sendes med!

Tilføjelse af todo-emner — filen todo_add.php:

```
<?php include("todo_lib.php");
    $year = $_REQUEST['year'];      $month = $_REQUEST['month'];
    $day = $_REQUEST['day'];        $text = $_REQUEST['text'];
    $pid = $_REQUEST['pid'];        $passwd = $_REQUEST['passwd'];

    if ( ! checkdate($month, $day, $year) ) {
        error("Illformed date");
    }
    chk_text_single_line($text);
    mydb_connect();
    validate_passwd($pid,$passwd);

    $epoch = mktime(0,0,0,$month,$day,$year);
    $cdate = date("Y-m-d", $epoch);
    mysql_query("insert into todo (pid,text,cdate)
                values ($pid, '$text', '$cdate')");
    header("Location: todo_list.php?pid=$pid&passwd=$passwd");
?>
```

Bemærk:

- ▶ Funktionen `checkdate` benyttes til at undersøge om en dato eksisterer
- ▶ Funktionen `mktime` benyttes til at beregne antal sekunder siden 1970
- ▶ Resultatet af kaldet til `mktime` benyttes til at formatere datoen med funktionen `date`

- ▶ Administrationssider?
- ▶ Bruger-features?

- ▶ Konstruktion af et web-baseret pladekartotek