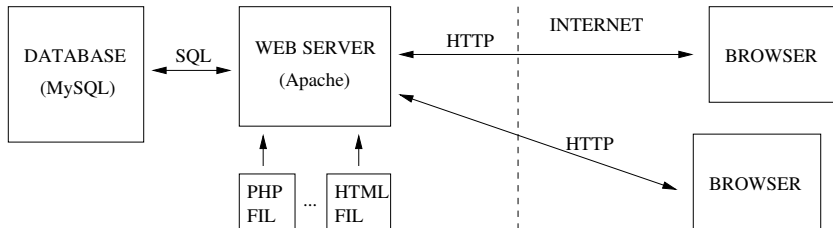


- ▶ Hvad har vi lært indtil nu?
- ▶ Overførsel af formvariable i links
- ▶ Radio-buttons og select-lister
- ▶ Tilstand i webprogrammer med skjulte formvariable
- ▶ Eksempel: Optælling og nedtælling
- ▶ Arrays
- ▶ Eksempel: Terningkast
- ▶ Eksempel: Frekvensdiagram
- ▶ Check-boxes med arrays

# Hvad har vi lært indtil nu?

## Oversigt:



## En PHP-fil:

```
<html>
  <head><title>Hello World</title>
</head>
<body>
  <?php echo "<b>Hello</b> ";
    echo "<i>WORLD</i>"; ?>
</body>
</html>
```

Indtil nu:

- ▶ Variabler, tal og strenge
- ▶ Beregninger
- ▶ if-sætninger og løkker
- ▶ Funktioner
- ▶ Formvariable

# Overførsel af formvariabler i links

Vi har set hvorledes formvariabler kan overføres ved hjælp af HTML forms — `veksle.html` Det er også muligt at overføre formvariabler i HTML links — `veksle_quick.html`

```
<h2>Quick Vekslebank</h2>
<a href="veksle_quick.php?kroner=500">Quick kr. 500</a> |
<a href="veksle_quick.php?kroner=1000">Quick kr. 1000</a>
<form action="veksle_quick.php">Indtast andet beløb i kroner:
  <p><input name="kroner"></p>
  <p><input type="submit" value="Beregn Dollarbeløb"></p>
</form>
```

## Filen veksle\_quick.php:

```
<html><head><title>Quick Vekslebank</title></head>
<body>
<h2>Quick Vekslebank</h2>
<?php
    $kroner=$_REQUEST['kroner'];
    $dollars = ($kroner - 20.0) / 5.43;
    $dollars = number_format($dollars, 2, ",", ".");
    echo "For kr. $kroner modtager du \$$dollars";
?>
<p><a href="veksle_quick.html">Ny beregning</a></p>
</body>
</html>
```

## Overførelse af formvariable i links — fortsat

Flere formvariable overføres i et link ved at adskille formvariablene med tegnet &.

Det er muligt at indtaste formvariable direkte i en browsers *location bar*.

*Eksempel* — `plus.php?a=5&b=47`

```
<html><head><title>Plus service</title></head>
  <body> <h2>Plus service</h2>
  <?php
    $a = $_REQUEST['a'];
    $b = $_REQUEST['b'];
    $sum = $a + $b;
    echo "Summen af $a og $b er $sum";
  ?>
</body>
</html>
```

For at overføre strenge indeholdende mellemrum og tegnene = og & skal disse tegn (og andre) escapes!

*Eksempel* — `velkommen.php?navn=Hans%20%26%20Grethe`

```
<html><head><title>Velkommen</title></head>
  <body> <h2>Velkommen</h2>
  <?php echo "Velkommen ".$_REQUEST['navn']." ..."; ?>
  </body>
</html>
```

I ovenstående link betyder %20 mellemrum og %26 betyder &  
Tip: PHP-funktionerne `urlencode()` og `urldecode` kan hjælpe

Hvorfor skal vi escape?

*Eksempel* — `velkommen.php?navn=Hans & Grethe`

## *Eksempel* — urlencode.php

```
<?php
$strangevar="Hans & Grethe";
$encoded = urlencode($strangevar);
echo "<p>
    <a href='velkommen.php?navn=$encoded'>Om bogen: $strangevar</a>
</p>";
echo "<p>Den encodede version af \$strangevar: $encoded</p>";

echo "<p><code>decode</code>:</p>";

echo ' <p>' .urldecode($encoded) . "</p>";
?>
```

Hvordan benytter vi radio-buttons og select-lister i forms?

```
<form action='format.php' method='post'>
  <b>Farve:</b>
  <select name=farve>
    <option value="red">Rød</option>
    <option value="green">Grøn</option>
    <option value="blue">Blå</option>
  </select>

  <b>type face:</b><br>
  Italic <input type="radio" name="face" value="i"><br>
  Bold <input type="radio" name="face" value="b"><br>
  Normal <input type="radio" name="face" value="normal" checked>

  <textarea name="tekst" cols="50" rows="4">Teksten</textarea>

  <input type="submit" value="Formatter tekst">
</form>
```

Filen format.php:

```
<h2>Tekstformattering: resultat</h2>
<?php
    $face = $_REQUEST['face'];
    $tekst = $_REQUEST['tekst'];
    $farve = $_REQUEST['farve'];
    if ( $face != "normal" ) {
        $tekst = "<$face>$tekst</$face>";
    }
    echo "<p style='color:$farve'>$tekst</p>";
?><p>
    <a href=format.html>Ny formattering</a>
</p>
```

Bemærk:

- ▶ Det er ikke muligt med denne service at formattere teksten som både **fed** og **kursiv**

# Tilstand i webprogrammer med skjulte formvariable

Skjulte (hidden) formfelter kan bruges til at implementere tilstand (state) i et webprogram, — `taeller.php`:

```
<?php
    if ( $t == "" ) {
        $t = 0;
    } else {
        $t = $t + 1;
    }
    echo "<h2>Taeller: $t</h2>
    <form action=\"taeller.php\" method=\"post\">
        <input type=\"hidden\" name=\"t\" value=\"$t\">
        <input type=\"submit\" value=\"OP\">
    </form>";
?>
```

Bemærk: Hvis formvariablen `$t` ikke er sat sættes variabelen til 0  
I øvelsen skal vi benytte en skjult formvariabel til at implementere spillet Gæt-et-tal, hvor et tilfældigt tal genereres og brugeren har et eller flere forsøg til at gætte tallet

Indtil nu har vi kun beskæftiget os med værdier som strenge, heltal og kommatal

Et *array* er en fjerde form for værdi, som kan indeholde en sekvens af andre værdier

Der findes to typer arrays, numeriske og associative

Numeriske arrays kan vi løbe igennem med `for`, `while` og `foreach`

Associative arrays kan vi løbe igennem med `foreach`

*Eksempel:* — taarne.php:

```
<?php $taarne = array();
    $taarne[0] = "Sears Tower"; // Konstruer array
    $taarne[1] = "Rundetårn";
    $taarne[2] = "Eifeltårnet";
    sort($taarne);                // Sorter alfabetisk      ?>
<h2>Tårne</h2>
<ul>
    <?php
    for ( $i=0 ; $i<sizeof($taarne) ; $i=$i+1 ) { // Udskriv indhold
        echo "<li>$taarne[$i] </li>";
    }
    ?>
</ul>
```

## Bemærk:

- ▶ For at tilgå element med *indeks* 2 i arrayet `$taarne` skriver vi `$taarne[2]`
- ▶ Vi *indekserer* et array fra 0 til  $N - 1$ , hvor  $N$  er antal elementer i arrayet
- ▶ Funktionen `sort` sorterer arrayets elementer alfabetisk
- ▶ Funktionen `sizeof` returnerer antal elementer i et array

# Indeksring med strenge (associativt)

Eksempel — foto.html:

```
<form action="foto.php" method="post">
  <p>Indtast navn: <input type="text" name="navn"><br />
    <input type="submit" value="Søg"></p>
</form>
```

Filen foto.php:

```
<?php
$navn=$_REQUEST['navn'];
$foto["Martin"] = "http://.../Elsman-Martin_jpg.jpg";
$foto["Peter"]  = "http://.../324.jpg";
$foto["Niels"]  = "http://.../Hallenberg-Niels_jpg.jpg";
if ( $foto[$navn] == "" ) {
  echo "Intet foto af $navn i systemet!";
} else {
  echo "<h3>Foto af $navn</h3><img src='$foto[$navn] ' >";
}
?> <p><a href='foto.html'>Ny søgning</a>
```

```
function menu($currentpage){
    $pages = array(
        "page1.php" => "Hjem",
        "page2.php" => "Om Foreach Menu",
        "page3.php" => "Kontakt"
    );

    echo "<ul>";
    foreach($pages as $url => $description){
        if($url == $currentpage){
            echo "<li>$description</li>";
        } else {
            echo "<li><a href='$url'>$description</a></li>";
        }
    }
    echo "</ul>";
}
```

Efterfølgende kan vi hurtigt lave nye sider, vi tilføjer dem til arrayet i menu() og....

```
<?php
//vi inkluderer filen med funktionen menu (og markup)
include("menu.php");

openHTML();

//Vi kalder funktionen med argumentet page1.php (denne side)
menu("page1.php");
?>
<p>Hej, velkommen til "hjem" siden</p>

<?php
closeHTML();
?>
```

# Eksempel: Terningkast

Lad os implementere en webservice til at slå med terninger  
Servicen giver mulighed for at slå et antal gange med en eller flere terninger og vise fordelingen af terningkastene

## Den gyldne plan:

Plan for filen `terning.html`:

- ▶ Brugeren indtaster antal terninger der skal kastes med i hvert kast
- ▶ Brugeren indtaster antal kast

Plan for filen `terning.php`:

- ▶ Konstruer et array `$fordeling` der viser hvor mange gange et bestemt antal øjne er slået.
- ▶ Udskriv fordelingen

Filen terning.html:

```
<form action="terning.php" method="post">
  <p>Antal terninger:
    <input type="text" name="terninger"></p>
  <p>Antal kast:
    <input type="text" name="kast"></p>
  <p>
    <input type="submit" value="Slå"></p>
</form>
```

## Eksempel: Terningkast — fortsat

Filen terning.php: (Klik for kommenteret version)

```
<?php
$terninger = $_REQUEST['terninger'];
$kast = $_REQUEST['kast'];
for ( $i=$terninger ; $i <= 6*$terninger ; $i=$i+1 ) {
    $fordeling[$i] = 0; // Nulstil fordeling
}
for ( $k=0 ; $k < $kast ; $k=$k+1 ) { // Konstruer fordeling
    $oejne = 0;
    for ( $t=0 ; $t < $terninger ; $t=$t+1 ) { // Kast terninger
        $oejne = $oejne + rand(1,6);
    }
    $fordeling[$oejne] = $fordeling[$oejne] + 1; // Optæl fordeling
}
// Udskriv fordeling
for ( $i=$terninger ; $i <= 6*$terninger ; $i=$i+1 ) {
    echo "$i : $fordeling[$i]<br>";
}
?>
```

```
<?php
//Initialisering og $_REQUESTs som forrige, se terning2.php

// Find det maksimale antal kast det samme antal øjne er slået
$max = 0;
for ( $i=$sterninger ; $i <= 6*$sterninger ; $i=$i+1 ) {
    if ( $fordeling[$i] > $max ) { $max = $fordeling[$i]; }
}

// Udskriv fordeling
for ( $i=$sterninger ; $i <= 6*$sterninger ; $i=$i+1 ) {
    $width = 60+(100 * $fordeling[$i]) / $max;
    echo "<div style='width: ".$width."px; background-color:red;'"
        $i : $fordeling[$i] </div>\n";
}
?>
```

Bemærk: For at beregne bredden af hver enkelt <div> finder vi først det maksimale antal kast der er kastet med det samme antal øjne

# Tekstformatting med check-boxes

Check-boxes giver mulighed for flere valg fra en liste af muligheder (i modsætning til radio-buttons) *Eksempel:* format\_check.html:

```
<form action='format_check.php' method='post'>
  <b>Farve:</b>
  <select name='farve'>
    <option value='red'>Rød</option>
    <option value='green'>Grøn</option>
    <option value='blue'>Blå</option>
  </select>
  <b>type face:</b>
  Italic <input type='checkbox' name="face[]" value='i'>
  Bold <input type='checkbox' name="face[]" value='b'>
  <p>
    <textarea name='tekst' cols='50' rows='4'>Teksten</textarea>
  </p>
  <p><input type='submit' value="formatter tekst"></p>
</form>
```

Bemærk: Der benyttes et array som formvariabel for face

## Tekstformattering med checkboxes — fortsat

Filen `format_check.php`:

```
<?php
$tekst = $_REQUEST['tekst'];
$farve = $_REQUEST['farve'];
$face = $_REQUEST['face'];
$prefix = "";
$postfix = "";
for ( $i=0 ; $i < count($face) ; $i=$i+1 ) {
    $prefix = $prefix . "<$face[$i]>";
    $postfix = "</$face[$i]>" . $postfix;
}
$tekst = "$prefix $tekst $postfix";
echo "<p style='color:$farve;'>$tekst</p>";
?>
```

Programmet løber igennem arrayet `$face` og tilføjer start tags til variabelen `prefix` og slut-tags til variabelen `postfix`:

```
<b><i> Teksten </i></b>
```

## Øvelsen:

- ▶ Tilfældig-drink generator
- ▶ Gæt et tal

## Næste gang:

- ▶ Regulære udtryk: et lille sprog til at klassificere strenge
- ▶ Med regulære udtryk kan et webscript afgøre om en bruger har indtastet et tal når det er ønsket og en email-adresse når det er ønsket
- ▶ Med regulære udtryk kan et website gøres mere robust
- ▶ Regulære udtryk giver mulighed for at klippe tekststrenge ud af fremmede websites