

Intro

- Siden sidst: evaluering på opgaver og virtuel kursus
- Kursussammensætning:
forelæsning – læse – arbejde selvstændigt – newsgroup – øvelsestid – aflevering af opgavesæt.

Databasestøttet webpublicering

Forelæsning 3

- Hvad har vi lært?
- Loops, arrays (løkker og variabelsæt) - og funktioner

Hvad har vi lært?

- Client/server – f.eks webserver m. php-fortolker som server til en webbrowser over internettet
- PHP-scripting-begreber og -syntaks
variabler
udskrivning af strenge
datatyper
regnearterne
- HTML: HTML-formularer sender indhold til webserver via GET eller POST-metoden.
- PHP-fortolkeren kan ”se” de indsendte datastrengene og anvende dem i afviklingen af scriptet.
- Udsagn og deres anvendelse sammen med if, if else og switch

Switch

```
$MinAlder = 18;  
switch ($MinAlder )    {  
    case ($MinAlder <= 15):  
        echo "Nej, dette her er ikke noget for dig";  
        break;  
    case ($MinAlder == 16):  
        echo "Tilbage til de sure øl hjemme på værelset";  
        break;  
    case ($MinAlder == 17):  
        echo "Du skal nok få lov til næste år";  
        break;  
    case ($MinAlder >= 18):  
        echo "<P>Du har lov til at komme ind på diskoteket IN";  
        break;  
}
```

Loops

- Hvorfor: for at kunne fremstille rutiner der kan håndtere mange ensartede opgaver og store mængder data - uden at skulle skrive en masse kode.
- At *loope* vil sige at gentage en kommando (eller mange) i loops som disse
 - **while**(udsagn) {udfør noget}; så test udsagnet igen
 - **do** {udfør noget} **while** (udsagn); så tilbage til **do** igen
 - **for** (sæt tællers start; tjek tæller; læg til eller træk fra tæller) {udfør noget};

Array

- **Egentlig: sæt af variabler**

Almindelig strengvariabel

```
$variabel = "etellerandet"
```

Array:

```
$variabelSaet[] = "etellerandet"
```

```
$variabelSaet[] = "nogethelttredje"
```

```
$variabelSaet[] = "syvende"
```

```
echo $variabelSaet[0];
```

```
echo $variabelSaet[1];
```

```
echo $variabelSaet[2];
```

Array kombineret med loop!

Arrayet kan gennemløbes ved *iterationen* i et loop:

```
echo $variabelSaet[0];  
echo $variabelSaet[1];  
echo $variabelSaet[2];
```

```
$laengdeAfArray = count($variabelSaet);  
for($i=0;i< $laengdeAfArray;$i++)    {  
    echo $variabelSaet[$i];  
}
```

At bygge i moduler

- Hvorfor: for at kunne fremstille rutiner (*procedurer*) der kan gentages mange gange – og dermed undgå at skulle skrive en masse kode.
- Anvendelse af *function* er en måde hvorpå man kan opbygge sit script i moduler:

```
function arrayUdskriver(array)    {  
    udfører loop og udskrivning af et hvilket som helst  
    array  
}
```