PHP

Grunnleggende programmering i PHP

Variabler (I)

www.hint.no

- En variabel er en plass avsatt i minnet som programmet kan benytte til å lagre data
- Hver variabel er identifisert ved et eget navn
- Variabelnavn starter alltid med tegnet $

```
$alder = 20;        // variabel som inneh. tall
$navn = ”Jo Råne”;  // variabel som inneh. tekststreng
```

- Merk! Tekststrenger må omsluttes av hermetegn
- Vi lagrer en verdi i en variabel med en tilordningssetning på formen *variabel = verdi*;

Variabler (II)

www.hint.no

- Følgende gjelder for variabelnavn:
  - Må begynne på en bokstav (a-z, A-Z) eller understrek (_)
  - Navnet kan fortsette med bokstaver (a-z, A-Z), sifre (0-9) eller _
  - Det er ingen begrensinger på lengden av navnet
  - Navnet kan ikke inneholde mellomrom eller æ, ø og å
  - Det er forskjell på store og små bokstaver

```
$vekt_i_kg        // OK
$kølapp           // Ikke OK. Inneholder ø
$sko nr           // Ikke OK. Inneholder mellomrom
$6pack            // Ikke OK. Begynner med et siffer
$catch22;         // OK
```

Variabler (III)

www.hint.no

- Hva kan vi lagre i variablene?

1) Tall

```
$heltall = 17;              // heltall (type integer)
$destall = 23.678;          // desimaltall (type float)
                            // Merk! Bruk punktum som
                            // desimalkomma
```

- Enkle regneoperasjoner på tall gjøres ved bruk av operatorene (+) addisjon, (-) subtraksjon, (/) divisjon og (*) multiplikasjon

```
$sum = 5 + $heltall + 186/6;       // $sum = 53
$svar = $destall * 2 – 6.35;       // $svar = 41.006
```

Variabler (IV)

www.hint.no

2) Tekststrenger

```
$karakter = ”B”;           // Tekststrenger skal
$tittel = ”Rubber Soul”;   // omsluttes av hermetegn
$fyord = ’I huleste’       // eller apostrofer
$telefon = ”74 12 34 56”;  // Ikke et tall
```

- Enkle operasjoner på tekststrenger gjøres ved bruk av operatoren (.) som skjøter sammen to tekststrenger

```
$merke = ”BMW”;
$modell = ”520iA”;
$bil = $merke . ” ” . $modell;     // $bil = ”BMW 520iA”
```

Variabler (V)

www.hint.no

3) Boolske verdier

- Dette er variabler som enten har verdien **true** (sann) eller **false** (usann). De er fine å bruke når programmet må huske om noe er sant/usant, på/av etc.

```
$er_mann = true;    // Obs! Ikke hermetegn her
$online = false;
```

- Enkle operasjoner på boolske variabler gjøres ved bruk av operatoren (!) ikke (eng. not) som inverterer verdien

```
$online = !$online;  // $online = true hvis $online var
                     // false fra før, eller false hvis
                     // den var true fra før
```

Spesialtegn i tekststrenger

www.hint.no

- Tekststrenger kan inneholde en del spesialtegn (escape-sekvenser). Disse tegnene skrives alltid inn ved hjelp av to tegn (\ og ett tegn til). Her er de vanligste:

\n	Linjeskift
\t	Tabulator
\"	Hermetegn
\'	Apostrof
\$	Dollartegn
\\	Omvendt skråstrek (bakslask)

Spesialtegn - eksempler

www.hint.no

- \n benytter vi når vi vil legge inn et linjeskift i programkoden (har samme effekt som om vi trykker ENTER når vi skriver HTML kode). \t er fin å bruke når vi ønsker innrykk i programkoden (har samme effekt som å trykke på TAB tasten)

```
echo "<ul>\n";
echo "\t<li>Sausnebb</li>\n";
echo "\t<li>Trøffelgaffel</li>\n";
echo "</ul>\n"
```

- Vil vises som:

```
<ul>
     <li>Sausnebb</li>
     <li>Trøffelgaffel</li>
</ul>
```

Spesialtegn - eksempler

www.hint.no

- \", \', \\ og \$ brukes når vi virkelig ønsker å skrive dette tegnet (sløyfer vi bakslasken har disse tegnene en spesiell betydning)

```
$lenke = "<a href=\"www.uff.no\fjert.htm\">Vindfull</a>";
$price = 1000;
$msg = "This unit costs more than \$ $price";
```

- I første eksempel må vi skrive \" fordi det ellers ville blitt tolket som start el. slutt på tekststreng. I siste eksempel betyr første $ at vi ønsker å skrive tegnet $, mens andre $ er starten på et variabelnavn

Blande ' og "

www.hint.no

- For å slippe å skrive \" kan vi bruke ' og " på en smart måte. Dersom vi får behov for å skrive ut hermetegn i en streng kan vi rett og slett omslutte denne tekststrengen med apostrofer.
- I stedet for;

```
$start_tab = "<table border=\"1\">";
```

- skriver vi (en av disse):

```
$start_tab = '<table border="1">';
$start_tab = "<table border='1'>";
```

Datatyper (I)

www.hint.no

- Hvordan kan vi finne ut hva slags type data vi har lagret i en variabel? Bruk *gettype($variabelnavn)*. Aktuelle typer er **integer** (heltall), **float** (desimaltall), **string** (tekststrenger) og **boolean** (boolske verdier)

```
$sant = false;
echo "Datatypen til \$sant er ", gettype($sant);
```

- Vi får utskriften: *Datatypen til $sant er boolean*
- gettype() er et eksempel på en funksjon (vil bli gjennomgått senere). En annen hendig funksjon er *var_dump($variabelnavn)*. Denne "dumper" ut datatypen, lengden og innholdet.

```
echo var_dump($sant);
```

Datatyper (II)

www.hint.no

- Det finnes også en rekke funksjoner av typen is_x() som kan brukes til å spørre om en variabel har den og den datatypen

```
$var1 = 60;
is_int($var1);      // Sjekker om $var1 er heltall
is_bool($var1);     // Sjekker om $var1 er boolean
is_string(var1);    // Sjekker om $var1 er tekststreng
is_float($var1);    // Sjekker om $var1 er desimaltall
is_numeric($var1);  // Sjekker om $var1 er et tall el.
                    // en tekststreng som inneholder
                    // et tall
```

- Vi skal komme tilbake til hvordan disse funksjonene kan brukes etter hvert

Datatyper (III)

www.hint.no

- Det er mulig å konvertere fra en datatype til en annen. Vi skriver navnet på datatypen vi vil ha (integer, boolean, float, string) i parenteser foran navnet på variabelen som skal konverteres

```
$str_vekt = ”120”;           // en strengvariabel
$vekt = (integer) $str_vekt; // konverter til integer
```

- Noen ganger skjer det en konvertering uten at vi har bedt om det, såkalt implisitt konvertering

```
$a = 1; $b = 2; // to heltallsvariabler
$c =$a . $b;    // konverterer heltallene til string før
                // de skjøtes sammen og lagres i $c.
                // Dette gjøres fordi operatoren . bare
                // virker på tekststrenger
```

Aritmetiske operatorer (I)

www.hint.no

- Aritmetiske operatorer brukes for å utføre matematiske operasjoner

Operator	Beskrivelse	Eksempel	Resultat
+	Addisjon	67 + 9.6	76.5
-	Subtraksjon	19 – 6	13
/	Divisjon	100 / 30	3.333333
*	Multiplikasjon	12 * 6	72
%	Modulo	12 % 5	2

Aritmetiske operatorer (II)

www.hint.no

- Eksempler. Antar at $a = 5 før hver operasjon utføres

Oppgave	Uttrykk	Alternativt	Resultat
Øke variabel med 1	$a = $a + 1	$a++ eller++$a	$a = 6
Redusere variabel med 1	$a = $a – 1	$a-- eller --$a	$a = 4
Øke variabel med 5	$a = $a + 5	$a += 5	$a = 10
Redusere variabel med 5	$a = $a – 5	$a -= 5	$a = 0
Multiplisere variabel med 4	$a = $a * 4	$a *= 4	$a = 20
Dividere variabel med 4	$a = $a / 4	$a /= 4	$a = 1.25

Sammenligningsoperatorer

www.hint.no

- Sammenligningsoperatorer brukes for å sjekke om to størrelser (f. eks. variabler) er like, ulike etc. Resultatet av en slik sammenligning er true eller false. Gitt $a = 5

Op.	Beskrivelse	Sant når	Eks.	Resultat
==	Likhet	v.s. er lik h.s	$a == 5	true
!=	Ulikhet	v.s. er ulik h.s.	$a != 5	false
===	Identisk likhet	v.s. og h.s. er like og av samme type	$a === ”5”	false
>	Større enn	v.s. > h.s.	$a > 5	false
>=	Større eller lik	v.s. >= h.s.	$a >= 5	true
<	Mindre enn	v.s. < h.s.	$a < 5	false
<=	Mindre eller lik	v.s. <= h.s.	$a <= 5	true

Logiske operatorer

www.hint.no

- Tre logiske operatorer er ! (ikke), && (og) og || (eller). Tabellen under viser hvordan de virker. Anta at $a og $b er to variabler av typen boolean

$a	$b	!$a	!$b	$a && $b	$a || $b
false	false	true	true	false	true
false	true	true	false	false	true
true	false	false	true	false	true
true	true	false	false	true	false

Prioriteten til operatorene

www.hint.no

- De ulike operatorene har forskjellig prioritet. F. eks. har operatorene *, / og % høyere prioritet enn + og – (+ og - har samme prioritet, og *, / og % har også samme prioritet). Dette kan ha betydning for hvordan PHP beregner et uttrykk.

```
$resultat = 6 + 2 * 2;      // $resultat = 10
```

- Her blir resultatet 10 pga at 2 * 2 regnes ut før 6 + 4. Dette skyldes at * har høyere prioritet enn +. Hvis vi ikke ønsker dette kan vi overstyre med parenteser:

```
$resultat = (6 + 2) * 2;    // $resultat = 16
```

- Bruk alltid parenteser hvis du er usikker

Konstanter

www.hint.no

- Dersom en variabel skal ha samme verdien hver gang programmet kjøres kan vi heller definere denne som en konstant (som sikrer den mot utilsiktet endring)

```
define(MVA, 0.25);
define(HILSEN, ”Velkomen, vyrde høvding”);

echo HILSEN;
$netto = 1200.0;
$pris = $netto + MVA * $netto;
echo ”Momsen er ” .MVA * 100. ”%. ”;
echo ”Det betyr at du må betale $pris kr”;
```

- Det er vanlig å bruke store bokstaver i navn på konstanter
- Forhåndsdefinerte konstanter finnes også f. eks. PHP_VERSION og PHP_OS

Lage lange tekststrenger

www.hint.no

- Lange tekststrenger kan skrives inn på følgende måte:

```
$tekst = <<<SLUTT
Her kan det komme mange
linjer etter hverandre. Du bestemmer selv
hvor mye du orker å skrive. Det hele
avsluttes ved å gjenta navnet du brukte i første
linje etterfulgt av et semikolon
SLUTT;
echo $tekst;        // Skriv ut hele greia
```

Litt om utskrift

www.hint.no

- Her følger tre forskjellige måter på skrive ut på ($navn=”Torunn”)

```
echo ”<b>Velkommen hit $navn</b><br>”;
echo ’<b>Velkommen hit ’ . $navn . ’</b><br>’;
echo ’<b>Velkommen hit ’, $navn, ’</b><br>’;
echo ’<b>Velkommen hit $navn</b><br>’;
```

- Når du benytter ” ” ved utskrift vil PHP lete gjennom tekststrengen for å se om det er noen variabler der.
- Hvis du bruker ’ ’ vil PHP ikke lete etter variabler i tekststrengen. Du kan da skrive ut slik som vist i linje 2. Men... siste linje vil da bli skrevet ut som Velkommen hit $navn
- Linje 3 viser hvordan du kan skrive ut flere ting ved å skille hver ting (variabel, tekst etc.) med komma. Den siste metoden er litt raskere enn linje 2 fordi der må tekstbitene skjøtes sammen i minnet før utskrift kan skje

Blande PHP og HTML

www.hint.no

- HTML og PHP kan fritt blandes som vist under:

```
<h1>Her kommer en overskrift</h1>
<?php echo ”<p>Her er det noe <b>PHP</b></p>”;
?>
<h2>Her kommer en ny overskrift</h2>
<p>Hallo i stuen unge venn</p>”
<?php
// Enda mer PHP her…
?>
```