

Webbutveckling med PHP

PHP

Utbildare: Mikael Olsson

mikael.olsson@emmio.se

076-174 90 43

NACKADEMIN

PHP-skript

- Ett skript är en serie instruktioner.
- Ett PHP-skript är som en vanlig text-fil där instruktionerna ligger i en PHP-tag:

```
01 > 🐘 01.php
1   Hello world 1
2   <?php
3
4   echo "Hello world 2";
5
6   ?>
7
```

```
→ 01 php 01.php
Hello world 1
Hello world 2%
```

(%-tecknet dyker bara upp för att jag kör ZSH:
<https://stackoverflow.com/questions/33901427/why-does-php-print-a-percent-sign-when-my-script-is-done>)

Starta med PHP

- PHP omges av den speciella PHP-taggen:
HTML-fil:

```
<?php // PHP code goes here ?>
```

```
<body>  
<h1>My first PHP page</h1>  
<?php // Do something ?>  
</body>
```

Variabler

- `$name = "Micke";`
- Behöver inte deklarerats innan den används.
- Svagt typat språk vilket betyder att variabler kan byta datatyp under körning.
- Startar med ett dollartecken följt av namnet.
- Namnet måste starta med en bokstav eller ett underscore: _
- Kan enbart innehålla stora eller små bokstäver (a-z), siffror och underscore.
- Kan inte innehålla mellanslag.
- Case sensitive, `$x` och `$X` är olika variabler.

Konstanter

- En konstant är som en variabel, men man kan inte byta värde på den efter att det är satt.

```
// Valid constant names
define( "FOO",      "something" );
define( "FOO2",     "something else" );
define( "FOO_BAR",  "something more" );
```

Skriva ut

- `echo "Micke";` `// Vanligast`

Kommentar.

- <http://php.net/manual/en/function.echo.php>

- `print "Micke";`

- <http://php.net/manual/en/function.print.php>

- "print is not actually a real function (it is a language construct) so you are not required to use parentheses with its argument list."
- "The major differences to echo are that print only accepts a single argument and always returns 1."

Datatyper

- PHP stöder följande datatyper:
 - String
 - Integer
 - Float (decimaltal, även kallat double)
 - Boolean
 - Array
 - Object
 - NULL
 - Resource

Aritmetiska operatörer

Exempel	Namn	Resultat
$+\$a$	Identitet	Konvertering av $\$a$ till int eller float.
$-\$a$	Negation	Motsatsen till $\$a$.
$\$a + \b	Addition	Summan av $\$a$ och $\$b$.
$\$a - \b	Subtraktion	Skillnaden mellan $\$a$ och $\$b$.
$\$a * \b	Multiplikation	Produkten av $\$a$ och $\$b$.
$\$a / \b	Division	Kvoten av $\$a$ och $\$b$.
$\$a \% \b	Modulus	Resten av $\$a$ delat med $\$b$.
$\$a ** \b	Exponent	$\$a$ upphöjt i $\$b$.

Ökande/minskande operatorer

Exempel	Namn	Effekt
<code>++\$a</code>	Pre-increment	Ökar \$a med ett, returnerar sedan \$a.
<code>\$a++</code>	Post-increment	Returnerar \$a, ökar sedan \$a med ett.
<code>--\$a</code>	Pre-decrement	Minskar \$a med ett, returnerar sedan \$a.
<code>\$a--</code>	Post-decrement	Returnerar \$a, minskar sedan \$a med ett.

Jämförelseoperatorer

Exempel	Namn	Resultat
<code>\$a == \$b</code>	Equal	TRUE if \$a is equal to \$b after type juggling.
<code>\$a === \$b</code>	Identical	TRUE if \$a is equal to \$b, and they are of the same type.
<code>\$a != \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a <> \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a !== \$b</code>	Not identical	TRUE if \$a is not equal to \$b, or they are not of the same type.
<code>\$a < \$b</code>	Less than	TRUE if \$a is strictly less than \$b.
<code>\$a > \$b</code>	Greater than	TRUE if \$a is strictly greater than \$b.
<code>\$a <= \$b</code>	Less than or equal to	TRUE if \$a is less than or equal to \$b.
<code>\$a >= \$b</code>	Greater than or equal to	TRUE if \$a is greater than or equal to \$b.
<code>\$a <=> \$b</code>	Spaceship	An integer less than, equal to, or greater than zero when \$a is respectively less than, equal to, or greater than \$b. Available as of PHP 7.

Övningar

- <https://www.w3schools.com/php/>
 - Variables
 - Echo / Print
 - Data Types
 - Constants
- Extra övningar för er som hinner:
<https://www.learn-php.org/>
 - Hello, World!
 - Variables and types

If-satser

- Låter oss styra programmet beroende på olika villkor.

```
$x=1;
```

```
if( $x==1 )  
    echo "x är lika med ett.";
```

- Vad händer om vi vill skriva ut två gånger?

```
if( $x==1 )  
    echo "x är lika med ett.";  
    echo "x är inte lika med två.";
```

Block

- Ett block utgörs av kod som står mellan måsvingar. All kod inom ett block hör ihop som en instruktion.

```
if($x==1) {  
    echo "x är lika med ett.";  
    echo "x är inte lika med två.";  
}
```

- Det är best practise att alltid använda sig av block, oavsett om man vill utföra en eller flera instruktioner.

If-else-satser

```
$x=1;
```

```
if($x==1) {  
    echo "x är lika med ett.";  
}
```

Else är inte obligatorisk

```
else if($x==2) {  
    echo "x är lika med två.";  
}
```

```
else {  
    echo "x är varken lika med ett eller två.";  
}
```

switch

- Det finns ett annat sätt att skriva if-satser som ser ut såhär:

```
if ($i == 0) {  
    echo "i equals 0";  
} elseif ($i == 1) {  
    echo "i equals 1";  
} elseif ($i == 2) {  
    echo "i equals 2";  
}
```

switch

Värde att kolla

↓

```
switch ($i) {  
  case 0:  
    echo "i equals 0";  
    break;  
  case 1:  
    echo "i equals 1";  
    break;  
  case 2:  
    echo "i equals 2";  
    break;  
}
```

```
if ($i == 0) {  
  echo "i equals 0";  
} elseif ($i == 1) {  
  echo "i equals 1";  
} elseif ($i == 2) {  
  echo "i equals 2";  
}
```


switch

Gäller för alla tre fallen.



```
switch ($i) {  
  case 0:  
  case 1:  
  case 2:  
    echo "i is less than 3 but not negative";  
    break;  
  case 3:  
    echo "i is 3";  
}
```

switch

```
switch ($i) {  
    case 0:  
        echo "i equals 0";  
        break;  
    case 1:  
        echo "i equals 1";  
        break;  
    case 2:  
        echo "i equals 2";  
        break;  
    default:  
        echo "i is not equal to 0, 1 or 2";  
}
```

Default gäller om inget annat matchar.



Loopar

- ”en konstruktion inom imperativa programmeringsspråk för att åstadkomma en iteration, så att en serie satser upprepas flera gånger”
 - for
 - while
 - do ... while

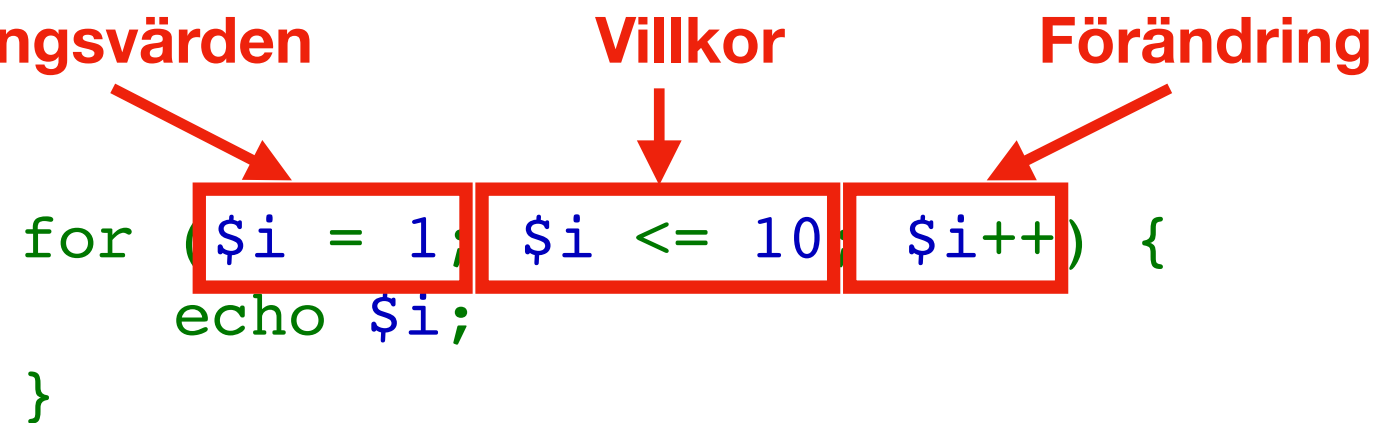
for

Initieringsvärden

Villkor

Förändring

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```



for

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```

Loop	\$i	Fortsätta?
1	1	true
2	2	true
3	3	true
4	4	true
5	5	true
6	6	true
7	7	true
8	8	true
9	9	true
10	10	true
11	11	false

while

Villkor
↓

```
$i = 1;  
while ($i <= 10) {  
    echo $i++; /* the printed value would be  
               $i before the increment  
               (post-increment) */  
}
```

do ... while

```
$i = 0;  
do {  
    echo $i;  
} while ($i > 0);
```

- Största skillnaden mot while är att innehållet i loopen alltid kommer att utföras minst en gång.

Övningar

- <https://www.w3schools.com/php/>
 - If ... Else ... Elseif
 - Switch
 - While Loops
 - For Loops
- Extra övningar för er som hinner:
<https://www.learn-php.org/>
 - For Loops
 - While Loops

Funktioner

- Funktioner är kod vi kan anropa flera gånger.
 - Systemfunktioner
 - Användarfunktioner

Systemfunktioner

- PHP har en massa inbyggda funktioner för att koppla upp databaser, hantera strängar osv.
- <http://php.net/manual/en/funcref.php>

```
$mystring = 'abc';  
$findme   = 'a';  
$pos = strpos($mystring, $findme);  
  
// Note our use of ===. Simply == would not work as expected  
// because the position of 'a' was the 0th (first) character.  
if ($pos === false) {  
    echo "The string '$findme' was not found in the string '$mystring';"  
} else {  
    echo "The string '$findme' was found in the string '$mystring';"  
    echo " and exists at position $pos";  
}
```

- <http://php.net/manual/en/function.strpos.php>

Användarfunktioner

- Vi kan skriva egna funktioner

```
function writeMsg() {  
    echo "Hello world!";  
}
```

```
writeMsg(); // call the function
```

Argument

- Vi kan skicka data, parametrar, argument till en funktion.

```
function familyName($fname, $year) {  
    echo "$fname Refsnes. Born in $year <br>";  
}
```

```
familyName("Hege", "1975");  
familyName("Stale", "1978");  
familyName("Kai Jim", "1983");
```

" VS '

- Skillnaden mellan tecknen i PHP är att med citattecken kommer PHP att utvärdera uttryck i strängen.

```
$name = "Micke";  
echo "Jag heter $name."; // Jag heter Micke.  
echo 'Jag heter $name.'; // Jag heter $name.  
echo 'Jag heter ' . $name . '.'; // Jag heter Micke.  
echo 'Jag heter ' . $name . " ."; // Jag heter Micke.
```

Argument

- Argument kan ha default-värden.

```
function setHeight($minheight = 50) {  
    echo "The height is : $minheight";  
}
```

```
setHeight(350);  
setHeight(); // will use the default value of 50
```

Returnera värden

- Funktioner kan även returnera värden.

```
function sum($x, $y) {  
    $z = $x + $y;  
    return $z;  
}  
  
echo "5 + 10 = " . sum(5, 10);  
echo "7 + 13 = " . sum(7, 13);  
echo "2 + 4 = " . sum(2, 4);
```

Övningar

1. Skriv en funktion som skriver ut en rektangels area. Låt den ta två sidor som argument.
2. Skriv en funktion som returnerar dubbla argumentets värde, dvs om du skickar in 4 till funktionen ska den returnera 8.
3. Skriv en funktion som tar ett startvärde och ett slutvärde som argument. Slutvärdet ska ha ett default-värde. Funktionen ska med hjälp av en loop anropa din andra funktion en gång för varje värde.
4. Skriv en funktion som konverterar grader i F till C.
5. Skriv en funktion som tar två värden och skriver ut det största värdet.
6. Extra research-uppgift: Skriv en rekursiv funktion som räknar ut fakulteten för ett heltal.

Mer övningar

- <https://www.w3resource.com/php-exercises/php-for-loop-exercises.php>
- <https://www.w3resource.com/php-exercises/php-function-exercises.php>

Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?