

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-tagga:

```
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

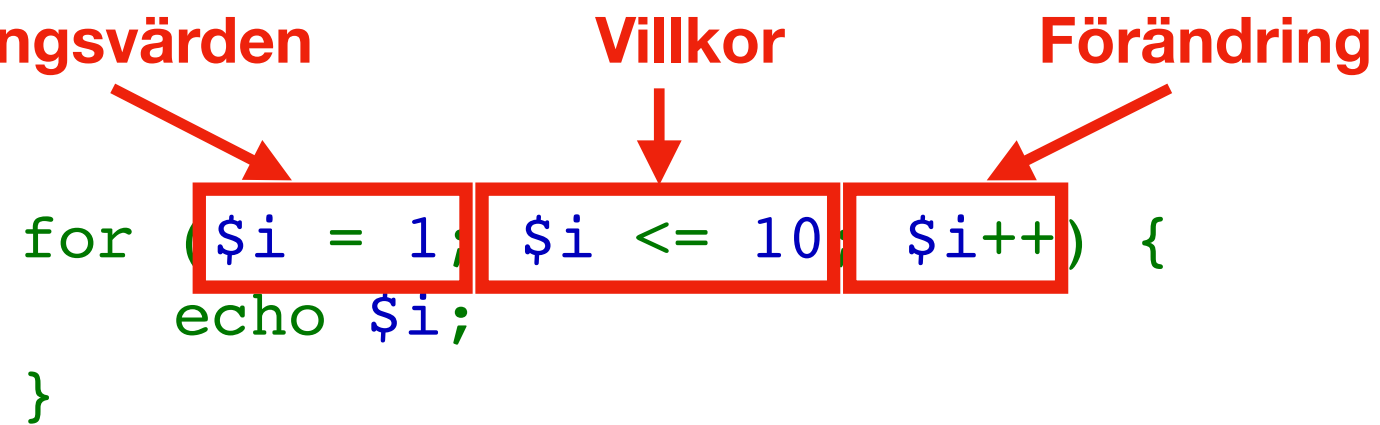
for

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>

PHP i webbmiljö

- PHP-filer kan innehålla HTML och/eller PHP-kod.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="css/main.css">
<script src="main.js"></script>
</head>
<body>
<?php
    echo date("Y-m-d H:i:s");
?>
</body>
</html>
```

PHP i webbmiljö

- PHP-filer kan innehålla hur många PHP-block som helst.

```
<body>
<h1>Hello world!</h1>
Dagens datum: <?php echo date("Y-m-d H:i:s"); ?><br>
<em>Hello world</em> baklänges blir: <?php echo strrev("Hello world"); ?>
</body>
```

Vad gör denna funktion?

PHP i webbmiljö

- Allt vi gjort hittills kan vi göra i webbmiljö.

```
<h1>Hello world:</h1>
<?php
    for ($i = 1; $i <= 5; $i++) {
        echo "$i<br>";
    }
?>
</body>
```

HELLO world!

1
2
3
4
5

PHP i webbmiljö

- Vad man kan göra i PHP beror på vilka moduler man har slagit på.
- Man slår på/av moduler i php.ini.
- Det finns en speciell funktion i PHP som visar information om installationen.

```
extension=imap.so  
extension=gettext.so  
extension=pgsql.so  
extension=pdo_pgsql.so
```

```
<body>  
<?php phpinfo(); ?>  
</body>
```

PHP Version 7.2.1

System	Darwin Mikael's-MacBook-Pro:root:xnu-4903.231.1
Build Date	Jan 15 2018 11:44:11
Configure Command	'./configure' '--with-apache=/Applications/MAMP/htdocs --with-apache2=/Applications/MAMP/htdocs --with-apache2-prefix=/Applications/MAMP/htdocs --with-apache2-sysconfdir=/Applications/MAMP/htdocs --with-apache2-path=/Applications/MAMP/htdocs --with-mysql=mysqlnd --with-mysqli=mysqlnd --with-bcmath --with-imagick=/Applications/MAMP/htdocs --with-pgsql=shared,/Applications/MAMP/htdocs --with-gettext=shared,/Applications/MAMP/htdocs --with-pdo-pgsql=shared,/Applications/MAMP/htdocs --with-pcre-dir=/Applications/MAMP/htdocs --with-iconv=/Applications/MAMP/htdocs --with-mbstring=/Applications/MAMP/htdocs

Extra övning

- Övning för hemmabruk för den som vill fördjupa sina kunskaper.
- <https://phpimagick.com/Tutorial>

Värden som hör ihop

- Hur ska man göra om man har flera studenter?
- Begränsande!

```
$student = "Kalle";
```

```
$student = "Lisa";
```

```
$student2 = "Jessica";
```

```
$student3 = "Lisa";
```

?

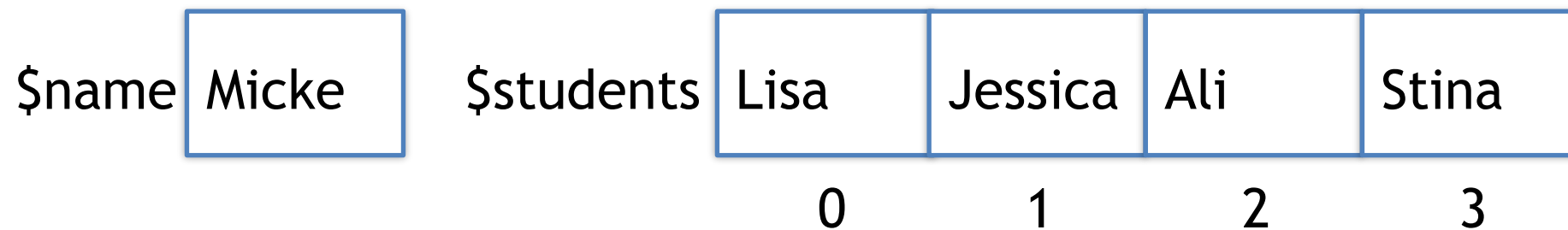
Array

En array är en speciell datatyp som kan innehålla flera värden.

```
$studenter = ["Lisa", "Jessica"];

// Kan stå på flera rader
$studenter = [
    "Lisa",
    "Jessica"
];
```


Array



- En array är en speciell datatyp som kan innehålla flera värden.

```
$name = "Micke";  
$students = [  
    "Lisa",  
    "Jessica",  
    "Ali",  
    "Stina"  
];  
  
echo $name . PHP_EOL;  
echo $students[0];
```

Array

- Det finns en speciell funktion för att skriva ut arrayer.

```
$students = [  
    "Lisa",  
    "Jessica"  
];  
  
print_r($students);
```

```
Array  
(  
    [0] => Lisa  
    [1] => Jessica  
)
```

Array

Man kan även skapa en array med funktionen array.

```
$students = array();  
print_r($students);
```

```
$students = array("kalle");  
print_r($students);
```

```
Array  
(  
)  
Array  
(  
    [0] => kalle  
)
```

Array

- Man kan även få den att returnera värdet.

```
$students = [  
    "Lisa",  
    "Jessica"  
];  
  
$arr = print_r($students, true);  
echo $arr;
```

```
Array  
(  
    [0] => Lisa  
    [1] => Jessica  
)
```

Array - lägga till värde

- Det är enkelt att lägga till nya värden.

```
$students = [  
    "Lisa",  
    "Jessica"  
];  
$students[] = "Kalle";  
  
print_r($students);
```

```
Array  
(  
    [0] => Lisa  
    [1] => Jessica  
    [2] => Kalle  
)
```

Associativ array

- En array kan även använda strängar som nycklar.

```
$students["name"] = "Kalle";  
$students["age"] = 25;  
$students["shoesize"] = 44;  
  
print_r($students);  
  
echo $students["name"];
```

```
Array  
(  
    [name] => Kalle  
    [age] => 25  
    [shoesize] => 44  
)  
Kalle
```

Associativ array

- Man kan även sätta värden direkt som nyckel => värde.

```
$students = [  
    "name" => "Kalle",  
    "age" => 25,  
    "shoesize" => 44  
];  
  
print_r($students);
```

```
Array  
(  
    [name] => Kalle  
    [age] => 25  
    [shoesize] => 44  
)
```

Multidimensionell array

- En array kan ha en annan array som värde.

```
$course = [  
  "teacher" => "Micke",  
  "students" => [  
    0 => [  
      "name" => "Kalle",  
      "age" => 25  
    ]  
  ]  
];  
  
$course["students"][] = [  
  "name" => "Stina",  
  "age" => 28  
];  
  
print_r($course);
```

```
Array  
(  
    [teacher] => Micke  
    [students] => Array  
        (  
            [0] => Array  
                (  
                    [name] => Kalle  
                    [age] => 25  
                )  
            [1] => Array  
                (  
                    [name] => Stina  
                    [age] => 28  
                )  
        )  
    )
```


Array - ändra värde

- Ange nyckeln / indexet för den nyckel du vill ändra.

```
$studerter = [  
    "Lisa",  
    "Jessica"  
];  
$studerter[0] = "Kalle";  
  
print_r($studerter);  
  
$course = [  
    "teacher" => "Micke",  
    "students" => []  
];  
  
$course["teacher"] = "Stina";  
  
print_r($course);
```

```
Array  
(  
    [teacher] => Stina  
    [students] => Array  
        (  
        )  
    )  
)
```

Array-funktioner i PHP

- PHP har många funktioner för att jobba med arrayer.
- <https://www.php.net/manual/en/ref.array.php>

Table of Contents

- [array_change_key_case](#) — Changes the case of all keys in an array
- [array_chunk](#) — Split an array into chunks
- [array_column](#) — Return the values from a single column in the input array
- [array_combine](#) — Creates an array by using one array for keys and another for its values
- [array_count_values](#) — Counts all the values of an array
- [array_diff_assoc](#) — Computes the difference of arrays with additional index check
- [array_diff_key](#) — Computes the difference of arrays using keys for comparison
- [array_diff_uassoc](#) — Computes the difference of arrays with additional index check which is function
- [array_diff_ukey](#) — Computes the difference of arrays using a callback function on the keys
- [array_diff](#) — Computes the difference of arrays
- [array_fill_keys](#) — Fill an array with values, specifying keys
- [array_fill](#) — Fill an array with values
- [array_filter](#) — Filters elements of an array using a callback function
- [array_flip](#) — Exchanges all keys with their associated values in an array
- [array_intersect_assoc](#) — Computes the intersection of arrays with additional index check
- [array_intersect_key](#) — Computes the intersection of arrays using keys for comparison
- [array_intersect_uassoc](#) — Computes the intersection of arrays with additional index check, function
- [array_intersect_ukey](#) — Computes the intersection of arrays using a callback function on the keys
- [array_intersect](#) — Computes the intersection of arrays
- [array_key_exists](#) — Checks if the given key or index exists in the array
- [array_key_first](#) — Gets the first key of an array
- [array_key_last](#) — Gets the last key of an array
- [array_keys](#) — Return all the keys or a subset of the keys of an array
- [array_map](#) — Applies the callback to the elements of the given arrays
- [array_merge_recursive](#) — Merge one or more arrays recursively
- [array_merge](#) — Merge one or more arrays
- [array_multisort](#) — Sort multiple or multi-dimensional arrays
- [array_pad](#) — Pad array to the specified length with a value
- [array_pop](#) — Pop the element off the end of array
- [array_product](#) — Calculate the product of values in an array
- [array_push](#) — Push one or more elements onto the end of array
- [array_rand](#) — Pick one or more random keys out of an array
- [array_reduce](#) — Iteratively reduce the array to a single value using a callback function
- [array_replace_recursive](#) — Replace elements from second array into the first array recursively

compact

- compact() looks for a variable with that name in the current symbol table and adds it to the output array such that the variable name becomes the key and the contents of the variable become the value for that key

```
$city = "San Francisco";  
$state = "CA";  
$event = "SIGGRAPH";
```

```
$location_vars = array("city", "state");
```

```
$result = compact("event", "nothing_here", $location_vars);  
print_r($result);
```

```
Array  
(  
    [event] => SIGGRAPH  
    [city] => San Francisco  
    [state] => CA  
)
```

implode

- Returns a string containing a string representation of all the array elements in the same order, with the glue string between each element.

```
$array = array('lastname', 'email', 'phone');  
$comma_separated = implode(",", $array);
```

```
echo $comma_separated; // lastname,email,phone
```

```
// Empty string when using an empty array:
```

```
var_dump(implode('hello', array())); // string(0) ""
```

array_values

- Return all the values of an array.

```
$array = array("size" => "XL", "color" => "gold");  
print_r(array_values($array));
```

```
Array  
(  
    [0] => XL  
    [1] => gold  
)
```

Övningar

- <https://www.w3schools.com/php/>
 - Arrays
 - Sorting arrays
 - Arrays multi (under PHP Advanced)
- Om du hinner:
 - <https://www.w3resource.com/php-exercises/php-array-exercises.php>

Objekt

- Ett objekt är ganska likt en array i PHP, åtminstone så långt vi vet om dem än.
- Ett standardobjekt skapas med `stdClass()`. Lägg också märke till nyckelordet *new*.

```
$course = new stdClass();  
  
$course->name = "System development in PHP";  
$course->points = 50;  
$course->approved = true;  
  
print_r($course);
```

```
stdClass Object  
(  
    [name] => System development  
    [points] => 50  
    [approved] => 1  
)
```

Objekt

Ett objekt är ganska likt en array i PHP, åtminstone så långt vi vet om dem än.

```
$course = new stdClass();  
  
$course->name = "System developme  
$course->points = 50;  
$course->approved = true;  
  
print_r($course);
```


Superglobala variabler

- Det finns ett par fördefinierad superglobala variabler som alltid finns tillgängliga.

- \$GLOBALS
- \$_SERVER
- \$_GET
- \$_POST
- \$_FILES
- \$_COOKIE
- \$_SESSION
- \$_REQUEST
- \$_ENV

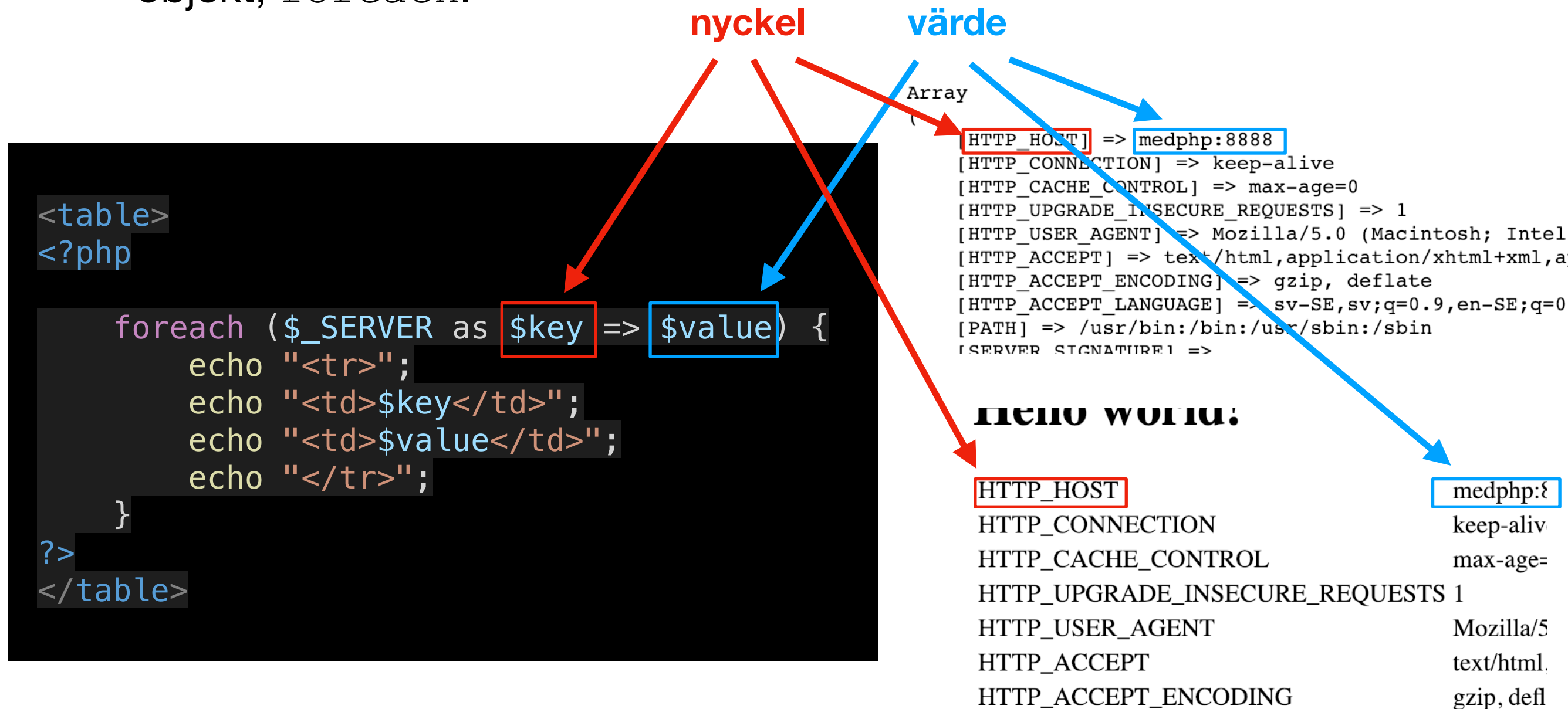
```
<body>
<pre>
<?php print_r ($_SERVER); ?>
</pre>
</body>
```

Array
(

```
[HTTP_HOST] => medphp:8888
[HTTP_CONNECTION] => keep-alive
[HTTP_CACHE_CONTROL] => max-age=0
[HTTP_UPGRADE_INSECURE_REQUESTS] => 1
[HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac
[HTTP_ACCEPT] => text/html,application/xhtml+xml,appli
[HTTP_ACCEPT_ENCODING] => gzip, deflate
[HTTP_ACCEPT_LANGUAGE] => sv-SE,sv;q=0.9,en-SE;q=0.8,e
[PATH] => /usr/bin:/bin:/usr/sbin:/sbin
[SERVER_SIGNATURE] =>
```

foreach

- Sist pratade vi om `for`-loopar.
- Det finns en speciell `for`-loop man kan använda för arrayer och objekt, `foreach`.



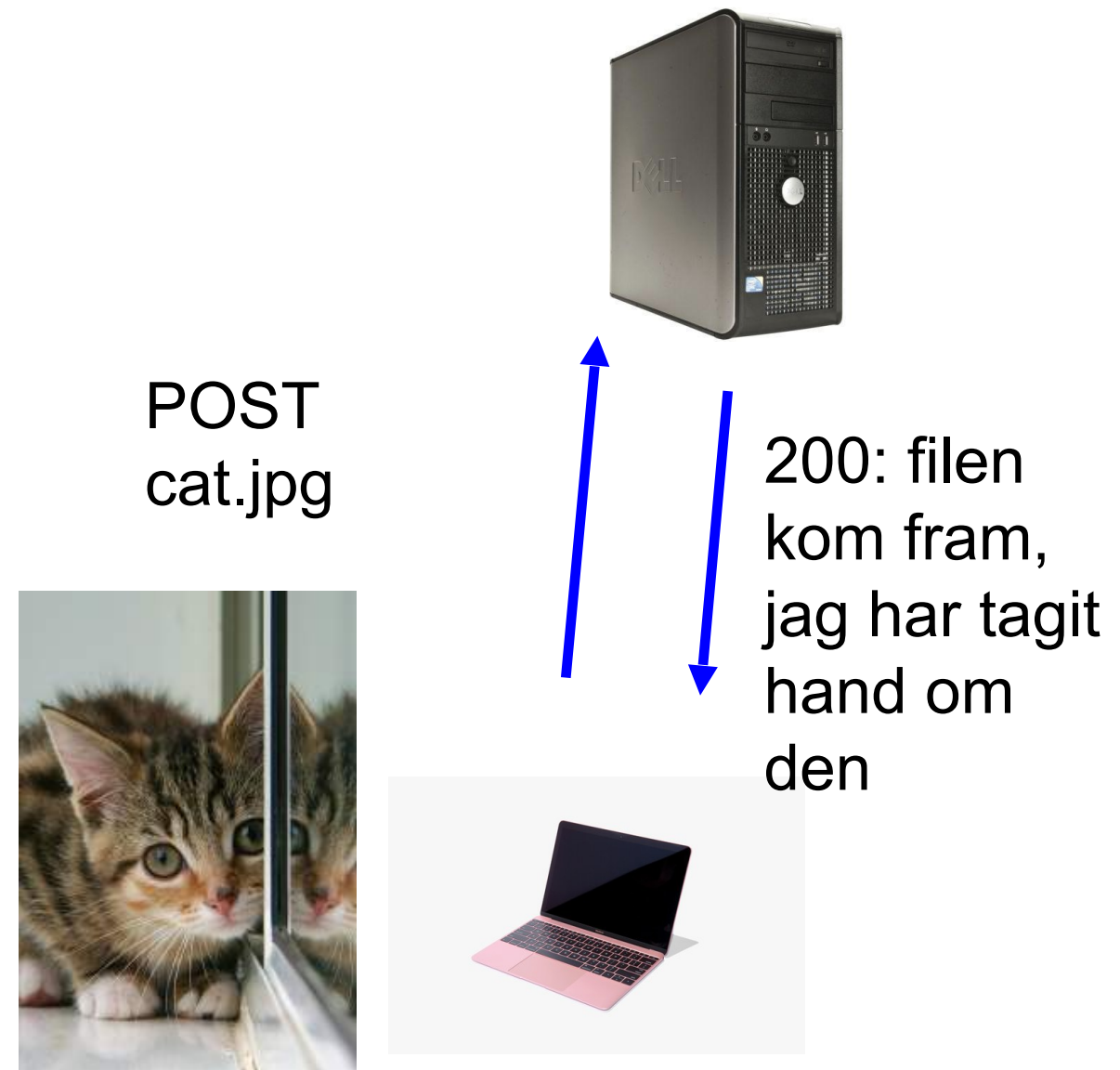
HTML - formulär

- Servern kan skicka information till besökaren (klienten) i form av html, css osv.



HTML - formulär

- Hur kan besökaren (klienten) skicka information till servern?
- Varför skulle vi vilja att besökaren kan skicka information till oss?
- Hur kan vi hantera informationen som besökaren skickar till oss?



HTML - formulär

- Saker vi kan göra med information besökaren skickar:
 - Posta ett inlägg a la facebook
 - Visa en kategori av böcker som besökaren valt
 - Autentisera användare
 - Mer?

HTML - formulär

```
<form method="post" action="some_page.php">
```

```
Namn:<br>
```

```
<input type="text" name="name" placeholder="Skriv  
ditt namn här"><br>
```

```
Ålder:
```

```
<select name="age">  
  <option value="age_group1">0-30</option>  
  <option value="age_group2">30-60</option>  
  <option value="age_group3">60-</option>  
</select>
```

```
</form>
```

HTML - formulär

```
<form method="post" action="some_page.php">
```

```
  Intressen:<br>
```

```
    <input type="checkbox" name="hobbies"
value="fotboll">Fotboll<br>
```

```
    <input type="checkbox" name="hobbies"
value="hockey">Hockey<br>
```

```
  Favorit-färg:
```

```
    <input type="radio" name="color"
value="yellow">Gul<br>
```

```
    <input type="radio" name="color" value="red">Röd<br>
```

```
</form>
```

HTML - formulär

```
<form method="post" action="some_page.php">  
  <input type="number" min="0" max="100" step="2"><br>  
  <input type="email"><br>  
  <input type="color"><br>  
</form>
```


Övning 1

- Skapa en array som innehåller alla superglobaler.
- För varje superglobal, skriv ut alla nycklar och dess värden.

Övning 2

- Skapa en sida med ett formulär med följande komponenter: textruta, textarea, select, radioknapp, checkbox, colorpicker
- Posta sidan till en ny sida och skriv där ut alla nycklar och värden som användaren har skickat.

Övning 3

- Strängfunktionerna i PHP:
<http://php.net/manual/en/ref.strings.php>
- Skapa ett formulär som låter användare använda följande funktioner:
 - htmlentities, explode, crypt, str_word_count, strip_tags, strlen

Övning 4

- Skapa spelet Nim. En hög har från början 21 pinnar. Två spelare turas om att ta 1, 2 eller 3 pinnar från högen. Den som tar den sista pinnen har förlorat.
- Håll reda på hur många pinnar som finns kvar i högen, vad spelarna heter och vems tur det är.