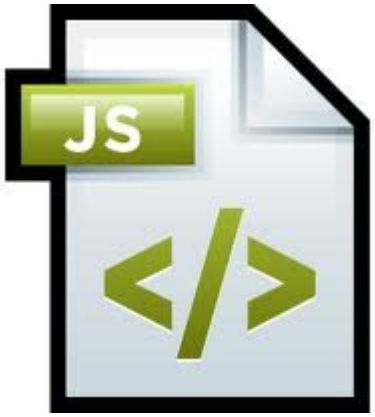


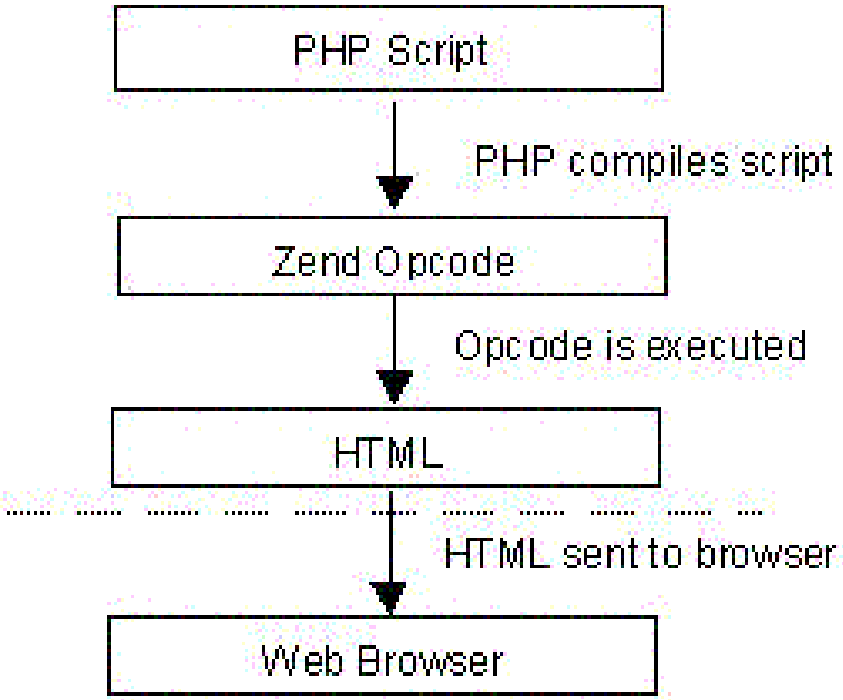
Lab 2 review

PHP intro

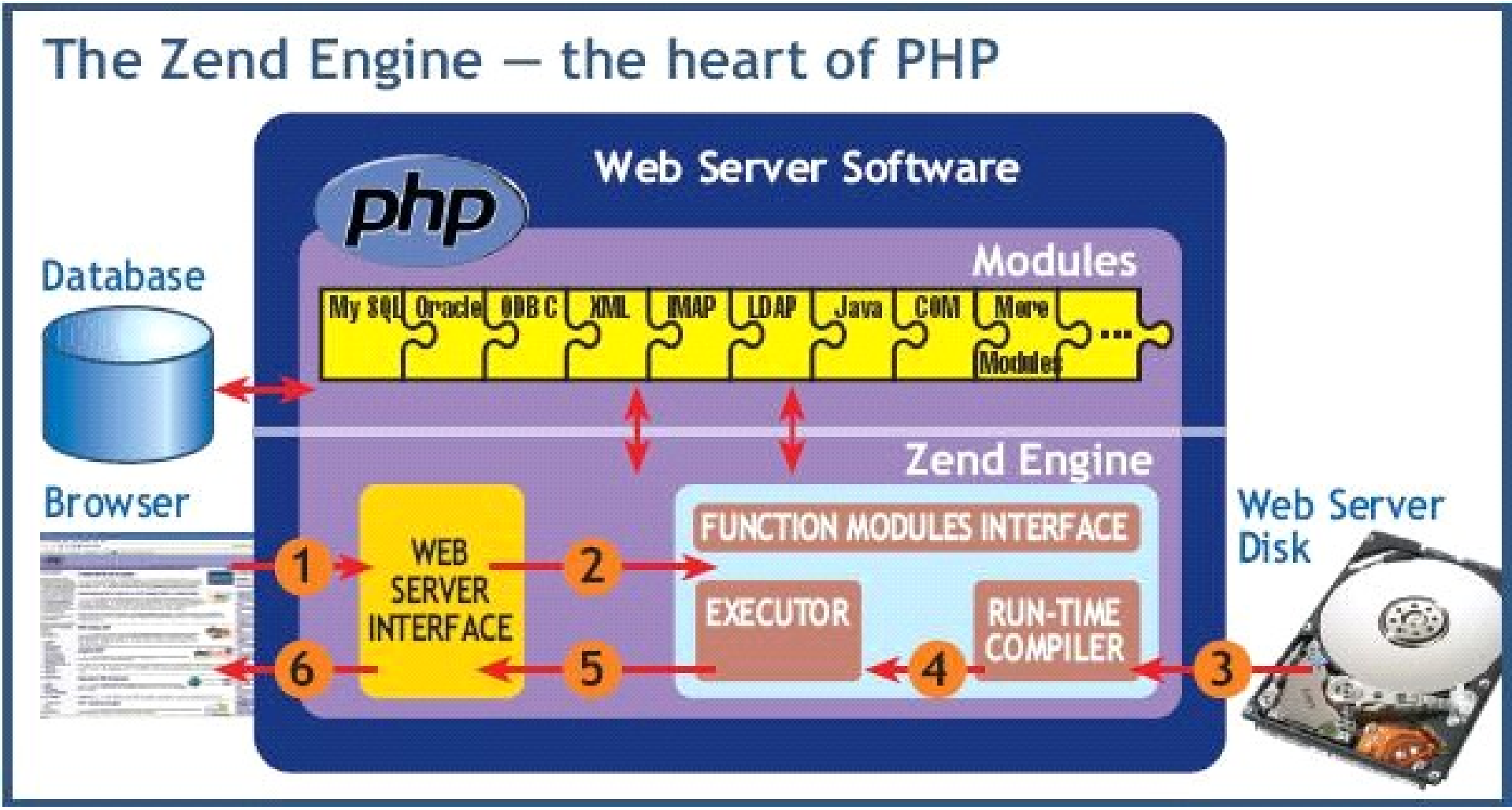
MVC (Model View controller)

What is PHP?

- Är ett allmänt open source server-side skriptspråk som ursprungligen utformats för webbutveckling
- Syfte att producera dynamiska webbsidor
- PHP-koden bäddas ofta in i HTML
- Källdokumentet, php sidan (filen), tolkas av en webbserver och dess installerade php motor (Zend Engine)
- Som efter tolkning (parsing), genererar och skickar innehållet till klienten



PHP how?


The Zend Engine — the heart of PHP
Web Server Software
php
Modules
My SQL
Oracle
ODB C
XML
IMAP
LDAP
Java
COM
More Modules
...
Database
Browser
Zend Engine
FUNCTION MODULES INTERFACE
WEB SERVER INTERFACE
EXECUTOR
RUN-TIME COMPILER
1
2
3
4
5
6
Web Server Disk

Execution of PHP?

- Diagrammet (förra sliden) visar arkitekturen och grundläggande dataflödet för PHP-baserade webbsidor.
- Browser initierar ett förfrågan av en php sida till webbservern, som skickar en begäran vidare till PHP Web Server Interface (1).
- Web Server Interfacet anropar Zend Engine (2), som har åtkomst till webserverns hårddisk. Från hårddisken hämtas koden PHP-sidan och skickar PHP-kod till Run-time kompilator (3).
- Zend Engines Run-time-kompilator skapar en komplierad version av scriptet, som skickas vidare till Zend Engine Executor (4). Executor genererar HTML som kan ses av webbläsaren.
- Om script behöver tillgång till andra moduler, såsom SQL-databas, XML etc, skickar Zend Engine den koden vidare lämplig PHP-moduler, som processer begäran, och skickar resultatet till webbserverns Interface (5).
- Som sedan skickar HTML till webbläsaren (6).

PHP - www.php.net

- PHP (Personal Home Page) or PHP Hypertext Processor is a server-side script language - you must have a running web server in order to try it out!
- Many Web CMS systems are based on PHP
- PHP code which can be mixed with HTML exist in .php files
- Remeber the client browser will NOT be able to se the code!
- It does not matter if you use PHP or HTML mode – a PHP file however always start off in HTML mode
- To insert CR (Carriage Return) and LF (Line Feed) put “\r\n”

```php
<?php
echo "<html>";
echo "<body>";
echo "This is some PHP";
echo "</body>";
echo "</html>";
?>
```

```php
<html>
<body>
<?php
echo "This is some PHP";
?>
</body>
</html>
```

Where? School and PHP help

- I en fil med filändelsen .php, tex. shopping.php
  - Oftast inbäddat i HTML koden
- Med hjälp av editor eller plugin
  - Notepad++, Dreamweaver, Codelobster, …, Eclipse
  - To install latest stable PDT build in Eclipse, open "Help -> Install New Software" and enter following URL: http://download.eclipse.org/tools/pdt/updates/release
- Skolans webbserver har stöd för php → phpinfo();
  - http://users.du.se/~hjo/cs/dt1040/phptest.php
  - PHP v5.3.x – 5.5.x
  - Du kan lägga dina php filer i din www mapp
- Webb resurser
  - http://www.tuxradar.com/practicalphp
  - http://php.net
  - http://w3schools.com/php/default.asp

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php
$result = phpinfo();
echo $result;
?>
</body>
</html>
```

PHP basics

- A PHP command does something, like open a file or display a message
- A PHP function returns information, such as the value of a database field or the sum of two numbers
- Every variable in PHP has to be preceded by the **$** (dollar) sign and can hold all sorts of data for the life of the script
- Every php snippet must start with **<?php** and end with **?>**
- PHP is very good to manipulate strings and dates
- The **.** (dot operator) concatenate strings

```
<!doctype html>
<html>
<body>
Welcome to our site.<br><br>
<?php
$rnd = rand(1,10);
if ($rnd == 7){
   echo "Thank you for visiting<br>";
}
?>
</body>
</html>
```

```
<?php
$to = "robert@the-web-book.com";
$subject = "Hello";
$message = "This is my Web server,
    sending me email on ";
$message .= date("D");
$message .= " at ";
$message .= date("H:i");
$m = mail($to, $subject, $message);
echo "Return code from mail was " . $m;
exit();
?>
```

PHP basics – operators

+	Binary. Adds operand one to operand two
-	Binary. Subtracts operand two from operand one
*	Binary. Multiples operand one by operand two
/	Binary. Divides operand one by operand two
.	Binary. Appends string operand two to operand one
!	Unary. Acts as an inverter (true becomes false and false becomes true)
++ --	Unary. Increments (++) or decrements (--) a variable
=	Binary. Assigns operand two to operand one
==	Binary. Tests equality between operand one and operands two
===	Binary. Tests *absolute* equality between operand one and operands two
< >	Binary. Less than and greater than, although also called the "Pulp Fiction" operators (if you don't get it, don't fret!)
&&	Binary. "Logical AND"; tests whether two sets of conditions are true
||	Binary. "Logical OR"; tests whether either of two sets of conditions are true

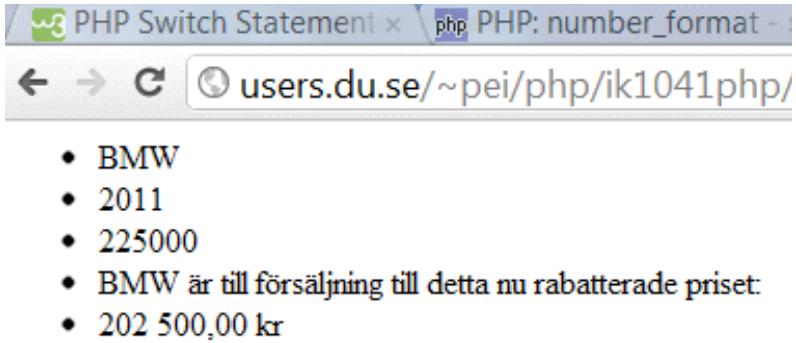
PHP basics

- Variable scope
  - Each variable has an area in which it exists, known as its scope
  - Any variables not set inside a function or an object are considered global
- To define strings use either ' or ” as 'Hans' or ”Hans”
- Constants
  - define(name, value);
- Pre set constants as
  - __FILE__, __LINE__, __FUNCTION__ etc.
- Checking if a variable is set
  - The isset() function
- Functions provide the core of PHP's programming functionality
  - Strings, Date and Time, Mathematics, Regular expressions
  - User functions

```php
<?php
    define("CURRENT_TIME", time());
    print CURRENT_TIME;

    $foo = 1;
    if (isset($foo)) {
        echo "Foo is set\n";
    } else {
        echo "Foo is not set\n";
    }
    if (isset($bar)) {
        echo "Bar is set\n";
    } else {
        echo "Bar is not set\n";
    }
?>
```

PHP basics – variables and echo

```php
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>PHP säger hej</title></head>
<body>
<ul>
<?php
    //skapar upp olika typer av variabler
    $bilMarke='BMW';
    $arsModell=2011;
    $pris=225000;
    $tillForsaljning=TRUE;
    $rabatt=0.10;
    //echo=skriver ut de olika variablernas värden(data) till sidan.
    //Skrivs som strängar till "outputstreamen"
    echo '<li>',$bilMarke,'</li>';
    echo '<li>',$arsModell,'</li>';
    echo '<li>',$pris,'</li>';
    if($tillForsaljning==TRUE)
    {
        echo '<li>',$bilMarke,' är till försäljning till detta nu rabatterade priset:</li>';
        $rabatteratPris=$pris-($pris*$rabatt);
        //number_format= Formattera ett tal antal decimaler, vilket kommatecken och tusentals separerare eg 1 234,45
        echo '<li>',number_format($rabatteratPris,2,',',' '),' kr </li>';
    }
?>
</ul>
</body>
</html>
```



- echo vs. print?
  - print cannot chain text output with the comma syntax
  - print returns a value

PHP loops and comments

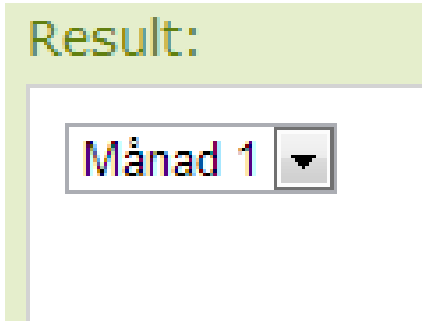
- The for and while loops works excactly as in other C derived or influenced languages as Java, C# etc.
- To comment your code in a php snippet use: #, /*…*/ or //

```php
<?php
echo "<!doctype html>";
echo "<html>";
echo "<body>";
echo "A web server that knows its
    7 times table!<br><br>";

# a for loop
for ($x = 1; $x <= 10; $x++)
{
  echo $x . " times 7 is " .
    $x * 7 . "<br>";
}
echo "</body>";
echo "</html>";
?>
```

```php
<!doctype html>
<html>
<body>
My web server still knows its 7 times table
<br><br>
<?php
$i=1;
# a while loop
while($i <= 10)
{
  echo $i;
  echo " times 7 is ";
  $product = $i * 7;
  echo $product . "<br>";
  $i = $i + 1;
}
?>
</body>
</html>
```

PHP basics – while loop

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4  <title>PHP loopar</title></head>
5  <body>
6  <form>
7  <select name="manader">
8  <?php
9      $i = 1;
10     while ($i <= 12) {
11         echo "<option value={$i}>Månad {$i}</option>";
12         $i++;
13     }
14 ?>
15 </select>
16 </form>
17 </body>
18 </html>
```

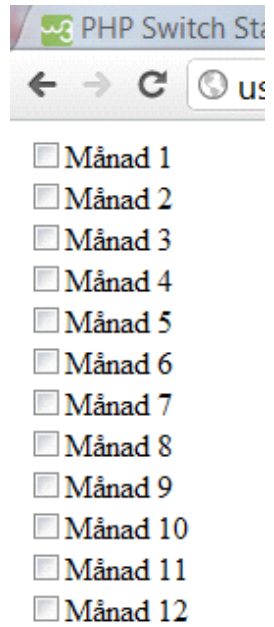


- The curly brackets are useful/needed if you want to have interpolated variables within a double-quoted string, but where the parser would have trouble identifying the variable name for one reason or another

PHP basics – for loop

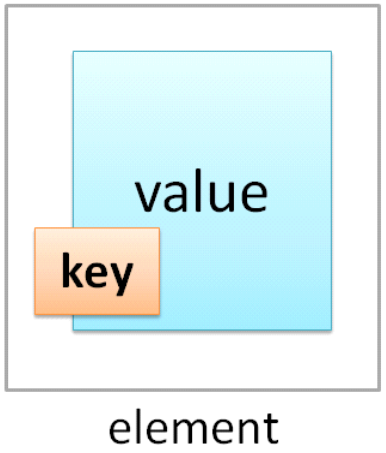
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4  <title>PHP loopar</title></head>
5  <body>
6  <form>
7  <?php
8
9      for ($i=1; $i <= 12;$++) {
10         echo "<input type="checkbox" name="{$i}" value="Milk">Månad {$i}<br>";
11         $i++;
12     }
13 ?>
14 </form>
15 </body>
16 </html>
```

```
<?php
$foo = 1;
$foobar = 2;
echo "{$foo}bar", "<br>"; // '1bar'
echo "$foobar"; // '2'
?>
```



Whats is an array

- En datastruktur som håller gemensamt data
- En array innehåller element som har en nyckel och ett värde
- Byggd enligt key-value principen
- Endera indexerad
  - $ar1 = array(value, …);
- Eller associativ
  - Use
  - $ar2 = array(key => value, ...);
  - or for PHP >= 5.4
  - $ar3 = [key => value, ...];



Indexed arrays

- Arrays of elements works much like in C derived or influenced languages as Java, C# etc.
- We could have initialized the array like this as well
  - $drink = array("Whiskey", "Brandy", ...); or
  - $drink[] = "Whiskey";
  - $drink[] = "Brandy";
- Deleting an element as unset($drink[3]); will create a ”hole”
  - Fix it with: $drink = array_values($drink);

```php
<?php
$drink[0] = "Whiskey";
$drink[1] = "Brandy";
$drink[2] = "Pint of lager";
$drink[3] = "Shandy";
$drink[4] = "White wine (large)";
$drink[5] = "White wine (small)";
$drink[6] = "Red wine (large)";
$drink[7] = "Red wine (small)";

echo "<html>";
echo "<body>";
echo "Drinks Menu<br><br>";
```

```php
$tot_drinks = count($drink);
// Elements will be arranged from
// lowest to highest when done
sort($drink);

for ($i = 0; $i < $tot_drinks; $i++) {
  echo $drink[$i];
  echo "<br>";
}

echo "</body>";
echo "</html>";
?>
```

Associative array

```
17  //Alla arrayer är byggd enligt key value principen
18  //endimensionell och indexerad
19  $bilMarken=array('BMW','Audi','Volvo','Saab','Cadilac');
20  //endimensionell och associativ
21  $bmw=array('Marke'=>'BMW','Modell'=>'633 CSI','Färg'=>'Svart','Pris'=>1235000,'Årsmodell'=>1989);
22  $markeLand = array('BMW' => 'Tyskland', 'Saab' => 'Sverige', 'Cadillac' => 'USA','Kia' => 'Syd Korea');
```

värde=Cadillac
nyckel=4

element

nyckel värde

element

Get data from an array

```php
//Alla arrayer är byggd enligt key value principen
//endimensionell och indexerad
$bilMarken=array('BMW','Audi','Volvo','Saab','Cadilac');
//endimensionell och associativ
$bmw=array('Marke'=>'BMW','Modell'=>'633 CSI','Färg'=>'Svart','Pris'=>1235000,'Årsmodell'=>1989);
$markeLand = array('BMW' => 'Tyskland', 'Saab' => 'Sverige', 'Cadillac' => 'USA','Kia' => 'Syd Korea');
```

```php
//Hämta ut data
//1. med hjälp av indexposition
echo $bilMarken[0],'<br />';
//2. med hjälp av nyckelvärde
echo $bmw['Modell'],'<br />';;
echo $markeLand['Saab'],'<br />';
```

output

Arrayer

BMW
633 CSI
Sverige

Foreach – looping arrays

```php
17  //Alla arrayer är byggd enligt key value principen
18  //endimensionell och indexerad
19  $bilMarken=array('BMW','Audi','Volvo','Saab','Cadilac');
20  //endimensionell och associativ
21  $bmw=array('Marke'=>'BMW','Modell'=>'633 CSI','Färg'=>'Svart','Pris'=>1235000,'Årsmodell'=>1989);
22  $markeLand = array('BMW' => 'Tyskland', 'Saab' => 'Sverige', 'Cadillac' => 'USA','Kia' => 'Syd Korea');
```

```php
foreach (array_expression as $value)
    statement
foreach (array_expression as $key => $value)
    statement
```

```php
43  //3. foreach loop
44  echo'<ul>';
45  foreach($bilMarken as $key=>$value)
46      echo '<li>',$key,'->',$value,'</li>';
47  echo '</ul>';
48
49  echo '<ul>';
50  foreach($bmw as $key=>$value)
51      echo '<li>',$key,'->',$value,'</li>';
52  echo '</ul>';
```

output

- 0->BMW
- 1->Audi
- 2->Volvo
- 3->Saab
- 4->Cadilac

- Marke->BMW
- Modell->633 CSI
- Färg->Svart
- Pris->1235000
- Årsmodell->1989

Strings and regular expressions

- It is common to split strings into arrays in some form
  - array **explode** (string $delimiter , string $string [, int $limit]);
  - Use: **array preg_split** (string $pattern , string $subject [, int $limit] [, int $flags]); for more complicated stuff

```php
<?php
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6"; // Example 1
$pieces = explode(" ", $pizza);
print_r($pieces); // print_r() prints human readable information about a variable
echo "<br>";

$data = "foo:*:1023:1000::/home/foo:/bin/sh";    // Example 2
// list assign variables as if they were an array
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $data);
echo $user, " and "; // foo
echo $pass, "<br>"; // *

//array_shift – Shift an element off the beginning of array - returns the shifted value
$country = "SE;SWEDEN,AL;ALBANIA,DK;DENMARK"; // Example 3
$result = array();
$lines = explode(',', $country);
foreach ($lines as $line) {
    $line = explode(';', $line);
    print_r($line);
    $result[array_shift($line)] = array_shift($line);
}
echo "<br>";
print_r($result);
?>
```

output

```
Array ( [0] => piece1 [1] => piece2 [2] => piece3 [3] => piece4 [4] => piece5 [5] => piece6 )
foo and *
Array ( [0] => SE [1] => SWEDEN ) Array ( [0] => AL [1] => ALBANIA ) Array ( [0] => DK [1] => DENMARK )
Array ( [SE] => SWEDEN [AL] => ALBANIA [DK] => DENMARK )
```

Passing Information to PHP

- By using REST (REpresentational State Transfer) and HTTP GET we can pass variables to the PHP script
- http://localhost/myhome/demo.php?fname=Hans&sname=Jones
- In the URL above we pass two strings, ”Hans” and ”Jones”
- We receive the parameters with $_GET[”variable-name”] in PHP
- For every new variable in the URL we put an ampersand (&) in between

- Error reporting is configured in php.ini
- It can be set in the file (first line) with the command error_reporting();
- error_reporting(E_ALL);

```php
<?php
if(isset($_GET["fname"]))
     $fname = $_GET["fname"];
else
     $fname = '';

if(isset($_GET["sname"]))
     $sname = $_GET["sname"];
else
     $sname = '';

$person = $fname . " " . $sname;

echo "<html>";
echo "<body>";
echo "Hello ";
echo $person;
echo ", how are you today?";
echo "</body>";
echo "</html>";
?>
```

Never forget to sanitize input!

- An attacker could put in *anything*, even scripts as parameters to your REST service!
  - http://localhost/myhome/demo.php?fname=Hans&sname=<h1>Jones</h1>
- We must get rid of tags (<) etc. and could for example use the str_replace() function
  - $person = str_replace("<","",$person);
- A better option is to use the preg_replace(); function
  - $person = preg_replace("/[^A-Z,a-z,0-9, ,.,',;,:,?]/", "", $str);
- It will filter out everything except the characters following the **^** until **]**
- It is much better to delete everything EXCEPT a specified range of characters than allow everything apart from the following ...
- Failure to do this will mean that your site WILL get hacked!

User-Defined Functions

- You can create your own user-defined functions
- Try out regular expressions at: http://www.functions-online.com/

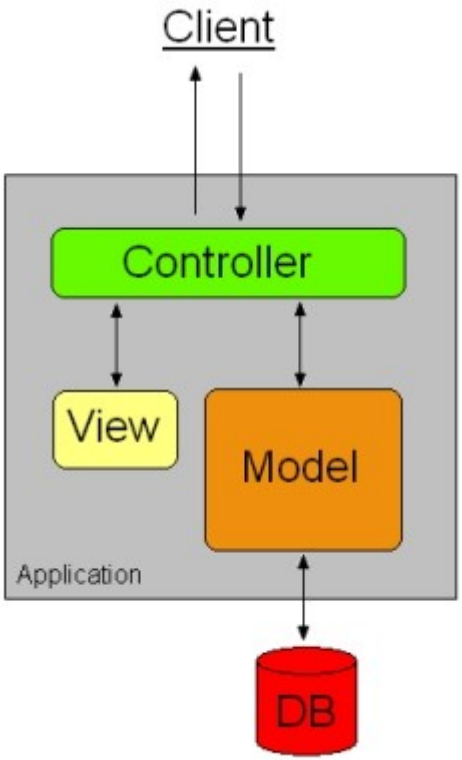
```php
<?php
echo "<html>";
echo "<body>";

if(isset($_GET["name"]))
     $person  = $_GET["name"];
else
     $person = '';
$person_san =
     sanitize_url_string($person);

echo strlen($person);
echo "<br>";

echo strlen($person_san);
echo "<br>";

echo "Hello ";
echo $person_san;
echo ", how are you today?";
exit();
```

- We need to specify exit(); otherwise the function may be executed
- Or just place your functions first

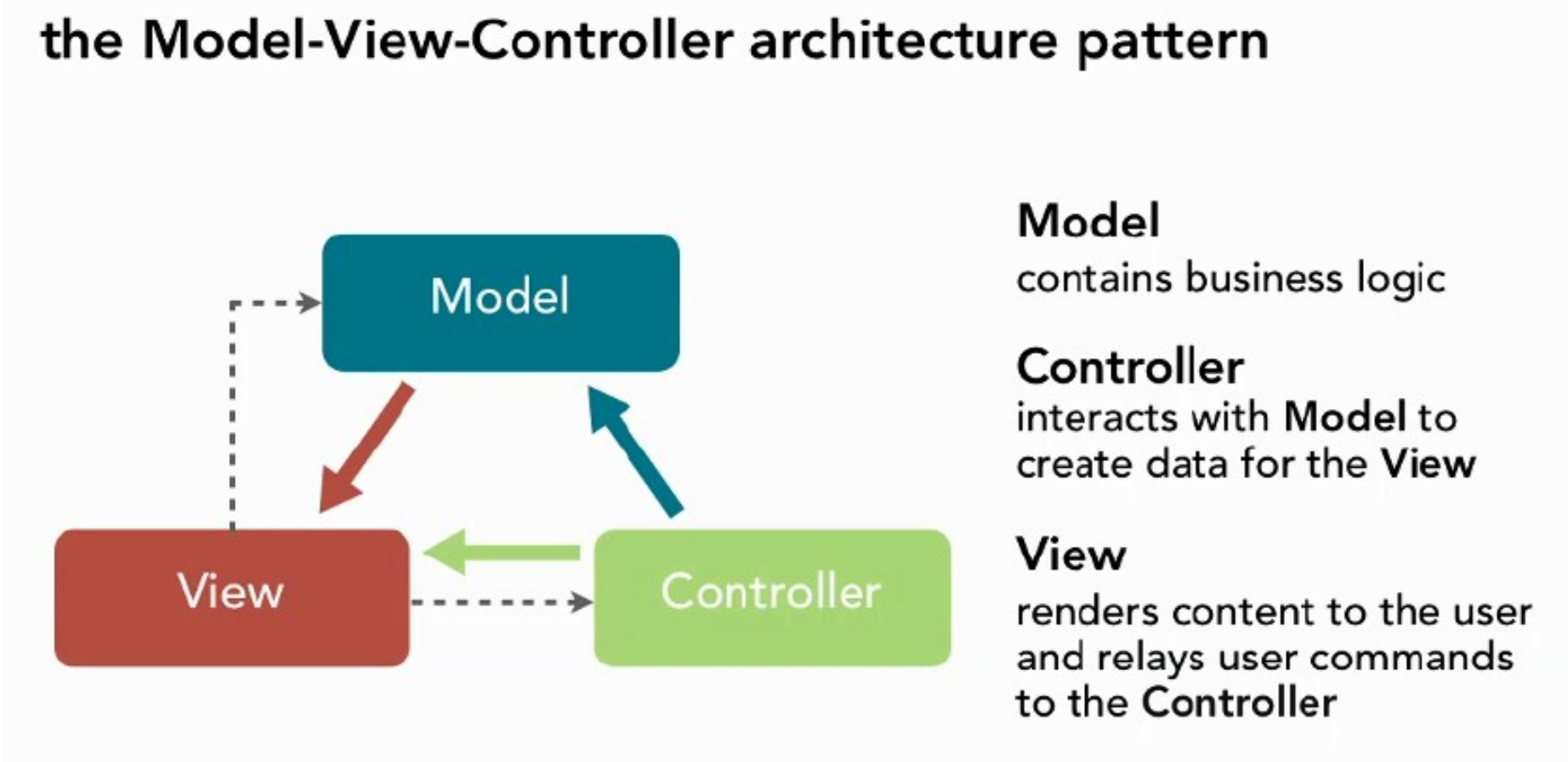
```php
function sanitize_url_string($str)
{
     #echo preg_replace($pattern, $replacement, $string);
     $s =
     preg_replace("/[^A-Z,a-z,0-9, ,.,',;,:,?]/", "", $str);
     return $s;
}
echo "</body>";
echo "</html>";
?>
```

Model View Controller (MVC)

- In order to do server-based programs we need the ability to interrogate or ”talk” with databases
- We can identify and solve this problem with a ”design pattern”
  - If we have a typical problem we may have a typical solution
  - http://en.wikipedia.org/wiki/Design_pattern_(computer_science)
- It is convinient to separate data (model) and presentation (view) so changes in the data management will not affect the presentation or vice versa
- MVC solves this by introducing the controller component
  - http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller



MVC architecture pattern



Model-View-Controller (MVC)

- The model layer
  - Contains business logic as the core site purpouse/usability, users, behaviour etc.
  - Ideally reusable between different solutions
- The view layer
  - Interacts with the user
  - Visualize the model and communicate to the user
  - Standard HTML / JavaScript / CSS
  - Can also render JSON, XML and custom return types

Model-View-Controller (MVC)

- The controller layer
  - Acts as the traffic cop
  - Controls interaction with the layers
  - Delegates work to the model and view layers

- Separation of concerns (benefits with MVC)
  - Components are only responsible for one thing
  - Loosely coupled
  - Modular components

MVC web

- MVC is often seen in web applications where the **view** is the HTML or XHTML generated by the app
- The **controller** receives GET or POST input and decides what to do with it, handing over to domain objects (i.e. the model) that contain the business rules and know how to carry out specific tasks such as processing a new subscription
- and which hand control to (X)HTML-generating components such as templating engines, XML pipelines, Ajax callbacks, etc...
- The **model** is not necessarily merely a database; the 'model' in MVC is both the data and the business/domain logic needed to manipulate the data in the application