DBW - PHP

PHP: "PHP: Hypertext Processor"

- PHP is a **“embedded” (template-like)** language.
  - Source code is included into normal HTML pages
  - Most text is written as HTML outside PHP code.
- PHP is a full featured language, but it is used mainly in web applications
- PHP has **extensions** for PDF, Flash, Databases,…
- PHP can be **procedural** or **Object-oriented** (Most libraries can be used in both styles)
- Most usually executed as Web server sub-process (server should be configured to recognize .php files)
- Syntax is "PERL" style
  - Variables begin with '$', lines end in ';', …
- **Source code** is enclosed in

  <?php ……?>

  <script language=“php”> …..</script> (not used in general)

  <? ….. ?> for short (not always available)

  <?= ….?> shortcut for <?php print|echo … ?>

Data types

- **Primitive:**
  - Boolean, integer, float, string
  - No need to declare
  - Numerical and string types can be interpreted as boolean.
- Variable's scope is always local!!
  - Global variables must be declared explicitly (global $...), or used from $GLOBALS[…]

- **Arrays**
  - Lists and Dictionaires/Hashes use the same format
  - Creating arrays
    - $array = [1,2,3,4,5,6];
    - $array = ['key1' => 1, 'key2' => 2, …];
    - Numeric keys are used as indexes!!
  - Values can be of any type
    - $array = ['key1' => [1,2,3,4,5], 'key2'=>3];
  - Elements are recovered as $array['key']
  - $array[] = New_element goes to the end of the array.
  - No need to create elements explicitly
  - **print_r(…)** gives full structure

Predefined variables (Superglobals)

- $GLOBALS (all of them)
  - Not needed in the main script but required on functions, classes, etc.
- $_SERVER (Server set vars, from CGI protocol)
- $_GET (arguments of GET requests)
- $_POST (arguments of POST requests)
- $_COOKIE (cookies sent by the browser)
- $_FILES (uploades files)
- $_ENV (S.O. environment vars)
- $_REQUEST ($_GET and $_POST)
- $_SESSION (persistent data)

Recovering data from HTML Forms

- Variables **$_GET** or **$_POST** and **$_REQUEST** are set automatically
- Special characters are “magically” escaped (\) to avoid problems
- Multiple values can be passed from forms if name is set to “something[]”
  - <input type=“text” name=“param[]” size=“30”>
  - <select name=“values[]” multiple>…</select>

- Do not trust in “magics”, however…

Control structures

- If (cond) {Block}
- If (cond1)
  …
  elseif (cond2)
  …
  else
  …
- while (cond) {BLOCK}
- do {BLOCK} while (cond)
- switch ($var) { case … case … case …}

- for (expr1; expr2; expr3;) {BLOCK}
- foreach (expr_array as $val) {BLOCK}
- foreach (expr_array as $key => $value) {BLOCK}
  - $key and $value are copies of the original array elements (could be modified)
  - break / continue
    - (last/next from PERL!)

Functions

- function name (parameters) {BLOCK}

- Output is given by “return”

- Parameters can be passed by value (default) or by reference (indicated by &$...)

```
<?php
function headerDBW($title) {
   return "
<html lang='en'>
<head>
<title>$title</title>
</head>
<body bgcolor='#ffffff'>
<div class='container'>
<h1>DBW - <?=$title?></h1>
";
}

function footerDBW() {
   return "</div></body></html>";
}

function errorPage($err) {
   return
headerDBW($err).$err.footerDBW();
}
```

Including external code

- require/require_once external_file
    - Stops run if file is not available
- include/include_once external_file
    - Does not stop execution (may give a warning)

- _once forms avoids errors when loading libraries more than once

- To be used for code that is common to several scripts
    - Setting global variables, database connections, common functions
    - Used also to load external libraries, etc.

Sessions in Apache/PHP

- Apache web servers and PHP automatically send identification Cookies, and have a Session space assigned to every ID.
- PHP can store data variables in Session space. Those variables are available as long as user ID is identified

```
session_start ();
a) session_register(“MySessionVar");
    $MySessionVar[“user"]=$user;
    session_register(“$user");
    $user=“Pepe”;
b) $_SESSION[‘user’] = “Pepe”;
```

- Session persistence is normally based in the used of cookies.
    - If cookies are not available, PHP adds a PHPSESSIONID hidden field to all forms