



Amirkabir University of Technology
(Tehran Polytechnic)

Database Systems Fundamentals

Using PHP Language

Arman Malekzade

Amirkabir University of Technology
(Tehran Polytechnic)

Notice: The class is held under the supervision of Dr.Shiri



github.com/arman-malekzade

arman.malekzade@aut.ac.ir





Chapter 3– HTML & PHP

Roadmap :

- Hypertext Markup Language
- Hypertext Preprocessor
 - * intval() , strval()
 - * Superglobals
 - * JSON & Serialize
 - * Working with Strings
 - * File Uploads
 - * Sessions and Cookies
 - * PDO Fetch Kinds
 - * Password Hashing & Encryption
 - * Date & Time
 - * isset() vs. empty()
 - * Include & Require
 - * Some useful functions



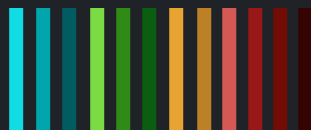
intval, strval, (string)

- General Form: `intval($var)`
- Purpose: Convert everything to an integer

<code>intval('10')</code>	10
<code>intval(10)</code>	10
<code>intval(10.2)</code>	10

- General Form: `strval($var)` or `(string)$var`
- Purpose: Convert everything to an integer

<code>strval(10)</code>	'10'
-------------------------	------





Superglobals: GLOBALS

- Superglobals:
Built-in variables that are always available in all scopes
- PHP Superglobal variables:
\$GLOBALS, \$_SERVER, \$_REQUEST, \$_POST, \$_GET, \$_FILES,
\$_ENV, \$_COOKIE, \$_SESSION
- ★ \$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

Ex: `function myfunc(){$GLOBALS['x']=2;}`

The variable is accessible from outside of the function as `$GLOBALS['x']`





Superglobals: SERVER

- ★ `$_SERVER` is a PHP super global variable which holds information about headers, paths, and script locations.

Consider the page <http://localhost/dbclass/index.php?x=1>

<code>\$_SERVER['PHP_SELF']</code>	<code>/dbclass/index.php</code>
<code>\$_SERVER['SERVER_ADDR']</code>	The IP Address of the server
<code>\$_SERVER['REQUEST_METHOD']</code>	GET
<code>\$_SERVER['REQUEST_TIME']</code>	1490386164





Superglobals: SERVER

<code>\$_SERVER['QUERY_STRING']</code>	<code>x=1</code>
<code>\$_SERVER['HTTP_HOST']</code>	localhost
<code>\$_SERVER['SERVER_PROTOCOL']</code>	HTTP/1.1
<code>\$_SERVER['HTTP_REFERER']</code>	The address of the page (if any) which referred the user agent to the current page.
<code>\$_SERVER['HTTP_USER_AGENT']</code>	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.101 Safari/537.36





Superglobals: REQUEST, GET & POST

- ★ `$_REQUEST` is an associative array that by default contains the contents of `$_GET`, `$_POST` and `$_COOKIE`.
- ★ `$_POST` is an associative array of variables passed to the current script via the HTTP POST method.
- ★ `$_GET` is an associative array of variables passed to the current script via the URL parameters.
- ★ `$_FILES` is an associative array of items uploaded to the current script via the HTTP POST method.





Superglobals: ENV, COOKIE & SESSION

- ★ `$_ENV` is an associative array of variables passed to the current script via the environment method.
- ★ `$_COOKIE` is An associative array of variables passed to the current script via HTTP Cookies.
- ★ `$_SESSION` is An associative array containing session variables available to the current script.





Superglobals: COOKIE vs. SESSION

★ Purpose:

Storing persistent data across page loads for a web visitor.

- ★ A user session (sometime referred to as a visit) is the presence of a user with a specific IP address who has not visited the site recently
- ★ Session is used to store per-user information for the current Web session on the server.
- ★ Cookie should be used to store per-user information for the current Web session or persistent information on the client.
- ★ Session variables will be expired when users close the browser. Expiration of a cookie can be set using `setcookie()`.





Working with JSON and Serialize

★ Convert Array to JSON:

```
$arr = array('name'=>'kara', 'job'=>'teacher');  
$arr_json = json_encode($arr); // {"name":"kara","job":"teacher"}
```

★ Convert JSON to Array

```
$jsd = json_decode($arr_json);  
echo $jsd->{'name'}; // 'kara'
```

★ Serialize: Generates a storable representation of a value

```
$sra = serialize($arr); // a:2:{s:4:"name";s:4:"kara";s:3:"job";s:7:"teacher";}  
$un_sra = unserialize($sra); // Array ( [name] => kara [job] => teacher )
```





Working with Strings

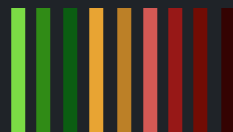
★ `strtolower()`: Make a string lowercase

★ `explode()`: Split a string by string

```
$str = 'abc def';  
$arr = explode(' ', $str);  
echo $arr[0]; // 'abc'      echo $arr[1]; // 'def'
```

★ `implode()`: Join array elements with a string

```
$arr = array('hello', 'world');  
$str = implode(' ', $arr); // 'hello world'
```





Working with Strings

- ★ `trim($str)`:

Strip whitespace (or other characters) from the beginning and end of a string

Ex: `$str=' abc'; echo trim($str); // 'abc'`

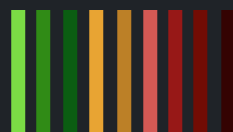
- ★ `str_replace($search, $replace, $str)`:

Replace all occurrences of the search string with the replacement string (Notice: This function is case-sensitive)

Ex: `echo str_replace("World", "Tom", "Hello World");`

- ★ `strpos()`: Find the position of the first occurrence of a substring in a string

`$str = 'hello world again world again world';`
`echo strpos($str, 'world'); // 6`





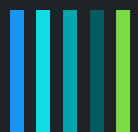
File Uploads

★ Step 1: Make a html form and receive the file through it

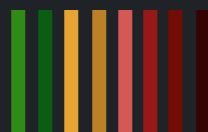
```
<form method="post" action="index.php" enctype="multipart/form-data">  
  <input type="file" name="myfile">  
  <input type="submit" name="submitfile" value="Upload">  
</form>
```

★ Step 2: Check if the file is received

```
if ( isset( $_FILES['myfile'] ) )
```



Uploading...





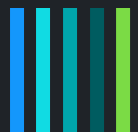
File Uploads

★ Step 3: Check if the file fits the rules

```
$file_size = $_FILES['myfile']['size']; // File Size (Bytes)
$filename = $_FILES['myfile']['name']; // pic1.jpg
$file_temp_name = $_FILES['myfile']['tmp_name'];
$file_type = $_FILES['myfile']['type']; // image/jpeg
$file_extension = end( explode('.', $filename) );
                    Array ( [0] => 02 [1] => jpg )
```

★ Step 4: Move the uploaded file

```
move_uploaded_file($file_temp_name, __DIR__ . '/uploads/' . $filename);
```



Uploading...





Working with Sessions

- ★ Step 1: Start a session

 - `session_start();` //put this code at the beginning of the page

- ★ Step 2:

 - Validate user credentials after he/she submits the login form
 - if (there exists such user in the database){

 - get his/her account details and save it in some variable

 - }

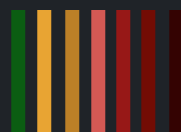
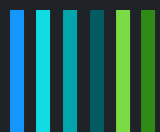
- ★ Step 3: Set the fields of `$_SESSION` (super global variable)

 - `$_SESSION['userid'] = 502;`

 - `$_SESSION['username'] = 'TomCarter96';`

- ★ Deleting a session variable

 - `session_destroy();` //unsets every session variable





Working with Cookies

★ Set a cookie

```
setcookie('name', 'Tom', time() + (86400 * 30), '/');
```

86400 = 1 day

A cookie with path '/' the cookie will be accessible via all pages in the domain

★ From the time a cookie is set, it can be accessed anywhere!

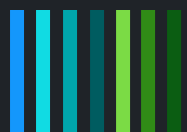
```
$_COOKIE['name']; // Contains 'Tom'. This is valid from now until the next month.
```

★ Delete a cookie

```
setcookie('name', "", time() - 3600);
```

the cookie will expire the next time the page loads.

another way: `unset($_COOKIE['name']);`





PDO: Fetch Kinds

★ fetchAll():

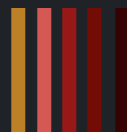
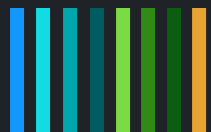
Returns an array containing all of the result set rows

★ fetchColumn():

Returns a single column from the next row of a result set

★ fetch():

Fetches the next row from a result set





Password Hashing & Encryption

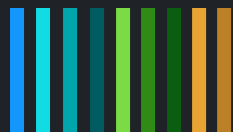
- ★ Crypt function: Each time a salt is generated automatically

```
$str = crypt('mypassword'); //$1$SFlqcAWy$Mv2iKG1LuTL8Nue3KSbc4.
```

The variable \$str can be stored in MySQL.

- ★ Check if the input is equal to the hash

```
if ( hash_equals( $str, crypt('mypassword', $str) ) ) echo 'yes';
```





Working with Date & Time

★ `time()`: Returns current Unix timestamp

```
echo time(); //1490398612
```

★ `date('Y-m-d')` // 2017-03-25

★ `date('y-m-d')` // 17-03-25

★ `date('Y-M-d')` // 2017-Mar-25

★ `date('Y-M-D')` // 2017-Mar-Sat

★ `date('H:i:s', time())` // 04:11:07

★ Set the default timezone:

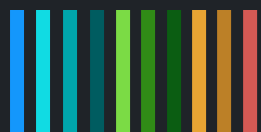
```
date_default_timezone_set("Asia/Tehran");
```





isset() vs. empty()

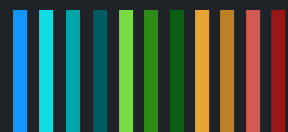
- ★ **isset():** Determine if a variable is set and is not NULL
 - `isset($_POST['submit']); // remember submitting a form...`
 - `isset($var1, $var2); // returns TRUE if (isset($var1) && isset($var2))`
- ★ **empty(\$var):**
 - A variable is considered empty if it does not exist or if its value equals FALSE.
- * Empty checks if the variable is set and if it is it checks it for null, "", 0, etc
- * Isset just checks if is it set, it could be anything not null





Include & Require

- ★ `include()`: Files are included based on the file path given
`include 'farsidate.php'; // Like including headers in C`
When a file is included with the include statement and PHP cannot find it, the script will continue to execute.
- ★ `include_once()`: If the code from a file has already been included, it will not be included again.
`include_once 'farsidate.php';`
- ★ `require()`: Files are included based on the file path given
`require 'farsidate.php'; // Like including headers in C`
When a file is included with the include statement and PHP cannot find it, the script won't continue to execute.





Some other useful functions

- ★ `header()`: Send a raw HTTP header

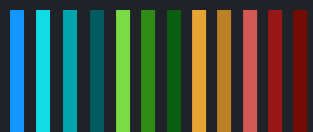
```
header("Location: http://www.example.com/"); /* Redirect browser */
```

- ★ `in_array()`: Checks if a value exists in an array

```
$arr = array(1,2,3,4,5);  
in_array(1,$arr); //TRUE
```

- ★ `file_get_contents()`: Reads entire file into a string

```
$str=file_get_contents('http://example.com/'); // <!doctype html>...
```





Some other useful functions

- ★ `file_put_contents()`: Write a string to a file

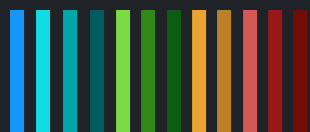
```
file_put_contents("test.txt",$str);
```

- ★ `is_dir($filename)`:

Tells whether the filename is a directory

- ★ `end($arr)`:

Advances array's internal pointer to the last element, and returns its value.

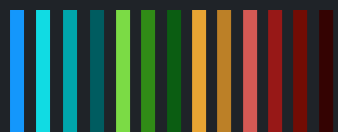




Amirkabir University of Technology
(Tehran Polytechnic)

References

- [1] PHP Cookbook - O'Reilly Media
shop.oreilly.com/product/0636920029335.do
- [2] PHP Documentation
- Retrieved Mar 25, 2017 from <http://php.net>



References