CSE417: WEB ENGINEERING

Daffodil International University

You Will Be Able To

- ✓ Work with
  - ✓ forms, cookies, files, time and date.
- ✓ Create a basic checker for user-entered data.

Contents

- Including Files, time and date
- Forms and Validation
- Cookies
- Session
- Handling Files

Including Files

The include() statement includes and evaluates the specified file.

vars.php

```php
<?php

$color = 'green';
$fruit = 'apple';

?>
```

test.php

```php
<?php

echo "A $color $fruit"; // A

include 'vars.php';

echo "A $color $fruit"; // A green apple

?>
```

```php
<?php

function foo()
{
    global $color;

    include ('vars.php');

    echo "A $color $fruit";
}

/* vars.php is in the scope of foo() so     *
 * $fruit is NOT available outside of this  *
 * scope.  $color is because we declared it *
 * as global.                                */

foo();                          // A green apple
echo "A $color $fruit";   // A green

?>
```

*The scope of variables in “included” files depends on where the “include” file is added!

You can use the include_once, require, and require_once statements in similar ways.

PHP Information

The `phpinfo()` function is used to output PHP information about the version installed on the server, parameters selected when installed, etc.

```
<html><head></head>
<!- info.php
<body>
<?php
// Show all PHP information
phpinfo();
?>
<?php
// Show only the general information
phpinfo(INFO_GENERAL);
?>
</body>
</html>
```

INFO_GENERAL	The configuration line, php.ini location, build date, Web Server, System and more
INFO_CREDITS	PHP 4 credits
INFO_CONFIGURATION	Local and master values for php directives
INFO_MODULES	Loaded modules
INFO_ENVIRONMENT	Environment variable information
INFO_VARIABLES	All predefined variables from EGPCS
INFO_LICENSE	PHP license information
INFO_ALL	Shows all of the above (default)

Server Variables

The $_SERVER array variable is a reserved variable that contains all server information.

```
<html><head></head>
<body>

<?php
echo "Referer: " . $_SERVER["HTTP_REFERER"]   . "<br />";
echo "Browser: " . $_SERVER["HTTP_USER_AGENT"] . "<br />";
echo "User's IP address: " .  $_SERVER["REMOTE_ADDR"] ;
?>

</body>
</html>
```

The $_SERVER is a super global variable, i.e. it's available in all scopes of a PHP script.

PHP Global Variables - Superglobals:
The PHP superglobal variables are:
$GLOBALS, $_SERVER, $_REQUEST, $_POST, $_GET, $_FILES, $_ENV, $_COOKIE, $_SESSION
What purpose do they serve?

Form Handling

Any form element is automatically available via one of the built-in PHP variables (provided the element has a "name" defined with it).

```
<html>
<body>
<form method="GET" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);
?>">
    <table>
        <tr>
            <td>Name:</td>
            <td><input type="text" name="name"></td>
        </tr>
    <tr>
            <td>Email:</td>
            <td><input type="text" name="email"></td>
        </tr>
     <tr>
            <td></td>
            <td><input type="submit" name="s" value="Send"></td>

        </tr>
</table>
</form>
    <?php
    if($_SERVER["REQUEST_METHOD"]=="POST"){$na =trim($_POST["name"]);
           $e =$_POST["email"];echo " name:" .$na."<br>";
        echo " email:" .$e."<br>"; }?></body></html>
```

Required Fields in User-Entered Data

A multipurpose script which asks users for some basic contact information and then checks to see that the required fields have been entered.

```
<?php
$ren=$rem='';
?>
<?php



    if($_SERVER["REQUEST_METHOD"]=="POST"){
        if(empty($_POST["name"])){
            $ren="name can't be empty";
        }
            else{
                $na =trim($_POST["name"]);
                 echo " name:" .$na."<br>";
                }

        if(empty($_POST["email"])){
            $rem="email can't be empty";
        }
        else{
            $e =$_POST["email"];
             echo " email:" .$e."<br>";
       }


    }
?>
```

Cookie Workings

`setcookie(name,value,expire,path,domain)` **creates cookies.**

```
<?php
setcookie("uname", $_POST["name"], time()+36000);
?>
<html>
<body>
<p>
Dear <?php echo $_POST["name"] ?>, a cookie was set on this
page! The cookie will be active when the client has sent the
cookie back to the server.
</p>
</body>
</html>
```

NOTE:

setcookie() **must appear BEFORE** <html> **(or any output) as it's part of the header information sent with the page.**

```
<html>
<body>
<?php
if ( isset($_COOKIE["uname"]) )
echo "Welcome " . $_COOKIE["uname"] . "!<br />";
else
echo "You are not logged in!<br />";
?>
</body>
</html>
```

`$_COOKIE`

contains all COOKIE data.

`isset()`

finds out if a cookie is set

use the cookie name as a variable

Session

- When you work with an application, you open it, do some changes, and then you close it. This is much like a Session.
  - HTTP address doesn't maintain state.
  - Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc).
  - By default, session variables last until the user closes the browser.
  - Session variables hold information about one single user
  - available to all pages in one application. ie, logged in

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

File Open

The `fopen("file_name","mode")` function is used to open files in PHP.

r	Read only.	r+	Read/Write.
w	Write only.	w+	Read/Write.
a	Append.	a+	Read/Append.
x	Create and open for write only.	x+	Create and open for read/write.

```
<?php
$fh=fopen("welcome.txt","r");
?>
```

For w, and a, if no file exists, it tries to create it (use with caution, i.e. check that this is the case, otherwise you'll overwrite an existing file).

For x if a file exists, it returns an error.

```
<?php
if
( !($fh=fopen("welcome.txt","r")) )
exit("Unable to open file!");
?>
```

If the `fopen()` function is unable to open the specified file, it returns 0 (false).

File Workings

`fclose()` **closes a file.**

`fgetc()` **reads a single character**

`fwrite(), fputs ()` **writes a string with and without \n**

`feof()` **determines if the end is true.**

`fgets()` **reads a line of data**

`file()` **reads entire file into an array**

```php
<?php
$myFile = "welcome.txt";
if (!($fh=fopen($myFile,'r')))
exit("Unable to open file.");
while (!feof($fh))
{
$x=fgetc($fh);
echo $x;
}
fclose($fh);
?>
```

```php
<?php
$myFile = "welcome.txt";
$fh = fopen($myFile, 'r');
$theData = fgets($fh);
fclose($fh);
echo $theData;
?>
```

```php
<?php
$lines = file('welcome.txt');
foreach ($lines as $l_num =>
$line)
{
 echo "Line #{$l_num}:"
.$line."<br/>";
}
?>
```

```php
<?php
$myFile = "testFile.txt";
$fh = fopen($myFile, 'a') or
die("can't open file");
$stringData = "New Stuff 1\n";
fwrite($fh, $stringData);
$stringData = "New Stuff 2\n";
fwrite($fh, $stringData);
fclose($fh);
?>
```

Getting Time and Date

`date()` and `time ()` **formats a time or a date.**

```
<!DOCTYPE html>
<html>
<body>
<?php
echo "Today is " . date("Y/m/d") . "<br>";
echo "Today is " . date("Y.m.d") . "<br>";
echo "Today is " . date("Y-m-d") . "<br>";
echo "Today is " . date("l");
?>
</body>
</html>
```

date() returns a string formatted according to the specified format.

```
<?php
$nextWeek = time() + (7 * 24 * 60 * 60);
                    // 7 days; 24 hours; 60 mins; 60secs
echo 'Now:       '. date('Y-m-d') ."\n";
echo 'Next Week: '. date('Y-m-d', $nextWeek) ."\n";
?>
```

time() returns current Unix timestamp

*Here is more on date/time formats: http://php.net/date