Working with Files

- Using copy()
- Using opendir()
- Using readdir()

Upload a File

First we need an HTML form:

```
<HTML>
<HEAD>
 <TITLE>Upload a File</TITLE>
</HEAD>
<BODY>
 <H1>Upload a File</H1>
<FORM METHOD="POST" ACTION="do_upload.php" ENCTYPE="multipart/form-data">
<p><strong>File to Upload:</strong><br>
 <INPUT TYPE="file" NAME="img1" SIZE="30"></P>
 <P><INPUT TYPE="submit" NAME="submit" VALUE="Upload File"></P>
</FORM>
</BODY>
</HTML>
```

Not nothing you haven't seen or used as of yet? Except for the ENCTYPE! So why do we have to use this? Because we are sending more than just text so we have to use the enctype attribute to specify how form-data should be encoded before sending it to the server.

The form-data is encoded to "application/x-www-form-urlencoded" by default. This means that all characters are encoded before they are sent to the server (spaces are converted to "+" symbols, and special characters are converted to ASCII HEX values). Taken on 3/4/2011from http://www.w3schools.com/tags/att_form_enctype.asp

Using the INPUT TYPE="file" will give you the browse button.

Now to create the .php script, in this case "do_upload.php". This is what the file is called in the ACTION.

```
<?
if ($_FILES["img1"] != "") {

    @copy($_FILES[img1][tmp_name],$_FILES[img1][name]) or die("Couldn't copy the file.");

} else {

    die("No input file specified");

}
?>
<HTML>
<HEAD>
<TITLE>Successful File Upload</TITLE>
</HEAD>
<BODY>
<H1>Success!</H1>

<P>You sent:<? echo $_FILES["img1"]["name"]; ?>, a <? echo $_FILES["img1"]["size"];
?> byte file with a mime type of <? echo $_FILES["img1"]["type"]; ?>.</P>

</BODY>
</HTML>
```

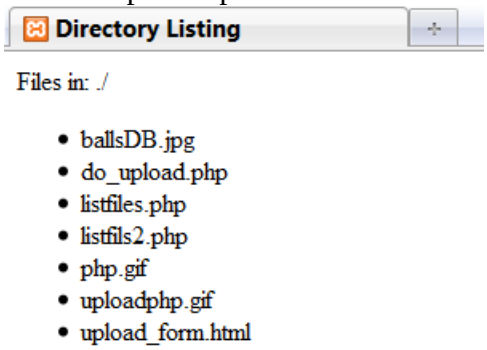
On line 4 we are using the @ symbol to suppress warnings and use the die() function that will cause the script to end and a message to display.

Also on line 4:
@copy($_FILES[img1][tmp_name],$_FILES[img1][name]) or die("Couldn't copy the file.");

$_FILES – the superglobal variable
($_FILES[img1][tmp_name] – A value that the file will be stored on the web sever as
$ _FILES[img1][name] – the name of the file being uploaded
$_FILES[img1][size] – the size of the file
$_FILES[img1][type]- the type of file such as text, jpg.

Listing a directory:

Just a simple script that uses the functions opendir() and readdir() to list a directory.



Take a look out at http://www.w3schools.com/php/func_directory_opendir.asp
I used their example with one very small modification to create the list below.

Lab 6

1. Recreate the up_load form and script
2. Using the opendir() and readdir() create a list that will display like the first list using <ul>.