AdChoices |

Today's Topics

Forums | Programming | Web Development | Computers | Tutorials | Snippets | Dev Blogs | Jobs | Lounge

#1

🔒 Login 🛛 Join!





▶ Dream.In.Code > Programming Tutorials > PHP Tutorials

(3 Pages) **▼1** 2 3 □

Uploading Images With PHP Build The Next Flickr! (DICr???)

Rate Topic: * * * * * 7 Votes

akozlik 🍙 🙇



D.I.C Addict DIC++



Posts: 797 **Joined:** 25-February 08 Intermediate Posted 10 December 2008 - 12:41 PM

There have been a lot of posts in the PHP forum regarding how to upload files to a server and put POPULAR them into a database. A lot of people were trying to put the entire binary file into the database, which was the incorrect way to do things. This tutorial is an attempt to explain how to properly upload images to your website, and store their records in the database using PHP and MySQL.

Please note that this tutorial has a lot of code. The code is well commented and will explain much of what we're trying to do. I'll give an overview in the tutorial, but you need to dig into the code to really understand how things work. Also, please note that this tutorial assumes you know some basics of PHP and MySQL and dont' have any problems writing and reading to and from a database.

The first thing we'll do is build a database. I have created a simple database named 'people', which will hold first and last names of people, as well as a picture of them. You can copy and paste this SQL statement into phpMyAdmin to build your table.

1	CREATE TABLE `people` (
2	`id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
3	`fname` VARCHAR(30) NOT NULL ,
4	`lname` VARCHAR(40) NOT NULL ,
5	`filename` VARCHAR(50) NOT NULL
6) ENGINE = MYISAM ;

Next, we will build our directory structure. We have our root folder which will hold all our directories, as well as our PHP files. Within this root directory we have a directory named 'images' and another named 'includes'. When we finish we will have a structure that looks like this

1	root-directory/images.php
2	root-directory/index.php
3	root-directory/upload.php
4	root-directory/images/
5	root-directory/includes/conn.php

Ask A Question

Join 500,000 dream.in.code® Developers



Follow & Share

Dream.In.Code













PHP Tutorials

- Basic Form Validation
- Securing login forms from brute-force attacks using queues.
- **Display Formatted** Code
- Simple Content **Formatting with Regular Expressions**
- <u>Limiting login attempts</u>
- **CodeIgniter Text** Based Mafia Game || **Login/Usersonline part**
- Adding Your Own **Functionality To Smarty w/ Plugins**
- CodeIgniter Text Based Mafia Game || Register part 2
- CodeIgniter Text **Based Mafia Game ||** Intro / Set-up Part 1

Programming Help

- ► C and C++
- ► VB.NET
- ▶ Java ► C#
- ► ASP.NET
- ► .NET Framework
- ► VB6
- ► PHP ► Rubv
- ► Pvthon
- ColdFusion
- **Databases**
- Other Languages
- Game Development Mobile Development
- 52 Weeks Of Code

Web Development

- ► Web Development
- HTML & CSS
- JavaScript



Within the includes/conn.php file, you should paste this code:

```
<?php
    // Input your information for the database
    here
    // Host name
   $host = "localhost";
   // Database username
    $username = "username";
    // Database password
12 $password = "password";
14
    // Name of database
    $database = "db name";
    $conn = mysql connect($host, $username,
    $password) or die ("Could not connect");
18 $db = mysql select db($database, $conn) or die
    ("Could not select DB");
```

On any page that connects to the database, you will require() this file. Be sure to save that file as conn.php within the includes folder.

Next we build our index file. For us, this will just be a form with three fields. One will be 'fname', one will be 'lname', and the next will be a file upload named 'image'. Please note I did some basic CSS styling, but if you don't want to style it that's fine.

```
02 // Start a session for displaying any form
    errors
   session start();
04 ?>
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0</pre>
    Strict//EN"
          "http://www.w3.org/TR/xhtml1/DTD/xhtml1
    strict.dtd">
08 <html xmlns="http://www.w3.org/1999/xhtml">
        <head>
           <title>Dream in code
    tutorial</title>
            <style type="text/css">
                label
                     float: left;
                    text-align: right;
                    margin-right: 10px;
                    width: 100px;
                     color: black;
```

- Functions that return booleans for fun and profit.
- 170 More PHP Tutorials...

▶ Graphic Design▶ Flash & ActionScript▶ Blogging

- ► SEO & Advertising
- ► Web Servers & Hosting
- ► Site Check

Reference Sheets



Code Snippets

- C Snippets
- C++ Snippets
- Java Snippets
- Visual Basic Snippets
- C# Snippets
- VB.NET Snippets
- PHP Snippets
- Python Snippets
- Ruby Snippets
- ColdFusion Snippets
- SQL Snippets
- Assembly Snippets
- <u>Functional</u><u>Programming Snippets</u>
- Perl Snippets
- HTML/CSS Snippets
- Javascript Snippets
- Flash/ActionScript Snippets
- Other Languages
 Snippets

DIC Chatroom

Join our IRC Chat

Bye Bye Ads





Make sure you notice that we are starting a session at the very top of the page. This will be handy for reporting any errors we have with the form. In addition, we have some code before the form to display any errors that are encountered. After displaying the error message we unset the error session variable.

Another thing to note is the MAX_FILE_SIZE. This can be used to ensure that the filesize of the file uploaded is less than that variable. I've left you on your own to write that portion of the script, but I promise you it is not hard.

Save that file as index.html in your root directory.

Now comes the important part. Here is the entire source file with comments. I will briefly go over how it works, but the details are within the source code:

```
01 <?php
02 // Start a session for error reporting
    session start();
    // Call our connection file
06 require ("includes/conn.php");
08 // Check to see if the type of file uploaded
    is a valid image type
 09 function is valid type($file)
10 {
        // This is an array that holds all the
    valid image MIME types
      $valid_types = array("image/jpg",
     "image/jpeg", "image/bmp", "image/gif");
      if (in array($file['type'],
     $valid types))
            return 1;
16
        return 0;
 19 // Just a short function that prints out the
     contents of an array in a manner that's easy
     to read
 20 // I used this function during debugging but
```

Alright, so we start out by starting a session and requiring our conn.php file. Nothing new there.

Next, we have a function that checks to see if the upload file is of a valid type. This function helps deter people from uploading exe files, as well as other potentially harmful stuff you don't want on your server. We have a variable that holds all the valid MIME types. That list is by no means exhaustive, but it works for our purposes. For a full list of MIME types check out the **Webmaster Tookit's list**

The next function is just something I use for debugging. This prints the contents of an array so that it's readable.

We create a constant variable that will hold our target path. I didn't use the define() function, but I could have.



Next block of code checks to make sure none of the fields were blank. This is just very basic form validation and can be adjusted for what you see fit.

The next block makes a call to the is_valid_type function and checks to make sure that the file is valid.

The next block of code checks to see if a file of that name already exists, in an effort to keep people from overwriting files. A good way to ensure a unique filename is to parse out the name part of the file name before the period, append a timestamp, and then hash it. After hashing it, add the extension back on. I'm attempting to keep this tutorial simple, so I did not show that, but you should be able to figure that out. Just use substrings.

Finally, we attempt to move the uploaded file to the new directory. Make sure the new directory has proper write permissions, or your script won't be able to save.

If moving the file was successful, we input the information into the database. Make sure you notice that we **do not input the binary file into the database**. What we have done is moved the file to a directory where we can access it, and then stored the images file name into the database. Whenever we want to print out the image we get its file name from the database and then build the img tag.

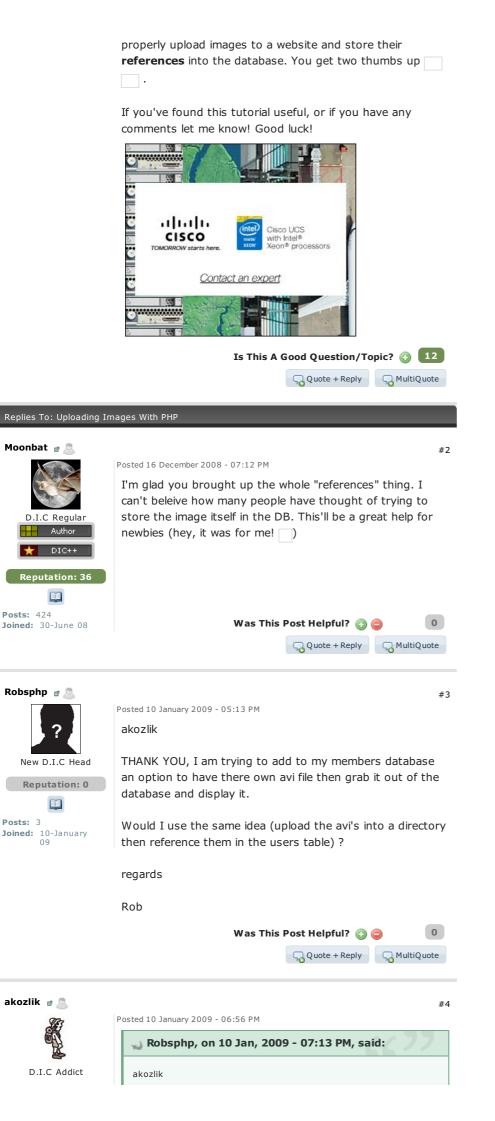
Lastly, we have our images.php page, which will display all those great images you've uploaded.

```
01 <?php
02 // Get our database connector
   require("includes/conn.php");
04 ?>
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0</pre>
    Strict//EN"
          "http://www.w3.org/TR/xhtml1/DTD/xhtml1
    strict.dtd">
08 <html xmlns="http://www.w3.org/1999/xhtml">
          <title>Dream in code tutorial - List
    of Images</title>
        </head>
        <body>
            <div>
                <?php
                    // Grab the data from our
    people table
                     $sql = "select * from
    people";
                    $result = mysql query($sql)
```

There's really nothing there that you shouldn't be able to understand.

Well, that's all I have for you! Now you know how to





Moonbat 🕝

Posts: 424

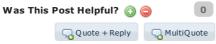
Robsphp 🕝 🙇

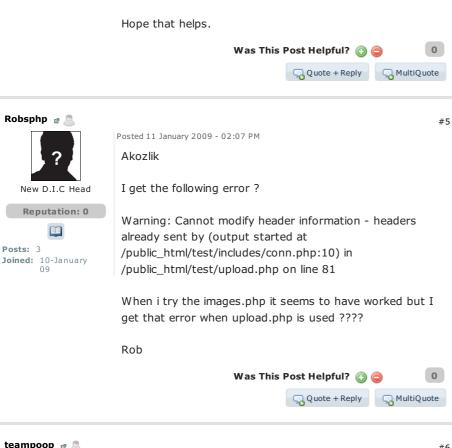
Posts: 3

akozlik 🍙 🔎



Yeah the idea's the same. I'm not sure how to best handle partially streamed videos or anything like that, but you can get the party started by using the same idea.







Hope this helps.







New D.I.C Head

Reputation: 0



Posts: 3 Joined: 10-January Posted 12 January 2009 - 03:01 AM

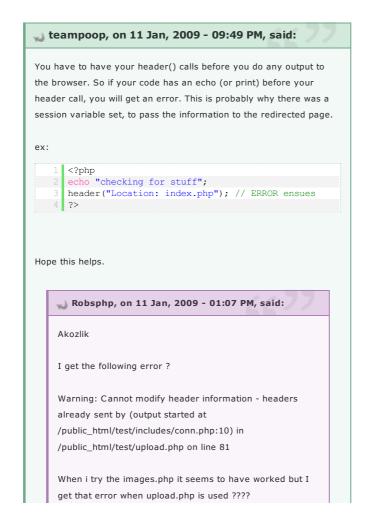
Still not working?

So does the session variable have to be cleared?

Is the code wrong?

regards

Rob







#8

akozlik 🖪 🙇



D.I.C Addict Expert

Reputation: 90



Posts: 797 Joined: 25-February Posted 12 January 2009 - 12:16 PM

The session variable does not need to be cleared.

Are you printing anything before you redirect? That includes any html tags, spaces, whatever. If you send anything to the browser, your header won't be sent.

If you start a thread in the PHP Forum and post your code, we can take a look. I'd like to keep comments on here specifically focused on the tutorials themselves. Each individual case can be handled in the thread.

Thanks and glad you like the tut!



goodmuyis 🍙 🙇



Reputation: 0



Posts: 9 Joined: 13 September 08 Posted 24 February 2009 - 02:55 AM

HelLo Mr, thanks for this stuff is just for me but am trying to change the file size but it doesn't respond to my changes, the size is it in byte or Kb(am sorry for that)

and thanks for ur stuff once again



chingkit 🍙 🙇



New D.I.C Head



Posts: 1 Joined: 09-May 09 Posted 09 May 2009 - 09:23 AM

Hi, reading this tutorial really caught my attention. I am now working on my personal website, and planning to add my own photo gallery..I try to experiment with the code you have given and I find it so cool and easy to understood to a newbie like me. I Just like ask for one part of the tutorial, because I'm a bit confused. How can I create the directory... Is it the folder which I'll be using to store all of my php files?..something like this.

root-directory/images.php

root-directory/index.php

root-directory/upload.php

root-directory/images/

root-directory/includes/conn.php

Hoping for your reply. Thanks in Advance=)

This post has been edited by **chingkit**: 09 May 2009 - 09:50 AM

#10





Posts: 1,065 Joined: 25-March 09 finished files), this is just another word for "any folder".

This post has been edited by kewlkreator: 25 September 2009 - 12:25 PM

Was This Post Helpful? 💿 😑





#15





mulson 🕝 🇷



New D.I.C Head

Reputation: 1



Posts: 11 Joined: 21 September 09

Posted 05 October 2009 - 04:56 PM

Robsphp, on 10 Jan, 2009 - 04:13 PM, said:

akozlik

THANK YOU, I am trying to add to my members database an option to have there own avi file then grab it out of the database and display it.

Would I use the same idea (upload the avi's into a directory then reference them in the users table)?

regards

Roh

i like the tutorial, but it is possible to display only one image at a time, not neccessary all...this is because i will like to dispaly only a profile picture for each user, when they login

akozlik, on 10 Dec, 2008 - 11:41 AM, said:

There have been a lot of posts in the PHP forum regarding how to upload files to a server and put them into a database. A lot of people were trying to put the entire binary file into the database, which was the incorrect way to do things. This tutorial is an attempt to explain how to properly upload images to your website, and store their records in the database using PHP and MySQL.

Please note that this tutorial has a lot of code. The code is well commented and will explain much of what we're trying to do. I'll give an overview in the tutorial, but you need to dig into the code to really understand how things work. Also, please note that this tutorial assumes you know some basics of PHP and MySQL and dont' have any problems writing and reading to and from a database.

The first thing we'll do is build a database. I have created a simple database named 'people', which will hold first and last names of people, as well as a picture of them. You can copy and paste this SQL statement into phpMyAdmin to build your table.

```
CREATE TABLE `people`
 'id' INT NOT NULL AUTO INCREMENT PRIMARY KEY ,
  fname` VARCHAR( 30 ) NOT NULL ,
 `lname` VARCHAR( 40 ) NOT NULL ,
`filename` VARCHAR( 50 ) NOT NULL
) ENGINE = MYISAM ;
```

Next, we will build our directory structure. We have our root folder which will hold all our directories, as well as our PHP files. Within this root directory we have a directory named 'images' and another

named 'includes'. When we finish we will have a structure that looks like this

```
1 root-directory/images.php
2 root-directory/index.php
3 root-directory/upload.php
4 root-directory/images/
5 root-directory/includes/conn.php
```

Within the includes/conn.php file, you should paste this code:

On any page that connects to the database, you will require() this file. Be sure to save that file as conn.php within the includes folder.

Next we build our index file. For us, this will just be a form with three fields. One will be 'fname', one will be 'lname', and the next will be a file upload named 'image'. Please note I did some basic CSS styling, but if you don't want to style it that's fine.

```
01 <?php
   // Start a session for displaying any form
   errors
   session_start();
04 ?>
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0</pre>
   Strict//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
   strict.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml">
       <head>
           <title>Dream in code tutorial</title>
            <style type="text/css">
                label
                    float: left;
                   text-align: right;
                   margin-right: 10px;
                   width: 100px;
                   color: black;
                #submit
                    float: left;
                    margin-top: 5px;
                   position: relative;
                    left: 110px;
                #error
```



Make sure you notice that we are starting a session at the very top of the page. This will be handy for reporting any errors we have with the form. In addition, we have some code before the form to display any errors that are encountered. After displaying the error message we unset the error session variable.

Another thing to note is the MAX_FILE_SIZE. This can be used to ensure that the filesize of the file uploaded is less than that variable. I've left you on your own to write that portion of the script, but I promise you it is not hard.

Save that file as index.html in your root directory.

Now comes the important part. Here is the entire source file with comments. I will briefly go over how it works, but the details are within the source code:

```
<?php
   // Start a session for error reporting
   session start();
   // Call our connection file
06 require("includes/conn.php");
08 // Check to see if the type of file uploaded is
   a valid image type
   function is_valid_type($file)
10 {
       // This is an array that holds all the valid
   image MIME types
       $valid_types = array("image/jpg",
   "image/jpeg", "image/bmp", "image/gif");
   if (in_array($file['type'], $valid_types))
           return 1;
       return 0;
   // Just a short function that prints out the
   contents of an array in a manner that's easy to
   read
20 // I used this function during debugging but it
   serves no purpose at run time for this example
   function showContents($array)
       echo "";
24
    print_r($array);
       echo "";
```

Alright, so we start out by starting a session and requiring our conn.php file. Nothing new there.

Next, we have a function that checks to see if the upload file is of a valid type. This function helps deter people from uploading exe files, as well as other potentially harmful stuff you don't want on your server. We have a variable that holds all the valid MIME types. That list is by no means exhaustive, but it works for our purposes. For a full list of MIME types check out the **Webmaster Tookit's list**

The next function is just something I use for debugging. This prints the contents of an array so that it's readable.

We create a constant variable that will hold our target path. I didn't use the define() function, but I could have.

Next block of code checks to make sure none of the fields were blank. This is just very basic form validation and can be adjusted for what you see fit.



The next block makes a call to the is_valid_type function and checks to make sure that the file is valid.

The next block of code checks to see if a file of that name already exists, in an effort to keep people from overwriting files. A good way to ensure a unique filename is to parse out the name part of the file name before the period, append a timestamp, and then hash it. After hashing it, add the extension back on. I'm attempting to keep this tutorial simple, so I did not show that, but you should be able to figure that out. Just use substrings.

Finally, we attempt to move the uploaded file to the new directory. Make sure the new directory has proper write permissions, or your script won't be able to save.

If moving the file was successful, we input the information into the database. Make sure you notice that we **do not input the binary file into the database**. What we have done is moved the file to a directory where we can access it, and then stored the images file name into the database. Whenever we want to print out the image we get its file name from the database and then build the img tag.

Lastly, we have our images.php page, which will display all those great images you've uploaded.

```
01
   <?php
   // Get our database connector
   require("includes/conn.php");
   ?>
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0</pre>
   Strict//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
   strict.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml">
       <head>
           <title>Dream in code tutorial - List of
   Images</title>
       </head>
       <body>
               <?php
                   // Grab the data from our people
   table
                   $sql = "select * from people";
                   $result = mysql_query($sql) or
   die ("Could not access DB: " . mysql_error());
                   while ($row =
   mysql_fetch_assoc($result))
   class=\"picture\">";
                       echo "";
```

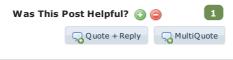
There's really nothing there that you shouldn't be able to understand.

Well, that's all I have for you! Now you know how to properly upload images to a website and store their **references** into the database. You get two thumbs up

If you've found this tutorial useful, or if you have any comments let me know! Good luck!



i like the tutorial,but it is possible to display only one image at a time,not neccessary all...this is because i will like to dispaly only a profile picture for each user,when they login.



 \square Previous Topic | **PHP Tutorials** | Next Topic \square

(3 Pages) **▼1** 2 3 □

Cloud Database System

trackvia.com/Cloud-Database

Create Your Own Cloud Database. Easy, Powerful & 100% Free to Try!



Related PHP Topics beta

- Creating A File Upload Form With PHP
- Help On Uploading Images With Php Please
- Uploading Images With GIF
 Extension
- Image Uploading In PHP
- Fckeditor With Php How To Use
 Fck Editor With Php
- Tpl Files With Php How To Map
 Tpl Files With Php
- Jquery-charts In Combination With PHP - How To Draw An Easy
 Jqearychart In Combination With PHP?
- How To Connect Crystal Reports
 With PHP? Hi,I Wanted To Know
 How We Can Connect Crystal
 Reports With Php. Actua
- Store Image Or Url In The Database+ Php
- Replace Pdf With Another Pdf

FAQ | Team Blog | Feedback/Support | Advertising | Terms of Use | Privacy Policy | About Us

Copyright 2001-2014 MediaGroup1 LLC, All Rights Reserved A MediaGroup1 LLC Production - Version 6.0.2.1.36
Server: secure3

