

PHP Email Validation and File Redirection

What makes up a Valid Address

- An email address can be broken up into four parts:

Local-part@subdomain.domain.tld

- Cannot be an empty or zero length string
- Domain portion cannot contain any of the following characters:
~ ' ^ ` " * + = \ |] [() { } \$ & ! # % / : , ; (spaces)
- NOTE: the local part (i.e. the part before the @ symbol) can
- Valid address cannot have a '.' in last or second last position
- There must be one, and only one, @ symbol in the address
- There cannot be a '.' Immediately following the '@' symbol
- There must be at least one '.' after the '@' symbol (just not immediately after it)
- Two periods ('.') cannot be side-by-side in the address
- There cannot be more than four characters after the last period in the address (or less than two, see fourth rule)

PHP Email Validation

- The majority of scripting or programming languages require that you code a "email validator"
 - Something that accepts a String argument and return a boolean depending on whether the string passed conform to the valid email rules
- PHP, as its sole purpose is for use on the web, provides a predefined function:
`filter_var()`

filter_var()

- The `filter_var()` function can "filter" out several things
- For the purposes of this course, it shall be used to "filter" invalid email address strings
- The syntax is:

```
filter_var($a_string_arg, FILTER_VALIDATE_EMAIL);
```
- This will either return a boolean false (if the string argument is an invalid email), or the string argument itself (if the string is a valid email address)
- **NOTE:** the `FILTER_VALIDATE_EMAIL` is a pre-defined constant, that tells the function what it is filtering
- See: <http://ca.php.net/manual/en/function.filter-var.php>

Simple filter_var() Example

```
<?php
```

```
//comes from http://www.w3schools.com/PHP/func\_filter\_var.asp
```

```
if(!filter_var("someone@example.com",  
              FILTER_VALIDATE_EMAIL))
```

```
{  
    echo("E-mail is not valid");
```

```
}
```

```
else
```

```
{
```

```
    echo("E-mail is valid");
```

```
}
```

```
?>
```

Realistic filter_var() Example

```
<?php
$error = ""; //an error message variable, initially empty
//embed in a page with an input box named 'email_address' and
//have the form submit here in POST mode
$email = trim($_POST['email_address']); //trim whitespace
if(strlen($email) == 0){
    //use strlen() to det. whether the var empty or not
    $error .= "You must enter an email address";
}else if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
    //note the ! not symbol, if valid no problem
    $error .= "<em>". $email . "</em> is not a valid email address";
    //NOTE: give the invalid address so user knows what they entered
    $email = ""; //empty out the variable so it is not sticky
}
?>
```

File/Page Redirection

- There are situations on the Web when you need to automatically redirect a user/visitor to another web page
 - Change in URL
 - Page done processing
 - Unauthorized attempt to access protected pages
- Due to these situations, PHP provides a predefined function that redirects a page:

```
header();
```

header()

- This function can send any raw HTTP header
 - This includes handling page exceptions
- Most popular use is to redirect a user from the executing page to a new page/location
- NOTE: that the `header()` function will fail if you have already called an echo before the `header("Location:...");` call
 - Error returned is: **Warning:** Cannot modify header information - headers already sent by... followed by a file name and line number the echo occurred
- See: <http://ca3.php.net/manual/en/function.header.php>

header() Error Work Around

- The work around to the redirect error is to empty the buffer (of http header data), to do this in the header.php file, at the very top of the page type the following:

```
<?php
    require "../includes/functions.php";
        //already there from assignment
    ob_start();
?>
```

- `ob_start();` //turns on output buffering
- Then when you make the header() function re-direct call:

```
<?php
    ob_flush();
    header("Location:lab9_login.php");
?>
```

- `ob_flush();` // flush (send) the output buffer

Simple header() Example

```
<?php
// this comes from:
http://www.w3schools.com/php/func\_http\_header.asp
header('Location: http://www.example.com');
    //absolute addresses work but...
header('Location: index.php');
    //so do relative addresses
?>
```

If you ever get the error when a page is executing the header function:

Warning: Cannot modify header information - headers already sent

Means the page has already started HTML output above the call or a PHP echo was already executed

Realistic header() Example

```
<?php
require("../includes/function.php");
                //need file for db stuff

ob_start();
include("../header.php");
$errors = "";    //initialize empty string for errors
//Do some error checking, concatenate error messages
//to $errors variable (no echoes!)
if(strlen($errors) == 0){
    //$errors is still empty, which means there are no //errors
after error checking
    //do database stuff like INSERT a new record
    ob_flush();
    header("Location:lab9.php");
    //send the user to a new page, we are done here
}
?>
```