



جامعة الانبار
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم أنظمة شبكات الحاسوب

برمجة صفحات الويب PHP
المرحلة الثالثة
الفصل الدراسي الاول

مدرس المادة
م.د. سميه عبدالله حمد

Web Programming Lectures (PHP)

Lectures:

١. lec١: PHP Fundamentals (اساسيات برمجة صفحات الانترنت)
٢. lec٢: PHP Data Types (أنواع البيانات)
٣. lec٣: PHP Arrays (المصفوفات)
٤. lec٤: PHP Logic Control Structures (عبارات السيطرة)
٥. lec٥: PHP Loops (عبارات التكرار)
٦. lec٦: PHP Strings PHP String Functions (السلاسل ودوالها)
٧. lec٧: PHP Function (الدوال)
٨. lec٨: PHP Date() & Time Function (دوال الوقت والتاريخ)
٩. lec٩: Regular Expressions (التعابير المنتظمة)
١٠. lec١٠: PHP File Handling & Functions (دوال وتحميل الملفات)
١١. lec١١: PHP Session & PHP Cookies (الجلسات وملفات تعريف الارتباط)
١٢. lec١٢: How to Send Email using PHP mail (كيفية ارسال الايميل)

2- How to Send Email using PHP mail() Function

What is PHP mail?

PHP mail is the built in PHP function that is used to send emails from PHP scripts.

The mail function accepts the following parameters:

- Email address
- Subject
- Message
- CC or BC email addresses
- It's a cost effective way of notifying users on important events.
- Let users contact you via email by providing a contact us form on the website that emails the provided content.
- Developers can use it to receive system errors by email
- You can use it to email your newsletter subscribers.
- You can use it to send password reset links to users who forget their passwords
- You can use it to email activation/confirmation links. This is useful when registering users and verifying their email addresses

Why/When to use the mail PHP

Sending mail using PHP

The PHP mail function has the following basic syntax

```
<?php  
mail($to_email_address,$subject,$message,[$headers],[$parameters])  
;  
?>
```

HERE,

- “\$to_email_address” is the email address of the mail recipient
- “\$subject” is the email subject
- “\$message” is the message to be sent.
- “[\$headers]” is optional, it can be used to include information such as CC, BCC

Web Programming Lectures (PHP)

- CC is the acronym for carbon copy. It's used when you want to send a copy to an interested person i.e. a complaint email sent to a company can also be sent as CC to the complaints board.
- BCC is the acronym for blind carbon copy. It is similar to CC. The email addresses included in the BCC section will not be shown to the other recipients.

Simple Mail Transmission Protocol (SMTP)

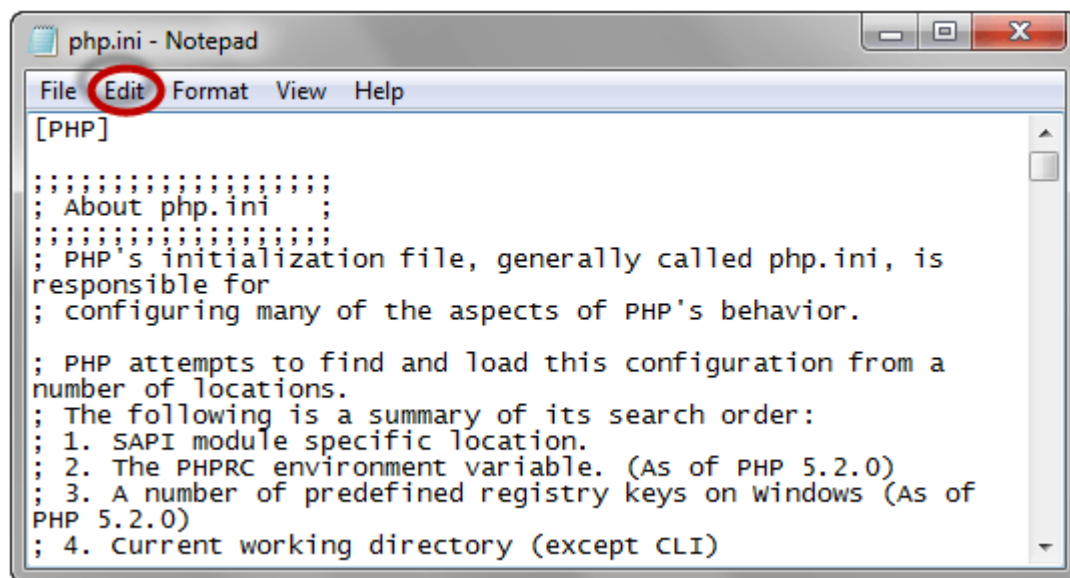
PHP mailer uses Simple Mail Transmission Protocol (SMTP) to send mail.

On a hosted server, the SMTP settings would have already been set.

The SMTP mail settings can be configured from “php.ini” file in the PHP installation folder.

Configuring SMTP settings on your localhost Assuming you are using xampp on windows, locate the “php.ini” in the directory “C:\xampp\php”.

- Open it using notepad or any text editor. We will use notepad in this example. Click on the edit menu

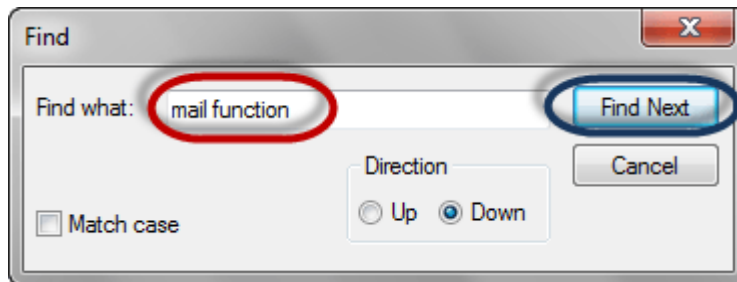


- Click on Find... menu

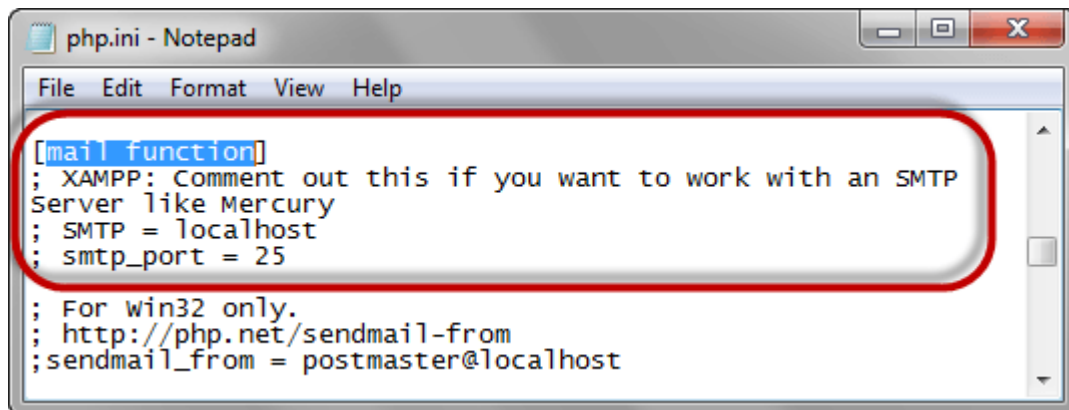
Web Programming Lectures (PHP)

Undo	Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Del
Find...	Ctrl+F
Find Next	F3
Replace...	Ctrl+H
Go To...	Ctrl+G
Select All	Ctrl+A
Time/Date	F5

- The find dialog menu will appear



- Click on Find Next button



- Locate the entries
- `[mail function]`
- `; XAMPP:` Don't remove the semi column if you want to work with an SMTP Server like Mercury
- `; SMTP = localhost`
- `; smtp_port = 25`

Web Programming Lectures (PHP)

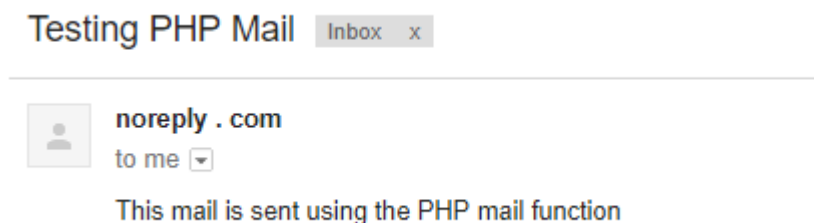
- Remove the semi colons before SMTP and smtp_port and set the SMTP to your [smtp server](#) and the port to your smtp port. Your settings should look as follows
- SMTP = smtp.example.com
- smtp_port = 25
- *Note the SMTP settings can be gotten from your web hosting providers.*
- If the server requires authentication, then add the following lines.
- auth_username = example_username@example.com
- auth_password = example_password
- Save the new changes.
- Restart [Apache](#) server.

PHP Mail Example

Let's now look at an example that sends a simple mail.

```
<?php
$to_email = 'name @ company . com';
$subject = 'Testing PHP Mail';
$message = 'This mail is sent using the PHP mail function';
$headers = 'From: noreply @ company . com';
mail($to_email,$subject,$message,$headers);
?>
```

Output:



Note: the above example only takes the 4 mandatory parameters.

You should replace the above fictitious email address with a real email address.

Sanitizing email user inputs

The above example uses hard coded values in the source code for the email address and other details for simplicity.

Web Programming Lectures (PHP)

Let's assume you have to create a contact us form for users fill in the details and then submit.

- Users can accidentally or intentional inject code in the headers which can result in sending spam mail
- To protect your system from such attacks, you can create a custom function that sanitizes and validates the values before the mail is sent.

Let's create a custom function that validates and sanitizes the email address using the filter_var built in function.

Filter_var function The filter_var function is used to sanitize and validate the user input data.

It has the following basic syntax.

```
<?php
filter_var($field, SANITIZATION TYPE);
?>
```

HERE,

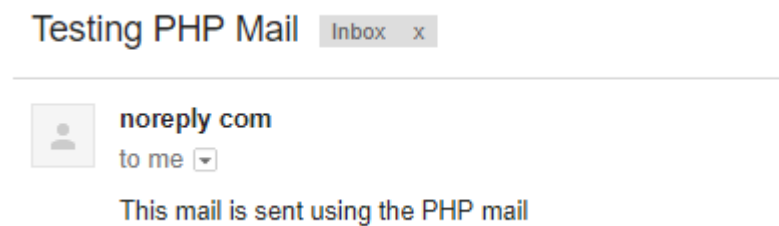
- “filter_var(…)” is the validation and sanitization function
- “\$field” is the value of the field to be filtered.
- “SANITIZATION TYPE” is the type of sanitization to be performed on the field such as;
- **FILTER_VALIDATE_EMAIL** – it returns true for valid email addresses and false for invalid email addresses.
- **FILTER_SANITIZE_EMAIL** – it removes illegal characters from email addresses. info\@domain.(com) returns info@domain.com.
- **FILTER_SANITIZE_URL** – it removes illegal characters from URLs. http://www.example@.comé returns >http://www.example@.com
- **FILTER_SANITIZE_STRING** – it removes tags from string values. am bold becomes am bold.

The code below implements uses a custom function to send secure mail.

```
<?php
function sanitize_my_email($field) {
    $field = filter_var($field, FILTER_SANITIZE_EMAIL);
    if (filter_var($field, FILTER_VALIDATE_EMAIL)) {
        return true;
    } else {
        return false;
    }
}
```

```
}  
}  
$to_email = 'name @ company . com';  
$subject = 'Testing PHP Mail';  
$message = 'This mail is sent using the PHP mail ';  
$headers = 'From: noreply @ company. com';  
//check if the email address is invalid $secure_check  
$secure_check = sanitize_my_email($to_email);  
if ($secure_check == false) {  
    echo "Invalid input";  
} else { //send email  
    mail($to_email, $subject, $message, $headers);  
    echo "This email is sent using PHP Mail";  
}  
?>
```

Output:



Secure Mail

Emails can be intercepted during transmission by unintended recipients.

This can expose the contents of the email to unintended recipients.

Secure mail solves this problem by transmitting emails via Hypertext Transfer Protocol Secure (HTTPS).

HTTPS encrypts messages before sending them.

Summary

- The PHP built-in function `mail()` is used to send mail from PHP scripts
- Validation and sanitization checks on the data are essential to sending secure mail
- The PHP built-in function `filter_var()` provides an easy-to-use and efficient way of performing data sanitization and validation