



PHP Guide Book

Moe Myint Shein

PHP GUIDE BOOK

Part (I)

Moe Myint Shein

အမှာစကား

မင်္ဂလာပါခင်ဗျာ။ အချိန်လေးသေချာပေးပြီးရေးလိုက်တော့ အခုလောက်ထိပြီးသွားပါတယ်...။ ကျနော် ကတိပေးထားခဲ့တာတွေ များနေပါပြီ။ ကျနော်ရေးနေတယ်ပြောထားလို့ လာမေးတဲ့သူတွေလည်း များလှပါပြီ။ ဒါကြောင့် အခုလောက်ထိပြီးသလောက်ကို Part (I) လို့ပဲအမည်တပ်ပြီး ထုတ်လိုက်ရပါတယ်။ ဒီမှာ မပါတာတွေ မပြည့်စုံတာတွေနဲ့ ထပ်ပြီးပါသင့်တာတွေကို ကျနော်နောက်စာအုပ်မှာ ထည့်ရေးသွားပါမယ်။ ရလာမယ့် အကြံ ပြုချက်တွေကိုလည်း မျှော်လင့်ပါတယ်။ အရင်တုန်းက ကျနော် Joomla install လုပ်တာနဲ့ google apps နဲ့ ဒီမိုင်းချိတ်တာရေးဖူးပါတယ်။ ပြီးတော့ Joomla နဲ့ပတ်သက်ပြီး စာအုပ်လေးရေးဖို့ စီစဉ်ခဲ့ပေမယ့် စိတ်ပြောင်းသွားခဲ့ပါတယ်။ Joomla ကလွယ်နေလို့ရယ် မြန်မာလို စာအုပ်ရေးရလောက်အောင်တော့ တစ်မျိုးဖြစ်နေလို့ ပိုအကျိုးရှိမယ့် PHP ကိုရွေးလိုက်ပါတယ်။ အဲဒီအတွက်လည်း ရေးမယ်ထင်ပြီးစောင့်နေခဲ့တဲ့ သူများကိုအားနာမိပါတယ်။ အခု Part (I) ဆိုတော့ Part (II) ကိုတော့ အကြာကြီးမစောင့်စေပါဘူး။ သေချာ test လုပ် edit လုပ်ဖို့ကျန်နေလို့ပါ။ ဒီရက်ပိုင်းမှာ အချိန်ပေးပြီးလုပ်နေတာဖြစ်ပါတယ်။ PHP Guide Book Part (II) ကို 2010 Aug, 13th မှာ www.moemyintshein.com မှာ လာရောက်ဒေါင်းလုပ်ရယူနိုင်မှာ ဖြစ်ပါတယ်။ အဲဒီအခါတော့ နှစ်အုပ်ပေါင်းပြီးတော့လည်း Pdf တစ်ဖိုင်တည်းပေါင်းပြီး အပြည့်တစ်အုပ်ပါလုပ်ထားမှာပါ။

ဒီစာအုပ်နဲ့လေ့လာဖို့ Programming အခြေခံရှိတာ အလွန်ကောင်းပါသည်။ မရှိပါက ဖြည်းဖြည်းချင်း ဖတ်ပါ။ code တိုင်းကို run ပါ လို့အကြံပြုပါရစေ။ ဝန်ခံချင်တာတွေကတော့ code တွေကို w3schools.com ကနေ reference လုပ်ပြီးသုံးထားပါတယ်။ အများစုက ဘာသာပြန်တာဖြစ်ပါတယ်။ နားလည်မယ်လို့တော့ မျှော်လင့်မိပါတယ်။ တစ်စုံတစ်ရာ အဆင်မပြေတာ ရှိခဲ့သော် ကျနော်ညွှန်လို့သာဖြစ်ကြောင်း။ ရေးနေစဉ်မှာ အားပေးခဲ့သူများအားလုံးကို ကျေးဇူးတင်ပါတယ်။ အားလုံးသော ဒေါင်းလုပ်ဆွဲကြသူ ဖတ်ရှုအားပေး လေ့လာကြသူ ညီအစ်ကို မောင်နှမ များ အားလုံးကိုလည်း ကျေးဇူးတင်ပါတယ်။ စကားမစပ် ကျနော် စာရေးဆရာမဟုတ်ပါ။ ကူညီချင်တဲ့ စိတ်သက်သက် နဲ့ရေးတာ ဖြစ်ပါတယ်။ နောက်လည်း စာတွေအများကြီး ရေးဖို့ အစီစဉ်မရှိပါခင်ဗျာ။ တတ်နိုင်သလောက်တော့ ကူညီသွားမှာပါ။ အားလုံးကိုကျေးဇူးတင်ပါတယ်။ အကြံပြုချက်များကို အီးမေး moemyintsheinster@gmail.com သို့ပေးပို့နိုင်ပါတယ်။

လေ့လာကြသူအားလုံးကို လေးစားလျက်

မိုးမြင့်ရှိန်

INTRODUCTION TO PHP

What is PHP?

PHP ဆိုတာ dynamic web pages တွေအတွက်ရည်ရွယ်ပြီးပေါက်လာခဲ့တဲ့ scripting language ဖြစ်ပါတယ်။ PHP ကို web scripts တွေရေးဖို့ဖန်တီးခဲ့တာသာဖြစ်ပါတယ်။ stand alone application တွေရေးဖို့မဟုတ်ပါဘူး (ဒါပေမယ့် အခုအချိန်မှာတော့ application တွေကိုလည်း PHP နဲ့ရေးလို့ရနေပါပြီ။) PHP ကို မူလပထမဆုံး ဖန်တီးခဲ့သူကတော့ Rasmus Lerdorf ပဲဖြစ်ပါတယ်။ ၁၉၉၄ ခုနှစ်မှာ PHP ရဲ့အရည်က Personal Home Page လို့ Rasmus Lerdorf ကခေါ်တွင်ခဲ့ပါတယ်။ နောက်ပိုင်းမှာတော့ အခုသိကြတဲ့အတိုင်း PHP: Hypertext Preprocessor ဖြစ်ပါတယ်။ PHP ဟာယနေ့အခါမှာတော့ အသုံးအများဆုံး programming language တစ်ခုအဖြစ်ရပ်တည်နေပါပြီ။ အခုအခါမှာ PHP ကို THE PHP GROUP ကနေ maintain လုပ်ပေးနေပါတယ်။ ဒါ့အပြင် PHP ဟာ PHP License ရဲ့အောက်မှာ ထွက်ပေါ်တဲ့ free software ဖြစ်ပါတယ်။ PHP ဟာ general-purpose scripting language တစ်ခုဖြစ်ပေမယ့် အထူးသဖြင့်တော့ web development အတွက်အဓိကတွင်တွင်ကျယ်ကျယ်အသုံးပြုကြပါတယ်။ သူ့ကို HTML coding တွေအထဲမှာ ထည့်သွင်းအသုံးပြုနိုင်ပြီး web server ပေါ်မှာ run တာဖြစ်ပါတယ်။ web servers အများစုမှာ အသုံးပြုနိုင်ပြီးတော့ Operating System တွေနဲ့ Platform တွေအားလုံးနီးပါးမှာ အခကြေးငွေပေးစရာမလိုပဲ ရယူ အသုံးပြုနိုင်ပါတယ်။ အခုအချိန်မှာ PHP ကို websites ပေါင်း သန်း ၂၅ ကျော် နဲ့ web server ၁ သန်းကျော်ပေါ်မှာ install လုပ်ကာအသုံးပြုနေကြပြီး ဒီအရေအတွက်ဟာ နေ့စဉ်နဲ့အမျှတိုးလို့လာနေပါတယ်။ သူ့ရဲ့ syntax ဟာ Perl နဲ့ C တို့နဲ့ဆင်တူပါတယ်။ operating system အတော်များများပေါ်မှာ Apache (web server) နဲ့ အတူတူ တွဲပြီး အသုံးပြုကြပါတယ်။ Windows ပေါ်မှာလည်း Microsoft ရဲ့ IIS နဲ့လည်းယှဉ်တွဲအသုံးပြုနိုင်ပါတယ်။ official current release ကတော့ PHP 5.3.2 (March 4, 2010) ဖြစ်ပါတယ်။ အခုဒီတစ်ခုရောက်ရှိနေတဲ့ latest preview release ကတော့ 5.3.3 rc3 (July 15, 2010) ဖြစ်ပါတယ်။ PHP 4 ကနေ PHP 5 ကိုပြောင်းလာတဲ့အခါမှာ object-oriented programming language ဖြစ်လာခဲ့ပါတယ်။ အခုအခါမှာတော့ ကမ္ဘာပေါ်မှာရှိတဲ့ ဘာသာစကားအားလုံးနီးပါး ရဲ့ အကွာရာ တွေကို ကိုင်တွယ်နိုင်ပြီလို့ဆိုနိုင်ပါတယ်။ PHP ဟာ သူ့ရဲ့ ယှဉ်ပြိုင်ဘက် language တွေထက် ပိုကောင်းတယ် ပိုမြန်တယ် သင်ယူဖို့ပိုမို လွယ်ကူတယ် လို့ အကြမ်းအားဖြင့် ပြောနိုင်ပါတယ်။ database အတော်များများနဲ့ယှဉ်တွဲအသုံးပြုနိုင်ခြင်း ပေါ့ပါးခြင်း အကန့် အသန့် မရှိဘူးလို့ ပြောနိုင်လောက်အောင် ချဲ့ထွင်အသုံးပြုနိုင်ခြင်း open source ဖြစ်သောကြောင့် ကုန်ကျစရိတ်မရှိခြင်း စတဲ့ကောင်းကျိုး များစွာ ရှိနေ ပါတယ်။ ကျွန်တော် PHP စာအုပ်ကိုအခုရေးပေးမယ့် JSP, ROR နဲ့ ASP.net တို့ကိုလည်း လေ့လာနေတာ ပဲဖြစ်ပါတယ်။ တစ်ခုစီတိုင်းမှာတော့ အားသာချက် အားနည်းချက်ကိုယ်စီရှိကြတာမျိုးဖြစ်ပါတယ်။ ဘယ်အရာကမှ ရာနှုန်းပြည့်ကောင်းမွန်တယ်လို့မဆိုနိုင်ပါဘူး...

PHP အလုပ်လုပ်ပုံ

PHP သည် server-side language ဖြစ်ပါတယ်။ ဆိုလိုတာက ကျွန်တော်တို့ရေးလိုက်တဲ့ code တွေဟာ server ပေါ်မှာ run တာဖြစ်ပါတယ်။ server-side language တွေက web page တွေဆီကနေ ပို့ပေးလိုက်တဲ့ အချက်အလက်တွေကိုရယူပါတယ်။ host ရဲ့ server ဆီကို ရောက်လာမယ်။ ရောက်လာတဲ့ information ပေါ်မှာမူတည်ပြီးတော့ server ပေါ်မှာ run ပါတယ်။ ပြီးမှ browser ဆီကို လိုအပ်တဲ့အချက်အလက်ကိုပြန်ပို့ပေးပါတယ်။ ဥပမာ ကျွန်တော်တို့က web page ပေါ်မှာ click တစ်ချက်နှိပ်လိုက်ရင် ကျွန်တော်တို့ အဲဒီ click နှိပ်လိုက်ခြင်းကြောင့် ရလာတဲ့ result ကိုမမြင်ရခင်မှာ server ဆီကို ကျွန်တော်တို့ရဲ့ Input ပေးလိုက်တဲ့ အချက်အလက်တွေကို ပို့တာ ရယ် server ကနေ ပြန်ပို့တာတွေ လုပ်နေတာဖြစ်ပါတယ်။ဒါက server-side ရဲ့သဘောပါ။

client-side language ကတော့ ကျွန်တော်တို့ user တွေအသုံးပြုတဲ့ client ပေါ်မှာပဲ တိုက်ရိုက် run တာပါ။ JavaScript ရဲ့ client ကတော့ web browser ပေါ့။ ဒါကြောင့် သူ့အနေနဲ့ web server ဆီကို information ပို့တာ ပြန်ထုတ်တာတွေ လုပ်ဖို့မလိုပါဘူး။ သဘောတရားကတော့ browser က code ကိုဖတ်တယ် ပြီးရင် interprets လုပ်တယ်။ ပြီးရင် ပြန်ထုတ်ပြတယ်။ မြန်မြန်ဆန်ဆန် ပြီးပါတယ်။ သူ့နေရာနဲ့ သူတော့ client-side ရော server-side ရော လိုအပ်ပါတယ်။ ဥပမာ form တစ်ခု ဆိုပါစို့ဗျာ။ ကျွန်တော် တို့ sign up လုပ်တဲ့အခါဖြစ်ဖြစ်ပေါ့ ကျွန်တော်တို့ text box ထဲမှာထည့်လိုက်တဲ့ data တွေကိုမှန် မမှန် ကို client side language တစ်ခုနဲ့စစ်မယ်။ valid ဖြစ်မဖြစ်ကိုပေါ့။ ပြီးမှ submit ကိုဖြစ်ဖြစ်နှိပ်လိုက်မှ

server ဆီကိုပို့မှာပေါ့။ server ဆီကိုမှန်ကန်တဲ့ information တွေပဲရောက်ရှိစေတာပေါ့ဗျာ။ web server ပေါ်မှာ အချက်အလက်တွေကို save လုပ်ပြီး သိမ်းထားဖို့လိုအပ်တာမျိုး ကြတော့ server side နဲ့ပဲလုပ်လို့ရတာပေါ့။

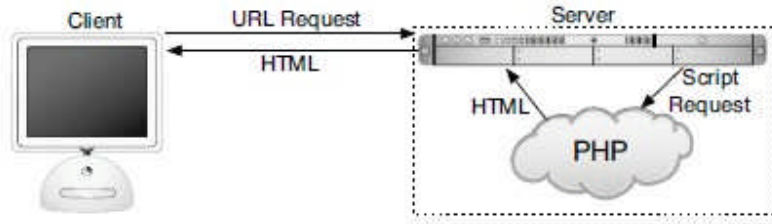


Fig 1.1 PHP ကိုအသုံးပြုထားသော server ထဲမှ user က web page ကို request လုပ်ပုံ
 PHP ကိုလေ့လာတော့မယ်ဆိုရင် Apache web server နဲ့ My SQL Database တို့ပါတွဲပြီးပါနေတာဖြစ်ပါတယ်။
 အခုကျွန်တော်တို့ PHP ကိုစတင်လေ့လာဖို့အတွက် လိုအပ်တဲ့ software ကိုသွင်းရပါမယ်။

localhost မှာသုံးဖို့အတွက် လိုအပ်တဲ့ software တွေအားလုံးကို ပေါင်းစုပြီးထည့်ထားတဲ့ package software တွေရှိပါတယ်။
 အဲဒီ software တွေကိုသုံးပြီး web server နဲ့ database ကိုကိုယ်စက်ထဲမှာပဲရရှိမှာဖြစ်ပါတယ်။
 အဲဒီ software တွေထဲမှာ windows အတွက်ထင်ရှားတာတွေကတော့ WAMP နဲ့ XAMPP တို့ပဲဖြစ်ကြပါတယ်။
 သူတို့မှာ apache, mysql, php စတဲ့ လိုအပ်တာတွေအားလုံးပါဝင်တာဖြစ်ပါတယ်။
 ကျွန်တော်တို့ အခု XAMPP ကိုပဲရွေးပြီးတော့ လေ့လာကြပါမယ်။
 ၁။ <http://www.apachefriends.org/en/xampp.html> ကိုသွားပြီးတော့ ကိုယ့်ရဲ့ windows version နဲ့ ကိုက်ညီတဲ့ xampp installer ကိုဒေါင်းလုပ်ဆွဲပါ။
 အဆင့်ဆင့်သွင်းသွားလိုက်ရင် ရသွားမှာပါ။ သာမန် software သွင်းသလိုပဲပဲခင်ဗျာ။ သွင်းပြီးသွားရင်တော့ C အောက်က xampp အောက်က htdocs အောက်မှာ php ဆိုတဲ့ folder လေးဆောက်သင့်ပါတယ်။ အဲဒီ အောက်မှာပဲကျနော်တို့ ရေးသမျှကို save သင့်ပါတယ်။ PHP စာအုပ်တစ်အုပ်ဖြစ်တဲ့အတွက် လုပ်တတ်သင့်တယ်လို့လည်း ယူဆတဲ့အတွက် PHP, Apache နဲ့ MySQL တို့ကို သီးခြားစီ Install လုပ်တဲ့ပုံစံကို Part (II) စာအုပ်မှာ ရေးဖော်ပါမယ်။ အခုကျနော်တို့ အလုပ်ဖြစ်ဖို့အတွက်နဲ့ လွယ်လွယ်ကူကူ မြန်မြန်ဆန်ဆန်လေ့လာနိုင်ဖို့ XAMPP နဲ့ပဲသွားဖို့ ကျနော် ရွေးလိုက်တာပါ။
 နောက်ထပ် ဒီစာအုပ်မှာပါတဲ့ သင်ခန်းစာ များကိုလေ့လာနိုင်ဖို့အတွက် Web Browser (firefox, IE) တစ်ခုလိုပါမယ်။ text editor တစ်ခုလိုပါမယ်။ PHP editor တစ်ခုဆိုပိုကောင်းပါတယ်။ ကျွန်တော်ကတော့ Dream weaver CS 4 ကို အသုံးပြုပါတယ်။ ဒါပေမယ့်လည်း Notepad ကိုပဲအသုံးပြုမယ်ဆိုရင်လည်း အဆင်ပြေနိုင်ပါသေးတယ်။

Chapter 1.The Building Blocks of PHP

Basic PHP Syntax

PHP ကိုရေးတဲ့အခါ <?php နဲ့စပြီး ?> နဲ့အဆုံးသတ်လေ့ရှိပါတယ်။ PHP block တစ်ခုကို document ရဲ့ဘယ်နေရာမဆို ထည့်လို့ရပါတယ်။ အတိုကောက်အနေနဲ့ <? နဲ့စပြီး ?> နဲ့ဆုံးရင်တော့ရပါတယ်။ ဒါပေမယ့် အရှည်ရေးတာကိုပဲအားပေးပါတယ်။

```
<?php
?>
```

PHP file တစ်ခုမှာ ပုံမှန်အားဖြင့် HTML ဖိုင်တစ်ခုလိုပဲ HTML tag တွေပါဝင်ပါတယ်။ PHP code တွေလည်းပါဝင်ပါတယ်။ အောက်က PHP script ကိုကြည့်မယ်ဆိုရင် ဒါဟာ browser ကို "Hello World" ဆိုတဲ့ စာသားလေးကို ထုတ်ပြပေးတဲ့ code လေးပဲဖြစ်ပါတယ်။

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

ဒီ code လေးတွေကို run မယ်ဆိုရင် Dreamweaver ကိုဖွင့် PHP ကိုရွေးပြီးတော့ အလိုအလျောက်နဲ့ရှိနေတဲ့ code တွေကိုဖြတ်နိုင်ပါတယ်။ Notepad ကိုပဲသုံးရင်လည်းရပါတယ်။ အဲဒီနောက် code တွေကို copy paste လုပ်ပါ။ ပြီးရင် save လုပ်ပါ.. save လုပ်ရမယ့်နေရာကတော့ C အောက်က xampp အောက်က htdocs အောက်က php ရဲ့အောက်မှာဖြစ်ပါတယ်။ အဆင်ပြေသလိုနာမည်ပေးနိုင်ပါတယ်။ ကျွန်တော်ကတော့ 1.php, 2.php အစရှိသဖြင့်ပဲဖေးထားပါတယ်။ အဲဒါကို browser မှာကြည့်မယ်ဆိုရင် <http://localhost/php/1.php>, <http://localhost/php/2.php> အစရှိသဖြင့်ခေါ်ပြီး ကြည့်လို့ ရပါပြီ။ PHP code တွေရဲ့အဆုံးသတ်တိုင်းမှာ semicolon နဲ့အဆုံးသတ်ရပါမယ်။ semicolon က စကြောင်းတစ်ကြောင်းနဲ့ နောက်တစ်ကြောင်းကို ခြားနားပေးတဲ့ separator သဘောပါပဲ။ PHP ကိုသုံးပြီးတော့ text စာသားတွေကို output ထုတ်တဲ့အခါ အသုံးပြုနိုင်တဲ့အခြေခံ statements နှစ်မျိုးရှိပါတယ်။ echo နဲ့ print ပါ။ အထက်ပါဥပမာမှာတော့ echo ကိုသုံးပြီး Hello World ဆိုတဲ့ စာသားကို output ထုတ်ထားပါတယ်။ မှတ်သားရမှာက တကယ်လို့ဖိုင်ကို save လုပ်တဲ့အခါ .php နဲ့ မသိမ်းပဲနဲ့ .html နဲ့သိမ်းမယ်ဆိုရင် အထဲက PHP code တွေဟာအလုပ်လုပ်မှာမဟုတ်ပါဘူး။

Comments in PHP

PHP မှာ single line comment အတွက် // ကိုအသုံးပြုပါတယ်။ စာကြောင်းရေများတယ်ဆိုရင်တော့ /* နဲ့စပြီး */ နဲ့ဆုံးပါတယ်။

```
<html>
<body>
<?php
//This is a comment
/*
This is
```

```
a comment
block
*/
?>
</body>
</html>
```

PHP Variables

variable တွေကို number တွေ၊ strings တွေ arrays တွေ function result တွေစတဲ့ တန်ဖိုးတွေကို သိမ်းဖို့အတွက်အသုံးပြုပါတယ်။ ဒီလို variable တစ်ခုကိုကြေငြာထားခဲ့မယ်ဆိုရင် သူ့ကို ကျွန်တော်တို့ရဲ့ php script ထဲမှာ ထပ်ခါထပ်ခါပြန်ခေါ်သုံးနိုင်တာဖြစ်ပါတယ်။

PHP ရဲ့ variable တွေအကုန်လုံးကို \$ သင်္ကေတနဲ့ စရပါတယ်။

```
$var_name = value;
```

တကယ်လို့ \$ သင်္ကေတနဲ့မစဘူးဆိုရင် အလုပ်လုပ်မှာမဟုတ်ပါဘူး။ အခု string variable တစ်ခုနဲ့ number variable တစ်ခုကို ရေးကြည့်ပါမယ်။

```
<?php
$txt = "Hello World!";
$number = 10;
?>
```

PHP မှာ variable တစ်ခုကို ဘာ data type ဖြစ်တယ်ဆိုတာ ကြိုတင်ကြေငြာစရာ မလိုပါဘူး။ အထက်က ဥပမာလေးမှာ ဒီ variable ဟာ ဘာ data type ဖြစ်တယ်ဆိုတာမျိုးကျနော်တို့ မကြေငြာခဲ့ပါဘူး။ သူ့ကိုကျနော်တို့ ပေးထားခဲ့တဲ့ တန်ဖိုးပေါ်မှာမူတည်ပြီး မှန်ကန်တဲ့ data type ကို PHP က အလိုအလျောက် ပြောင်းပေးမှာဖြစ်ပါတယ်။ Java လိုမျိုး Strong type language မှာဆိုရင်တော့ အသုံးမပြုခင်မှာ သူ့ရဲ့ data type ကိုကြိုတင်ကြေငြာပေးရတာဖြစ်ပါတယ်။ variable ကိုအမည်ပေးတဲ့အခါ သူ့ရဲ့ ဥပဒေတွေရှိပါတယ်။

- variable name တစ်ခုကို letter တစ်ခုခု ဒါမှမဟုတ် underscore "_" နဲ့စတင်ရပါမယ်။
- variable name တစ်ခုမှာ alpha-numeric characters နဲ့ underscores (ဆိုလိုတာက a-z, A-Z, 0-9, နဲ့ _) တို့ပဲပါဝင်ရပါမယ်။
- variable name တစ်ခုမှာ space တွေမပါဝင်ရပါဘူး။ variable name စာလုံး (word) တစ်ခုထက်ပိုတယ်ဆိုရင် underscore (_) နဲ့ (\$my_string) ဒါမှမဟုတ် capital letter နဲ့ (\$myString) ပြားနားထားသင့်ပါတယ်။

Strings in PHP

String variable တစ်ခုကို text (character strings)တွေကို သိမ်းဆည်း ဖို့ အလုပ်လုပ်ဖို့ အတွက် အသုံးပြုပါတယ်။ string တွေနဲ့ ပတ်သတ်ပြီး အသုံးများတဲ့ function တွေနဲ့ operators တွေကိုနဲ့လေ့လာကြည့်ပါမယ်။ String တစ်ခုကို ဖန်တီးပြီးတဲ့အခါ သူ့ကို တွက်ချက်နိုင်ပါတယ်။ string တစ်ခုကို function တစ်ခုထဲမှာ တိုက်ရိုက်အသုံးပြုနိုင်သလို variable တစ်ခုအဖြစ်နဲ့လည်း သိမ်းဆည်းနိုင်ပါတယ်။

အောက်က ဥပမာမှာ Hello World ဆိုတဲ့ String ကို \$txt ဆိုတဲ့ string variable ထဲကိုတန်ဖိုး assign လုပ်ထားတာဖြစ်ပါတယ်။

```
<?php
$txt = "Hello World";
echo $txt;
```

```
?>
```

သို့ရဲ့ output ကတော့ **Hello World** ဖြစ်ပါတယ်။ အခု string တွေကို function နဲ့ operators တစ်ချို့ကိုသုံးပြီး manipulate လုပ်တဲ့ပုံစံကိုကြည့်ပါမယ်။

The Concatenation Operator

PHP မှာ string နဲ့ပတ်သတ်တဲ့ operator တစ်ခုပဲရှိပါတယ်။ concatenation operator ဆိုတဲ့ (.) ကို string value နှစ်ခုကို အတူတကွဖြစ်သွားအောင် လုပ်ဖို့အတွက် အသုံးပြုပါတယ်။ dot (.) operator ကိုအသုံးပြုပြီး string နှစ်ခုကို ပေါင်းတဲ့ပုံစံကို အောက်မှာကြည့်ပါ။

```
<?php
$txt1="Hello World";
$txt2="12345";
echo $txt1 . " " . $txt2;
?>
```

သို့ရဲ့ output ကတော့ **Hello World 12345** ပဲဖြစ်ပါတယ်။ ကျနော်တို့ရဲ့ code ကိုကြည့်မယ်ဆိုရင် concatenation operator (.) ကို နှစ်ကြိမ်အသုံးပြုထားတာကိုတွေ့မှာဖြစ်ပါတယ်။ တတိယ string ဖြစ်တဲ့ " "(empty space) ကိုထည့်ချင်လို့ပဲဖြစ်ပါတယ်။

Using the strlen() function

strlen() function ကို string ရဲ့အရှည် length ကိုသိရှိလိုတဲ့အခါမှာ အသုံးပြုပါတယ်။ "Hello World!" ဆိုတဲ့ string ရဲ့ length ကိုရှာမယ်ဆိုရင်

```
<?php
echo strlen("Hello World!");
?>
```

သို့ရဲ့ output ကတော့ **12** ပဲဖြစ်ပါတယ်။ string ရဲ့ length ကို looping တွေ့နဲ့ တခြား function အတော်များများမှာအသုံးပြုဖို့လိုအပ်ပါတယ်။

Using the strpos() function

strpos() function ကို string တစ်ခုအထဲမှာရှိတဲ့ character တစ်ခု ဒါမှမဟုတ် နောက် string တစ်ခုကို ရှာဖို့အတွက် အသုံးပြုပါတယ်။

```
<?php
echo strpos("Hello World!", "World");
?>
```

သို့ရဲ့ output ကတော့ **6** ပဲဖြစ်ပါတယ်။ World ရဲ့နေရာ တနည်းအားဖြင့် W ရဲ့နေရာက 6 ဖြစ်နေပါတယ်။ ဘာလို့ 7 မဖြစ်တာလဲဆိုတော့ string တစ်ခုရဲ့ ပထမ position ဟာ 1 မဟုတ်ပါဘူး 0 ဖြစ်လို့ပါပဲ။ PHP မှာ string အတွက် function ပေါင်းမြောက်များစွာရှိပါတယ်။ အခုတော့ အခြေခံဖြစ်လို့ ဒီလောက်ထိပဲထည့်လိုက်ပါတယ်။

PHP Operators

အောက်မှာ PHP မှာသုံးတဲ့ operator တွေပဲဖြစ်ပါတယ်။

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Arithmetic Operators တွေအနေနဲ့ ဥပမာတစ်ခုလောက် run ကြည့်ပါမယ်။

```
<?php
$no1=2;
$no2=3;
$no3= $no1+$no2;
echo $no3;
?>
```

ဒီ code ကို run ရင် output 5 ရပါမယ်။

အပေါင်း၊ အနှုတ်၊ အမြှောက်၊ ကတော့ ရှင်းပါတယ်။ Division ဆိုတဲ့ အစားမှာတော့ 15 ကို 5 နဲ့စားရင် သုံးရပါမယ်။ 5 ကို 2 နဲ့စားရင် 2.5 ရလဒ်ရပါမယ်။

```
<?php
$no1=5;
$no2=2;
$no3= $no1/$no2;
echo $no3;
?>
```

Modulus ဆိုတာကတော့ အကြွင်းကို ရှာတာဖြစ်ပါတယ်။ 5%2 ဆိုရင် 1 ရပါမယ်.. 10%2 ဆိုရင် 0 ရပါမယ်။

```
<?php
$no1=5;
$no2=2;
$no3= $no1%$no2;
echo $no3;
?>
```

Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Assignment Operators တွေကတော့ အတိုရေးတယ်လို့ပြောရမှာပါ..။ PHP မှာသုံးတဲ့ ပုံစံတွေပေါ့။

x=x+y ဆိုရင် xနဲ့ y နဲ့ပေါင်းပါ ရတဲ့တန်ဖိုးကို x ထဲထည့်ပါလို့ဆိုလိုပါတယ်။ အဲဒါကို PHP မှာ assignment operator သုံးပြီးတော့ x+=y လို့ရေးပါတယ်။

ဥပမာ တစ်ပုဒ်လောက် run ကြည့်ပါ။

```
<?php
$x=5;
$y=2;
$z= $x+=$y;
echo $z;
?>
```

Output က 7 ရပါမယ်။

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Comparison Operators တွေကတော့ တန်ဖိုးခြင်းနှိုင်းယှဉ်ပါမယ်... အမှန်နဲ့ အမှားဆိုပြီး ရလဒ်နှစ်ခု ပြန်ထုတ်ပေးပါတယ်။

== ဆိုရင် ညီသလားလို့စစ်ပါတယ်။ ဥပမာ 5 က 8 နဲ့ညီသလားဆိုရင် 5==8 လို့စစ်ပါမယ် result ကတော့ false ထွက်ပါမယ်။

```
$x=5;
$y=2;
if ($x== $y)
echo "true";
else echo "false";
?>
```

အထက်ပါ code ကို run မယ်ဆို 5 နဲ့ 2 နဲ့ မညီတဲ့အတွက် false ရပါမယ်။

အောက်ကအတိုင်း variable x ရော y ရော 3 ဆိုပြီး တူနေတဲ့အတွက် run တဲ့အခါ output true ရပါမယ်။

```
$x=3;
$y=3;
if ($x==$y)
echo "true";
else echo "false";
?>
```

Logical Operators

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Logical Operators တွေအတွက်လည်း အထက်ပါ ဇယားကိုကြည့်ရင် နားလည်မှာပါ.. ဒီ Operator တွေကို အရှေ့မှာ တဖြည်းဖြည်းနဲ့အသုံးပြုရင်း သေချာနားလည်လာပါလိမ့်မယ်။

Chapter 2. Flow Control Functions in PHP

PHP If... Else Statements

PHP မှာ if, elseif နဲ့ else statements တွေကို မတူညီတဲ့ condition တွေပေါ်မှာ မတူညီတဲ့ လုပ်ဆောင်ချက်တွေကို ဆောင်ရွက်ဖို့အတွက် အသုံးပြုပါတယ်။

Conditional Statements

ကျနော်တို့ coding ရေးတဲ့အခါမှာ မကြာခဏဆိုသလို မတူညီတဲ့ ဆုံးဖြတ်ချက်တွေအတွက် မတူညီတဲ့ လုပ်ဆောင်မှုတွေကို ဆောင်ရွက်လိုကြပါတယ်။ အဲဒီလိုလုပ်ဆောင်နိုင်ဖို့အတွက် conditional statements တွေကိုအသုံးပြုကြရပါတယ်။

The if...else statement

အခြေအနေတစ်ခုက မှန်တယ်ဆိုရင် ဒီ code ကိုဆက် run ပါ.. တကယ်လို့ မှားတယ်ဆိုရင် ဒီဘက်က code ကိုဆက် run ပါဆိုတာမျိုးအတွက် ဒီ if...else statement ကိုအသုံးပြုပါတယ်။ သူ့ရဲ့ syntax ကတော့ အောက်ဖော်ပြပါပုံစံမျိုးဖြစ်ပါတယ်။

if (condition)

code to be executed if condition is true;

else

code to be executed if condition is false;

ဒီဟာကို ဥပမာတစ်ခုနဲ့ကြည့်မယ်ဆိုရင်

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "Have a nice weekend!";
else
echo "Have a nice day!";
?>
</body>
</html>
```

ဒီ code ကို run ကြည့်မယ်ဆိုရင် တကယ်လို့ ဒီနေ့က သောကြာနေ့ဖြစ်ရင် Have a nice weekend လို့ပေါ်မှာဖြစ်ပြီးတော့ တခြားနေ့ဆိုရင်တော့ Have a nice day လို့ပေါ်မှာဖြစ်ပါတယ်။ ကျနော်အခု run တာတော့ Have a nice weekend လို့ပေါ်ပါတယ် ဒီနေ့က သောကြာနေ့ဖြစ်နေလို့ပါ.. အဲဒါနဲ့ Computer ရဲ့ System Date ကို Tuesday လို့ပြောင်းပြီး ကျနော်ပြန် run ပါတယ်။ Have a nice day လို့ပေါ်ပါတယ်ခင်ဗျာ... :) ။

တကယ်လို့များ condition တစ်ခုက မှန်တယ် ဒါမှမဟုတ် မှားတယ်ဆိုတဲ့အခါမှာ တစ်ကြောင်းထက်ပိုတဲ့ code တွေကို run စေချင်တယ်ဆိုရင်

တော့ curly braces ဆိုတဲ့ { } ထဲမှာထည့်ထားရမှာဖြစ်ပါတယ်။ဥပမာ...

```

<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
{
echo "Hello! <br/>";
echo "Have a nice weekend!";
echo " See you on Monday!";
}
?>
</body>
</html>

```

ဒီ code လေးရဲ့ output ကတော့

Hello!

Have a nice weekend! See you on Monday!

ဖြစ်ပါတယ်ခင်ဗျာ။

Computer System Date က သောကြာဖြစ်နေဘူးဆိုရင်တော့ပေါ်မှာမဟုတ်ဘူးနော်...။

The elseif statement

တစ်ခုထက်ပိုတဲ့ condition တွေအများကြီးထဲက တစ်ခုခုကမှန်နေရင် code ကို run စေချင်တယ်ဆိုရင် elseif statement ကိုအသုံးပြုပါတယ်။

သူ့ရဲ့ syntax ကတော့

if (*condition*)

code to be executed if condition is true;

elseif (*condition*)

code to be executed if condition is true;

else

code to be executed if condition is false;

ဒီလိုပုံစံ ဖြစ်ပြီးတော့ ဥပမာပြရရင်...

```

<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "Have a nice weekend!";
elseif ($d=="Sun")
echo "Have a nice Sunday!";
else
echo "Have a nice day!";
?>
</body>
</html>

```

တကယ်လို့အနေက သောကြာဖြစ်ရင် Have a nice weekend! လို့ပြပါမယ်။ တနင်္ဂနွေဖြစ်နေရင် Have a nice Sunday! လို့ output ပြမှာဖြစ်ပြီးတော့ တခြားနေ့တစ်ရက်ရက်ဆိုရင်တော့ Have a nice day! လို့ပဲပြမှာဖြစ်ပါတယ်။

The Switch Statement

အခြေအနေတစ်ခုကိုမူတည်ထားမယ် အဲဒီအခြေအနေပေါ်မှာမူတည်ပြီး မှန်ကန်တဲ့အခါ ဘယ် code ကို run ရမယ်... ဘယ် case နဲ့ကိုက်ညီတယ်ဆိုတာကို စစ်တာဖြစ်ပါတယ်။ switch case statement ကိုသုံးတဲ့အခါမှာ if...elseif...else statement တွေအရည်ကြီးသုံးရတာတွေကနေလည်း သက်သာစေပါတယ်။သူ့ရဲ့ Syntax ကအောက်ဖော်ပြပါအတိုင်းဖြစ်ပါတယ်။

```
switch (expression)
{
case label1:
    code to be executed if expression = label1;
    break;
case label2:
    code to be executed if expression = label2;
    break;
default:
    code to be executed
    if expression is different
    from both label1 and label2;
}
```

ဥပမာကို run ကြည့်မှသာ သဘောပေါက်နိုင်မှာဖြစ်ပါတယ်။

```
<html>
<body>
<?php
$x=4;
switch ($x)
{
case 1:
echo "Number 1";
break;
case 2:
echo "Number 2";
break;
case 3:
echo "Number 3";
break;
default:
echo "No number between 1 and 3";
}
?>
</body>
</html>
```

ဒီ code ကို run လိုက်ရင် No number between 1 and 3 လို့ရမှာဖြစ်ပါတယ်။ \$x=4 လို့တန်ဖိုးသက်မှတ်ပေးထားခဲ့တာကြောင့်ပါ။။ တကယ်လို့ \$x=2 လို့ assign လုပ်ခဲ့ရင် Number 2 လို့ output ပြမှာဖြစ်ပါတယ်။ သူ့ရဲ့ အလုပ်လုပ်ပုံကတော့ single expression တစ်ခု အများအားဖြင့်တော့ variable တစ်လုံးကို တစ်ကြိမ် evaluate လုပ်တယ်.. code တွေထဲမှာပါတဲ့ case တစ်ခုစီထဲက တန်ဖိုးနဲ့ expression ရဲ့ တန်ဖိုးကို နှိုင်းယှဉ်တယ်။ ကိုက်ညီတယ်ဆိုလို့ရှိရင် အဲဒီ case နဲ့ဆက်နွှယ်တဲ့ code ကို execute (run) တယ်။ execute လုပ်ပြီးတဲ့အခါ နောက် case တစ်ခုကို ဆက်ပြီး မ run တော့လေအောင် break ကိုသုံးပြီးထွက်တယ်။ တကယ်လို့ case တွေအားလုံးက မမှန်ဘူးဆိုရင် default statement ကို execute လုပ်သွားမှာပဲဖြစ်ပါတယ်။

PHP Arrays

array တစ်ခုက variable name တစ်ခုလေးထည့်မှာပဲ တစ်ခုသို့မဟုတ် တစ်ခုထက်ပိုတဲ့ တန်ဖိုးတွေကို သိမ်းဆည်းပေးပါတယ်။

ကျနော်တို့အနေနဲ့ တခါတရံမှာ ဆင်တူတဲ့ variable တွေအများကြီးကို ဖန်တီးလိုတဲ့အခါမျိုးရှိလာပါတယ်။ အဲဒီအခါမှာ variable တွေအများကြီး မဖန်တီးတော့ပဲနဲ့ array တစ်ခုထဲမှာပဲ elements တွေအနေနဲ့ သိမ်းလိုက်နိုင်ပါတယ်။ array ထဲမှာရှိတဲ့ element တစ်ခုစီမှာ သူ့ရဲ့ ကိုယ်ပိုင် အမှတ် (ID) ရှိမှာဖြစ်တဲ့အတွက် လွယ်ကူစွာခေါ်သုံးနိုင်ပါတယ်။

array အမျိုးအစား သုံးမျိုးရှိပါတယ်။

-Numeric array – ID key ကိုနံပါတ်နဲ့ပေးတဲ့ array မျိုး

-Associative array- ID key တွေတစ်ခုစီတိုင်းက တန်ဖိုးတစ်ခုနဲ့ ဆက်စပ်နေတဲ့ array မျိုး

-Multidimensional array- တစ်ခု သို့မဟုတ် တစ်ခုထက်ပိုတဲ့ array တွေပါဝင်နေတဲ့ array မျိုး

နားလည်သွားအောင် တစ်ခုချင်းစီကို ဥပမာနဲ့ကြည့်ကြပါမယ်။

Numeric array

numeric array တစ်ခုကို နည်းလမ်းအမျိုးမျိုးနဲ့ ဖန်တီးနိုင်ပါတယ်။"

```
$names = array("MgMg","MaMa","KoKo");
```

ဒီ စာကြောင်းမှာဆိုရင် ID key ကိုအလိုအလျောက်သတ်မှတ်ပေးမှာဖြစ်ပါတယ်။ အောက်က code မှာတော့ ကျနော်တို့က manual သတ်မှတ်ပေးတာဖြစ်ပါတယ်။

```
$names[0] = "MgMg";
```

```
$names[1] = "MaMa";
```

```
$names[2] = "KoKo";
```

အဲဒီ နှစ်မျိုးစာလုံးရဲ့ အကျိုးသက်ရောက်မှုကတော့ အတူတူပါပဲ။ PHP coding ထဲမှာ ထည့်သုံးကြည့်မယ်ဆိုရင်...

```
<?php>
```

```
$names[0] = "MgMg";
```

```
$names[1] = "MaMa";
```

```
$names[2] = "KoKo";
```

```
echo $names[1] . " and " . $names[2] . " are " . $names[0] . "'s neighbours";
```

```
?>
```

output အနေနဲ့ MaMa and KoKo are MgMg's neighbours ဆိုပြီး ရလာမှာဖြစ်ပါတယ်။

Associative Arrays

associative array မှာ ID key တစ်ခုစီဟာ တန်ဖိုးတစ်ခုစီနဲ့ဆက်နွယ်ပါတယ်။ တိကျတဲ့ တန်ဖိုး တစ်ချို့နဲ့ပတ်သတ်တဲ့ အတာတွေကို သိမ်းဆည်းတဲ့အခါ numerical array ကအမြဲတမ်းအကောင်းဆုံးနည်းလမ်းမဟုတ်ပါဘူး...။ associative arrays ကိုသုံးတဲ့အခါမှာ တန်ဖိုးတွေကို key အနေနဲ့ပါ သုံးနိုင်ပြီးတော့ array မှာ တန်ဖိုးတွေကို assign လုပ်နိုင်ပါတယ်။ ရှင်းသွားအောင် ဥပမာကိုကြည့်ပါမယ်။

ဒီဥပမာမှာကျနော်တို့ဟာ လူတစ်ယောက်စီကို သူတို့ရဲ့အသက်ကို array ကိုသုံးပြီး assign (တန်ဖိုးသတ်မှတ်) ပါတယ်။

```
$ages = array ("MgMg"=>18,"MaMa"=>20,"KoKo"=>19);
```

အောက်ကပုံစံနဲ့လည်းရေးနိုင်ပါတယ်.. အဓိပ္ပါယ်ကတော့အတူတူပါပဲ ရေးတဲ့ပုံစံကွာသွားတာပါ။ သူကလည်း array ကိုဖန်တီးတာပါပဲ။

```
$ages['MgMg'] = "18";
$ages['MaMa'] = "20";
$ages['KoKo'] = "19";
```

ဒီ code တွေကို PHP ထဲမှာ ထည့်ရေးတဲ့အခါ...

```
<?php>
$ages['MgMg'] = "18";
$ages['MaMa'] = "20";
$ages['KoKo'] = "19";
echo "KoKo is" . $ages['KoKo'] . " years old.";
?>
```

သူ့ရဲ့ output ကတော့ KoKo is19 years old. ပဲဖြစ်ပါတယ်။

Multidimensional Arrays

multidimensional array တစ်ခုထဲမှာ အဓိက array ကြီးရှိမယ်.. အဲဒီ အဓိက array ကြီးထဲက element တစ်ခုစီကလဲ array ဖြစ်နိုင်တယ်။ အဲဒီ sub array တစ်ခုစီထဲက element တွေက လည်း array ထပ်ဖြစ်နိုင်တယ်.. ထပ်ဆင့်ထပ်ဆင့်ပေါ့...။

```
<?php
$families = array
(
    "Griffin"=>array
    (
        "Peter",
        "Lois",
        "Megan"
    ),
    "Quagmire"=>array
    (
        "Glenn"
    ),
    "Brown"=>array
```



```
(
    "Cleveland",
    "Loretta",
    "Junior"
);
echo "Is " . $families['Griffin'][2] .
"a part of the Griffin family?";
?>
```

အထက်ပါ php code ကိုကြည့်ရင် \$families ဆိုတဲ့ variable ထဲမှာ array တစ်ခုဆောက်ပါတယ်။ Griffin, Quagmire, Brown ဆိုတဲ့သုံးခုပါဝင်ပါတယ်။ Griffin ရဲ့အထဲမှာ Peter, Lois, Megan ဆိုတဲ့ array သုံးခု ထပ်ဆင့်ဆောက်ထားသလို Quagmire ထဲမှာ Glenn Brown ထဲမှာ Cleveland, Loretta, Junior ဆိုပြီးသုံးခုထပ်ပါနေတာတွေ့ရပါတယ်။ ဒီဟာကို run လိုက်ရင် Output က

Is Megan a part of the Griffin family?

ရှာဖြစ်ပါတယ်။ Griffin array ရဲ့ နံပါတ်နှစ်အခန်းက Megan ဖြစ်ပါတယ်။ Peter က နံပါတ် ၀ ၊ Lois က နံပါတ် ၁ ဖြစ်ပါတယ်။

PHP Looping

code အပိုင်းလေးတစ်ခုကို အကြိမ်များစွာ run ဖို့အလိုရှိတဲ့အခါ looping ကိုသုံးရပါတယ်။ PHP မှာ အောက်ဖော်ပြပါ looping statements များရှိပါတယ်။

while - အခြေအနေတစ်ခုမှန်နေပါက ဖော်ပြပါ code ကို run ပါမယ် ... မှန်နေသရွေ့ဆက် run နေမှာဖြစ်ပါတယ်။

do...while - တစ်ကြိမ်တော့ ဖော်ပြပါ code ကို သေချာပေါက် run ပါမယ်...။ အဲဒီနောက်မှာမှ အခြေအနေက မှန်နေသရွေ့ထပ်ခါ ထပ်ခါ ဆက် run ပါမယ်။

for - သက်မှတ်ထားတဲ့အကြိမ်အရေအတွက်အတိုင်း ဖော်ပြပါ code ကို ထပ်ခါထပ်ခါ run နေပါမယ်။

foreach – array တစ်ခုထဲမှာရှိတဲ့ element တစ်ခုစီအတွက် ဖော်ပြပါ code ကို run ပါမယ်။

တစ်ခုချင်းဆီကိုအသေးစိတ်ကြည့်ကြရအောင်...။

The while Statement

အခြေအနေတစ်ခုမှန်နေပါက ဖော်ပြပါ code ကို run ပါမယ် ... မှန်နေသရွေ့ဆက် run နေမှာဖြစ်ပါတယ်။

Syntax

```
while (condition)
code to be executed;
```

Example

ဖော်ပြပါ code လေးက variable i အနေနဲ့ ၅ ထက် ငယ်နေ ဒါမှမဟုတ် ညီနေသရွေ့ ဆက်ပြီးတော့ တစ်ကြိမ်ပြီးတစ်ကြိမ် looping ပတ် နေမှာဖြစ်ပါတယ်။ တစ်ကြိမ် loop ပတ် ပြီးတိုင်းမှာ i ရဲ့တန်ဖိုးက ၁ တိုးတိုးလာပါမယ်။

```
<html>
<body>
<?php
$i=1;
while ($i<=5)
{
    echo "The number is " . $i . "<br/>";
    $i++;
}
```

```
?>
</body>
</html>
```

output အနေနဲ့-

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

ဆိုပြီးရလာမှာပါ...။ အရင်ဆုံး variable i လေးကို 1 assign လုပ်ထားပေးပါတယ်။။ အဲဒီနောက် i က ငါးထက်ငယ်လား ညီလားလို့စစ်ပါတယ်.. i ရဲ့တန်ဖိုးက ၁ ဖြစ်နေတဲ့အတွက် ငယ်ပါတယ် ဒါကြောင့် { } ထဲက code တွေကို အလုပ်လုပ်ပါတယ်။ The number is 1 ဆိုတဲ့တစ်ကြောင်း output ထွက်ပါတယ်...
 ကြောင့် အောက်တစ်ကြောင်းဆင်းပါတယ်.. i ရဲ့တန်ဖိုးကို တစ်တိုးတဲ့အတွက် i တန်ဖိုးက 2 ဖြစ်သွားပါတယ်.. } ကိုရောက်တဲ့အခါ while မှာတစ်ခါထပ်သွားစစ်ပါတယ်.. i က ငါးထက် ငယ်နေသေးတဲ့ အတွက် ထပ်အလုပ်လုပ်ပါတယ်.. The number is 2 ဆိုပြီးထွက်ပါတယ်..။ ဒီလိုနဲ့ i ရဲ့တန်ဖိုး ၆ ဖြစ်သွားတဲ့အထိလုပ်ပါတယ်... ၆ ဖြစ်သွားတဲ့အခါ while မှာထပ်စစ်တော့ ငါးထက်မယ်သလို မညီတော့တဲ့အတွက်.. { } ထဲက code ကိုအလုပ်မလုပ်တော့ပါဘူး.. နားလည်မယ်လို့မျှော်လင့်ပါတယ်...။

The do...while Statement

တစ်ကြိမ်တော့ ဖော်ပြပါ code ကို သေချာပေါက် run ပါမယ်..။ အဲဒီနောက်မှာမှ အခြေအနေက မှားနေရင် ဆက် မ run တော့ပါဘူး.. မှန်နေသရွေ့တော့ ထပ်ခါ ထပ်ခါ ဆက် run ပါမယ်။

Syntax

```
do
{
code to be executed;
}
while (condition);
```

Example

ဖော်ပြပါ ဥပမာမှာ အရင်ဦးဆုံး a ကိုတစ်တိုးပါတယ် အဲဒီနောက်မှာ a ရဲ့တန်ဖိုးက ၅ ထက်ငယ်နေသရွေ့ ဆက်ပြီး တစ်တိုးသွားမှာ ဖြစ်ပါတယ်။

```
<html>
<body>
<?php
$a=0;
do
{
$a++;
```

```

echo "The number is " . $a . "<br/>";
}
while ($a<5);
?>
</body>
</html>

```

အသေးစိတ်အလုပ်လုပ်ပုံကတော့ variable a ထဲကို သုညထည့်ပါတယ် အဲဒီနောက် do { } ထဲက code တွေကိုအလုပ်လုပ်ပါတယ်.. a ကို တစ်တိုးတဲ့အတွက် a က တစ်ဖြစ်သွားပါတယ်... echo ကြောင့် The number is 1 လို့ထွက်မှာဖြစ်ပါတယ်..။
 ကြောင့် အောက်တစ်လိုင်းဆင်းပါတယ်..။ a ကငါးထက်ငယ်နေသေးတဲ့အတွက် do ဆီကိုပြန်သွားပြီး do { } ကိုထပ် အလုပ်လုပ်ပါတယ်..။ a ကငါးထက်ကြီးသွားတဲ့အခါကြမှ looping ကနေထွက်သွားပါတယ်.. သူ့ရဲ့ output က...

```

The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
ပဲဖြစ်ပါတယ်။

```

The for Statement

ရိုးရိုးပြောရရင်တော့ code အပိုင်းလေးတစ်ခုကို ဘယ်နှစ်ကြိမ် run စေချင်တယ်ဆိုတာမျိုးကိုကျနော်တို့ သိထားတဲ့အခါမှာ for statement ကိုသုံးပါတယ်။ သူ့ရဲ့ syntax ကတော့

```

for (init; cond; incr)
{
code to be executed;
}

```

init ဆိုတာက initial value ကိုပြောတာပါ.. မူလအစတန်ဖိုးပေါ့... cond ကတော့ condition ပေါ့... အဲဒီ condition မှန်နေသမျှ အလုပ်လုပ်
ငါမယ်...မှားသွားတာနဲ့ loop ထဲကနေထွက်ပါမယ်။ incr ကတော့ မူလတန်ဖိုး init ကို တိုးပေးတဲ့ increment ပေါ့... ။
အောက်ကတော့ Hellow World! ဆိုတဲ့ စာသားကို ဆယ်ကြိမ် ပြပေးမယ် ဥပမာလေးပါ...

```

<html>
<body>

<?php
    for ( $i=1; $i<=10; $i++)
    {
echo "Hello World!<br />";
    }
?>
</body>
</html>

```

variable `i` ရဲ့မူလတန်ဖိုးကို 1 လို့ပေးပြီးစလိုက်ပါတယ်။ `i` က 10 ထက် ယ်လား ညီနေလား စစ်ပါတယ်.. မှန်တဲ့အတွက် Hello World! တစ်ကြိမ်ပြပါတယ် အောက်တစ်ကြောင်းဆင်းပါတယ်.. `i` ရဲ့တန်ဖိုးကို တစ်တိုးပါတယ်..။ `i` က 2 ဖြစ်သွားပါတယ်.. 10 ထက်ယ်လား ညီလား ထပ်ပြီးစစ်ပါတယ်.. ငယ်နေသေးတဲ့အတွက် နောက်ထပ် Hello World! တစ်ကြောင်းပြပါမယ်.. ဒါပေမယ့် program မပြီးသေးတဲ့အတွက် output တွေကမပြဘူးနော်.. အကုန်ပြီးသွားမှ တခါတည်းပြမှာ...။ ဟုတ်ပါပြီ... အဲဒီလိုနဲ့ `i` ကိုတစ်တိုးလိုက် 10 ထက်ငယ်လား ညီလား စစ်လိုက်နဲ့ နောက်ဆုံး `i` ရဲ့တန်ဖိုး 11 ဖြစ်သွားတဲ့အခါမှာမှ.. Looping ကနေထွက်သွားပါတယ်..။ output အနေနဲ့ Hello World! ဆိုတဲ့ စာသား ဆယ်ကြောင်းရရှိပါတယ်။

The foreach Statement

foreach statement ကိုတော့ array တွေ ကို looping ပတ်ဖို့အတွက်အသုံးပြုပါတယ်။ သူ့ရဲ့ Syntax (ပုံစံ) ကတော့

```
foreach (array as value)
```

```
{
```

```
code to be executed;
```

```
}
```

ဖြစ်ပါတယ်။ အောက်က ဥပမာကိုကြည့်ရင်နားလည်မှာပါ။

```
<html>
```

```
<body>
```

```
<?php
```

```
$arr=array("one", "two", "three");
```

```
foreach ($arr as $value)
```

```
{
```

```
    echo "Value: " . $value. "<br />";
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

output အနေနဲ့-

Value: one

Value: two

Value: three

ဆိုပြီးရပါမယ်။

looping တစ်ခုစီတိုင်းအတွက် array element ရဲ့တန်ဖိုးကို `$value` ထဲကို assign လုပ်တယ်။ array ရဲ့ pointer ကိုတစ်နေရာပြီးတစ်နေရာရွှေ့သွားတယ်။ နောက်တစ်ကြိမ်ထပ်ပတ်တဲ့အခါ နောက်ထပ် element တစ်ခုပေါ်မှာအလုပ်လုပ်ပါတယ်။

Chapter 3. Working With Functions & Forms

PHP Functions

အမှန်တကယ် PHP ရဲ့စွမ်းအားက function တွေပါပဲ...။ PHP မှာ built-in နှိုင်းတိုင်းကပါတဲ့ functions ပေါင်း 700 ကျော်ရှိပါတယ်။

ဒီ ကျူတိုရီရယ်မှာတော့ function တစ်ခုကိုဘယ်လိုဖန်တီးတယ်ဆိုတာကိုရေးသားသွားပါမယ်။

Creating a PHP Function

function ဆိုတာ code အစုအဝေးတစ်ခုပါပဲ.. ကျနော်တို့လိုအပ်တဲ့အချိန်မှာသူ့ကို ခေါ်ပြီးသုံးရမှာဖြစ်ပါတယ်။

function နဲ့ပတ်သတ်ပြီးမှတ်သားရမှာတွေကတော့ -

- function တွေအားလုံးကို "function ()" ဆိုတဲ့စကားလုံးနဲ့ စတင်ရပါမယ်။
- function ကိုအမည်ပေးတဲ့နေရာမှာ သူ့ရဲ့နာမည်ကိုကြည့်လိုက်တာနဲ့ ဒီ function ဟာဘာလုပ်တဲ့ဟာလည်းဆိုတာကို သိသာထင်ရှားအောင်အမည်ပေးသင့်ပါတယ်။ နာမည်ကို letter ဒါမှမဟုတ် underscore နဲ့စတင်သင့်ပါတယ် နံပါတ်နဲ့မဟုတ်ပါဘူး။
- "{ "ကို function ရဲ့အစမှာထည့်ရပါမယ်။" { " ရဲ့အနောက်မှာတော့ function ရဲ့ code တွေကိုရေးရပါမယ်..။
- function ကိုပိတ်တဲ့အခါ "}" နဲ့ပိတ်ရပါမယ်။

အရှင်းဆုံးပုံစံနဲ့ function ရဲ့ပုံစံလေးကိုရေးကြည့်ပါမယ်။

Example

```
<html>
<body>
<?php
function writeMyName()
{
    echo "Moe Myint Shein";
}
writeMyName();
?>
</body>
</html>
```

writeMyName() ဆိုတဲ့ function လေးကိုအရင်ဆောက်လိုက်တယ်.. သူ့ရဲ့ code ကတစ်ကြောင်းပဲရှိပါတယ်။ နာမည်ကို screen မှာပြနေမှာပါ။ ဒီအတိုင်းပဲထားလို့ ရှိရင် မပေါ်ပါဘူး.. function ကိုပိတ်တယ်။ ပြီးမှ writeMyName(); ဆိုပြီးခေါ်လိုက်တော့မှ အလုပ်လုပ်တာဖြစ်ပါတယ်။ ပိုပြီးနားလည်သွားအောင်နောက်ထပ် ဥပမာတစ်ခုထပ်ရေးပါမယ်။

```
<html>
<body>
<?php
function writeMyName()
{
    echo "Moe Myint Shein";
}
```

```

echo "Hello World! <br />";
echo "My name is ";
writeMyName();
echo ". <br /> That's right, ";
writeMyName();
echo " is my name.";
?>
</body>
</html>

```

သို့ရဲ့ output ကတော့

```

Hello World!
My name is Moe Myint Shein.
That's right, Moe Myint Shein is my name.

```

အခုလောက်ဆို function ရဲ့အလုပ်လုပ်ပုံကိုနားလည်မယ်လို့မျှော်လင့်ပါတယ်။

PHP Functions- Adding parameters

အခုကျနော်တို့ရေးလိုက်တဲ့ writeMyName() ဆိုတဲ့ function လေးကတော့ အရိုးရှင်းဆုံးပုံစံလေးပါ...။ သူက string လေးတစ်ခုကိုပဲ echo လုပ်ပေးတာလေးပါ။ function တစ်ခုကိုပိုမိုသက်ဝင်လှုပ်ရှားလာအောင် parameters တွေကိုထည့်ပေးနိုင်ပါတယ်။ parameter ဆိုတာ variable လိုသဘောတရားပါပဲ။ writeMyName() ဆိုတဲ့ function name ကိုကြည့်မယ်ဆိုရင် parentheses လို့ခေါ်တဲ့ () ကိုသတိထားမိမှာပါ။ parameter တွေကို အဲဒီ parentheses ရဲ့အထဲမှာထည့်ပေးရမှာဖြစ်ပါတယ်။

Example 1

အခုဥပမာမှာ တူညီတဲ့မျိုးရိုးနာမည်တွေနဲ့ မတူညီတဲ့ first name နဲ့ကိုရေးမှာဖြစ်ပါတယ်။

```

<html>
<body>
<?php
function writeMyName($fname)
{
    echo $fname. "Moe. <br />";
}
echo "My name is ";
writeMyName("Shwe");
echo "My name is ";
writeMyName("Ngwe");
echo "My name is ";
writeMyName("Sein");
?>
</body>
</html>

```

အခုလို output ထွက်လာပါမယ်။

My name is ShweMoe.

My name is NgweMoe.

My name is SeinMoe

အလုပ်လုပ်ကတော့ writeMyName () ဆိုတဲ့ function ထဲမှာ \$fname ဆိုတဲ့ parameter တစ်လုံးထည့်ပေးထားပါတယ်။ function ထဲမှာပါဝင်တဲ့ code ကလည်းတစ်ကြောင်းတည်းပါပဲ.. \$fname ဆိုတဲ့ parameter ရဲ့တန်ဖိုးရယ် သူ့ရဲ့နောက်မှာ Moe ဆိုတဲ့ မျိုးရိုးနာမည်ရယ်ကို echo နဲ့ရိုက်ခိုင်းပြီး အောက်တစ်ကြောင်းဆင်းစေထားတာဖြစ်ပါတယ်။ ကျနော်တို့က writeMyName() ဆိုတဲ့ function ကို Shwe ဆိုတဲ့ parameter လေးကိုထည့်စေပြီးတော့ writeMyName("Shwe"); ဆိုပြီးလှမ်းခေါ်လိုက်တဲ့အခါ \$fname ရဲ့တန်ဖိုးက Shwe ဖြစ်သွားပြီးတော့ ShweMoe ဆိုတဲ့ နာမည်ကိုဖြစ်သွားစေပါတယ်။ ကျန်တာတွေလည်းအလားတူပါပဲ။ အခုဆက်ပြီးတော့ parameter နှစ်လုံးပေးတဲ့ ဥပမာကိုကြည့်ကြပါမယ်။

```
<html>
<body>
<?php
function writeMyName($fname, $punctuation)
{
    echo $fname. "Moe". $punctuation . "<br />";
}
echo "My name is ";
writeMyName("Shwe", ".");
echo "My name is ";
writeMyName("Ngwe", "!");
echo "My name is ";
writeMyName("Sein", "...");
?>
</body>
</html>
```

run လိုက်တဲ့အခါ အခုအတိုင်းထွက်လာပါတယ်။

My name is ShweMoe.

My name is NgweMoe!

My name is SeinMoe...

ထမညပမာအတိုင်း အပါအဝင် \$punctuation ဆိုတဲ့ parameter နေရာမှာပါ ".", "!", နဲ့ "..." တို့က function ကိုခေါ်လိုက်တဲ့အခါတိုင်း တလှည့်စီ နေရာထင်ယူသွားတာဖြစ်ပါတယ်။

PHP Functions – Return Values

function တွေကို တန်ဖိုးတွေကို return ပြန်ဖို့အတွက်လည်း အသုံးပြုနိုင်ပါတယ်။

```
<html>
<body>
<?php
function add($x,$y)
{
    $total= $x + $y;
    return $total;
}
echo "1+16=" . add(1,16);
```

```
?>
</body>
</html>
```

output - **1+16=17** ဖြစ်ပါတယ်။ add () function ထဲမှာ \$x နဲ့ \$y parameter နှစ်လုံးပါပါတယ်။ အဲဒီ parameter နှစ်လုံးက ရလာမယ့် တန်ဖိုးတွေကို ပေါင်းပြီး \$total ထဲကိုထည့်တယ်.. အဲဒီနောက် \$total ကို return ပြန်ပါတယ်.. add() function ကိုခေါ်လိုက်ရင် \$total ရဲ့တန်ဖိုးကိုပြပေးမှာဖြစ်ပါတယ်။ add(1,16); ဆိုပြီးတော့ function ကိုခေါ်လိုက်တဲ့အခါမှာ... function အလုပ်လုပ်ပါပြီ.. 1 နဲ့ 16 နဲ့ကိုပေါင်းပြီးတော့ \$total ထဲကို 17 ရောက်သွားတယ်.. အဲဒီတန်ဖိုးကို return ပြန်ထားတာဖြစ်တဲ့အတွက် output မှာ 1+16=17 ဆိုပြီးရလာတာဖြစ်ပါတယ်။

PHP Forms and User Input

user ကရေးသွင်းလိုက်တဲ့အချက်အလက် တနည်းအားဖြင့် Form တွေက အချက်အလက်တွေကို ရယူဖို့အတွက် PHP မှာ \$_GET နဲ့ \$_POST ဆိုတဲ့ variable တွေကို အသုံးပြုပါတယ်။ HTML form တွေ PHP form တွေအလုပ်လုပ်တဲ့အခါ အရေးအကြီးဆုံးသတိထားရမှာကတော့ HTML Page တစ်ခုထဲမှာ form element တွေအကုန်လုံးဟာ PHP code ထဲမှာ အလိုအလျောက် အသုံးပြုနိုင်မှာဖြစ်ပါတယ်။ ဥပမာ-

```
<html>
<body>
<form action = "welcome.php" method="post">
Name: <input type="text" name="name"/>
Age:<input type="text" name = "age"/>
<input type="submit"/>
</form>
</body>
</html>
```

ဒီ code လေးကို form1.html လို့ save လိုက်ပါမယ်။ ဥပမာလေးမှာ input field နှစ်ခုနဲ့ button လေးတစ်ခုပါပါတယ်။ user က form ထဲမှာဖြည့်ပြီးတော့ submit button ကိုနှိပ်လိုက်ရင် form ထဲက အချက်အလက်တွေကို welcome.php ဆိုတဲ့ file လေးဆီကိုပို့ပေးတာဖြစ်ပါတယ်။ အခုတော့ welcome.php မရေးရသေးတဲ့အတွက် button ကိုနှိပ်လိုက်ရင် object not found လို့ပေါ်ပါလိမ့်မယ်။ welcome.php ဖိုင်လေးကိုပါရေးလိုက်ပါ မယ်။

```
<html>
<body>
Welcome <?php echo $_POST ["name"]; ?>.<br />
You are <?php echo $_POST ["age"]; ?> years old.
</body>
</html>
```

ဒါလေးကို welcome.php လို့အမည်ပေးပြီး ခုနက form1.html နဲ့နေရာအတူတူ folder တစ်ခုထဲမှာ save လိုက်မယ်.. ပြီးရင် form1.html ကို <http://localhost/php/form1.html> လို့ခေါ်ပြီး run မယ် အမည်နဲ့ အသက်ကို ထည့်ပြီး submit button နှိပ်လိုက်ရင် ဒီလိုပုံစံမျိုး output ရလာပါလိမ့်မယ်။

```
Welcome Moe Myint Shein
You are 20 years old.
```


PHP \$_GET

PHP \$_GET variable ကို get method ကို သုံးပြီးတော့ form တစ်ခုက တန်ဖိုးတွေကို ရယူဖို့အတွက် အသုံးပြုပါတယ်။
\$_GET variable က HTTP GET method ကနေပို့လိုက်တဲ့ variable names တွေနဲ့ တန်ဖိုးတွေရဲ့ array တစ်ခုဖြစ်ပါတယ်။
GET method ကိုအသုံးပြုပြီး အချက်အလက်တွေကို ပို့တဲ့အခါမှာ (browser ရဲ့ address bar မှာပြလိုရှိရင်) လူတိုင်းမြင်ရပါတယ်။ ပြီးတော့ အများဆုံး 100 characters ပဲပို့လို့ရတယ်ဆိုတဲ့ ကန့်သန့်ချက်ရှိပါတယ်။

Example

```
<form action="welcome1.php" method="get">
Name:<input type="text" name="name1"/>
Age: <input type="text" name="age1"/>
<input type="submit"/>
</form>
```

ဒီ code လေးကို form2.php လို့ save လိုက်ပါတယ်။(ကျနော်ကတော့ စာအုပ်ရဲ့အစပိုင်းမှာပြောခဲ့သလိုပဲ php နဲ့ html file တွေအားလုံးကို C: အောက်က xampp အောက်က htdocs အောက်က php ရဲ့အောက်ထဲမှာပဲ save တာဖြစ်ပါတယ်။ ဒါမှ http://localhost/php/form2.php လို့ browser address bar မှာ ခေါ်ပြီး run ရတာဖြစ်ပါတယ်) welcome1.php ဆိုတဲ့ file ကိုအောက်ပါအတိုင်းရေးပြီးတော့ save လိုက်ပါတယ်

```
Welcome <?php echo $_GET["name1"]; ?>.<br />
You are <?php echo $_GET["age1"]; ?> years old!
```

အခုအချိန်မှာ form2.php ကို run လိုက်တဲ့ပါတယ်။ text box နှစ်ခုကိုဖြည့်ပြီးတော့ submit button နှိပ်လိုက်တဲ့အခါ address bar မှာ http://localhost/php/welcome1.php?name1=Moe+Myint+Shein&age1=20 ဆိုပြီးတော့ ပေါ်သွားတာတွေ့ရပါတယ်.. welcome1.php ဖိုင်ထဲမှာ name1 နဲ့ age1 ရဲ့တန်ဖိုးတွေကို \$_GET variable နဲ့ catch လုပ်ထားတဲ့အတွက် browser မှာ အခုလိုပြပါတယ်။

```
Welcome moemyintshein.
You are 20 years old!
```

\$_GET variable ကိုအသုံးပြုတဲ့အခါ variable name တွေနဲ့ တန်ဖိုးတွေကို address bar မှာပြတာကိုခံရမှာဖြစ်တဲ့အတွက် password တွေနဲ့ လူသိမခံစေလိုတဲ့ အချက်အလက်တွေကို ပို့ပေးတဲ့အခါမှာ မသုံးသင့်ပါဘူး... ဒါပေမယ့် တချို့နေရာတွေမှာတော့ အခုလို variable တွေကို URL မှာမြင်ရတဲ့အတွက် အကျိုးရှိပါတယ်။ အရမ်းကြီးတဲ့ variable values တွေကိုလည်းမကိုင်တွယ်နိုင်ပါဘူး character အလုံး 100 ထက်ပိုလို့မရပါ။

THE \$_REQUEST Variable

\$_REQUEST variable က \$_GET, \$_POST နဲ့ \$_COOKIE တို့အလုပ်လုပ်သလိုလုပ်ပေးပါတယ်။ Get method နဲ့ဖြစ်ဖြစ် POST method နဲ့ဖြစ်ဖြစ် ပို့လိုက်တဲ့ data တွေရဲ့ result ကိုရရှိစေဖို့အတွက် အသုံးပြုနိုင်ပါတယ်။
အကယ်လို့ ခုနက welcome1.php ဖိုင်လေးမှာ \$_GET နေရာမှာ \$_REQUEST လို့အောက်ပါအတိုင်းပြောင်းရေးရင်လည်း အတူတူပဲအလုပ်လုပ်တာဖြစ်ပါတယ်။

```
Welcome <?php echo $_REQUEST["name1"]; ?>.<br />
You are <?php echo $_REQUEST["age1"]; ?> years old!
```

PHP \$_POST

\$_POST variable ကတော့ HTTP POST method ကိုသုံးပြီး variable name နဲ့ values တွေရဲ့ array တစ်ခုပဲဖြစ်ပါတယ်။ method= "post" ဆိုပြီးတော့ form တစ်ခုက တန်ဖိုးတွေကို ရယူဖို့အတွက် သုံးပါတယ်။ POST method ကိုအသုံးပြုရင် ပို့လိုက်တဲ့ အချက်အလက်တွေကို browser address bar မှာမမြင်ရတဲ့အပြင် အကန့်အသန့်မရှိလည်း ပို့နိုင်ပါတယ်။ ကျနော်ကတော့ form3.php လို့ save ပါတယ်။

Example

```
<form action="welcome2.php" method="post">
Enter your name: <input type="text" name="name" />
Enter your age: <input type="text" name="age" />
<input type="submit" />
</form>
```

welcome2.php ရဲ့ code မှာ POST ကိုသုံးပြီးလက်ခံပါမယ်။

```
Welcome <?php echo $_POST["name"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old!
```

form3.php ကို run ပြီးတော့ အမည် အသက်တို့ကိုဖြည့်ပြီး submit လုပ်တဲ့အခါ GET method နဲ့မတူပဲ address bar မှာ <http://localhost/php/welcome2.php> ဒီလိုလေးပဲပြပေးပါတယ်။ output ကတော့ အထက်က GET method တုန်းကလိုပဲဖြစ်မှာပါ။

\$_REQUEST Variable ကို \$_POST နေရာမှာထည့်ပြီးသုံးရင်လည်း တူတူပဲရမှာဖြစ်ပါတယ်..။ တခါတရံမှာ GET method ကိုအသုံးပြုပြီးတော့ တခါတရံမှာ POST method ကိုသုံးတာကပိုသင့်လျော်ပါတယ်။

Chapter 4.PHP Advanced

The PHP Date() Function

အချိန်နဲ့ နေ့ရက်တွေကို ပုံစံချဉ်းအတွက် ဒီ function ကိုအသုံးပြုပါတယ်။ သူ့ရဲ့ Syntax ကတော့ date(format, timestamp) ဖြစ်ပါတယ်။ format ဆိုတဲ့ parameter ကတော့ မဖြစ်မနေပါဝင်ဖို့လိုအပ်ပါတယ်။ အချိန်ရဲ့ format ကို သက်မှတ်ပေးတာဖြစ်ပါတယ်။ timestamp ကတော့ optional ပါ... default အနေနဲ့ကတော့ လက်ရှိ နေ့ရက်နဲ့ အချိန်ကို timestamp အနေနဲ့ သုံးသွားမှာဖြစ်ပါတယ်။ timestamp ဆိုတာကတော့ 1970 ဇန်နဝါရီ တစ်ရက် GMT 00:00:00 ကနေစတဲ့ စက္ကန့်အရေအတွက် ဖြစ်ပါတယ်။ Unix Timestamp လို့လည်း ခေါ်ကြပါတယ်။

date() function ရဲ့ ယထာရပ် parameter ဖြစ်တဲ့ format က date/time တစ်ခုကို ဘယ်လို format ပေးရမလဲဆိုတာ သတ်မှတ်ပေး ပါတယ်။ နေ့ရက်နဲ့ အချိန်ပုံစံတွေကို ကိုယ်စားပြုဖော်ပြအတွက် letter တွေကိုအသုံးပြုပါတယ်။ အသုံးပြုနိုင်တဲ့ ပုံစံတွေကတော့

- d - လတစ်လရဲ့ရက် တွေဖြစ်ပါတယ်။ (01-31)
- m - လက်ရှိလဖြစ်ပါတယ်။ (01-12)
- Y - digit လေးလုံးနဲ့ရေးတဲ့ လက်ရှိနှစ်ပဲဖြစ်ပါတယ်။

"/", ".", "-" အစရှိတဲ့ character တွေကိုလည်း letter တွေကြားထဲမှာထည့်သွင်း အသုံးပြုနိုင်ပါတယ်။

```
<?php
echo date("Y/m/d");
echo "<br />";
echo date("Y.m.d");
echo "<br />";
echo date ("Y-m-d");
?>
```

အထက်ပါ code ကို run လိုက်မယ်ဆိုရင် အောက်ပါပုံစံအတိုင်း output ရလာမှာပဲဖြစ်ပါတယ်။

2009/09/13

2009.09.13

2009-09-13

date() function ထဲမှာ format ဆိုတဲ့ parameter တစ်ခုထဲပါဝင်တဲ့ ပုံစံဖြစ်ပါတယ်။ timestamp အနေနဲ့ အခုရောက်နေတဲ့အချိန်နဲ့နေ့ကို အလို အလျောက် ခေါ်သုံးသွားတာဖြစ်ပါတယ်။

PHP Date- Adding a Timestamp

date() function ထဲက ဒုတိယ parameter ဖြစ်တဲ့ timestamp ကိုပါထည့်ပြီးသုံးကြည့်ပါမယ်။ အခုထပ်ပြီးကြည့်ကြမယ် ဥပမာလေးမှာ mktime () ဆိုတဲ့ function လေးကိုသုံးပြီးတော့ မနက်ဖန်အတွက် timestamp တစ်ခုကိုဖန်တီးပါမယ်။ mktime() function အနေနဲ့ specified date တစ်ခုအတွက် Unix timestamp ကို return ပြန်ပေးပါတယ်။ သူ့ရဲ့ Syntax ကတော့

mktime(hour, minute, second, month, day, year, is_dat) ဖြစ်ပါတယ်။ အနာဂတ်က မနက်ဖန်ဖြစ်တဲ့အတွက် တစ်ရက်ပဲထပ်ပေါင်း ပေးရပါမယ်။ mktime() function ရဲ့ day ဆိုတဲ့ argument မှာ တစ်ရက်ထပ်ပေါင်းပေးလိုက်ရုံနဲ့ မနက်ဖန် အချိန်နဲ့ နေ့ကိုဖော်ပြမှာ ဖြစ်ပါတယ်။

```
<?php
$tomorrow = mktime(0,0,0,date("m"),date("d")+1, date("Y"));
echo "Tomorrow is ". date("Y/m/d", $tomorrow);
?>
```

သူ့ရဲ့ output ကတော့ Tomorrow is 2009/09/14 ပဲဖြစ်ပါတယ်။ ဒီနေ့ လက်ရှိအချိန်က စက်တင်ဘာ 13 ဖြစ်တဲ့အတွက် ဖြစ်ပါတယ်။

PHP Include File

Server Side Includes ဆိုတဲ့ SSI ကို စာမျက်နှာပေါင်းများစွာပေါ်မှာ ထပ်ခါထပ်ခါ အသုံးပြုမယ့် functions တွေ header တွေ footer တွေ elements တွေကို ဖန်တီးဖို့အတွက် အသုံးပြုပါတယ်။

Server Side Includes

include() ဒါမှမဟုတ် require() ဆိုတဲ့ function ကိုသုံးပြီးတော့ PHP file တစ်ခုထဲကို ဆာဗာက မ run ခင်မှာ file ရဲ့ content ကို ထည့်ပေးနိုင်ပါတယ်။ ဒီ function နှစ်ခု ရဲ့ အလုပ်လုပ်ပုံက နေရာအတော်များများမှာအတူတူဖြစ်ပါတယ်။ ကွဲသွားတာကတော့ include () function အနေနဲ့က warning ကို ထုတ်ပေးမယ်.. ဒါပေမယ့် script ကိုတော့ ဆက်ပြီး execute လုပ် သွားမှာဖြစ်ပြီးတော့ require () function ကတော့ error ထုတ်ပေးမယ် ပြီးရင် script ကိုလည်း ဆက်ပြီး execute မလုပ်တော့ဘဲ ရပ်လိုက်မှာဖြစ်ပါတယ်။ ဒီ function နှစ်ခုကို စာမျက်နှာတွေအများကြီးပေါ်မှာ ထပ်ခါထပ်ခါ အသုံးပြုနိုင်တဲ့ function တွေ header တွေ footer တွေ နဲ့ elements တွေကို ဖန်တီးဖို့ အသုံး ပြုကြတာဖြစ်ပါတယ်။ ဒီလိုလုပ်တာဟာ developer အတွက် အချိန်ကုန်သက်သာစေပါတယ်။ ဆိုလိုတာက ကျနော်တို့ရဲ့ web page တွေအားလုံးမှာပါဝင်စေချင်တဲ့ header ဒါမှမဟုတ် menu file လေးတစ်ခုကို ဖန်တီးနိုင်ပါတယ်။ header ကို update လုပ်ဖို့လိုအပ်တဲ့အခါ ဒီ include file တစ်ခုထဲကိုပဲ update လုပ်ဖို့လိုမှာဖြစ်သလို site ထဲကို စာမျက်နှာအသစ်တစ်ခုထပ်ပေါင်းတဲ့အခါမျိုးမှာလည်း web page တွေအားလုံးရဲ့ link တွေကို update လုပ်ရမဲ့အစား menu file လေးကိုပဲ ပြောင်းလဲပြင်ဆင်ပေးရုံပဲဖြစ်ပါတယ်။

Example

ကျနော်တို့မှာ header.php ဆိုတဲ့ header file တစ်ခုရှိတယ်လို့ ယူဆကြည့်ပါမယ်။ ဒီ header file ကို page တစ်ခုမှာပေါ်ဖို့အတွက် include() function ကိုအောက်ပါအတိုင်းအသုံးပြုနိုင်ပါတယ်။

```
<html>
<body>
<?php include("header.php"); ?>
<h1> Welcome to my home page</h1>
<p> Some text</p>
</body>
</html>
```

Example 2

include file တွေဟာအများအားဖြင့် .php နဲ့ပဲအဆုံးသတ်တာဖြစ်ပါတယ်။ အခုတစ်ခါ web page တွေအားလုံးမှာအသုံးပြုမယ့် menu file လေးတစ်ခု ရှိတယ်လို့ ယူဆပါမယ် အောက်က menu.php ကိုလေ့လာကြည့်ပါမယ်။

```
<html>
<body>
<a href ="http://www.moemyintshein.com/default.php"> Home </a> |
<a href= "http://www.moemyintshein.com/about.php"> About Me</a> |
<a href="http://www.moemyintshein.com/contact.php"> Contact Me</a>
</body>
</html>
```

default.php , about.php နဲ့ contact.php ဆိုတဲ့ file သုံးခုစလုံးမှာ menu.php ဆိုတဲ့ file လေးပါဝင်သင့်ပါတယ်။ ဥပမာ ဒါကတော့ default.php ထဲက code ပဲဖြစ်ပါတယ်။

```
<html>
<body>
<?php include("menu.php");?>
<h1> Welcome to my home page</h1>
<p> Some text</p>
</body>
</html>
```

ကျနော်တို့ menu.php ကို ပဲ run တဲ့အခါ output အနေနဲ့

[Home](#) | [About Us](#) | [Contact Us](#)

ဆိုပြီးတော့ ရလာမှာဖြစ်ပါတယ်။ default.php ကို run တဲ့အခါမှာတော့

[Home](#) | [About Us](#) | [Contact Us](#)

Welcome to my home page

Some text

ယခုလိုပဲပေါ်မှာဖြစ်ပါတယ်။ default.php နဲ့ menu.php ကတော့ directory(folder) တစ်ခုထဲမှာပဲရှိရပါမယ်။ အကယ်၍ ကျနော်တို့က about.php ကို အောက်ပါအတိုင်းရေးမယ်ဆိုရင်

```
<html>
<body>
<?php include("menu.php");?>
<h1> Welcome to my about me page.</h1>
<p> Some text</p>
</body>
</html>
```

အောက်ပါအတိုင်း Output ရမှာဖြစ်ပါတယ်။

[Home](#) | [About Us](#) | [Contact Us](#)

Welcome to my about me page.

Some text

သူ့ရဲ့ source ကို ကြည့်မယ်ဆိုရင်လည်း

```
<html>
<body>
<html>
<body>
<a href="http://www.w3schools.com/default.php">Home</a> |
<a href="http://www.w3schools.com/about.php">About Us</a> |
<a href="http://www.w3schools.com/contact.php">Contact Us</a><h1> Welcome to my about me
page.</h1>
<p> Some text</p>
</body>
</html>
```

အခုလိုပဲတွေ့ရမှာဖြစ်ပါတယ်။ ကျနော်တို့အနေနဲ့ contact.php မှာလည်း အခုလိုပဲ include.php ကိုပြန်ခေါ်ပြီး ရေးနိုင်တာဖြစ်ပါတယ်။ include file ကိုအသုံးပြုခြင်းအားဖြင့် menu link တွေရဲ့အစီအစဉ်ကို ပြောင်းလဲချင်တဲ့အခါ အမည်ပြောင်းချင်တဲ့အခါ ဒါမှမဟုတ် site ထဲကို နောက်ထပ် web page ထပ်ထည့်ချင်တဲ့အခါတွေမျိုးမှာ page တစ်ခုစီ file တစ်ခုစီမှာပြင်ဖို့မလိုဘဲနဲ့ menu.php ထဲမှာရှိတဲ့ code ကိုပဲ ပြင်လိုက်ရုံပဲဖြစ်ပါတယ်။

The require() Function

require () function က include() function အလုပ်လုပ်သလိုပဲ ပုံစံတူအလုပ်လုပ်ပါတယ်။ error ကိုကိုးကွယ်တဲ့နေရာမှာပဲကွာခြားမှု ရှိတာဖြစ်ပါတယ်။ include() function ကိုသုံးရင် warning ပေးရုံသာပေးပြီး script ကိုဆက် run မှာဖြစ်ပေမယ့် require() function ကိုသုံး ထားရင်တော့ error message ပြပြီးတာနဲ့ ဆက်ပြီးတော့လည်း execute လုပ်မှာမဟုတ်တော့ပါဘူး။

include() function နဲ့ ဖိုင်တစ်ခုကိုခေါ်သုံးထားပြီး အဲဒီဖိုင်ကိုရှာမတွေ့ရင်သော်လည်းကောင်း error တစ်ခုခုဖြစ်တယ်ဆိုရင် အောက်ပါအတိုင်း error message တတ်မှာဖြစ်ပါတယ်။

PHP CODE

```
<html>
<body>
<?php
include("wrongFile.php");
echo "Hello World!";
?>
</body>
</html>
```

ERROR MESSAGE

Warning: include(wrongFile.php) [[function.include](#)]: failed to open stream: No such file or directory in C:\xampp\htdocs\php\10.php on line 5

Warning: include() [[function.include](#)]: Failed opening 'wrongFile.php' for inclusion (include_path='.;C:\xampp\php\pear\') in C:\xampp\htdocs\php\10.php on line 5

Hello World!

error message ဘယ်လိုပုံပြပုံ echo ကြောင့် ရလာတဲ့ Hello World ကိုတော့ execute လုပ်တာပဲဖြစ်ပါတယ်။ ဒါဟာ include ကိုသုံးတဲ့အတွက် warning ပေးရုံပဲရှိပြီး execute လုပ်တာကိုမရပ်လိုက်တဲ့အတွက်ဖြစ်ပါတယ်။ အခုဒီ ဥပမာကိုပဲ require() function နဲ့ပြောင်းပြီး ရေးကြည့်ပါမယ်။

PHP CODE

```
<html>
<body>

<?php
require("wrongFile.php");
echo "Hello World!";
?>
```

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

Error Message

Warning: require(wrongFile.php) [function.require]: failed to open stream: No such file or directory in C:\xampp\htdocs\php\10.php on line 5

Fatal error: require() [function.require]: Failed opening required 'wrongFile.php' (include_path='.;C:\xampp\php\pear\') in C:\xampp\htdocs\php\10.php on line 5

echo statement ကိုတောင် execute မလုပ်ပါဘူး require ကိုသုံးထားတဲ့ဖြစ်ပါတယ်။ error ပြပြီးတာနဲ့ execute လုပ်တာကိုပါရပ်လိုက်တာ ဖြစ်ပါတယ်။ ကျနော်တို့ တကယ် ရေးတဲ့အခါမှာတော့ include() ကိုသုံးမယ့်အစား require() ကိုသုံးဖို့ recommend လုပ်ပါတယ်။ အကယ်၍ file တွေ missing ဖြစ်နေတာမျိုး နာမည်မှားနေတာမျိုးဖြစ်တဲ့အခါမှာ script ကို ဆက်လက်ပြီး execute မလုပ်စေသင့်တဲ့အတွက်ဖြစ်ပါတယ်။

PHP File Handling

Opening a File

PHP မှာ fopen() function ကို file တွေကိုဖွင့်ဖို့အတွက် အသုံးပြုကြပါတယ်။ function ရဲ့ ပထမ parameter မှာဖွင့်မယ့်ဖိုင်ရဲ့ အမည်ပါဝင် မှာဖြစ်ပြီး.. ဒုတိယ parameter မှာတော့ ဘယ် လိုပုံစံ မျိုးနဲ့ ဖိုင်ကိုဖွင့်ခိုင်းမယ်ဆိုတာ သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။

```
<html>
<body>
<?php
$file=fopen("welcome.txt","r");
?>
</body>
</html>
```

file ကို ဘယ်လိုပုံစံမျိုးနဲ့ ဖွင့်ခိုင်းမယ်ဆိုတဲ့ ဒုတိယ parameter မှာရွေးရတဲ့ mode ပေါင်းများစွာရှိပါတယ်။ အခု ကျနော်သုံးလိုက်တဲ့ r ကတော့ read only ကိုဆိုလိုတာပါ။

အောက်က ဥပမာ မှာ အကယ်၍ welcome.txt ဆိုတဲ့ file က သူ့ရဲ့ directory မှာရှိမနေဘူးဆိုရင် fopen() function အလုပ်မလုပ်ပဲနဲ့ Unable to Open File ! ဆိုတဲ့ error message ကို ပြပေးမှာဖြစ်ပါတယ်။

```
<html>
<body>
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
?>
</body>
</html>
```

Closing a File

`fclose()` function ကတော့ ဖွင့်ထားတဲ့ ဖိုင်ကို ပိတ်ဖို့အတွက်သုံးတာပဲပေါ့..။

```
<?php
$file = fopen("welcome2.txt","r");
//some code to be executed
fclose($file);
?>
```

Checking End-of-File

`feof()` function ကိုသုံးပြီးတော့ ဖိုင်ရဲ့အဆုံး (End of File) ရောက်ပြီလားဆိုတာကို စစ်နိုင်ပါတယ်။ ဒီ function က ဘယ်လောက် data length ရှိတယ်ဆိုတာကိုမသိတဲ့ looping တွေအတွက် အသုံးဝင်ပါတယ်။

```
if (feof($file)) echo "End of file";
```

Reading a File Line by Line

`fgets()` function ကိုသုံးပြီးတော့ ဖိုင်တစ်ခု က တစ်ကြောင်းခြင်းကို read လုပ်ဖို့သုံးပါတယ်။ အောက်က ဥပမာမှာ တစ်ကြောင်းပြီးတစ်ကြောင်း ဖိုင်ရဲ့အဆုံးကိုမရောက်မချင်း read လုပ်သွားတာဖြစ်ပါတယ်။

အကယ်၍ `fopen()` function က file ကိုမဖွင့်နိုင်ဘူးဆိုရင် `false(0)` ကို return ပြန်မှာဖြစ်ပါတယ်။

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
//Output a line of the file until the end is reached
while(!feof($file))
{
    echo fgets($file). "<br />";
}
fclose($file);
?>
```

ဒီဟာကို run လိုက်တဲ့အခါ... welcome.txt ထဲကစာတွေ တစ်ကြောင်းပြီးတစ်ကြောင်းကို read သွားပြီးတော့ echo နဲ့ပြထားတာဖြစ်တဲ့အတွက် browser မှာ welcome.txt ဖိုင်ထဲက စာတွေအကုန်လုံးပေါ်မှာဖြစ်ပါတယ်။

Reading a File Character by Character

`fgetc()` function ကိုတော့ ဖိုင်တစ်ခုက character တစ်လုံးချင်းစီကို read လုပ်ဖို့သုံးပါတယ်။ အောက်က ဥပမာလေးက ဖိုင်ရဲ့ အဆုံးကို ရောက်သွားတဲ့အထိ character တစ်လုံးချင်းစီကို read လုပ်သွားတာပဲဖြစ်ပါတယ်။

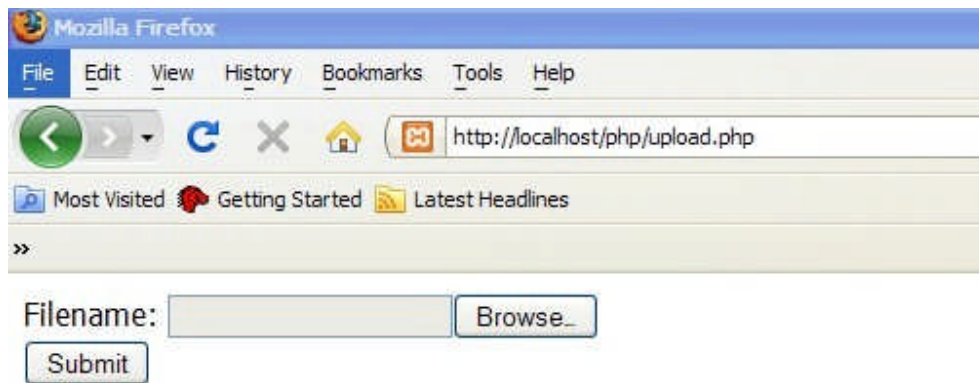
```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```


PHP File Upload

အခု ကျနော်တို့ PHP ကိုသုံးပြီးတော့ server ပေါ်ကို file တွေ upload တင်တာကိုကြည့်ပါမယ်။ user တွေအနေနဲ့ ဖိုင်တွေကို upload တင်နိုင်ဖို့အတွက် form ကို သုံးမယ်ဆိုရင် အဆင်ပြေပါတယ်။ အောက်က HTML code ကို ဖိုင်တွေ upload တင်တဲ့နေရာမှာသုံးနိုင်ပါတယ်။

```
<html>
<body>
<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file"/>
<br/>
<input type="submit" name="submit" value="Submit" />
</form>
```

သူ့ကို run လိုက်မယ်ဆိုရင် browser မှာအောက်ကအတိုင်း browse လုပ်ဖို့ရယ် submit လုပ်ဖို့ရယ် အတွက် upload တင်တဲ့ ပုံစံလေးကိုရပါပြီ။..



ဒီ form မှာကျနော်တို့ submit button ကိုနှိပ်လိုက်မယ်ဆိုရင် upload_file.php ဆိုတာလေးကို run မှာဖြစ်ပါတယ်။ upload_file.php ဖိုင်လေးကိုအောက်ပါအတိုင်းရေးပြီးသိမ်းလိုက်ပါမယ်။

```
<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```

PHP \$_FILES array ကိုသုံးပြီးတော့ remote server ဆီကို client computer ကနေ ဖိုင်တွေတင်တာပဲဖြစ်ပါတယ်။ ပထမဆုံး parameter ကတော့ ဖောင် ရဲ့ input name ဖြစ်ပါတယ်.. ဒုတိယ index အနေနဲ့ကတော့ name, type, size, tmp_name, error စသည်ဖြင့်တစ်ခုခုဖြစ်နိုင်ပါတယ်။

- \$_FILES["file"]["name"]- upload တင်လိုက်တဲ့ ဖိုင်ရဲ့အမည်၊
 - \$_FILES["file"]["type"]- upload တင်လိုက်တဲ့ ဖိုင်ရဲ့အမျိုးအစား၊
 - \$_FILES["file"]["size"]- upload တင်လိုက်တဲ့ ဖိုင်ရဲ့ size (bytes နဲ့ပြပေးပါတယ်)၊
 - \$_FILES["file"]["tmp_name"]- server မှာသိမ်းယူ ဖိုင်ရဲ့ ယာယီ ကော်ပီရဲ့နာမည်၊
 - \$_FILES["file"]["error"]- ဖိုင် upload လုပ်ရကနေဖြစ်လာနိုင်တဲ့ error code၊ စသည်ဖြင့်ဖြစ်ပါတယ်။
- အခုဒီပုံစံက ဖိုင်ကို php နဲ့အရိုးရှင်းဆုံး upload တင်တဲ့ပုံစံပဲဖြစ်ပါတယ်။ security အတွက် user တွေကို ဖိုင် upload တင်ခွင့်ပေးတဲ့အခါမှာ ကန့်သန့်ချက်တွေ ထားရှိသင့်ပါတယ်။

Restrictions on Upload

အခုအောက် က script ထဲမှာ ဖိုင် upload လုပ်တဲ့နေရာမှာ ကန့်သတ်ချက်အချို့ထည့်ပေးမှာဖြစ်ပါတယ်။ user အနေနဲ့ gif ဒါမှမဟုတ် jpeg ဖိုင်တွေကိုပဲ upload ခွင့်ပေးပါမယ်။ ဖိုင်အရွယ်အစားကိုလည်း 20 kb ထက်မကြီးစေရဘူးလို့ ကန့်သတ်ပါမယ်။

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Error: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
}
else
{
    echo "Invalid file";
}
?>
```

Saving the uploaded File

အပေါ်ကဥပမာက server မှာရှိတဲ့ PHP temp folder ထဲမှာပဲ upload တင်လိုက်တဲ့ ဖိုင်ရဲ့ ယာယီ ကော်ပီကိုပဲ ဖန်တီးပေးတာဖြစ်ပါတယ်။ php script ဆုံးသွားတာနဲ့ အဲဒီကော်ပီကလည်း ပျောက်သွားမှာဖြစ်ပါတယ်။ upload လုပ်လိုက်တဲ့ ဖိုင်ကို store လုပ်ဖို့အတွက် location တစ်ခုကို ကူးယူပေးဖို့လိုအပ်ပါတယ်။

```

<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
}
else
{
echo "Invalid file";
}
?>

```

upload_file.php ဖိုင်လေးကို အထက်ပါ code အတိုင်း ရေးပြီးသိမ်းလိုက်တဲ့အခါ upload.php ကို run လို့ရပါပြီ။ php directory (C:\xampp\htdocs\php)အောက်မှာ upload ဆိုတဲ့ folder လေးကို ဆောက်ပေးလိုက်ပါ။ အဲဒီနောက် upload တင်မယ်ဆိုရင်တော့ upload ဆိုတဲ့ folder အောက်ထဲကို ကိုယ်တင်လိုက်တဲ့ file လေးရောက်နေမှာဖြစ်ပါတယ်။ အခုက ကျနော်တို့က ကိုယ့်ရဲ့ local စက်ထဲမှာပဲစမ်းနေတာဖြစ်ပါတယ်။ web hosting ပေါ်မှာစမ်းရင်လည်း သဘောတရားကတော့ အတူတူပဲဖြစ်ပါတယ်။

PHP Cookies

cookie ကို user တစ်ယောက်ရဲ့ identity အတွက် အသုံးပြုပါတယ်။ cookie ဆိုတာ user ရဲ့ ကွန်ပျူတာပေါ်မှာ ဆာဗာကနေ ချန်ထားခဲ့တဲ့ ဖိုင်လေးတစ်ခုပဲဖြစ်ပါတယ်။ browser တစ်ခုကိုသုံးပြီး တူညီတဲ့ ကွန်ပျူတာတစ်လုံးကနေ web page တစ်ခုကို request လုပ်လိုက်တဲ့အခါတိုင်းမှာ cookie ကိုပို့ပေးပါတယ်။ PHP ကိုသုံးပြီးတော့ cookie ဖန်တီးပုံ ရယူပုံတွေကိုလေ့လာကြပါမယ်။

How to Create a Cookie?

cookie တစ်ခုကို ဖန်တီးဖို့အတွက် setcookie() function ကိုသုံးပါတယ်။ setcookie() function ကို <html> tag ရဲ့အရှေ့မှာ ရေးရမှာဖြစ်ပါတယ်။

setcookie (name,value,expire,path,domain); ဒါကတော့ cookie ကို set လုပ်တဲ့ syntax ဖြစ်ပါတယ်။

Example

အောက်က ဥပမာလေးမှာ ကျနော်တို့ user ဆိုတဲ့ cookie လေးတစ်ခုကို ဖန်တီးပါတယ်။ Alex Porter ဆိုတဲ့ တန်ဖိုးကို ပေးပါတယ်။ တစ်နာရီကြာတဲ့အခါ expire ဖြစ်ဖို့ကို သက်မှတ်ပေးပါတယ်။

```
<?php
setcookie("user", "Alex Porter", time()+3600);
?>
<html>
```

....

cookie ရဲ့ expire time ကို နောက်ထပ်နည်းတစ်မျိုးနဲ့အခုလိုသက်မှတ်နိုင်ပါတယ်။ စက္ကန့်ကိုသုံးတာထက်ပိုလွယ်ပါတယ်။

```
<?php
$expire=time()+60*60*24*30;
setcookie("user", "Alex Porter", $expire);
?>
<html>
```

.....

ဒီ ဥပမာမှာဆိုရင် expire time ကို တစ်လအဖြစ်သတ်မှတ်ထားခဲ့တာဖြစ်ပါတယ်။ (60 sec * 60 min * 24 hours * 30 days).

Cookie တန်ဖိုးတစ်ခုကို Retrieve လုပ်ခြင်း

```
<?php
//Print a cookie
echo $_COOKIE["user"];
// A way to view all cookies
print_r($_COOKIE);
?>
```

ဒီဥပမာမှာ user ဆိုတဲ့ cookie လေးရဲ့တန်ဖိုးကို ရယူပြီးတော့ ပြန်ပြပေးပါတယ်။

အောက်က ဥပမာမှာတော့ isset() function ကိုသုံးပြီး cookie တစ်ခုဟာ တန်ဖိုးသတ်မှတ်လိုက်ခြင်းရှိမရှိကို စစ်ဆေးမှာဖြစ်ပါတယ်။

```
<html>
<body>
<?php
if (isset($_COOKIE["user"]))
```

```

echo "Welcome" . $_COOKIE["user"] . "<br/>";
else
echo "Welcome guest!<br />";
?>
</body>
</html>

```

Cookie တစ်ခုကို ဖျက်ပစ်ခြင်း

cookie တစ်ခုကိုဖျက်ပစ်တဲ့အခါမှာ expiration date က အတိတ်မှာကျန်ခဲ့တာသေချာစေဖို့သင့်ပါတယ်။ ဥပမာ

```

<?php
//set the expiration date to one hour ago
setcookie("user", "", time()-3600);
?>

```

Browser တစ်ခုက Cookie ကို support မလုပ်တဲ့အခါ အခြား နည်းလမ်းတွေကိုအသုံးပြုပြီး စာမျက်နှာတစ်ခုကနေတစ်ခုကို အချက်အလက်တွေကို ပို့ပေးဖို့လုပ်ရပါတယ်။ နည်းလမ်းတစ်ခုကတော့ form ကိုအသုံးပြုခြင်းဖြစ်ပါတယ်။ အောက်ကဥပမာ လေးမှာ Submit button ကို user က နှိပ်လိုက်တဲ့အခါ welcome.php ဆီကို user input တွေကိုလှမ်းပို့ပေးတာဖြစ်ပါတယ်။

```

<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name"/>
Age : <input type="text" name="age"/>
<input type="submit"/>
</form>
</body>
</html>

```

အထက်ပါ code တွေကို form1.php လို့ save လိုက်တယ်ဆိုပါစို့။
welcome.php ဖိုင်ထဲမှာအောက်ပါအတိုင်း တန်ဖိုးတွေကို retrieve လုပ်ယူရမှာဖြစ်ပါတယ်။

```

<html>
<body>
Welcome <?php echo $_POST["name"]; ?>. <br />
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>

```

အထက်ပါဥပမာမှာ forum1.php ကို Browser မှာ run လိုက်တဲ့အခါ အမည်နဲ့ အသက်ကိုဖြည့်ဖို့ text box နှစ်ခုရှိမှာ ဖြစ်ပါတယ်။ ဖြည့်ပြီးတော့ submit ကိုနှိပ်လိုက်ရင် အောက်ပါအတိုင်း ဖြည့်လိုက်တဲ့အပေါ်မှာမူတည်ပြီး ပြန်ပြပေးပါလိမ့်မယ်။ ကျနော်ကတော့ အမည်မှာ moemyintshein အသက်မှာ 20 ကိုထည့်ပြီး submit လုပ်လိုက်တဲ့အခါ...

```

Welcome moemyintshein.
You are 20 years old.

```

PHP Sessions

PHP session variable တစ်ခုကို user session တစ်ခုအတွက် အချက်အလက်တွေကိုသိမ်းဆည်းဖို့ သော်လည်းကောင်း ဆက်တင်တွေကို ပြောင်းလဲဖို့ရန်သော်လည်းကောင်း အသုံးပြုရပါတယ်။ user တစ်ယောက်ရဲ့ အချက်အလက်တွေကို ကိုင်တွယ်ပြီးတော့ application တစ်ခုရဲ့ စာမျက်နှာတွေအားလုံးမှာ ပြန်သုံးနိုင်ပါတယ်။ application တစ်ခုကိုသုံးတဲ့အခါမှာ ကျနော်တို့က အဲဒါကိုဖွင့်တယ် အပြောင်းအလဲအချို့ကိုလုပ်တယ်.. ပြီးရင်ပိတ်လိုက်တယ်။ ဒါဟာ session ပါပဲ။ computer က ကျနော်တို့ဘယ်သူဆိုတာကိုသိတယ်။ ကျနော်တို့ ဘယ်အချိန်မှာ ဒါကို စဖွင့်ပြီးတော့ ဘယ်အချိန်မှာ ပိတ်လိုက်တယ်ဆိုတာကို သူကသိပါတယ်။ အင်တာနက်ပေါ်မှာဆိုရင်တော့ ပြဿနာတစ်ခုရှိလာပါတယ်။ web server ကတော့ ကျနော်တို့ ဘယ်သူဆိုတာ ဘာတွေလုပ်တယ်ဆိုတာကို မသိနိုင်ပါဘူး.. ဘာလို့လည်းဆိုတော့ HTTP အနေနဲ့ state ကို maintain မလုပ်ပေးပါဘူး။

ဒီပြဿနာကို PHP session ကိုသုံးပြီးတော့ ဖြေရှင်းပါတယ်။ သူ့ကို သုံးပြီးတော့ နောင်အခါမှာပြန်လည်အသုံးပြုတဲ့အခါ remember ဖြစ်နေစေဖို့အတွက် user ရဲ့ အချက်အလက်တွေကို server ပေါ်မှာသိမ်းဆည်းထားနိုင်ပါတယ်။ ဒါပေမယ့် session information တွေက ယာယီပဲဖြစ်ပါတယ်။ user က website ကနေပြန်လည်ထွက်ခွာသွားတာနဲ့ ပျက်သွားမှာဖြစ်ပါတယ်။ အကယ်၍ အမြဲတမ်းရှိနေစေချင်တယ်ဆိုရင်တော့ database ထဲမှာ data တွေကိုသိမ်းရမှာဖြစ်ပါတယ်။ session အနေနဲ့ unique id(UID) တစ်ခုကို visitor တစ်ဦးစီတိုင်းအတွက် ဖန်တီးပေးပါတယ်။ ပြီးတဲ့အခါ ဒီ UID ပေါ်မှာအခြေခံပြီးတော့ variable တွေကိုသိမ်းဆည်းပါတယ်။ UID ကိုတော့ cookie တစ်ခုထဲမှာသော်လည်းကောင်း URL ထဲမှာ propagated လုပ်ပြီးသော်လည်းကောင်း သိမ်းဆည်းပါတယ်။

Starting a PHP Session

PHP session တစ်ခုထဲမှာ user information တွေကိုမသိမ်းခင် အရင် session တစ်ခုကိုအစ ပြုပေးရပါမယ်။ session_start() function ကို <html> tag မလာခင်မှာ ရေးရပါတယ်။

```
<?php session_start(); ?>
<html>
<body>
</body>
</html>
```

အခုလိုအတိုင်းရေးပေးလိုက်ခဲ့မယ်ဆိုရင် server ပေါ်မှာ user session ကို register လုပ်လိုက်ပြီးတော့ user Information တွေ စတင် သိမ်းမှာဖြစ်ပါတယ်။ ဒါ့အပြင် user session အတွက် UID တစ်ခုကိုလည်း assign လုပ်ပေးပါတယ်။

Storing a Session Variable

session variable တွေကို သိမ်းဖို့ ထုတ်ယူဖို့အတွက် သုံးရတာကတော့ PHP \$_SESSION variable ပဲဖြစ်ပါတယ်။

```
<?php
session_start();
//store session data
$_SESSION['views']=1;
?>
<html>
<body>
<?php
//retrieve session data
echo "Pageviews=" . $_SESSION['views'];
?>
</body>
</html>
```

output အနေနဲ့ Pageviews=1 ဆိုပြီးရမှာဖြစ်ပါတယ်။

အခုထပ်ပြမယ် ဥပမာမှာတော့ ရိုးရှင်းတဲ့ page view counter တစ်ခုကို ဖန်တီးမှာဖြစ်ပါတယ်။ isset() function ကိုသုံးပြီးတော့ views ဆိုတဲ့ variable ကို သတ်မှတ်ထားမထားစစ်ပါမယ်...။ သတ်မှတ်ထားပြီးသားဆိုရင် counter ကို တစ်တိုးပါမယ်။ မဖန်တီးရသေးဘူးဆိုရင်တော့ views variable ကို ဖန်တီးပြီးတော့ တန်ဖိုးကို တစ်လို့ သတ်မှတ်ပေးပါမယ်။

```
<?php
session_start();

if(isset($_SESSION['views']))
    $_SESSION['views']=$_SESSION['views']+1;
else
    $_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

ဒါကို run ကြည့်မယ်ဆိုရင်.... relode လုပ်လိုက်တဲ့အခါတိုင်းမှာ views ကတစ်တိုးတိုးသွားတာတွေမှာဖြစ်ပါတယ်။

Destroying a Session

session data ကိုပြန်ဖျက်မယ်ဆိုရင်တော့ unset() function ဒါမှမဟုတ် sessopm_destroy() function ကိုအသုံးပြုနိုင်ပါတယ်။ unset() function ကိုတော့ ဖော်ပြထားတဲ့ session variable ကို free လုပ်ပစ်ဖို့အတွက်အသုံးပြု ပါတယ်။

```
<?php
unset($_SESSION['views']);
?>
```

session_destroy()function ကို ခေါ်ပြီးတော့ session ကိုလုံးဝ ဖျက်ဆီးလို့လည်းရပါတယ်

```
<?php
session_destroy();
?>
```

PHP Mail Function

PHP script ရေးပြီး အီးမေးတွေကို တိုက်ရိုက်ပို့လို့ရပါတယ်။

PHP ရဲ့ mail() function ကိုအသုံးပြုရမှာဖြစ်ပါတယ်။

Syntax

mail(to,subject,message,headers,parameters)

အောက်ကတော့ သူ့မှာထည့်လို့ရတဲ့ parameter တွေနဲ့ သူတို့ရဲ့ဖော်ပြချက်တွေဖြစ်ပါတယ်။

Parameter	Description
to	Required. Specifies the receiver / receivers of the email
subject	Required. Specifies the subject of the email. Note: This parameter cannot contain any newline characters
message	Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
headers	Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
parameters	Optional. Specifies an additional parameter to the sendmail program

to, subject, message သုံးခုကတော့ မဖြစ်မနေထည့်ပေးရမှာဖြစ်ပြီး headers နဲ့ parameters ဆွဲကတော့ Optional ပဲဖြစ်ပါတယ်။

PHP Simple E-Mail

PHP နဲ့အီးမေးပို့မယ်ဆိုရင် အရှင်းဆုံးနဲ့ အလွယ်ဆုံးကတော့ text စာသားတွေကို ပို့တာပဲဖြစ်ပါတယ်။ အောက်က ဥပမာလေးမှာ အရင်ဆုံး \$to, \$subject, \$message, \$from, \$headers ဆိုတဲ့ variable တွေကိုကြေငြာပြီးတော့ mail() function ထဲမှာ အီးမေးတစ်ခုကိုပို့ဖို့အတွက် ခေါ်သုံးတာဖြစ်ပါတယ်။

```
<?php
$to= "mail@moemyintshein.com";
$subject="Test mail";
$message="Hello! This is a simple email message.";
$from= "gtalk@moemyintshein";
$headers="From: $from";
mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

PHP Mail Form

PHP ကိုသုံးပြီးတော့ ကျနော်တို့ရဲ့ website တွေပေါ်မှာ feedback-form လိုမျိုးဖန်တီးနိုင်ပါတယ်။ အောက်ကဥပမာမှာ form ကိုသုံးပြီး အီးမေးပို့တာကိုပြထားတာဖြစ်ပါတယ်။

```
<html>
<body>

<?php
if (isset($_REQUEST['email']))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail( "someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
```



```

else
//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message: <br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>

</body>
</html>

```

Email:

Subject:

Message:

code တွေကို save လုပ်တဲ့အခါ mailform.php လို့ save လုပ်လိုက်ပါတယ်။ သူ့ရဲ့အလုပ်လုပ်ပုံကတော့ ပထမဆုံး email ရဲ့ input field ကိုဖြည့်ထားမထားစစ်ပါတယ်။ မဖြည့်ထားသေးဘူးဆိုရင်တော့ ဆိုလိုတာက ပထမဆုံးအကြိမ် ဒီစာမျက်နှာကိုရောက်လာတာဆိုရင်တော့ HTML form လေးကိုပြပေးပါတယ်။ အကယ်၍ ဖြည့်ပြီးသား ဖြစ်နေပြီးဆိုရင်တော့ form ကနေ အီးမေးကို ပို့ပေးပါတယ်။ submit button ကိုနှိပ်လိုက်တာနဲ့ ဒီ page ကိုပဲ reload လုပ်ပါတယ်။ input field မှာ ဖြည့်ထားပြီးဖြစ်တဲ့အတွက် အီးမေးကိုပို့ပေးပါတယ်။ အီးမေးပို့တာတွေဖြစ်တဲ့အတွက် SMTP နဲ့ smtp_post စတဲ့ ဆက်တင်တွေကို php.ini file မှာသတ်မှတ်တာတွေ.. လုပ်ရမှာဖြစ်ပါတယ်။အခုတော့ အခြေခံဖြစ်လို့ သဘောတရားကို နားလည်ရင်ရပါပြီ.. အသေးစိတ်ကို ကိုယ်တိုင် ဆက်လေ့လာသင့်ပါတယ်။

အထက်မှာဖော်ပြခဲ့တာတွေကတော့ အရှင်းဆုံး အီးမေးပို့တဲ့ပုံစံဖြစ်ပါတယ် ဒါပေမယ့် တကယ်တမ်း တော့ လက်တွေ့ လုပ်ရတဲ့အခါ အခုအတိုင်းရေးလို့မဖြစ်ပါဘူး.. အဲဒါက လုံခြုံမှုမရှိပါဘူး...။ ရှေ့အခန်းမှာ ကျနော်တို့ ပိုပြီး security ကောင်းအောင် ရေးကြပါမယ်။

PHP Secure E-mails

PHP E-mail Injections

အထက်ကအခန်းမှာကျနော်တို့ဖော်ပြခဲ့တဲ့ code ရဲ့အားနည်းချက်ကတော့ unauthorized user တွေက input form ရဲ့ mail header ထဲမှာ data တွေကိုထည့်သွင်းနိုင်ပါတယ်။ ဥပမာ အောက်မှာပြထားသလိုမျိုး user က form ရဲ့ input field ထဲမှာ ထည့်မယ်ဆိုရင် အဲဒီလိပ်စာတွေအကုန်လုံးဆီကို အီးမေးရောက်မှာဖြစ်ပါတယ်။

```
someone@example.com%0ACc:person2@example.com
%0ABcc:person3@example.com,person3@example.com,
anotherperson4@example.com,person5@example.com
%0ABTo:person6@example.com
```

အောက်ပါ code က form ထဲက email field ကိုစစ်ဆေးဖို့အတွက် input ကို validate လုပ်ဖို့ပါရေးထားတာဖြစ်ပါတယ်။

```
<html>
<body>
<?php
function spamcheck($field)
{
    //filter_var() sanitizes the e-mail
    //address using FILTER_SANITIZE_EMAIL
    $field=filter_var($field, FILTER_SANITIZE_EMAIL);

    //filter_var() validates the e-mail
    //address using FILTER_VALIDATE_EMAIL
    if(filter_var($field, FILTER_VALIDATE_EMAIL))
    {
        return TRUE;
    }
    else
    {
        return FALSE;
    }
}

if (isset($_REQUEST['email']))
{
    //if "email" is filled out, proceed

    //check if the email address is invalid
    $mailcheck = spamcheck($_REQUEST['email']);
    if ($mailcheck==FALSE)
    {
        echo "Invalid input";
    }
    else
    {
        //send email
        $email = $_REQUEST['email'] ;
```

```

$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail("someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
}
else
{
    //if "email" is not filled out, display the form
    echo "<form method='post' action='mailform.php'>
    Email: <input name='email' type='text' /><br />
    Subject: <input name='subject' type='text' /><br />
    Message:<br />
    <textarea name='message' rows='15' cols='40'>
    </textarea><br />
    <input type='submit' />
    </form>";
}
?>

</body>
</html>

```

ဒီ code မှာကျနော်တို့ PHP Filter ကိုအသုံးပြုထားပါတယ်။ input ကိုမှန်မမှန် စစ်ဆေးဖို့ဖြစ်ပါတယ်။

- The FILTER_SANITIZE_EMAIL filter က string တစ်ခုက illegal email character တွေအားလုံးကို ဖယ်ရှားပေးပါတယ်။
- The FILTER_VALIDATE_EMAIL ကတော့ email လိပ်စာရဲ့ အရည်အချင်း... format ကိုက်မကိုက်ကို စစ်ဆေးပေးပါတယ်။ PHP Filter အခန်းမှာ filter အကြောင်းပိုရေးသွားပါမယ်။

PHP Error Handling

PHP ရဲ့ default error handling ကတော့ ရိုးရိုးလေးပါပဲ။ ဖိုင်ရဲ့နာမည် လိုင်းနံပါတ် ပါတဲ့ error message နဲ့ ဘာ error ဆိုတာ ပြောပြီးတော့ browser ကိုပို့ပေးလိုက်တာပါပဲ။ web application တွေရေးတဲ့အခါ error handling ကအရေးကြီးတဲ့ အပိုင်းပါပဲ။ ကျနော်တို့ရေးလိုက်တဲ့ code မှာ error checking code မပါဘူးဆိုရင် ကျနော်တို့ရဲ့ ပရိုဂရမ်က ပရော်ဖက်ရှင်နယ် မဆန်ပါဘူး ပြီးတော့ security risk တွေလည်ရှိလာပါမယ်။ ဒီ အခန်းမှာတော့ PHP ရဲ့ error စစ်တဲ့နည်းလည်း အချို့ကိုရေးပါမယ်။ အောက်ပါ နည်းလမ်း တွေပါဝင်ပါမယ်။

- Simple "die()" statements
- Custom errors and error triggers
- Error reporting

Basic Error Handling: Using the die() function

ယထမဆုံးဥပမာအနေနဲ့ text file တစ်ခုကို ဖွင့်တဲ့ ရိုးရှင်းတဲ့ code လေးပါ...။ အကယ်၍ ဖိုင်ရှိရင် ဘာ error မှပေါ်မှာမဟုတ် ပါဘူး ဖိုင်မရှိခဲ့ရင်တော့ ကျနော်တို့ဆိုမှာတော့ အခုအတိုင်းပေါ်ပါတယ်။

Warning: fopen(welcome.txt) [function.fopen]: failed to open stream: No such file or directory in C:\xampp\htdocs\php\20.php on line 2

ဒီလိုမျိုး error message မျိုးကို user ကိုမရရှိစေချင်တဲ့အတွက်.. အရင်ဆုံး code ထဲမှာ ဖိုင်ရှိမရှိ အရင်စစ်ခိုင်းမယ် ပြီးမှ access လုပ်ခိုင်းပါမယ်။

```
<?php
if(!file_exists("welcome.txt"))
{
die("File not found");
}
else
{
    #file=fopen("welcome.txt"."r");
}
?>
```

အခုလိုရေးလိုက်တဲ့အတွက် welcome.txt ဖိုင်သာရှိမနေဘူးဆိုရင် File not found ဆိုတဲ့ error message ပဲပေါ်မှာ ဖြစ်ပါတယ်။ ယထာပတ္တိ code ထက်စာရင်တော့ အများကြီးပို ပရော်ဖက်ရှင်နယ်ဆန်သွားတယ်ပေါ့..။ error ကိုပြပြီးတာနဲ့ ကျန်တဲ့ code တွေကိုဆက်မ run တော့ဘဲ ရှိရင်းတဲ့ error handling တစ်ခုကိုသုံးခဲ့တာပါ။ ဒါပေမယ့် အခြေတမ်းဒီပုံစံကြီးပဲသုံးလို့ တော့မဖြစ်ပါဘူး.. တခြား error handling အတွက် php function တစ်ချို့ကိုဆက်လေ့လာကြပါမယ်။

Creating a Custom Error Handler

ဒါဟာလည်း ရှိရင်းပါတယ်။ error တစ်ခုဖြစ်တဲ့အခါမှာ function တစ်ခုကိုခေါ်ဖို့အတွက် လုပ်ပေးရတာပါပဲ။ ဒီ function အနေနဲ့ အနည်းဆုံးတော့ parameters နှစ်ခု(error level နဲ့ error message) ကိုကိုင်တွယ်နိုင်ရပါမယ်။ ဒါပေမယ့် သူ့အနေနဲ့ parameter ငါးလုံးအထိလက်ခံနိုင်တာဖြစ်ပါတယ်။ (optional: file, line number, error context)။

Syntax

`error_function (error_level, error_message, error_file, error_line, error_context)`

Parameter	Description
error_level	Required. Specifies the error report level for the user-defined error. Must be a value number. See table below for possible error report levels
error_message	Required. Specifies the error message for the user-defined error
error_file	Optional. Specifies the filename in which the error occurred
error_line	Optional. Specifies the line number in which the error occurred
error_context	Optional. Specifies an array containing every variable, and their values, in use when the error occurred

Error Report Levels

user က သတ်မှတ်တဲ့ error handler တွေကိုအသုံးပြုနိုင်တဲ့ error အမျိုးအစားတွေရဲ့ error report levels တွေဖြစ်ပါတယ်

Value	Constant	Description
2	E_WARNING	Non-fatal run-time errors. Execution of the script is not halted
8	E_NOTICE	Run-time notices. The script found something that might be an error, but could also happen when running a script normally
256	E_USER_ERROR	Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()
512	E_USER_WARNING	Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()
1024	E_USER_NOTICE	User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()
4096	E_RECOVERABLE_ERROR	Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
8191	E_ALL	All errors and warnings, except level E_STRICT (E_STRICT will be part of E_ALL as of PHP 6.0)

နားမလည်လို့ အရမ်းစိတ်မညစ်နေပါနဲ့ဦး...။ အေးအေးဆေးဆေးပေါ့နော်...။ ဒါတွေရှိတယ်လို့တော့သိသွားတာပေါ့ အခု ကျနော်တို့ error တွေကို ကိုင်တွယ်ဖို့အတွက် function တစ်ခုကို ဖန်တီးပါမယ်။

```
function customError($errno, $errstr)
{
    echo "<b>Error:</b> [$errno] $errstr<br />";
    echo "Ending Script";
    die();
}
```

ဒီ code လေးကတော့ ရှိရင်းတဲ့ error handling function တစ်ခုပဲ ဖြစ်ပါတယ်။ အဲဒါလေး ကို o run တဲ့အခါ မှာ error level နဲ့ error message ကိုရယူပြီးတော့ output ပြပေးပြီးတော့ script ကိုအဆုံးသတ်တာပဲဖြစ်ပါတယ်။ ကျနော်တို့ error handling function ကို ဖန်တီးပြီးပြီဆိုရင် ဘယ်အချိန်မှာ trigger လုပ်မယ် (o run မယ်) ဆိုတာကို ဆုံးဖြတ် ရပါမယ်။

Set Error Handler

PHP ရဲ့ default error handler ကတော့ built in error handler ပါပဲ။ အပေါ်မှာ ရေးခဲ့တဲ့ function ကို default error handler အနေနဲ့ script အတောအတွင်းမှာ ထားပါမယ်။ တကယ်တော့ error တစ်ချို့အတွက်ပဲ ဒီ error handler ကိုသုံးဖို့ သတ်မှတ်ပေးနိုင်ပါတယ်။ ဒီလိုနည်းနဲ့ မတူညီတဲ့ error တွေကို မတူညီတဲ့ နည်းလမ်းတွေနဲ့ ကိုင်တွယ် ဖြေရှင်း နိုင်တာ ဖြစ်ပါတယ်။ ဒီဥပမာမှာတော့ error တွေအားလုံးအတွက် ကျနော်တို့ရဲ့ error handler ကိုသုံးပါမယ်။

```
set_error_handler("customError");
```

error တွေအားလုံးကိုကိုင်တွယ်ဖို့ ကျနော်တို့ရဲ့ function ကိုသုံးမှာဖြစ်လို့ set_error_handler() function အနေနဲ့ parameter တစ်လုံးပဲလိုအပ်ပါတယ်။ ဒုတိယ parameter ကိုတော့ error level ကို သတ်မှတ်ပေးဖို့အတွက် ထပ်ပေါင်းထည့်နိုင်ပါတယ်။

အောက်က ဥပမာလေးမှာတော့ မရှိတဲ့ variable ကို Output ပြပေးဖို့ ကြိုးစားမှာဖြစ်ပါတယ်။ ကျနော်တို့ရဲ့ error handler ကိုစမ်းကြည့်ကြတာပေါ့။

```
<?php
//error handler function
function customError($errno, $errstr)
{
    echo "<b>Error:</b> [$errno] $errstr";
}
//set error handler
set_error_handler("customError");
//trigger error
echo($test);
?>
```

သူ့ကို run မယ်ဆိုရင်- **Error: [8] Undefined variable: test** ဆိုပြီးတော့ output ရမှာပါ။

Trigger an Error

user က data တွေကို input ပေးနိုင်တဲ့ script တစ်ခုမှာ illegal input တွေတဲ့အခါမှာ errors တွေကို trigger လုပ်ဖို့အသုံးဝင်ပါတယ်။ PHP မှာတော့ trigger_error() function ကိုသုံးပြီး လုပ်ဆောင်နိုင်ပါတယ်။ အောက်က ဥပမာမှာတော့ test variable က 1 ထက်ကြီးမယ်ဆိုရင် error တစ်ခု ဖြစ်မှာဖြစ်ပါတယ်။

```
<?php
$test=2;
if ($test>1)
{
    trigger_error("Value must be 1 or below");
}
?>
```

သူ့ကို run တဲ့အခါမှာ ကျနော်ကတော့ အောက်ပါအတိုင်း output ရပါတယ်။

Notice: Value must be 1 or below in C:\xampp\htdocs\php\30.php on line 5

Chapter 5. PHP & AJAX

PHP and AJAX

AJAX INTRODUCTION

AJAX= Asynchronous JavaScript And XML

AJAX က web browser နဲ့ web server ကြားထဲကနေ data တွေကို ပို့ဖို့နဲ့ လက်ခံဖို့အတွက် JavaScript ကိုအသုံးပြုပါတယ်။ AJAX နည်းပညာကြောင့် user က အပြောင်းအလဲတစ်ခု လုပ်လိုက်တဲ့အခါမှာ web page တစ်ခုလုံးကို reload လုပ်လိုက်တာမျိုးမဟုတ်ပဲနဲ့ နောက်ကွယ်မှာ web server နဲ့ data exchange လုပ်တဲ့အတွက် web page တွေကိုပိုပြီးတော့ responsive ဖြစ်ပါတယ်။ AJAX ကအောက်ပါ standards တွေပေါ်မှာအခြေခံထားပါတယ်။

-JavaScript

-XML

-HTML

-CSS

PHP ကိုလေ့လာတဲ့အခါမှာ AJAX ကိုမသိလို့မဖြစ်ပါဘူး... ။ JavaScript, XML, HTML,CSS ဒါတွေမသိရင် PHP လေ့လာလည်း သိပ်တော့မထူးပါဘူး.. အဲဒါအကုန်သိရမှာဖြစ်ပါတယ်.. အားလုံးဆက်စပ်ပြီးအသုံးပြုရမှာဖြစ်လို့ပါ။

AJAX အနေနဲ့ major browser တွေအားလုံးက support လုပ်ကြပါတယ်။ ဒါကြောင့် AJAX application တွေက browser နဲ့ platform ပေါ်မှာမှီခိုမှုမရှိဘဲ... Cross Platform, Cross Browser Technology လို့ဆိုနိုင်ပါတယ်။

AJAX Uses XML And HTTP Requests

ရိုးရိုးသာမန်နည်းနဲ့ရေးထားတဲ့ web application တစ်ခုမှာ html form တစ်ခုကိုသုံးပြီး input တစ်ခုကို web server ဆီကို submit လုပ်ပါမယ်။ web server က data ကို process လုပ်ပြီးတဲ့အခါမှာ user ဆီကို web page အသစ်တစ်ခုကို return ပြန်လာပါမယ်။ user က input ပေးတဲ့အခါတိုင်းမှာ web page အသစ်တစ်ခုကို server က ပြန်ပေးရတာဖြစ်တဲ့အတွက် ဒီရိုးရိုးနည်းနဲ့ရေးတာက နှေးပါတယ် ပြီးတော့ user friendly မဖြစ်ပါဘူး။ AJAX နည်းပညာကိုသုံးရင် web application တွေက web page တစ်ခုကို reload လုပ်ရာမလိုပဲနဲ့ data တွေကို ပို့နိုင် ထုတ်နိုင်တာဖြစ်ပါတယ်။ ဒီကိစ္စကို နောက်ကွယ်မှာ server ဆီကို HTTP request တွေပေးပို့ခြင်းနဲ့အလုပ်လုပ်တာဖြစ်ပါတယ်။ server က data ပြန်လာတဲ့အခါ JavaScript ကိုသုံးပြီး web page ရဲ့အစိတ်အပိုင်းလေးကိုပဲ modify လုပ်တာဖြစ်ပါတယ်။ ရှေ့လာမယ့် ဥပမာတွေကိုကြည့်ရင် နားလည်မှာပါ။

PHP and AJAX

AJAX မှာတော့ server ရယ်လို့မရှိပါဘူး.. ။ AJAX ဆိုတာ ကျနော်တို့ရဲ့ browser ပေါ်မှာပဲ run တဲ့ နည်းပညာတစ်ခုဖြစ်ပါတယ်။ browser နဲ့ web server ရဲ့ကြားမှာ HTTP requests (asynchronous data transfer) ကိုသုံးပါတယ်။ ဆိုလိုတာက web page တစ်ခုလုံးကို server ဆီကိုပို့တာမဟုတ်ပဲ အဲဒီ web page ထဲက အချက်အလက် အနည်းငယ်အစိတ်အပိုင်းလေးကိုပဲ ပို့တာဖြစ်ပါတယ်။ ပြောရရင်တော့ AJAX က web server software တွေပေါ်မှာမမှီခိုတဲ့ web browser technology တစ်ခုဖြစ်ပါတယ်။ ဒီ tutorial ထဲမှာတော့ PHP server ပေါ်မှာ run တဲ့ဥပမာတွေကိုအသားပေးဖော်ပြပေးပါမယ်။

AJAX XMLHttpRequest

XMLHttpRequest object ကတော့ AJAX ရဲ့အဓိကသော့ချက်ဖြစ်ပါတယ်။ July 2000 Internet Explorer ထွက်ကတည်းက သုံးလို့ရတာဖြစ်ပါတယ်။ ဒါပေမယ့် 2005 မှာ AJAX နဲ့ Web 2.0 အကြောင်းပြောဖြစ်ကြတဲ့အခါမှ အပြည့်အဝအသုံးပြုကြတာ ဖြစ်ပါတယ်။ မတူညီတဲ့ browser တွေအနေနဲ့ XMLHttpRequest object ကိုဖန်တီးဖို့အတွက် ကွဲပြားတဲ့နည်းလမ်းတွေကို သုံးကြပါတယ်။ Internet Explorer က ActiveXObject ကိုအသုံးပြုပါတယ်။ တခြား browser တွေကတော့ build in JavaScript object တစ်ခုဖြစ်တဲ့ XMLHttpRequest ကိုအသုံးပြုပါတယ်။ အောက်က code လေးက ဒီပြဿနာကိုဖြေလည် စေတဲ့အရိုးရှင်းဆုံး နည်းလမ်းပါ။


```
var XMLHttpRequest=null
if (window.XMLHttpRequest)
{
XMLHttpRequest=new XMLHttpRequest()
}
else if (window.ActiveXObject)
{
XMLHttpRequest=new ActiveXObject("Microsoft.XMLHTTP")
}
```

အရင်ဆုံး XMLHttpRequest object ကိုအသုံးပြုဖို့အတွက် XMLHttpRequest variable တစ်ခုကို ဖန်တီးပါတယ်။ တန်ဖိုးကိုတော့ null လို့ သတ်မှတ်ပေးခဲ့ပါတယ်။ ပြီးတဲ့နောက်မှာ window.XMLHttpRequest ဆိုတဲ့ object လေး available ဖြစ်မဖြစ် စစ်ပါတယ်။ Firefox, Opera, Safari နောက်ပိုင်း version တွေမှာဆိုရင် available ဖြစ်မှာဖြစ်ပါတယ်။ တကယ်လို့ available ဖြစ်တယ်ဆိုရင် XMLHttpRequest=new XMLHttpRequest() ဆိုပြီးတော့ object အသစ်တစ်ခုဖန်တီးပါတယ်။ available မဖြစ်ရင်တော့ window.ActiveXObject က available ဖြစ်မဖြစ်စစ်ပါတယ်။ Internet Explorer 5.5 နဲ့နောက်ပိုင်းတွေ အားလုံးမှာ available ဖြစ်ပါတယ်။ available ဖြစ်တယ်ဆိုရင် XMLHttpRequest=new ActiveXObject() ဆိုပြီးတော့ object အသစ်တစ်ခုကို ဖန်တီးပါတယ်။ ပရိုဂရမ်မာတွေအနေနဲ့ ဒီထက်ပိုကောင်းတဲ့ ပိုမြန်တဲ့ XMLHttpRequest object ကို ပိုနှစ်သက် နိုင်ပါတယ်။ အောက်က ဥပမာလေးကတော့ Microsoft.XMLHTTP အစား Xsxml2.XMLHTTP ဆိုတဲ့ Microsoft ရဲ့ latest version ကို load လုပ်တာဖြစ်ပါတယ်။ အဲဒါကတော့ IE 6 နဲ့နောက်ပိုင်းမှာ မှ available ဖြစ်တာဖြစ်ပါတယ်။

```
function GetXmlHttpRequestObject()
{
var xmlhttp=null;

try
{
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlhttp;
}
```

ဥပမာက ဘာကိုပြောတာလဲဆိုရင် အရင်ဆုံးXMLHttpRequest object ကိုအသုံးပြုဖို့အတွက် XMLHttpRequest ဆိုတဲ့ variable လေးကို ဖန်တီးပါတယ်။ အဲဒီနောက် web standarads (Mozilla, Opera နဲ့ Safari)အရ XMLHttpRequest= new XMLHttpRequest() object ကိုဖန်တီးဖို့ ကြိုးစားပါတယ်။ နောက်တစ် Microsoft နည်းလမ်းနဲ့ IE 6 နဲ့နောက်ပိုင်း အတွက် XMLHttpRequest=new ActiveXObject("Msxml2.XMLHTTP object ကိုဖန်တီးဖို့ကြိုးစားပါတယ်။ အဲဒီမှာ error ဖြစ်တယ်ဆိုမှ အဟောင်းနည်းလမ်း (IE 5.5) XMLHttpRequest=new ActiveXObject(" Microsoft.XMLHTTP") ကိုကြိုးစားပါတယ်။

AJAX Suggest

အောက်က AJAX ဥပမာမှာ user က web form တစ်ခုထဲကို data ရိုက်ထည့်ပြီးတော့ web page တစ်ခုက web server နဲ့ ဘယ်လို communicate လုပ်တယ်ဆိုတာကို သရုပ်ပြသွားမှာဖြစ်ပါတယ်။ အောက်ကလိုပုံစံမျိုးလေး ရေးမှာပါ။ box လေးထဲမှာ character လေးတစ်ခုရိုက်ထည့်တာနဲ့ suggestion မှာပေါ်လာမယ်။

Type a Name in the Box Below

First Name:

Suggestions:

ဒီ ဥပမာမှာ page သုံးခုပါဝင်ပါမယ်။

- ရိုးရိုး HTML page
- JavaScript page
- PHP page

The HTML Form

အခုကတော့ HTML page ဖြစ်ပါတယ်။ သူ့မှာ form အတွက် HTML code ရယ် JavaScript ကိုခေါ်တဲ့ link ရယ်ပါဝင်ပါတယ်။ ဒီ html file ကို ajaxtesting.html လို့ပေးပါမယ်။

```
<html>
<head>
<script src="clienthint.js"></script>
</head>

<body>

<form>
First Name:
<input type="text" id="txt1"
onkeyup="showHint(this.value)">
</form>

<p>Suggestions: <span id="txtHint"></span></p>

</body>
</html>
```

ဒီဥပမာမှာဆိုရင် txt1 ဆိုတဲ့ Input field လေးပါတဲ့ ရိုးရိုး HTML form လေးပါပါတယ်။ အလုပ်လုပ်ပုံကတော့

1. user က input field လေးထဲမှာ keyboard ပေါ်က key တစ်ခုကို နှိပ်ပြီးတော့ release လုပ်လိုက်တဲ့အခါ event တစ်ခုကို အစပျိုးပေးပါတယ်။
2. event ကိုအစပျိုးပြီးတဲ့အချိန်မှာ showHint() ဆိုတဲ့ function လေးကိုခေါ်ပြီးတော့ execute လုပ်ပါတယ်။
3. form ရဲ့အောက်မှာ txtHint ဆိုတဲ့ လေးရှိပါတယ်။ အဲဒါကတော့ showHint() ကပြန်လာတဲ့ return data လေး ပေါ်မဲ့နေရာလေးပဲဖြစ်ပါတယ်။

အောက်က JavaScript code ကိုတော့ clienthint.js လို့သိမ်းမှာဖြစ်ပြီး HTML document နဲ့ လင့်ခ်လုပ်ထားတာဖြစ်ပါတယ်။

```
var xmlhttp;
```

```
function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
xmlHttp=GetXmlHttpRequest();
if (xmlHttp==null)
{
alert ("Browser does not support HTTP Request");
return;
}
var url="gethint.php";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlHttp.onreadystatechange=stateChanged;
xmlHttp.open("GET",url,true);
xmlHttp.send(null);
}

function stateChanged()
{
if (xmlHttp.readyState==4 || xmlHttp.readyState=="complete")
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText;
}
}

function GetXmlHttpRequest()
{
var xmlHttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
```

```
}
return xmlhttp;
} // JavaScript Document
```

ဒီဥပမာလေးကိုရှင်းရရင်တော့ showHint() function က input field ထဲမှာ ကျန်ရှိတဲ့ character တစ်လုံးရှိလာလို့ကတော့ အခါတိုင်းအလုပ်လုပ်ပါတယ်။ အကယ်၍ input filed ထဲမှာ input ရှိပြီးသား ဆိုလိုတာက str.length>0 ဖြစ်သွားတယ်ဆိုရင် အောက်ပါအတိုင်း execute လုပ်ပါတယ်။

1. server ကိုဖော်မယ် url filename ကို သတ်မှတ်ပါတယ်။ ဒီမှာတော့ gethint.php ပါ။
2. input filed ထဲမှာရောက်နေမယ့် content (character) နဲ့ ယှဉ်တွဲပြီးတော့ parameter (q) လေးကိုပေါင်းထည့်ပါတယ်။
3. server က cached file ကိုသုံးမိတာကနေ ကာကွယ်ဖို့ random number တစ်ခုကို ထပ်ပေါင်းထည့်ပါတယ်။
4. GetXmlHttpRequest function ကိုခေါ်ပြီးတော့ XMLHttpRequest object ကိုဖန်တီးယူပါတယ်။ ပြီးတော့ ပြောင်းလဲမှုတစ်ခု ဖြစ်တဲ့အခါ အဲဒီ object အနေနဲ့ StateChanged ဆိုတဲ့ function လေးကို execute လုပ်ပေးပါတယ်။
5. ဖော်လိုက်တဲ့ url အတိုင်း XMLHttpRequest object ကို ဖွင့်လိုက်ပါတယ်
6. server ဆီကို HTTP request ကိုလှမ်းပို့လိုက်ပါတယ်။

input filed က ဗလာဖြစ်နေတယ်ဆိုရင် function က txtHint ရဲ့ placeholder (suggestion ပေါ်တဲ့နေရာ) မှာရှိနေတဲ့ content (စာတွေ) အားလုံးကို ရှင်းလင်းပစ်လိုက်ပါတယ်။

The stateChanged() Function

XMLHttpRequest object ရဲ့ state ပြောင်းလဲတဲ့အခါတိုင်းမှာ ဒီ function ကို execute(တွက်ချက်) လုပ်ပါတယ်။

state က 4 ဒါမှမဟုတ် complete အဖြစ်ပြောင်းတဲ့အခါ txtHint ရဲ့ content ကို သူ response လုပ်တဲ့ text နဲ့ဖြည့်ပါတယ်။

The GetXmlHttpRequest() Function

AJAX applications အနေနဲ့ XML support လုပ်တဲ့ browser နဲ့ပဲ run မှာဖြစ်တဲ့အတွက် GetXMLHttpRequest() ကိုခေါ်ပြီးတော့ ဒီပြဿနာနည်းလမ်းတစ်ခု မဟုတ်တစ်ခုနဲ့ဖြေရှင်းနိုင်စေဖို့ XMLHttpRequest object အမျိုးမျိုးကိုဖန်တီးပါတယ်။

အခု ကျန်တဲ့ PHP page အတွက်ဖြစ်ပါတယ်။ JavaScript code ကခေါ်တဲ့လိုအပ်တဲ့ server page ကတော့ gethint.php လို့ခေါ်တဲ့ Php file လေးပါ။ ဒီ gethint.php ဖိုင်လေးထဲက code မှာတော့ နာမည်တွေရဲ့ array တစ်ခုကို စစ်ပြီးတော့ client ဆီကို သက်ဆိုင်ရာ အမည်ကို return ပြန်ပေးပါမယ်။

```
<?php
// Fill up array with names
$a[]="Anna";
$a[]="Brittany";
$a[]="Cinderella";
$a[]="Diana";
$a[]="Eva";
$a[]="Fiona";
$a[]="Gunda";
$a[]="Hege";
$a[]="Inga";
$a[]="Johanna";
$a[]="Kitty";
$a[]="Linda";
$a[]="Nina";
$a[]="Ophelia";
$a[]="Petunia";
$a[]="Amanda";
```

```

$a[]="Raquel";
$a[]="Cindy";
$a[]="Doris";
$a[]="Eve";
$a[]="Evita";
$a[]="Sunniva";
$a[]="Tove";
$a[]="Unni";
$a[]="Violet";
$a[]="Liza";
$a[]="Elizabeth";
$a[]="Ellen";
$a[]="Wenche";
$a[]="Vicky";
//get the q parameter from URL
$q=$_GET["q"];
//lookup all hints from array if length of q>0
if (strlen($q) > 0)
{
$hint="";
for($i=0; $i<count($a); $i++)
{
if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
{
if ($hint=="")
{
$hint=$a[$i];
}
else
{
$hint=$hint." , ".$a[$i];
}
}
}
}

//Set output to "no suggestion" if no hint were found
//or to the correct values
if ($hint == "")
{
$response="no suggestion";
}
else
{
$response=$hint;
}

//output the response

```

```
echo $response;
?>
```

JavaScript ဖိုင်ကနေ (strlen(\$q)>0) ဖြစ်တဲ့အခါတိုင်း text တစ်ခုခုကို လှမ်းပို့တဲ့အခါ အောက်ပါအတိုင်းဖြစ်ပေါ်ပါတယ်။

1. JavaScript ကနေလှမ်းပို့တဲ့ character နဲ့ match ဖြစ်တဲ့ အမည်ကိုရှာတယ်။
2. နာမည်တစ်ခုထက်ပိုပြီးအတွဲမိတယ်ဆိုရင် response string မှာအမည်တွေအားလုံးပါလာမယ်။
3. ဘာမှ match ဖြစ်တာမတွေ့ရင်တော့ no suggestion ဆိုပြီး response ဖြစ်မယ်။
4. တစ်ခုထက်ပိုပြီးတွေ့ရင် ဒီနာမည်တွေကို response ဖေးမယ်
5. ဒီ response တွေအားလုံးကို txtHint placeholder ကိုပို့ပေးမယ်။

အခုအချိန်မှာ clienthint.js , gethint.php ajaxtesting.html ဖိုင်သုံးခု ကတူညီတဲ့ folder directory အောက်မှာ save ခဲ့တာဖြစ်ရပါမယ်.. အဲဒါဆိုရင်တော့ ajaxtesting.html ကို browser မှာ run တဲ့အခါ PHP နဲ့ AJAX နဲ့တွဲသုံးထားတဲ့ ပျော်စရာ web application ဖိစီးလေးတစ်ခုကိုရပါပြီခင်ဗျာ။ :D

PHP AND AJAX XML EXAMPLE

အခုတစ်ခါကတော့ AJAX ကို XML file နဲ့ယှဉ်တွဲသုံးနိုင်ပါတယ်။ အောက်က ဥပမာမှာ web page တစ်ခုက နေ့ AJAX နည်းယူကုန်သုံးပြီးတော့ XML file တစ်ခုစီက အချက်အလက်တွေကို ဘယ်လိုခေါ်ယူသုံးနိုင် (fetch) လုပ်နိုင်တယ်ဆိုတာကို လေ့လာမှာဖြစ်ပါတယ်။ Select a CD: ဆိုပြီးတော့ စာသားလေးရှိမယ်ဘေးမှာ drop down box လေးရှိမယ် အဲဒီထဲမှာ ရွေးချယ်စရာသုံးခုရှိမယ်.. တစ်ခုကိုရွေးလိုက်ရင် အောက်မှာ အဲဒါနဲ့ ပတ်သက်တဲ့ CD info ကိုပြပေးမယ်။ ဒါမျိုးလေးပါ။ ဒီ ဥပမာလေးမှာလည်း HTML form တစ်ခု ၊ XML file တစ်ခု ၊ JavaScript တစ်ခု ၊ PHP page တစ်ခု စုစုပေါင်းလေးခုပါဝင်ပါမယ်။

THE HTML FORM

ဒီအောက်က ဥပမာလေးမှာတော့ HTML form လေးပါမယ်။ JavaScript ဖိုင်ဆီကိုလည်း လင့်ချက်ထားတာဖြစ်ပါတယ်။

ဒီ HTML code တွေကိုတော့ ajaxxml.html လို့ save လိုက်ပါတယ်။

```
<html>
<head>
<script src="selectcd.js"></script>
</head>
<body>
<form>
Select a CD:
<select name="cds" onchange="showCD(this.value)">
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bee Gees">Bee Gees</option>
<option value="Cat Stevens">Cat Stevens</option>
</select>
</form>
<p>
<div id="txtHint"><b>CD info will be listed here.</b></div>
</p>
</body>
</html>
```

ဒါက ရိုးရှင်းတဲ့ HTML form လေးပဲဖြစ်ပါတယ်။ cds ဆိုတဲ့ drop down box လေးပါမယ်။ အဲဒါရဲ့အောက်မှာတော့ txtHint လို့ခေါ်တဲ့ div တစ်ခုပါမယ်။ အဲဒီ div ကို web server ကနေပြန်လာမယ့် အချက်အလက်တွေကို ပြပေးမယ်။ နေရာ

placeholder အနေနဲ့အသုံးပြုတာဖြစ်ပါတယ်။ user က data တစ်ခုကို ရွေးလိုက်တဲ့အခါ showCD ဆိုတဲ့ function ကို execute(run) လုပ်မှာဖြစ်ပါတယ်။ onchange event ပေါ်မှာမူတည်ပြီးတော့ function ကိုစ run မှာဖြစ်ပါတယ်။ ဆိုလိုတာက user က drop down box ထဲက တန်းဖီးကို ပြောင်းလိုက်တဲ့အခါတိုင်း ရွေးချယ်မှုပြောင်းလိုက်တဲ့အခါတိုင်းမှာပေါ့ showCD ဆိုတဲ့ function ကိုသွားခေါ်မှာဖြစ်ပါတယ်။ XML file အနေနဲ့ကတော့ အောက်က code လေးကို cd_catalog.xml လို့ save ပြီး ခုနက ajaxxml.html ဆိုတဲ့ file နဲ့တူညီတဲ့ folder အောက်မှာထားလိုက်ပါတယ်။

```
<CATALOG>

<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
-
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
-
<CD>
<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>
</CD>
-
<CD>
<TITLE>Still got the blues</TITLE>
<ARTIST>Gary Moore</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Virgin records</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1990</YEAR>
</CD>
-
<CD>
<TITLE>Eros</TITLE>
<ARTIST>Eros Ramazzotti</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>BMG</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
-
<CD>
<TITLE>One night only</TITLE>
<ARTIST>Bee Gees</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1998</YEAR>
</CD>
-
<CD>
<TITLE>Sylvias Mother</TITLE>
<ARTIST>Dr.Hook</ARTIST>
```

```

<COUNTRY>UK</COUNTRY>
<COMPANY>CBS</COMPANY>
<PRICE>8.10</PRICE>
<YEAR>1973</YEAR>
</CD>
-
<CD>
<TITLE>Maggie May</TITLE>
<ARTIST>Rod Stewart</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Pickwick</COMPANY>
<PRICE>8.50</PRICE>
<YEAR>1990</YEAR>
</CD>
-
<CD>
<TITLE>Romanza</TITLE>
<ARTIST>Andrea Bocelli</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.80</PRICE>
<YEAR>1996</YEAR>
</CD>
-
<CD>
<TITLE>When a man loves a woman</TITLE>
<ARTIST>Percy Sledge</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Atlantic</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1987</YEAR>
</CD>
-
<CD>
<TITLE>Black angel</TITLE>
<ARTIST>Savage Rose</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Mega</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1995</YEAR>
</CD>
-
<CD>
<TITLE>1999 Grammy Nominees</TITLE>
<ARTIST>Many</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Grammy</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1999</YEAR>
</CD>
-
<CD>
<TITLE>For the good times</TITLE>
<ARTIST>Kenny Rogers</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Mucik Master</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1995</YEAR>
</CD>
-
<CD>
<TITLE>Big Willie style</TITLE>
<ARTIST>Will Smith</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
-

```

```

<CD>
<TITLE>Tupelo Honey</TITLE>
<ARTIST>Van Morrison</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>8.20</PRICE>
<YEAR>1971</YEAR>
</CD>
-
<CD>
<TITLE>Soulsville</TITLE>
<ARTIST>Jorn Hoel</ARTIST>
<COUNTRY>Norway</COUNTRY>
<COMPANY>WEA</COMPANY>
<PRICE>7.90</PRICE>
<YEAR>1996</YEAR>
</CD>
-
<CD>
<TITLE>The very best of</TITLE>
<ARTIST>Cat Stevens</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Island</COMPANY>
<PRICE>8.90</PRICE>
<YEAR>1990</YEAR>
</CD>
-
<CD>
<TITLE>Stop</TITLE>
<ARTIST>Sam Brown</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>A and M</COMPANY>
<PRICE>8.90</PRICE>
<YEAR>1988</YEAR>
</CD>
-
<CD>
<TITLE>Bridge of Spies</TITLE>
<ARTIST>T'Pau</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Siren</COMPANY>
<PRICE>7.90</PRICE>
<YEAR>1987</YEAR>
</CD>
-
<CD>
<TITLE>Private Dancer</TITLE>
<ARTIST>Tina Turner</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Capitol</COMPANY>
<PRICE>8.90</PRICE>
<YEAR>1983</YEAR>
</CD>
-
<CD>
<TITLE>Midt om natten</TITLE>
<ARTIST>Kim Larsen</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Medley</COMPANY>
<PRICE>7.80</PRICE>
<YEAR>1983</YEAR>
</CD>
-
<CD>
<TITLE>Pavarotti Gala Concert</TITLE>
<ARTIST>Luciano Pavarotti</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>DECCA</COMPANY>
<PRICE>9.90</PRICE>

```



```
<YEAR>1991</YEAR>
</CD>
-
<CD>
<TITLE>The dock of the bay</TITLE>
<ARTIST>Otis Redding</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Atlantic</COMPANY>
<PRICE>7.90</PRICE>
<YEAR>1987</YEAR>
</CD>
-
<CD>
<TITLE>Picture book</TITLE>
<ARTIST>Simply Red</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Elektra</COMPANY>
<PRICE>7.20</PRICE>
<YEAR>1985</YEAR>
</CD>
-
<CD>
<TITLE>Red</TITLE>
<ARTIST>The Communards</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>London</COMPANY>
<PRICE>7.80</PRICE>
<YEAR>1987</YEAR>
</CD>
-
<CD>
<TITLE>Unchain my heart</TITLE>
<ARTIST>Joe Cocker</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>EMI</COMPANY>
<PRICE>8.20</PRICE>
<YEAR>1987</YEAR>
</CD>
</CATALOG>
```

အောက်က code ကတော့ selectcd.js ဆိုပြီးတော့ သိမ်းမယ်။ JavaScript code ပဲဖြစ်ပါတယ်။

```
var xmlHttp

function showCD(str)
{
xmlHttp=GetXmlHttpRequest()
if (xmlHttp==null)
{
alert ("Browser does not support HTTP Request")
return
}
var url="getcd.php"
url=url+"?q="+str
url=url+"&sid="+Math.random()
xmlHttp.onreadystatechange=stateChanged
xmlHttp.open("GET",url,true)
xmlHttp.send(null)
}

function stateChanged()
{
if (xmlHttp.readyState==4 || xmlHttp.readyState=="complete")
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText
}
}

function GetXmlHttpRequest()
{
```

```

var xmlhttp=null;
try
{
    // Firefox, Opera 8.0+, Safari
    xmlhttp=new XMLHttpRequest();
}
catch (e)
{
    // Internet Explorer
    try
    {
        xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch (e)
    {
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
}
return xmlhttp;
}

```

stateChanged() နဲ့ GetXmlHttpRequest တွေကတော့ ရှေ့ဥပမာတစ်ခုကအတိုင်းပါပဲ။

showCD() function ရဲ့အလုပ်လုပ်ပုံကတော့ drop down box ထဲက item တစ်ခုကိုရွေးလိုက်ပြီးဆိုတာနဲ့ အလုပ်လုပ်တာဖြစ်ပါတယ်။ သူလည်းပဲ အရှေ့မှာလေ့လာခဲ့တဲ့ showHint() လိုမျိုးပဲအလုပ်လုပ်တာဖြစ်ပါတယ်။

JavaScript ကနေလှမ်းခေါ်မယ့် server page ကတော့ getcd.php ဆိုတဲ့ဖိုင်လေးဖြစ်ပါတယ်။ အောက်ပါအတိုင်း ရေးပြီး သိမ်းပါမယ်။ XML document cd_catalog.xml ကို load လုပ်ဖို့အတွက် XML DOM ကိုသုံးပြီး PHP နဲ့ရေးထားတာဖြစ်ပါတယ်။ XML file ထဲကနေ query run ပြီးတော့ HTML အနေနဲ့ result ကို return ပြန်လာတာဖြစ်ပါတယ်။

```

<?php
$q=$_GET["q"];
$xmlDoc = new DOMDocument();
$xmlDoc->load("cd_catalog.xml");
$x=$xmlDoc->getElementsByTagName('ARTIST');
for ($i=0; $i<=$x->length-1; $i++)
{
    //Process only element nodes
    if ($x->item($i)->nodeType==1)
    {
        if ($x->item($i)->childNodes->item(0)->nodeValue == $q)
        {
            $y=($x->item($i)->parentNode);
        }
    }
}
$cd=($y->childNodes);
for ($i=0;$i<$cd->length;$i++)
{
    //Process only element nodes
    if ($cd->item($i)->nodeType==1)
    {
        echo($cd->item($i)->nodeName);
        echo(": ");
        echo($cd->item($i)->childNodes->item(0)->nodeValue);
        echo("<br />");
    }
}
?>

```

JavaScript ကနေ PHP page ကို query လှမ်းပို့ပြီး ခေါ်လိုက်တဲ့အခါ အောက်ပါအတိုင်းဖြစ်ပွားပါတယ်။

1. PHP က cd_catalog.xml file ရဲ့ XML DOM object ကို ဖန်တီးလိုက်ပါတယ်။
2. artist elements (nodetypes=1) တွေအကုန်လုံးက JavaScript ကနေပို့လိုက်တာနဲ့ match ဖြစ်တဲ့ အမည်ကို loop ပတ်ပြီးရှာပါတယ်။
3. မှန်ကန်တဲ့ artist ပါတဲ့ CD ကိုတွေ့သွားပါတယ်။
4. album အချက်အလက်တွေကို Output အနေနဲ့ရပြီးတော့ txtHint placeholder ဆီကိုလှမ်းပို့လိုက်ပါတယ်။

နောက်ထပ် ဥပမာတစ်ခုပြီးသွားပါပြီ။ ဒီဟာလည်းပျော်စရာကောင်းမယ်လို့ထင်ပါတယ်။

PHP and AJAX RSS Reader

RSS Reader ကိုသုံးပြီးတော့ RSS Feeds တွေကိုဖတ်လိုရပါတယ်။ RSS ဆိုတာကတော့ သတင်းတွေ up to date အချက်အလက်တွေကို မြန်မြန်ဆန်ဆန် သိရှိစေတဲ့ဟာ ဖြစ်ပါတယ်။

AJAX RSS Reader

အခုအောက်က ဥပမာမှာတော့ webpage ကို refresh လုပ်စရာမလိုဘဲနဲ့ RSS ရဲ့ content တွေကို load လုပ်ပေးတဲ့ RSS reader တစ်ခုကို ရေးမှာဖြစ်ပါတယ်။

Select an RSS-Feed:

RSS Feed will be listed here.

ဒီဥပမာမှာတော့ HTML page တစ်ခု၊ JavaScript တစ်ခု နဲ့ PHP တစ်ခုပါဝင်မှာဖြစ်ပါတယ်။

The HTML Form

ဒီဖောင်လေးမှာ getrss.js ဆိုတဲ့ JavaScript ကိုလှမ်းချိတ်ထားပါတယ်။

```
<html>
<head>
<script type="text/javascript" src="getrss.js"></script>
</head>
<body>
<form>
Select an RSS-Feed:
<select onchange="showRSS(this.value)">
<option value="Google">Google News</option>
<option value="MSNBC">MSNBC News</option>
</select>
</form>
<p><div id="rssOutput">
<b>RSS Feed will be listed here.</b></div></p>
</body>
</html>
```

ကျနော်တို့တွေတဲ့အတိုင်းပဲ ဒီ HTML page မှာ drop-down box တစ်ခုနဲ့ HTML form တစ်ခုပါဝင်ပါတယ်။

user က drop down box ထဲကနေ option တစ်ခုကိုရွေးလိုက်တဲ့အခါ event တစ်ခု စတင်ဖြစ်ပါတယ်။ အဲဒီအခါမှာ showRSS() ဆိုတဲ့ function ကိုလှမ်းခေါ်ပါတယ်။

အောက်မှာပါတဲ့ rssOutput ဆိုတဲ့ <div> ကတော့ showRSS() function ကနေ return ပြန်လာတဲ့ data တွေကို ပေါ်ပေးမယ့် placeholder အနေနဲ့သုံးတာဖြစ်ပါတယ်။

The JavaScript

အောက်က code ကတော့ getrss.js ဆိုတဲ့ JavaScript code ပဲဖြစ်ပါတယ်။

```
var xmlHttp
function showRSS(str)
{
```

```

xmlHttp=GetXmlHttpRequestObject()
if (xmlHttp==null)
{
    alert ("Browser does not support HTTP Request")
    return
}
var url="getrss.php"
url=url+"?q="+str
url=url+"&sid="+Math.random()
xmlHttp.onreadystatechange=stateChanged
xmlHttp.open("GET",url,true)
xmlHttp.send(null)
}

function stateChanged()
{
    if (xmlHttp.readyState==4 || xmlHttp.readyState=="complete")
    {
        document.getElementById("rssOutput")
        .innerHTML=xmlHttp.responseText
    }
}

function GetXmlHttpRequestObject()
{
    var xmlHttp=null;
    try
    {
        // Firefox, Opera 8.0+, Safari
        xmlHttp=new XMLHttpRequest();
    }
    catch (e)
    {
        // Internet Explorer
        try
        {
            xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)
        {
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttp;
}

```

JavaScript code ကနေလှမ်းခေါ်လိုက်တဲ့ getrss.php ဆိုတဲ့ PHP file လေးကတော့ အောက်ပါအတိုင်းပါ.. သူကတော့ server page ပေါ့။

```

<?php
//get the q parameter from URL
$q=$_GET["q"];
//find out which feed was selected
if($q=="Google")
{
    $xml=( "http://news.google.com/news?ned=us&topic=h&output=rss" );
}
elseif($q=="MSNBC")
{
    $xml=( "http://rss.msnbc.msn.com/id/3032091/device/rss/rss.xml" );
}
$xmlDoc = new DOMDocument();
$xmlDoc->load($xml);
//get elements from "<channel>"
$channel=$xmlDoc->getElementsByTagName('channel')->item(0);
$channel_title = $channel->getElementsByTagName('title')
->item(0)->childNodes->item(0)->nodeValue;
$channel_link = $channel->getElementsByTagName('link')
->item(0)->childNodes->item(0)->nodeValue;

```

```

$channel_desc = $channel->getElementsByTagName('description')
->item(0)->childNodes->item(0)->nodeValue;
//output elements from "<channel>"
echo("<p><a href='" . $channel_link
. "'>" . $channel_title . "</a>");
echo("<br />");
echo($channel_desc . "</p>");
//get and output "<item>" elements
$xmlDoc->getElementsByTagName('item');
for ($i=0; $i<=2; $i++)
{
    $item_title=$xmlDoc->item($i)->getElementsByTagName('title')
->item(0)->childNodes->item(0)->nodeValue;
    $item_link=$xmlDoc->item($i)->getElementsByTagName('link')
->item(0)->childNodes->item(0)->nodeValue;
    $item_desc=$xmlDoc->item($i)->getElementsByTagName('description')
->item(0)->childNodes->item(0)->nodeValue;
    echo ("<p><a href='" . $item_link
    . "'>" . $item_title . "</a>");
    echo ("<br />");
    echo ($item_desc . "</p>");
}
?>

```

JavaScript ကနေ option တစ်ခုလွှမ်းမိုးလိုက်တဲ့အခါ အောက်ပါအတိုင်း အလုပ်လုပ်ပါတယ်။

- ဘယ် RSS feed ကိုရွေးထားတယ်ဆိုတာ PHP ကစစ်ပါတယ်။
- XML DOM object တစ်ခုကို ရွေးချယ်ထားတဲ့ RSS feed အတွက် ဖန်တီးလိုက်ပါတယ်။
- RSS channel ထဲက တွေ့ရှိလာတဲ့ element တွေကို output ပြပေးပါတယ်။
- RSS items တွေထဲက ပထမဆုံးသုံးခုက loop ပတ်နေပြီးတော့ output ပြပေးပါတယ်။

အခုအချိန်မှာ rssreader.html ကို run မယ်ဆိုရင် ရပါပြီခင်ဗျာ။

PHP AND AJAX POLL

အောက်က Ajax example မှာ web page က reload လုပ်စရာမလိုဘဲနဲ့ result ကို ပြန်ပြနေနိုင်တဲ့ Poll တစ်ခုကို ပြသသွားမှာ ဖြစ်ပါတယ်။

ဒီဥပမာမှာလည်း စာမျက်နှာလေးခုပါဝင် ပါမယ်။

- HTML form တစ်ခု
- JavaScript တစ်ခု
- PHP page တစ်ခု နဲ့
- Yes, No result တွေကို သိမ်းဖို့ text file တစ်ခု တို့ပါဝင်ပါတယ်။

The HTML Form

အခုအရင်ဆုံး ကတော့ HTML စာမျက်နှာဖြစ်ပါတယ်။ ရိုးရိုး HTML form တစ်ခုပဲဖြစ်ပြီးတော့ JavaScript ဖိုင်ကိုလွှမ်းမိုး link ချိတ်ထားပါတယ်။

```

<html>
<head>
<script src="poll.js"></script>
</head>
<body>
<div id="poll">
<h2>Do you like PHP and AJAX so far?</h2>
<form>
Yes:
<input type="radio" name="vote"
value="0" onclick="getVote(this.value)">
<br />No:
<input type="radio" name="vote"
value="1" onclick="getVote(this.value)">
</form>
</div>
</body>
</html>

```

ဒါလေးကို ajaxpoll.html ဆိုပြီး save ထားလိုက်ပါတယ်။

သူ့ရဲ့အလုပ်လုပ်ပုံလေးကတော့ <div> tag ထဲမှာ Yes နဲ့ No ဆိုတဲ့ radio button လေး နှစ်ခုပါဝင်တဲ့ form လေးရှိပါတယ်။ user က yes ဒါမှ မဟုတ် no ကိုရွေးလိုက်တဲ့အခါမှာ event တစ်ခု အလုပ်လုပ်ပါတယ်။ အဲဒီ event ဖြစ်တဲ့အခါမှာ getVote() ဆိုတဲ့ function ကိုခေါ်ပြီး execute လုပ်ပါတယ်။ getVote() function ကနေပြန်လာတဲ့ data တွေကို form မှာ replace ပြန်လုပ်ပါတယ်။

The Text File

text file ကိုတော့ poll_result.txt ဆိုပြီးတော့ save ပါမယ်။ poll ကရတယ်.. data တွေကိုသိမ်းဖို့ပဲဖြစ်ပါတယ်။

အောက်ကပုံစံအတိုင်းသိမ်းမှာပါ။

0||0

ထဲမလာတဲ့ နံပါတ်က Yes votes တွေကို ကိုယ်စားပြုပြီးတော့ || ရဲ့နောက်က နံပါတ်ကတော့ No votes တွေကိုကိုယ်စားပြုပါတယ်။ ဥပမာ.. ကျနော်တို့ Yes ကိုတခါ vote လိုက်ရင် အရှေ့က 0 က 1 ဖြစ်သွားပါမယ်။

The JavaScript

JavaScript code တွေကိုတော့ poll.js နဲ့သိမ်းပါမယ်။

```
var xmlhttp

function getVote(int)
{
    xmlhttp=GetXmlHttpRequest()
    if (xmlhttp==null)
    {
        alert ("Browser does not support HTTP Request")
        return
    }
    var url="poll_vote.php"
    url=url+"?vote="+int
    url=url+"&sid="+Math.random()
    xmlhttp.onreadystatechange=stateChanged
    xmlhttp.open("GET",url,true)
    xmlhttp.send(null)
}

function stateChanged()
{
    if (xmlhttp.readyState==4 || xmlhttp.readyState=="complete")
    {
        document.getElementById("poll").
        innerHTML=xmlhttp.responseText;
    }
}

function GetXmlHttpRequest()
{
    var objXMLHttpRequest=null
    if (window.XMLHttpRequest)
    {
        objXMLHttpRequest=new XMLHttpRequest()
    }
    else if (window.ActiveXObject)
    {
        objXMLHttpRequest=new ActiveXObject("Microsoft.XMLHTTP")
    }
    return objXMLHttpRequest
}
```

Yes ဒါမှမဟုတ် No ကို HTML form ထဲမှာ select လုပ်လိုက်တဲ့အခါ getVote() function ကိုလာအလုပ်လုပ်ပါတယ်။

အခုနောက်ဆုံးကတော့ JavaScript ထဲကနေလှမ်းခေါ်ထားတဲ့ poll_vote.php ပဲဖြစ်ပါတယ်။

```
<?php
$vote = $_REQUEST['vote'];
//get content of textfile
$filename = "poll_result.txt";
$content = file($filename);
//put content in array
$array = explode("||", $content[0]);
$yes = $array[0];
$no = $array[1];
if ($vote == 0)
{
    $yes = $yes + 1;
}
if ($vote == 1)
{
    $no = $no + 1;
}
//insert votes to txt file
$insertvote = $yes."||".$no;
$fp = fopen($filename,"w");
fputs($fp,$insertvote);
fclose($fp);
?>
<h2>Result:</h2>
<table>
<tr>
<td>Yes:</td>
<td>
'
height='20'>
<?php echo(100*round($yes/($no+$yes),2)); ?>%
</td>
</tr>
<tr>
<td>No:</td>
<td>
'
height='20'>
<?php echo(100*round($no/($no+$yes),2)); ?>%
</td>
</tr>
</table>
```

Yes (0) , No (1) စတဲ့တန်ဖိုးတွေကို JavaScript ကနေလှမ်းပို့ပေးပါတယ်။ အဲဒီနောက်မှာတော့

- poll_result.txt file ရဲ့ content ကိုယူပါတယ်။
- အဲဒီ content တွေကို variable ပြောင်းပြီးတော့ variable တစ်ခုကို JavaScript ကပို့လာတဲ့အတိုင်း တစ်တိုးပါတယ်။
- အဲဒီ တစ်တိုးပြီးတဲ့ result ကို poll_result.txt ဖိုင်မှာရေးလိုက်ပြီး update လုပ်ပါတယ်။
- poll result ရဲ့ graphical representation ကိုထုတ်ပြလိုက်ပါတယ်။

အခုအချိန်မှာ ajaxpoll.html ကို run လိုက်မယ်ဆိုရင် မျှော်လင့်ထားသလိုပဲ Do you like PHP & AJAX so far? ဆိုတဲ့ poll လေးကို Yes, No button နှစ်ခုနဲ့တွေ့ရမှာဖြစ်ပါတယ်။ ajaxpoll.html ဖိုင်ကို လိုသလိုပြင်ပြီးတော့ ဥပမာ Do you like PHP & AJAX so far ဆိုတဲ့နေရာမှာ နှင်ငါ့ကိုချစ်လား... Yes နေရာမှာ ချစ်တယ် လို့ ထားလိုက်No နေရာမှာ Yes လို့ပြောင်းပြီးရေးလိုက်ပေါ့ . ပြီးရင် run ကြည့်လိုက်ပေါ့... :P

Chapter 6.PHP Database

ကျွန်တော်တို့အနေနဲ့ တကယ်ဆို PHP, My SQL Database နဲ့ Apache Web Server တို့ကို တစ်ခုစီသီးသန့် သွင်းပြီးချိတ်ဆက် အသုံးပြုသင့်ပေမယ့် ဒီစာအုပ်ရဲ့ ရည်ရွယ်ချက်က မြန်မြန် နဲ့ လွယ်လွယ် PHP ရဲ့အခြေခံတွေကို နားလည်စေချင်တာ ဖြစ်တဲ့အတွက် XAMPP ကိုသာ အဆင်သင့် Install လုပ်ပြီး လေ့လာကြတာဖြစ်ပါတယ်..။ သင်ခန်းစာတွေမှာ လွယ်ကူစေဖို့အတွက် ကျနော်တို့ရဲ့ PHP ကို လေ့လာရာမှာ... အသုံးပြုနေတဲ့ XAMPP နဲ့ သူ့ရဲ့ တွဲပါတဲ့ PHP Myadmin စတဲ့ tool တွေနဲ့ ပါတ်သတ်ပြီး ဒီအခန်းထဲမှာပဲ My SQL Database ကိုလေ့လာရင်း တစ်ပါတည်း လေ့လာသွားမှာဖြစ်ပါတယ်..။

PHP Database

PHP MySQL Introduction

What is MySQL?

MySQL ဆိုတာကတော့ ပေါ်ပျူလာအဖြစ်ဆုံး open source database system ပဲဖြစ်ပါတယ်။ MySQL ဆိုတာ database ပါ။ MySQL ထဲမှာ ဒေတာတွေကို tables လို့ခေါ်တဲ့ database objects တွေထဲမှာသိမ်းဆည်းပါတယ်။ table ဆိုတာကတော့ ဆက်နွယ်နေတဲ့ ဒေတာတွေစုပေါင်းပါဝင်တဲ့ column တွေ rows တွေပါဝင်တဲ့အစုအဝေးတစ်ခုပါ။ အခုအချိန်မှာ Database ရဲ့အခြေခံလေးနည်းနည်းကို နားလည်အောင်ကြည့်ကြရအောင်လား..။

Database Tables

database တစ်ခုမှာ table တစ်ခုဒါမှမဟုတ် အများကြီးပါဝင်ပါတယ်။ table တစ်ခုစီမှာ အမည်ရှိပါမယ် ဥပမာ (Customers, Orders). Table တွေမှာ rows လို့ခေါ်တဲ့ records တွေပါဝင်ပါတယ်။အောက်ကဥပမာကတော့ Persons လို့ခေါ်တဲ့ table ရဲ့ ဥပမာလေးပါ။

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

ဒီ table မှာဆိုရင် လူတစ်ယောက်အတွက် တစ်ခု စုစုပေါင်း သုံးခု records (rows) သုံးခုနဲ့ column(LastName, firstName, Address နဲ့ City) ဆိုပြီးတော့ လေးခုပါဝင်ပါတယ်။

Queries

query ဆိုတာကတော့ မေးခွန်း ဒါမှမဟုတ် တောင်းဆိုမှုလို့ပြောနိုင်ပါတယ်။

အောက်က query လေးကိုကြည့်ပါ။

SELECT LastName FROM Persons

ဒီ query လေးက Persons table ထဲက LastName ဆိုတဲ့ column ထဲက အချက်အလက်တွေအားလုံးကိုရွေးထုတ်ပေးမှာ ဖြစ်ပါတယ်။ အောက်ကပုံစံအတိုင်းထွက်လာပါမယ်။

LastName
Hansen
Svendson
Pettersen

Create a Connection to a MySQL Database

database တစ်ခုထဲက data တွေ access မလုပ်နိုင်ခင်မှာပဲ database အတွက် connection ကိုဖန်တီးပေးရပါမယ်။ PHP မှာတော့ mysql_connect() function ကိုသုံးပြီးလုပ်နိုင်ပါတယ်။

Syntax

mysql_connect(servername,username,password);

Parameter	Description
servername	Optional. Specifies the server to connect to. Default value is "localhost:3306"
username	Optional. Specifies the username to log in with. Default value is the name of the user that owns the server process
password	Optional. Specifies the password to log in with. Default is ""

အောက်က code ထဲမှာ ဆိုရင် ကျနော်တို့အနေနဲ့ connection ကို (\$con) ဆိုတဲ့ variable လေးထဲမှာသိမ်းပါတယ်.. script ထဲမှာ နောက်ပိုင်းမှာပြန်လည်းခေါ်အသုံးပြုဖို့အတွက်ပါ။ အကယ်၍ connection fail ဖြစ်ခဲ့ရင်တော့ die ကိုအလုပ်လုပ်ပါမယ်။

```
<?php
$con=mysql_connect("localhost","user","password");
if (!$con)
{
    die('Could not connect:' > mysql_error());
}
//some code
?>
```

PHP Script ဆုံးသွားတဲ့အခါမှာ connection အလိုအလျောက် ပိတ်သွားပါတယ်။ ကြိုပိတ်ချင်ရင်တော့ mysql_done() function ကိုသုံးနိုင်ပါတယ်။

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// some code
mysql_close($con);
?>
```

PHP MySQL Create Database and Tables

Create a Database

MySQL မှာ database တစ်ခုကို ဖန်တီးဖို့အတွက် CREATE DATABASE statement ကိုအသုံးပြုပါတယ်။

Syntax

CREATE DATABASE database_name

အခုက PHP tutorial ဖြစ်တဲ့အတွက် လိုအပ်သလောက် database အပိုင်းကိုပဲရေးမှာဖြစ်ပါတယ်..။ ဒီအတွက် SQL language ကို သီးသန့် ထပ်မံလေ့လာသင့်ပါကြောင်း...။

အထက်ပါ စာကြောင်းကို execute လုပ်ဖို့အတွက် mysql_query() function ကိုအသုံးပြုရပါမယ်။ ဒီ function ကတော့ MySQL connection ဆီကို query ဒါမှမဟုတ် command တစ်ခုပို့ဖို့အတွက် အသုံးပြုတာပါ။

အောက်ပါဥပမာကတော့ my_db လို့ခေါ်တဲ့ database တစ်ခုကို ဖန်တီးတာပါ။

```
<?php
$con = mysql_connect("localhost","user","moemyintshein");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

if (mysql_query("CREATE DATABASE my_db",$con))
```

```

{
    echo "Database created";
}
else
{
    echo "Error creating database: " . mysql_error();
}

```

```
mysql_close($con);
```

```
?>
```

Table တစ်ခုတည်ဆောက်ခြင်း

CREATE TABLE statement ကိုသုံးပြီးတော့ MySQL မှာ table တစ်ခုကိုတည်ဆောက်ရပါတယ်။ သူ့ရဲ့ syntax ကတော့

```
CREATE TABLE table_name
```

```

(
    column_name1 data_type,
    column_name2 data_type,
    column_name3 data_type,

```

```
...
```

```
)
```

အခုလိုပုံစံပဲဖြစ်ပါတယ်။ SQL အကြောင်း သင်ချင်ရင် SQL Tutorial ကိုလေ့လာပါခင်ဗျာ။

PHP ထဲမှာထည့်သုံးဖို့အတွက် mysql_query() function ထဲကို CREATE TABLE statement ကိုပေါင်းထည့်ပြီး execute လုပ်မှာဖြစ်ပါတယ်။

ဥပမာ။ ။

အောက်ပါဥပမာမှာ Persons ဆိုတဲ့ table တစ်ခုကိုဖန်တီးပြီးတော့ သူ့မှာ FirstName, LastName နဲ့ Age ဆိုတဲ့ Column သုံးခု ပါဝင်မှာဖြစ်ပါတယ်။

```

<?php
$con = mysql_connect("localhost","root","moemyintshein");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

// Create database
if (mysql_query("CREATE DATABASE my_db1",$con))
{
    echo "Database created";
}
else
{
    echo "Error creating database: " . mysql_error();
}

// Create table
mysql_select_db("my_db1", $con);
$sql = "CREATE TABLE Persons
(
    FirstName varchar(15),

```

```

LastName varchar(15),
Age int
);

// Execute query
mysql_query($sql,$con);

```

```

mysql_close($con);
?>

```

မှတ်သားရန်- table တစ်ခုကို မဖန်တီးခင်မှာ database ကိုရွေးပေးရပါတယ်။ mysql_select_db() ဆိုတဲ့ function ကိုသုံးပြီးတော့ database ကိုရွေးယူနိုင်ပါတယ်။ varchar type ကိုသုံးပြီးတော့ database field တစ်ခုကို ဖန်တီးတဲ့အခါမှာ အဲဒီ filed ရဲ့ maximum length ဥပမာ varchar(15) ဆိုပြီးတော့ သတ်မှတ်ပေးရပါမယ်။ data type ကနေပြီးတော့ ဒီ column ကကိုင်တွယ်နိုင်တဲ့ data အမျိုးအစားကိုသတ်မှတ်ပေးတာဖြစ်ပါတယ်။

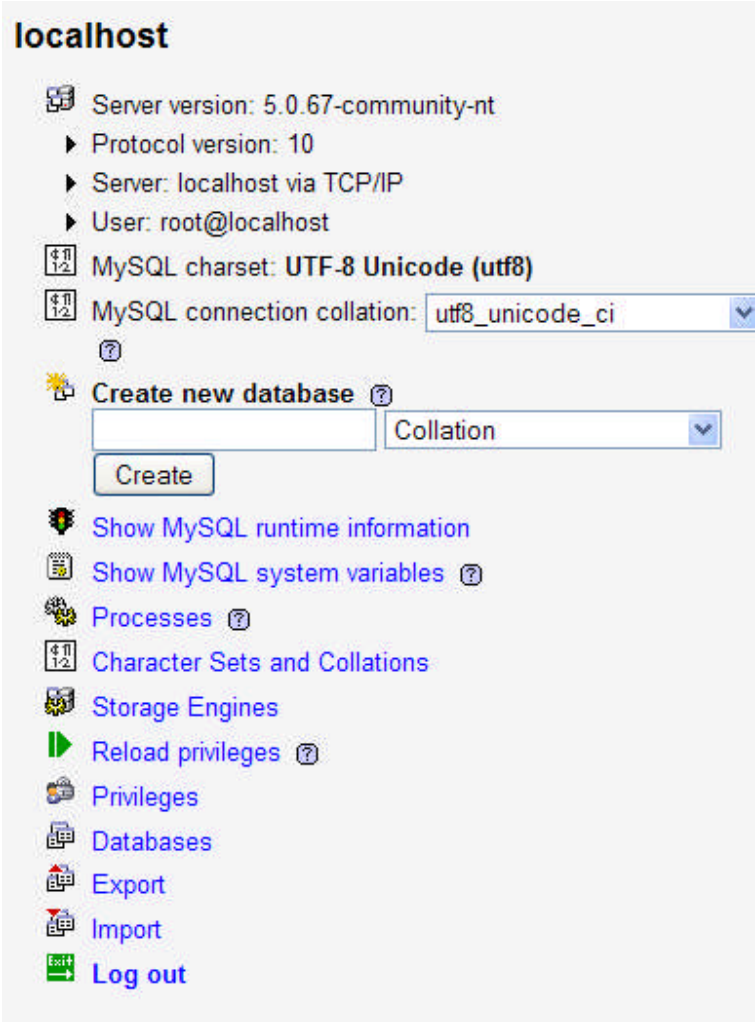
Working with User Privileges in MySQL

အပေါ်မှာ ပုံစံပြပြီးတဲ့အခါ အခုတော့ ကျနော်တို့ လက်တွေ့ လေ့ကျင့်ကြရအောင်လား...။ ကျနော်တို့အနေနဲ့ PHP, MySQL နဲ့ Apache အားလုံးပါဝင်ပြီးသား Package ဖြစ်တဲ့ XAMPP ကိုအသုံးပြုနေတာဖြစ် ပါတယ်။ အလိုအလျောက်အနေနဲ့ root ဆိုတဲ့ username နဲ့ password မပါဘဲ ဖန်တီးထားပေးမှာဖြစ်ပါတယ်...။ အခု Database တွေနဲ့ ကျနော်တို့ ထောင်အလုပ်လုပ်တော့မှာဖြစ်လို့ အမှန်တကယ် username နဲ့ password တစ်ခုကို ဖန်တီးသင့်ပါတယ်...။ MySQL ရဲ့ User အခွင့်အရေးတွေကို အနည်းငယ်လောက် ဖြစ်ဖြစ် နားလည်ဖို့လိုအပ်ပါတယ်...။

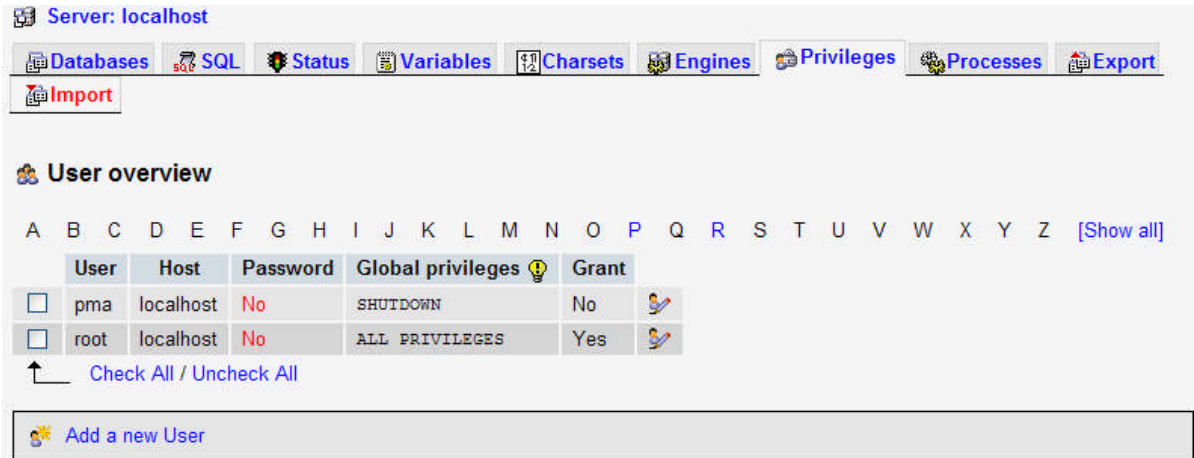
Creating a New User


Database တွေနဲ့ အလုပ်လုပ်တော့မယ်ဆိုရင် <http://localhost/phpmyadmin/> ဆိုပြီးဝင်လိုက်ပါ.... PHP My Admin ဆိုတဲ့ tool ကိုအသုံးပြုပြီး Database တွေနဲ့ အလုပ်လုပ်ရမှာဖြစ်ပါတယ်...။ User အသစ်တစ်ယောက်ကို SQL Statement ကိုသုံးပြီး ဖန်တီးနိုင်သလို ဒါမှမဟုတ် PHP Admin ထဲမှာအသင့်ပါတဲ့ Create New User ကိုနှိပ်ပြီးတော့လဲ လွယ်ကူစွာဖန်တီးနိုင်ပါတယ်...။

<http://localhost/phpmyadmin/> လို့ Web Browser မှာရိုက်တဲ့အခါ အောက်ကနေရာကိုရောက်လာမှာဖြစ်ပါတယ်... အဲဒီမှာ Privileges ဆိုတဲ့နေရာကိုဝင်လိုက်ပါ...။



Privileges ဆိုတဲ့နေရာကိုဝင်လိုက်တဲ့အခါ အောက်ပါတိုင်း Create a new User ဆိုတာကိုတွေ့ပါမယ်.. အဲဒီ လင့်လေးကို နှိပ်ပြီး user အသစ်ကိုဖန်တီးနိုင်ပါတယ်။



 **Add a new User**

Login Information

User name:

Use text field: ▼

moemyintshein

Host:

Local ▼

localhost

Password:

Use text field: ▼

●●●●●●●●●●

Re-type:

●●●●●●●●●●

Generate Password:

Generate

Copy

Database for user

☒ None

☐ Create database with same name and grant all privileges

☐ Grant all privileges on wildcard name (username_%)

Global privileges (Check All / Uncheck All)

Note: MySQL privilege names are expressed in English

Data

☒ SELECT

☒ INSERT

☒ UPDATE

☒ DELETE

☒ FILE

Structure

☒ CREATE

☒ ALTER

☒ INDEX

☒ DROP

☒ CREATE TEMPORARY TABLES

☒ CREATE VIEW

Administration

☒ GRANT

☒ SUPER


☒ PROCESS

☒ RELOAD

☒ SHUTDOWN

☒ SHOW DATABASES

ကျနော်အနေနဲ့ Username နေရာမှာ moemyintshein host ကို localhost နဲ့ password ကို moemyintshein လို့ပေးလိုက်ပြီးတော့ Global privileges မှာ Check All လုပ်ပေးပါတယ်.. ပြီးတဲ့ အခါ Go ကိုနှိပ်ပြီးတော့ user အသစ်ကိုဖန်တီးလိုက်ပါတယ် ခင်ဗျာ..။ အဲဒီအခါမှာ အောက်ပါအတိုင်း SQL Statement တွေကို အလိုအလျောက်ဖန်တီးပေးပြီးတော့ execute လုပ်ထားကာ.. you have added a new user ဆိုတဲ့ Message ပြမှာဖြစ်ပါတယ်.. ဒါဆိုရင်တော့ user အသစ်ကိုဖန်တီးပြီးသွားပါပြီ..။

 You have added a new user.

SQL query:

```
CREATE USER 'moemyintshein'@'localhost' IDENTIFIED BY '*****';

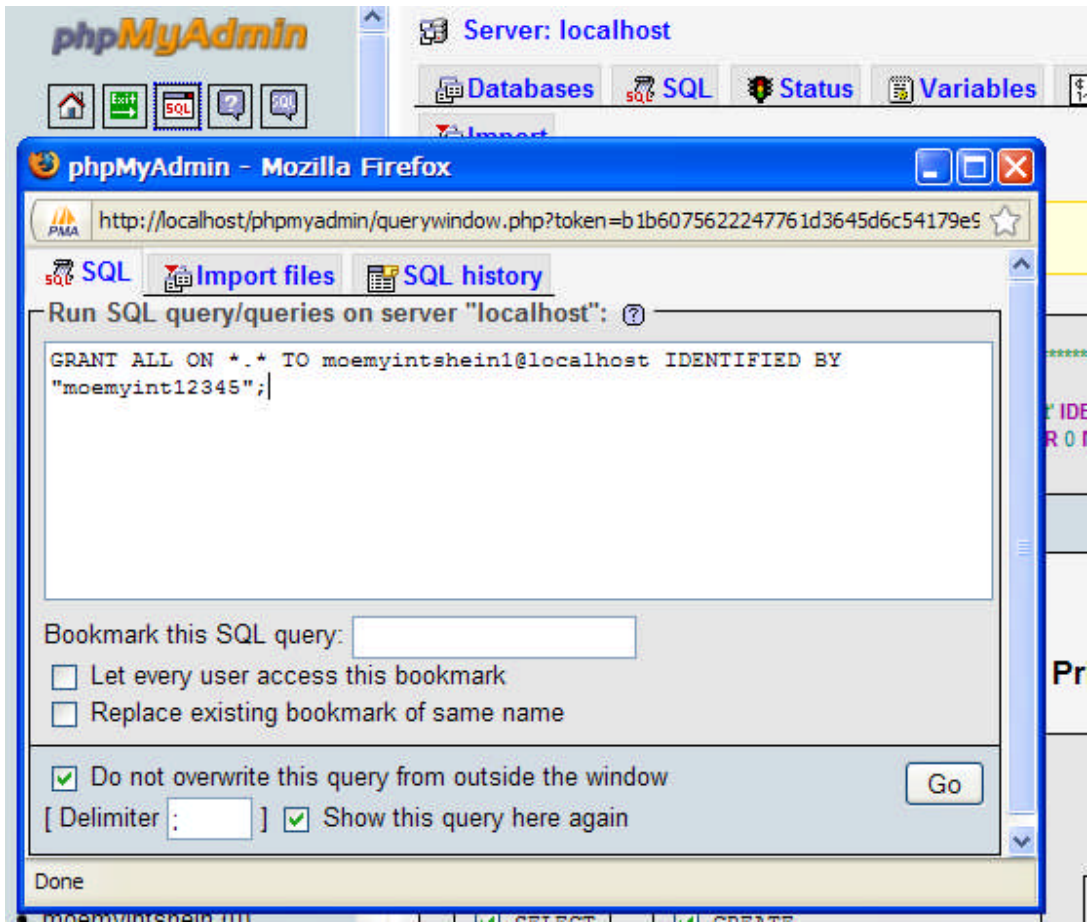
GRANT ALL PRIVILEGES ON * . * TO 'moemyintshein'@'localhost' IDENTIFIED BY '*****' WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
```

အကယ်လို့ ကိုယ့်အနေနဲ့ အစကတည်းက SQL statement ကိုပဲ သုံးပြီးဖန်တီးမယ်ဆိုရင်လဲရပါတယ်..။ ခုနက moemyintshein ဆိုတဲ့ နာမည်နဲ့ localhost ဆိုတဲ့ ကျနော်တို့ Server မှာ user အသစ်ဖန်တီးထားတဲ့အတွက် အခုတစ်ခါတော့ SQL statement ကိုသုံးပြီး username ကို moemyintshein1 password ကို moemyint12345 လို့ပေးပြီး အသစ်ထပ်ဖန်တီးပါမယ်ခင်ဗျာ...

69

www.moemyintshein.com

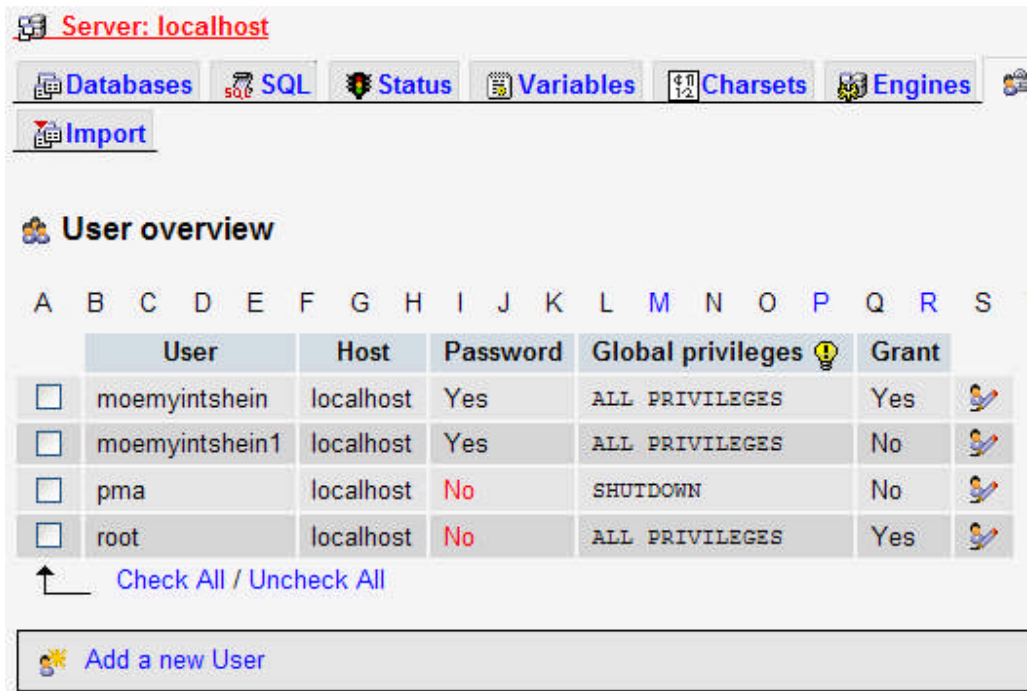
69



အထက်ပါပုံအတိုင်း PHP Admin main page ရဲ့ ညာဘက်ထောင့်အပေါ်နားလေးမှာ button ငါးခုရှိတဲ့အနက် အလယ်က SQL ဆိုတာလေးကို နှိပ်လိုက်တဲ့အခါ သူ့အောက်မှာ SQL query ကို execute လုပ်နိုင်တဲ့ pop up page တစ်ခု တတ်လာပါတယ်.. အဲဒီအထဲမှာ ကျနော်က

GRANT ALL ON *.* TO moemyintshein1@localhost IDENTIFIED BY "moemyint12345";
လိုရှိတာကို Go ကိုနှိပ်လိုက်တဲ့အခါ localhost ဆိုတဲ့ Server မှာ moemyint12345 ဆိုတဲ့ password နဲ့ moemyintshein1 ဆိုတဲ့ user အသစ်တစ်ယောက်ကိုလည်း ထပ်ဖန်တီးလိုက်ပြီဖြစ်ပါတယ်။

အခုအချိန်မှာ privileges ထဲမှာ ဝင်ကြည့်တဲ့အခါ အောက်ပါတိုင်း အစက root ပဲရှိရာကနေ user အသစ်နှစ်ယောက်ပါ ရှိနေတာဖြစ်ပါတယ်...။



PHP Guide Book Part (I) ကိုဒီနေရာမှာပဲရပ်နားလိုက်ပါတယ်။

PHP Guide Book Part (II) ကို 2010 Aug, 13th မှာ www.moemyintshein.com မှာ လာရောက်ဒေါင်းလုပ်ရယူနိုင်မှာ ဖြစ်ပါတယ်။ကျေးဇူးတင်ပါတယ်။ အကြံပြုချက်များကို အီးမေး moemyintsheinster@gmail.com သို့ပေးပို့နိုင်ပါတယ်။

Moe Myint Shein