



PHP Guide Book

First Edition

Moe Myint Shein

PHP GUIDE BOOK

First Edition

Moe Myint Shein

အမှာစကား

မင်္ဂလာပါခင်ဗျာ။ ကျနော် Part (I) ပါဆိုပြီးတော့ ပြီးခဲ့တဲ့ လက တစ်အုပ် ထုတ်ခဲ့ပါတယ်။ ၂၀၁၀ ဩဂုတ် ၁၃ မှာ Part (II) ထုတ်မယ်လို့ပြောခဲ့ပါတယ်။ အဲလိုမျိုးဆို နှစ်အုပ်ဖြစ်ပြီး ရှုပ်နေမယ်ထင်ပါတယ်။ တကယ်ရေးပြီးတဲ့အခါမှာ First edition လို့ ထုတ်တာက ပိုပြီး သင့်တော်တယ်လို့ ယူဆမိပါတယ်။ ဘာရယ်မဟုတ်ပါဘူး အရှေ့က တစ်အုပ်နဲ့ အမည်ကွဲအောင်ပါ။ Second edition လာလိမ့်မယ်လို့တော့ မထင်ပါဘူး။ ဒီလိုပဲ နာမည်ပေးတာပါ။ ပထမစာအုပ်မှာ စာမျက်နှာ ၇၀ ဝန်းကျင်နဲ့ ထုတ်ပေးခဲ့ပြီးတော့ ဒီ First Edition မှာ ကျနော် နောက်ထပ် စာမျက်နှာ ၇၀ နီးပါး လောက်ထပ်ပေါင်းထည့်လိုက်ပါတယ်။ ဒီစာအုပ်မှာတော့ တတ်နိုင်သမျှ ပထမစာအုပ်မှာ အလောတကြီး လုပ်လိုက်လို့ Chapter တွေသေချာ မစီထားဖြစ်တာတွေ ကအစ ပုံစံတွေပြန်ပြင်ထားပြီး ကြိုးစားရေး ထားပါတယ်။ စာတွေတော့ မပြောင်း သွားပါဘူး။ အဆင်ပြေအောင် အရှေ့အနောက် ပြန်စီစဉ်ထားတာတော့ရှိပါတယ်။

ဒီစာအုပ်ကို လေ့လာပြီးတဲ့အခါ သင် PHP မတတ်သွားပါဘူး။ Web Developer လည်းမဖြစ်ပါ။ ဒါပေမယ့် မဖတ်ခင်က နဲ့ ယှဉ်ရင်တော့ သင်အများကြီး ကွာခြားသွားမှာပါ။ PHP နဲ့ပတ်သတ်ပြီးအခြေခံကောင်းတွေရပြီး ဆက်လက်လေ့လာဖို့ လမ်းပွင့် သွားမယ်လို့ ယုံကြည်ပါတယ်။

ဒီစာအုပ်နဲ့လေ့လာဖို့ Programming အခြေခံရှိက အလွန်ကောင်းပါသည်။ မရှိပါက ဖြည်းဖြည်းချင်း ဖတ်ပါ။ code တိုင်းကို တတ်နိုင်သမျှ 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။ ဘာတွေ အသစ်ရှိနေမလဲဆိုတာကိုလည်း ကျနော် ဘလော့ www.moemyintshein.com မှာလာရောက် လည်ပတ်နိုင် ကြောင်းပါ။

လေ့လာကြသူအားလုံးကို လေးစားလျက်

မိုးမြင့်ရှိန်

Contents at a Glance

Chapter 1	Introducing PHP	5
Chapter 2	The Building Blocks of PHP	7
Chapter 3	Flow Control Functions in PHP	14
Chapter 4	Working With Functions and Forms	25
Chapter 5	PHP Date, Include, File Handling & Some PHP Advanced	32
Chapter 6	PHP and AJAX	54
Chapter 7	PHP MyAdmin Basic	74
Chapter 8	PHP and Database	83
Chapter 9	XML and PHP	97
Chapter 10	Exception Handling and Filter in PHP	124
Conclusion		136

Chapter 1 Introducing 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 ဝဲဖြစ်ပါတယ်။ စတင်ပေါ်ချင်း 1994 မှာ 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။ အခုဒီစာရေးနေစဉ် 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 သင်ယူဖို့မို 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 ဝဲဖြစ်ပါတယ်။ 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။ ညမာ 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။

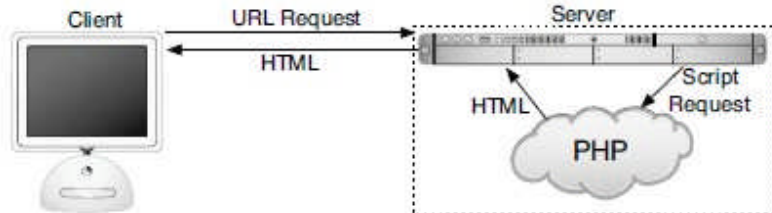


Fig 1.1 PHP ကိုအသုံးပြုထားသော server ထဲမှ user က web page ကို request လုပ်ပုံ
 PHP ကိုလေ့လာတာမယ်ဆိုရင် Apache web server နဲ့ My SQL Database တို့ပါတွဲပြီးပါနေတာဖြစ်ပါတယ်။
 အခုကျွန်တော်တို့ PHP ကိုစတင်လေ့လာဖို့အတွက် လိုအပ်တဲ့ software ကိုသွင်းရပါမယ်။

localhost မှာသုံးဖို့အတွက် လိုအပ်တဲ့ software တွေအားလုံးကို ပေါင်းစုပြီးထည့်ထားတဲ့ package software တွေရှိပါတယ်။ အဲဒီ software တွေကိုသုံးပြီး web server နဲ့ database ကိုကိုယ့်စက်ထဲမှာပဲရရှိမှာဖြစ်ပါတယ်။

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 သွင်းသလိုပါပဲခင်ဗျာ။ သွင်းပြီးသွားရင်တော့ C အောက်က xampp အောက်က htdocs အောက်မှာ php ဆိုတဲ့ folder လေးဆောက်သင့်ပါတယ်။ အဲဒီ အောက်မှာပဲကျနော်တို့ ဇေးသမျှကို save သင့်ပါတယ်။ PHP စာအုပ်တစ်အုပ်ဖြစ်တဲ့အတွက် လုပ်တတ်သင့်တယ်လို့လည်း ယူဆတဲ့အတွက် PHP, Apache နဲ့ MySQL တို့ကို သီးခြားစီ Install လုပ်တဲ့ပုံစံကို Part (II) စာအုပ်မှာ ရေးပေးပါမယ်။ 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 ကိုအသုံးပြုမယ်ဆိုရင်လည်း အဆင်ပြေနိုင်ပါသေးတယ်။

Chapter 2.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 အလိုအလျောက်နဂိုရှိနေတဲ့ 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 စာကြောင်းတစ်ကြောင်းနဲ့ 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) ခြားနားထားသင့်ပါတယ်။

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;
?>
```

သူ့ရဲ့ 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;
?>
```

သူ့ရဲ့ 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 ဆွဲနဲ့ 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 မဟုတ်ဘူး 0 ဖြစ်လို့ပါ။ PHP မှာ string အတွက် function ပေါင်းမြောက်များစွာရှိပါတယ်။ အခုတော့ အခြေခံဖြစ်လို့ ဒီလောက်ထိပဲထည့်လိုက်ပါတယ်။

PHP Operators

အောက်မှာ PHP မှာသုံးတဲ့ operator ဆွဲပဲဖြစ်ပါတယ်။

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 ဆွဲအနေနဲ့ 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 3. 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

ကျွန်တို့ 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 တနားနေ့ဆိုရင်တော့ Have a nice day လို့ပေါ်မှာဖြစ်ပါတယ်။ ကျွန်တို့အခု run တာတော့ Have a nice weekend လို့ပေါ်ပါတယ် ဒီနေ့က သောကြာနေ့ဖြစ်နေလို့ပါ.. အဲဒါနဲ့ Computer ရဲ့ System Date ကို Tuesday လို့ပြောင်းပြီး ကျွန်တို့ပြန် 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 ဆွဲအရှည်ကြီးသုံးရတာတွေကနေလည်း 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 တန်ဖိုးနဲ့ကို 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 တန်ဖိုးတစ်ခုစီနဲ့ဆက်နွှယ်ပါတယ်။ 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);
```

အောက်ကပုံနဲ့လည်းရေးနိုင်ပါတယ်.. 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?

ရမှာဖြစ်ပါတယ်။ 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 4. Working With Functions & Forms

PHP Functions

အမှန်တကယ် PHP ရဲ့ စွမ်းအားက function တွေပါပဲ...။ PHP မှာ built-in နဂိုတည်းကပါတဲ့ 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 သူ့ရဲ့ နာမည်ကိုကြည့်လိုက်တာနဲ့ 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 မှာပြဖို့အတွက်ပါ။ 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>

```

သူ့ရဲ့ 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
```

အလုပ်လုပ်ပုံကတော့ 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"); ဆိုပြီးလွှဲပေးလိုက်တဲ့အခါ \$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...

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

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 နဲ့ပေါင်းပြီးတော့ \$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 တွေနဲ့အလုပ်လုပ်တဲ့အခါ 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။ ဒီဥပမာလေးမှာ 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 ဖော်သွားတာတွေ့ရပါတယ်.. 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"]; ?>.

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"]; ?>.

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 5 .PHP Date, Include & File Handling

The PHP Date() Function

အချိန်နဲ့ နေ့ရက်တွေကို ပုံစံချဖို့အတွက် 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 ယူမ parameter ဖြစ်တဲ့ format က date/time တစ်ခုကို ဘယ်လို format ပေးရမလဲဆိုတာ သတ်မှတ်ပေး ပါတယ်။ နေ့ရက်နဲ့ အချိန်ပုံစံတွေကို ကိုယ်စားပြုဖို့အတွက် 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 ဖြစ်ပါတယ်။ 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 နှေးနိုင်တာဖြစ်ပါတယ်။ include file ကိုအသုံးပြုခြင်းအားဖြင့် menu link တွေရဲ့အစီအစဉ်ကို ပြောင်းလဲချင်တဲ့အခါ အမည်ပြောင်းချင်တဲ့အခါ ဒါမှမဟုတ် site ထဲကို နောက်ထပ် web page ထပ်ထည့်ချင်တဲ့အခါတွေမျိုးမှာ page တစ်ခုစီ file တစ်ခုစီမှာပြင်ဖို့လိုအပ်ဘဲနဲ့ menu.php ထဲမှာရှိတဲ့ code ကိုပဲ ပြင်လိုက်ရုံပဲဖြစ်ပါတယ်။

The require() Function

require () function က include() function အလုပ်လုပ်သလိုပဲ ပုံစံတူအလုပ်လုပ်ပါတယ်။ error ကိုကိုင်တွယ်တဲ့နေရာမှာပဲကွာခြားမှု ရှိတာဖြစ်ပါတယ်။ include() function ကိုသုံးရင် warning ဖေးရုံသာပေးပြီး 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 ဘယ်လိုပဲပြော 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);
?>
```

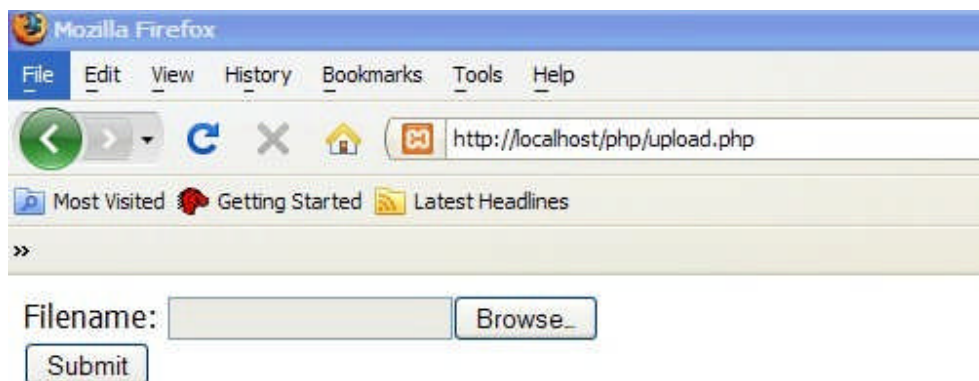
PHP File Upload

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 မှာသိမ်းမယ့် ဖိုင်ရဲ့ ယာယီ ကော်ပီရဲ့နာမည်၊
- \$_FILES["file"]["error"]- ဖိုင် upload လုပ်ရကနေဖြစ်လာနိုင်တဲ့ error code၊ စသည်ဖြင့်ဖြစ်ပါတယ်။

အခုဒီပုံစံက ဖိုင်ကို php နဲ့အရှိရင်းဆုံး 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

အဆင်ကဉ္စမာက 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=to,e()+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 ပြုစားရပါမယ်။ 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။

အခုထပ်ပြမယ့် 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.... reload လုပ်လိုက်တဲ့အခါတိုင်းမှာ views ကတစ်တိုးတိုးသွားတာတွေမှာဖြစ်ပါတယ်။

Destroying a Session

session data ကိုပြန်ဖျက်မယ်ဆိုရင်တော့ unset() function ဒါမှမဟုတ် session_destroy() function ကိုအသုံးပြုနိုင်ပါတယ်။ unset() function ကိုတော့ ဖော်ပြထားတဲ့ session variable ကို free လုပ်ပစ်ဖို့အတွက်အသုံးပြုပါတယ်။

```
<?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 တွေကတော့ Optional ဖြစ်ပါတယ်။

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

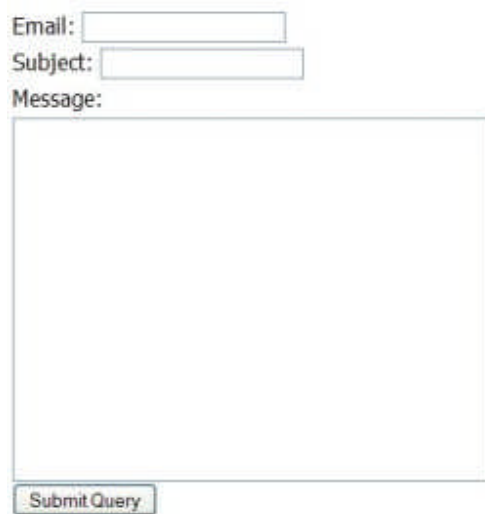
PHP ကိုသုံးပြီးတော့ ကျနော်တို့ရဲ့ website တွေပေါ်မှာ feedback-form လိုမျိုးဖန်တီးနိုင်ပါတယ်။ အောက်ကဥပမာမှာ form ကိုသုံးပြီး အီးမေးပို့တာကိုပြထားတာဖြစ်ပါတယ်။

```
<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>
```



The screenshot shows a web form with three input fields: 'Email:', 'Subject:', and 'Message:'. The 'Message:' field is a large text area. Below the fields is a button labeled 'Submit Query'.

code တွေကို save လုပ်တဲ့အခါ mailform.php လို့ save လုပ်လိုက်ပါတယ်။ သူ့ရဲ့အလုပ်လုပ်ပုံကတော့ ပထမဆုံး email ရဲ့ input field ကိုဖြည့်ထားမထားစစ်ပါတယ်။ မဖြည့်ထားသေးဘူးဆိုရင်တော့ ဆိုလိုတာက ပထမဆုံးအကြိမ် ဒီ စာမျက်နှာကိုရောက်လာတာဆိုရင်တော့ 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 အီးမေးပို့တဲ့ပုံစံပဲဖြစ်ပါတယ် 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 ဆန်ပါဘူး ပြီးတော့ security risk တွေလည်းရှိလာပါမယ်။ 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

ပထမဆုံးဥပမာအနေနဲ့ text file တစ်ခုကို ဖွင့်တဲ့ ရိုးရှင်းတဲ့ code လေးပါ..။ အကယ်၍ ဖွင့်ရုံရင် ဘာ error မှပေါ်မှာမဟုတ် ငါဘူး ဖွင့်မရှိခဲ့ရင်တော့ ကျနော်ဆီမှာတော့ 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 ဖွင့်ရုံမရှိ အရင်စစ်ခိုင်းမယ် ပြီးမှ access လုပ်ခိုင်းပါမယ်။

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

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

Creating a Custom Error Handler

ဒါဟာလည်း ရိုးရှင်းပါတယ်။ 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 ရိုးရှင်းတဲ့ error handling function တစ်ခုပဲ ဖြစ်ပါတယ်။ အဲဒါလေး ကို စ run တဲ့အခါ မှာ error level နဲ့ error message ကိုရယူပြီးတော့ output ပြပေးပြီးတော့ script ကိုအဆုံးသတ်တာပဲဖြစ်ပါတယ်။ ကျနော်တို့ error handling function ကို ဖန်တီးပြီးပြီဆိုရင် ဘယ်အချိန်မှာ trigger လုပ်မယ် (စ 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 လုပ်ဖို့အသုံးဝင် ပါတယ်။ 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 6. 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 လုပ်လိုက်တာမျိုးမဟုတ်ဘဲ 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 လုပ်စရာမလိုဘဲနဲ့ data တွေကို ပို့နိုင် ထုတ်နိုင်တာဖြစ်ပါတယ်။ ဒီကိစ္စကို နောက်ကွယ်မှာ server ဆီကို HTTP request တွေပေးပို့ခြင်းနဲ့အလုပ်လုပ်တာဖြစ်ပါတယ်။ server က data ပြန်လာတဲ့အခါ JavaScript ကိုသုံးပြီး web page ရဲ့အစိတ်အပိုင်းလေးကိုပဲ modify လုပ်တာဖြစ်ပါတယ်။ ရှေ့လာမယ့် ညာမာတွေကိုကြည့်ရင် 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 ဆီကိုပို့တာမဟုတ်ဘဲ အဲဒီ web page ထဲက အချက်အလက် အနည်းငယ်အစိတ်အပိုင်းလေးကိုပဲ ပို့တာဖြစ်ပါတယ်။ ပြောရရင်တော့ AJAX က web server software တွေပေါ်မှာမမှီခိုဘဲ web browser technology တစ်ခုဖြစ်ပါတယ်။ ဒီ tutorial ထဲမှာတော့ PHP server ပေါ်မှာ run တဲ့ညာမာတွေကိုအသားဝေးဖော်ပြပေးပါမယ်။

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 ပိုမိုသိတဲ့ XMLHttpRequest object ကို ပိုနှစ်သက် နိုင်ပါတယ်။ အောက်က ဥပမာလေးကတော့ Microsoft.XMLHTTP အစား Xml2.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;
}
```

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

```

ဒီဥပမာမှဆိုရင် 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 ပေါ်မို့နေရာလေးပဲဖြစ်ပါတယ်။

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=GetXmlHttpRequestObject();
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 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 Document

```

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

1. server ကိုဖေဖို့မယ် 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 လုပ်စေပါတယ်။
5. ဖေးလိုက်တဲ့ url အတိုင်း XMLHttpRequest object ကို ဖွင့်လိုက်ပါတယ်
6. server ဆီကို HTTP request ကိုလှမ်းပို့လိုက်ပါတယ်။

input filed က ယာယီနောက်ဆုံးရင် 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 ပေးမယ်
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 နေ AJAX နည်းပညာကိုသုံးပြီးတော့ XML file တစ်ခုစီက အချက်အလက်တွေကို ဘယ်လိုခေါ်ယူသုံးနိုင် (fetch) လုပ်နိုင်တယ်ဆိုတာကို လေ့လာမှာဖြစ်ပါတယ်။ Select a CD: ဆိုပြီးတော့ စာသားလေးရှိမယ်ဆားမှာ 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 ရှိရင်းတဲ့ HTML form လေးပဲဖြစ်ပါတယ်။ cds ဆိုတဲ့ drop down box လေးပါမယ်။ အဲဒါရဲ့အောက်မှာတော့ txtHint လို့ခေါ်တဲ့ div တစ်ခုပါပါမယ်။ အဲဒီ div ကို web server ကနေပြန်လာမယ့် အချက်အလက်တွေကို ပြပေးမယ့် 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 တန်ဖိုးကို 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 တွေကိုဖတ်လို့ရပါတယ်။ RSS ဆိုတာကတော့ သတင်းတွေ up to date အချက် အလက်တွေကို 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 ဖော်ပေးမယ့် 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
$x=$xmlDoc->getElementsByTagName('item');
for ($i=0; $i<=2; $i++)
{
$item_title=$x->item($i)->getElementsByTagName('title')
->item(0)->childNodes->item(0)->nodeValue;
$item_link=$x->item($i)->getElementsByTagName('link')
->item(0)->childNodes->item(0)->nodeValue;
$item_desc=$x->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 ပြန်ပြမေးနိုင်တဲ့ 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 ဖိုင်ကိုလှမ်းပြီး 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

ထာမလာတဲ့ နံပါတ်က Yes votes တွေကို ကိုယ်စားပြုပြီးတော့ || ရဲ့နောက်က နံပါတ်ကတော့ No votes တွေကိုကိုယ်စား ပြုပါတယ်။ 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 objXMLHttp=null
    if (window.XMLHttpRequest)
    {
        objXMLHttp=new XMLHttpRequest()
    }
    else if (window.ActiveXObject)
    {
        objXMLHttp=new ActiveXObject("Microsoft.XMLHTTP")
    }
    return objXMLHttp
}

```

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 ဇယားကို Yes, No button နှစ်ခုနဲ့တွေ့ရမှာဖြစ်ပါတယ်။ ajaxpoll.html ဖိုင်ကို လိုသလိုပြင်ပြီးတော့ ဥပမာ Do you like PHP & AJAX so far ဆိုတဲ့နေရာမှာ နင်ငါ့ကိုချစ်လား... Yes နေရာမှာ ချစ်တယ် လို့ ထားလိုက်No နေရာမှာ Yes လို့ပြောင်းပြီးရေးလိုက်ပေါ့ . ပြီးရင် run ကြည့်လိုက်ပေါ့... :P

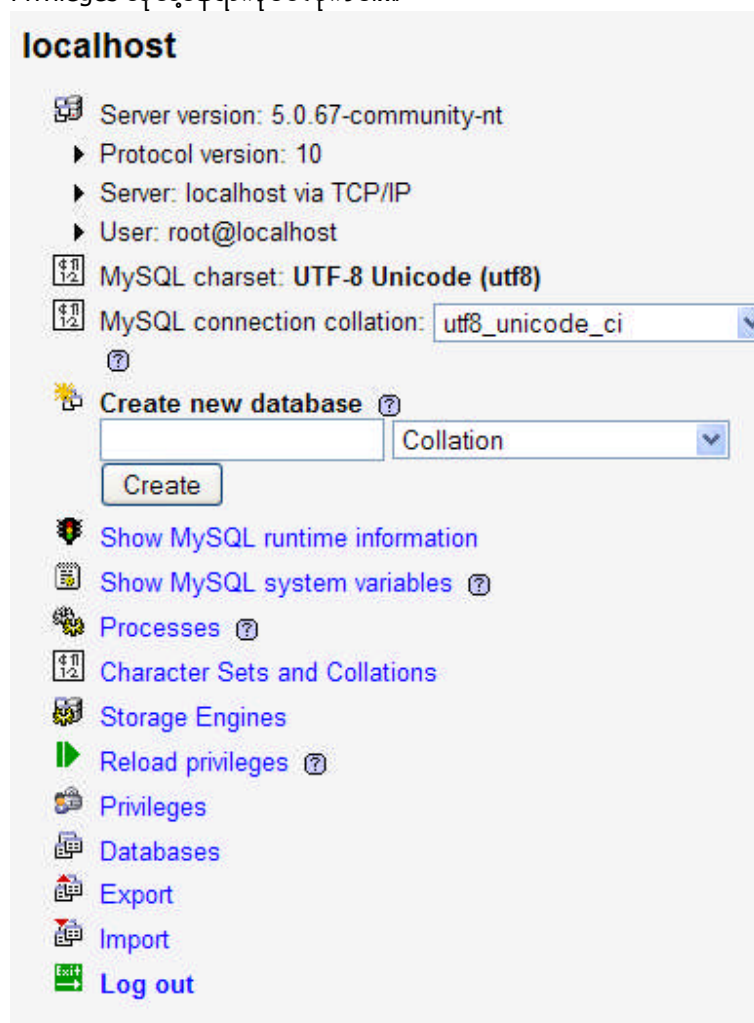
Chapter 7. PHP MyAdmin Basic

ကျွန်ုပ်တို့အနေနဲ့ PHP, MySQL နဲ့ Apache အားလုံးပါဝင်ပြီးသား Package ဖြစ်တဲ့ XAMPP ကိုအသုံးပြုနေတာဖြစ်ပါတယ်။ အလိုအလျောက်အနေနဲ့ root ဆိုတဲ့ username နဲ့ password မပါဘဲ ဖန်တီးထားပေးမှာဖြစ်ပါတယ်။။ အခု Database တွေနဲ့ ကျွန်ုပ်တို့ စတင်အလုပ်လုပ်တော့မှာဖြစ်လို့ အမှန်တကယ် username နဲ့ password တစ်ခုကို ဖန်တီးသင့်ပါတယ်။။ MySQL ရဲ့ User အခွင့်အရေးတွေကို အနည်းငယ်လောက် ဖြစ်ဖြစ် နားလည်ဖို့လိုအပ်ပါတယ်။။

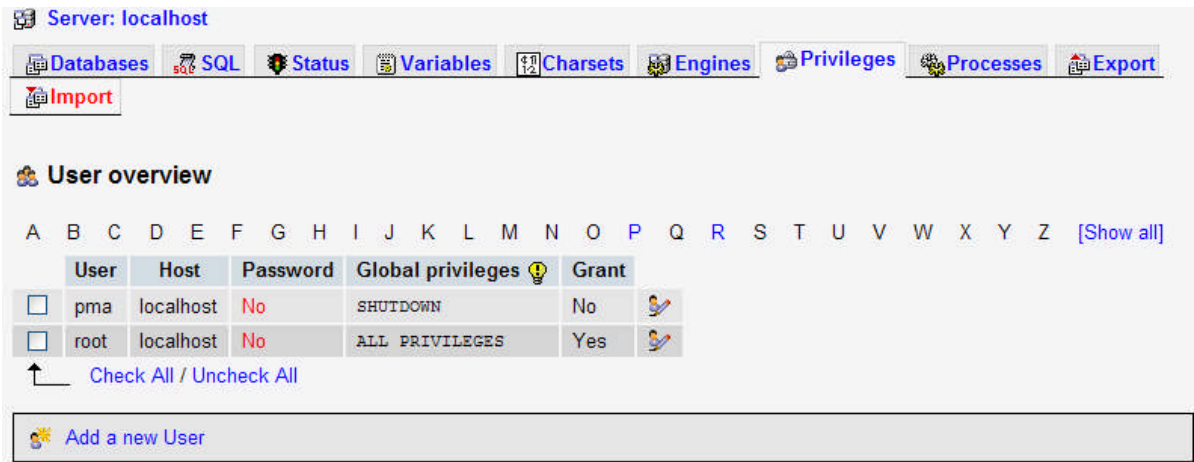
Creating a New User

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 ဆင်ဗျာ..။ အဲဒီအခါမှာ အောက်ပါအတိုင်း SQL Statement တွေကို အလိုအလျောက်ဖန်တီးပေးပြီးတော့ execute လုပ်ထားကာ.. you have added a new user ဆိုတဲ့ Message ပြမှာဖြစ်ပါတယ်.. ဒါဆိုရင်တော့ user အသစ်ကိုဖန်တီးပြီးသွားပါပြီ..။

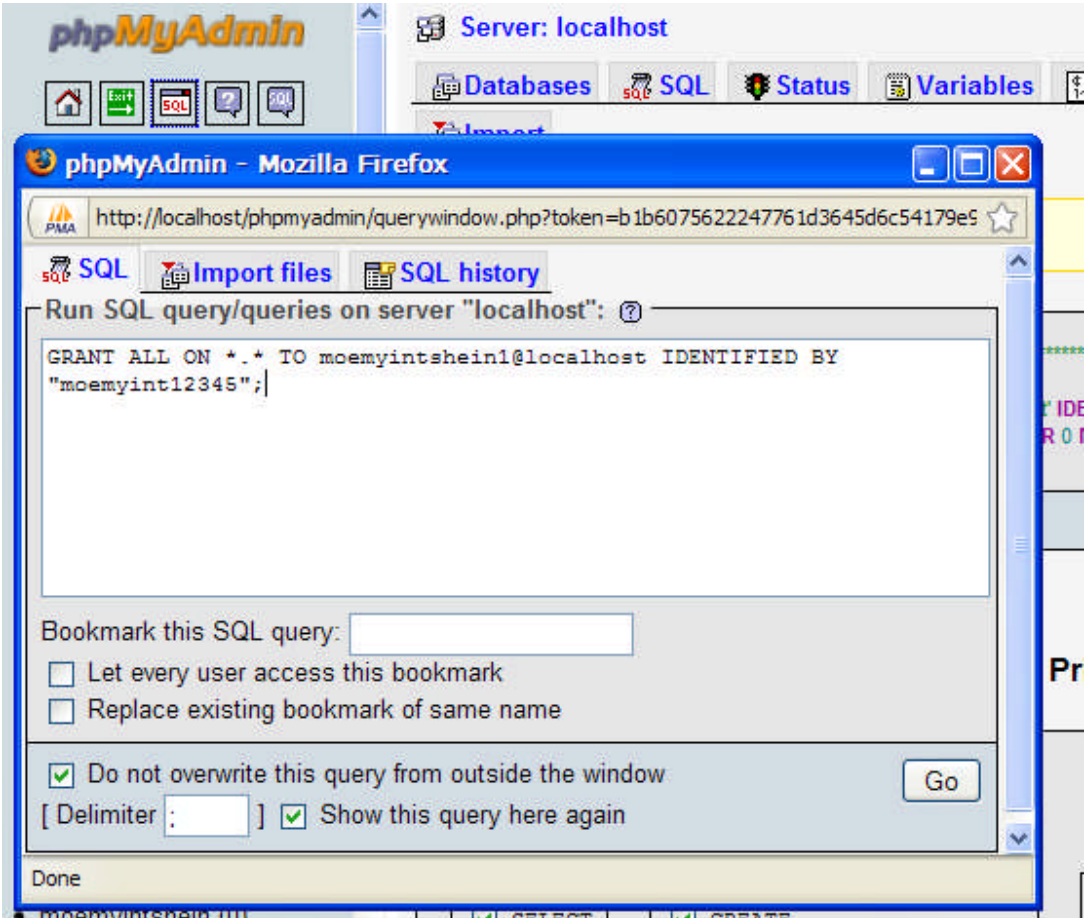
You have added a new user.

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...

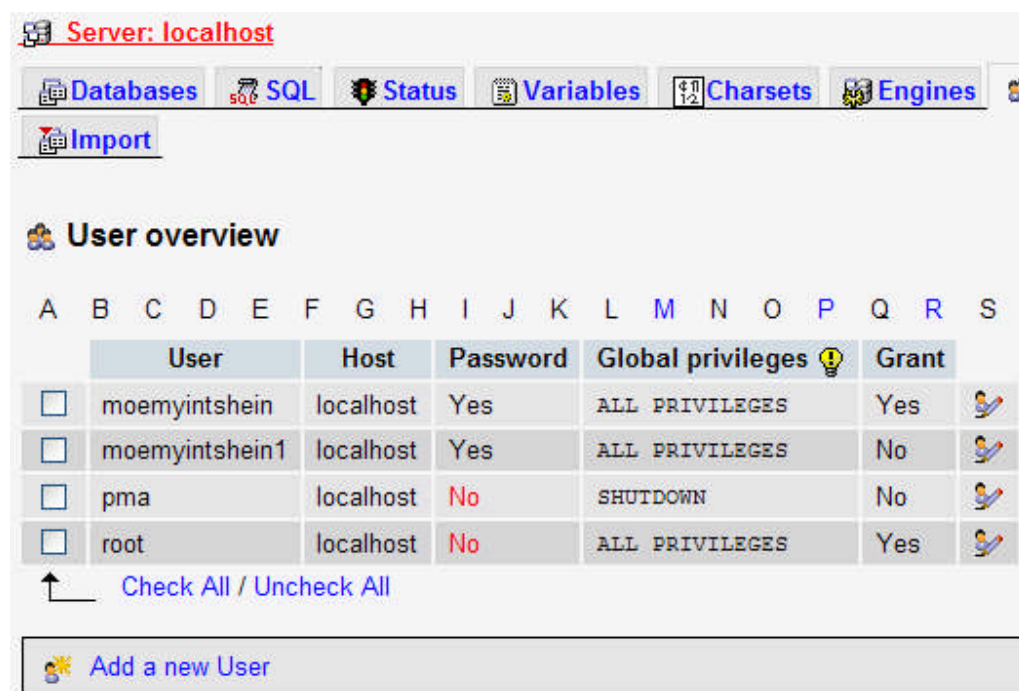


အထက်ပါပုံအတိုင်း PHP Admin main page ရဲ့ ညာဘက်ထောင့်အပေါ်နားလေးမှာ button ငါးခုရှိတဲ့အနက် အလယ်က SQL ဆိုတာလေးကို နှိပ်လိုက်တဲ့အခါ သူ့အောက်မှာ SQL query ကို execute လုပ်နိုင်တဲ့ pop up page တစ်ခု တတ်လာပါတယ်.. အဲဒီအထဲမှာ ကျနော်က

`GRANT ALL ON *.* TO moemyintshein1@localhost IDENTIFIED BY "moemyint12345";`

လိုက်ပြီး 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 ရှိနေမှာဖြစ်ပါတယ်...



The screenshot shows the 'User overview' section of PHP MyAdmin. It features a table with columns for User, Host, Password, Global privileges, and Grant. Below the table are links for 'Check All / Uncheck All' and 'Add a new User'.

	User	Host	Password	Global privileges	Grant
<input type="checkbox"/>	moemyintshein	localhost	Yes	ALL PRIVILEGES	Yes
<input type="checkbox"/>	moemyintshein1	localhost	Yes	ALL PRIVILEGES	No
<input type="checkbox"/>	pma	localhost	No	SHUTDOWN	No
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES	Yes

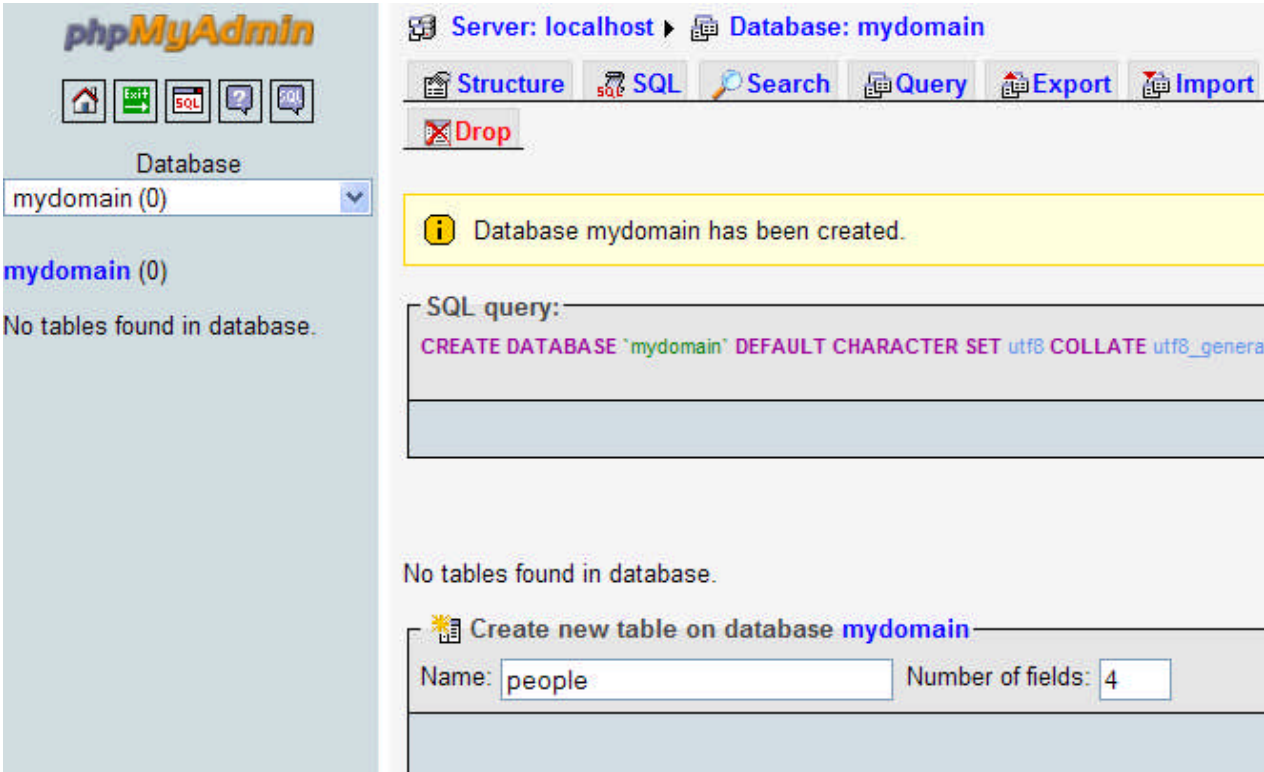
Check All / Uncheck All

Add a new User

user အသစ်ဖန်တီးပြီးတဲ့နောက်မှာ PHP MyAdmin ထဲမှာ Table ဆောက်ရအောင်ဗျ။

Database အရင်ဆောက်ဖို့လိုပါတယ်။ Home ပုံစံလေးကို နှိပ်လိုက်ရင် Databases ဆိုတာ သူ့ရဲ့ menu တွေထဲမှာ ရှာလို့တွေ့ပါတယ်။ အဲဒီမှာ Create New Database ဆိုတဲ့အောက်လေးမှာ ဆောက်နိုင်ပါတယ်။ Database နာမည်ကို mydomain လို့ပေးလိုက်ပါတယ်။ Collation ဆိုတဲ့အထဲမှာတော့ အဆင်ပြေတာကိုပေးပေါ့။ ကျနော်က utf8_general_ci နဲ့ဆောက်လိုက်တယ်။

ပြီးတဲ့နောက်မှာအောက်ပါပုံအတိုင်းပေါ်လာပြီးတော့ ကျနော်က mydomain ဆိုတဲ့ အဲဒီ database ထဲမှာ people ဆိုတဲ့ table ကို field လေးခု ပေးပြီး Go ကိုနှိပ်ပြီး ဆောက်လိုက်ပါတယ်။



Go ကိုနှိပ်ပြီးဆောက်လိုက်တဲ့အခါ mydomain ဆိုတဲ့ database ထဲမှာ people ဆိုတဲ့ table ရောက်နေပါပြီ အောက်ပါအတိုင်း ဖြည့်ပြီး table ကို save လုပ်လိုက်ပါမယ်။

Field	Type	Length	Default	Extra
id	int	6	0	auto_increment
name	char	100		
telephone	char	50		
birthday	char	50		

PHP MyAdmin ထဲမှာတော့ အခုလိုသွားဖြည့်မယ်။

phpMyAdmin

Server: localhost Database: mydomain Table: people

Field	Type	Length/Values	Collation
id	INT	6	
name	VARCHAR	100	
telephone	VARCHAR	50	
birthday	VARCHAR	50	

Table comments:

Storage Engine: MyISAM

Table: people

Collation	Attributes	Null	Default	Extra
		not null	0	auto_increment
		not null		
		not null		
		not null		

Storage Engine: MyISAM

Collation:

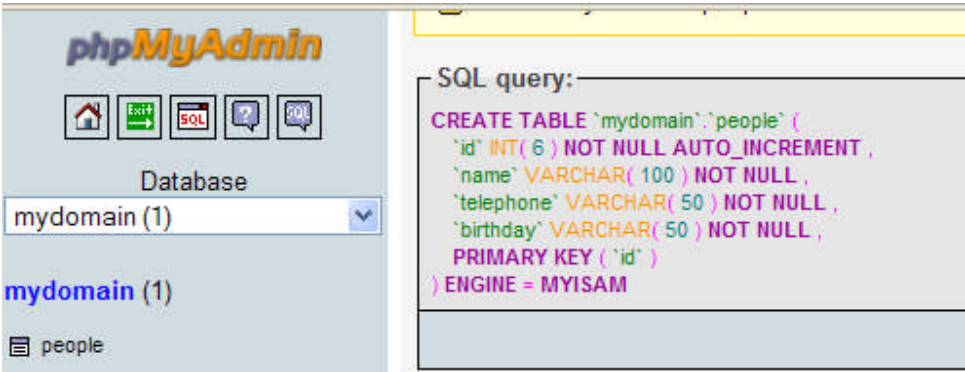
Save Or Add 1 field(s) Go

























ဒီအတိုင်းတွေဖြည့်ပြီးတော့ save လိုက်ပါတယ်။







length value က input character တွေရဲ့ လက်အခံနိုင်ဆုံး အရှည်ကို သတ်မှတ်ပေးတာဖြစ်ပါတယ်။ Type အတွက် ထားလို့ ရတာတွေကလည်း အများကြီးပါ။ ကျနော်ဒီမှာလျှောက်ထည့်ထားတာပါ။ efficient ဖြစ်ချင်မှဖြစ်မယ်။ PHP MyAdmin ရဲ့ function တစ်ချို့ကို အခြေခံလောက် သုံးတတ်အောင် ဒီဟာကို ထည့်ပေးထားတာပါ။ ပြီးရင် Drop ကိုသုံးပြီး အချိန်မရွေး ဒီ table တွေဖျက်လို့ရပါတယ်။




Primary Key အနေနဲ့ သုံးမယ့် id field ကို auto_increment ပေးထားပါတယ်။ record တွေကို Input ထပ်ထည့်တဲ့အခါ ကျနော်တို့ ရိုက်ရိုက်နေစရာမလိုတော့ဘူးပေါ့။ Default ကိုတော့ 0 ပေးလိုက်ပါတယ်။


အဲတန်ဖိုးတွေ အားလုံးထည့်ပြီး save လုပ်လိုက်တော့ အောက်ပါ screen ပေါ်လာပါတယ်။



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id	int(6)			No		auto_increment	     
<input type="checkbox"/>	name	varchar(100)	utf8_general_ci		No			     
<input type="checkbox"/>	telephone	varchar(50)	utf8_general_ci		No			     
<input type="checkbox"/>	birthday	varchar(50)	utf8_general_ci		No			     

Check All / Uncheck All With selected:      











Print view  Relation view  Propose table structure 





Add 1 field(s) ☒ At End of Table ☐ At Beginning of Table ☐ After id  Go

သင် table ဖန်တီးတတ်ပါပြီ။ အဲဒီ table ကိုဖြစ်စေတဲ့ SQL command တွေကိုလည်း တွေ့ရမှာပါ။ အခုတော့ သင် သိချင်မှ သိပါလိမ့်မယ်။ ဒါပေမယ့် ဘယ်ဟာကို ဘာကိုဆိုလိုတယ်ဆိုတာ PHP Database အခန်းကို လေ့လာလိုရင်တော့ သဘောပေါက်လာမှာပါ။

Table ထဲကို Data တွေထည့်ရအောင်ဗျာ။
Insert ဆိုတာ အပေါ်မှာတွေ့မယ်ထင်ပါတယ်။ နှိပ်လိုက်ကြရအောင်။
ပြီးတဲ့အခါ အခုလိုပဲဖြည့်ထည့်နိုင်ပါတယ်။

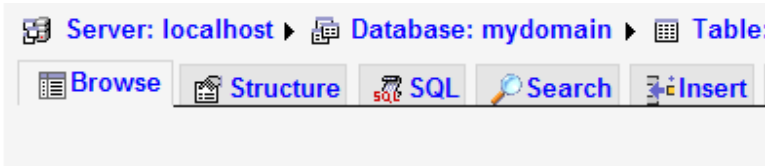
Server: localhost Database: mydomain Table: people

 Browse  Structure  SQL  Search  Insert  Export  Import  Operations  Empty  Drop

Field	Type	Function	Null	Value
id	int(6)			<input type="text"/>
name	varchar(100)			Joe Blow
telephone	varchar(50)			604 777 9999
birthday	varchar(50)			24 June

Go

သူ့ရဲ့ အပေါ်မှာရှိတဲ့ tab တွေ ကိုလိုက်သားနှိပ်ကြည့်ပါ။



Insert ကိုပြန်နှိပ်ပြီး ဒေတာတွေ field တစ်ခုစီအတွက် ထပ်ထပ်ဖြည့်သွားနိုင်ပါတယ်။
အောက်နားလေးမှာ သူ့ setting တွေချိန်လို့ရတာရှိပါတယ်။

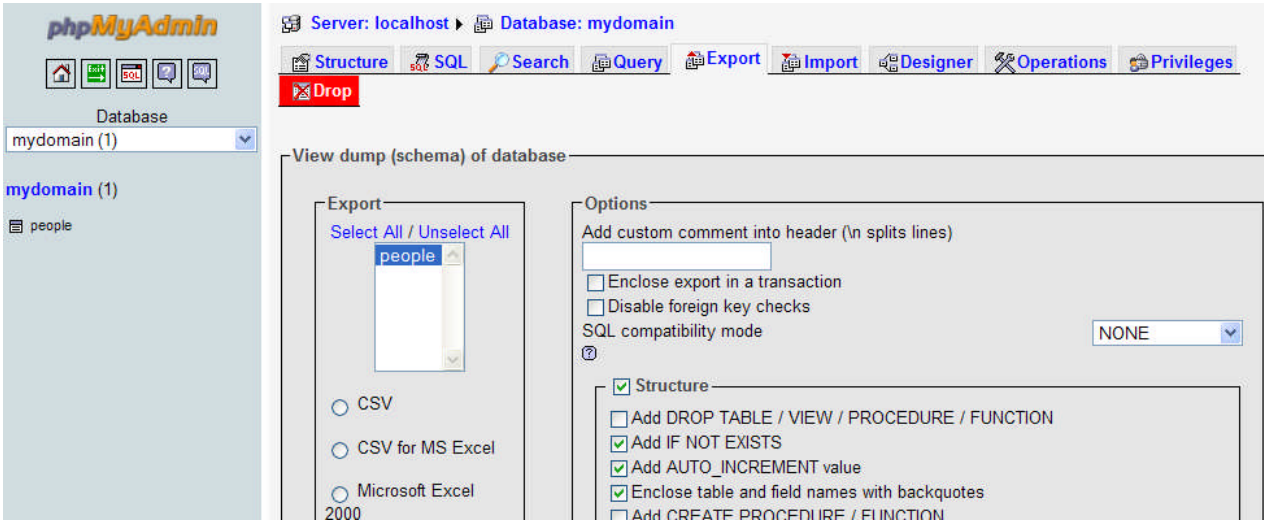


တခုဖြည့်ပြီးလို့ save လုပ်တိုင်း SQL code တွေပါတဲ့ window လည်းပေါ်လာပါတယ်။
id column က auto increment ဖြစ်တဲ့အတွက် ဘာမှမထည့်ပေးပါဘူး။
အောက်ပါပုံစံအတိုင်းဖြစ်သွားအောင် ဖြည့်ပေးကြည့်ပါလား။

		id	name	telephone	birthday
<input type="checkbox"/>		1	Joe Blow	604 777 9999	24 June
<input type="checkbox"/>		2	Jane Silly	519 222 2222	
<input type="checkbox"/>		3	Jane Silly	519 222 2222	April 1
<input type="checkbox"/>		4	Gray Cat	250 911 9999	March 5
<input type="checkbox"/>		5	Brown Horse	406 888 8888	May 12
<input type="checkbox"/>		6	Animal Keeper	905 111 1111	October 30

With selected:

အဲလိုဖြစ်ပြီးလားမဖြစ်သေးဘူးလားဆိုတာ Browse Tab ကိုဝင်ပြီး ကြည့်နိုင်ပါတယ်ခင်ဗျာ။ အဲဒီမှာဘေးနားက ခဲတံပုံနဲ့ ဟာလေးက တစ်ခုချင်းဆီကို edit ပေးလုပ်ပြီးတော့ အမှိုက်ပုံးလေးကတော့ Delete ပေးလုပ်တာလေးပါ။
အခု ဘယ်လို back up လုပ်လဲလေ့လာရအောင်။
Database ထဲကိုသွားရအောင် ဒီဘက် ဘယ်ဘက်ပေါ့ အခု database ကတော့ mydomain ပေါ့။
အပေါ်က Tab တွေထဲက Export ကိုရွေးမယ်။



အခုကတော့ people တစ်ခုပဲ ဆောက်ထားတော့ သူပဲရှိတယ် အဲဒီအကွက်လေးဆဲမှာ export လိုချင်တဲ့ backup လုပ်ချင်တဲ့ table တွေကိုထပ်ရွေးနိုင်တယ်။ ကျနော်ကတော့ ပြီးတဲ့အခါ save as file ကို checked လုပ်ပြီးတော့ zipped ကိုထပ်ရွေးပြီးတော့ GO လုပ်လိုက်ပါတယ်။ အဲဒီအခါ ဒေါင်းလုပ်ဆွဲရပါပြီ mydomain.sql ဆိုတဲ့ ဖိုင်လေးကို Zip နဲ့ရပါတယ် ဆင်ဗျာ။ အဲဒါ backup ပါပဲ။

အခုလောက်ဆို လေ့လာသူအနေနဲ့ Database ဆောက်တတ်ပြီ။ table တွေ ဆောက်တတ်ပြီ။ fields တွေဖန်တီးတတ်ပြီး records တွေထဲ အချက်အလက်ထည့်တတ်တယ်။ record ကိုပြန်ကြည့်တတ်တယ်။ edit လုပ်တတ်တယ်။ ဖျက်တတ်တယ် နောက်ဆုံး back up ပါလုပ်တတ်ပါပြီ။ အဲဒါအခြေခံပါပဲ။ ကျန်တာတွေကို တဖြည်းဖြည်းနဲ့ ဆက်လေ့လာသွားပါဆင်ဗျာ။ ကျနော်ကတော့ ဒီလောက်ပဲ အစပြုပေးလိုက်ပါတယ်။

Chapter 8.PHP and 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 အရှေ့ကအခန်းမှာ အနည်းငယ် လေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။

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 အမည်ရှိပါမယ့် 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 အပိုင်းကိုပေးမှာဖြစ်ပါတယ်..။ တတ်နိုင်သလောက်တော့ SQL ကိုလည်း ထည့်ရေးပေးထားပါတယ်။ သီးခြား SQL ကိုထပ်လေ့လာနိုင်ရင်လည်းကောင်းပါတယ်။

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 field ရဲ့ maximum length ဥပမာ varchar(15) ဆိုပြီးတော့ သတ်မှတ်ပေးရပါမယ်။ data type ကနေပြီးတော့ ဒီ column ကကိုင်တွယ်နိုင်တဲ့ data အမျိုးအစားကိုသတ်မှတ်ပေးတာဖြစ်ပါတယ်။

Primary Keys and Auto Increment Fields

table တစ်ခုထဲမှာရှိတဲ့ rows တွေကို ခွဲခြမ်းစိတ်ဖြာနိုင်ဖို့အတွက် primary key တစ်ခုကို အသုံးပြုရပါတယ်။ primary key တစ်ခုစီဟာ table ထဲမှာ တစ်ခုထည်းရှိရပါမယ်။ primary key ဟာ null (ဘာမှမရှိတဲ့တန်ဖိုး) မဖြစ်ရပါဘူး ဘာဖြစ်လို့လည်းဆိုတော့ database engine အနေနဲ့ record တွေကို ရှာဖွေဖို့အတွက် တန်ဖိုး တစ်ခုခုတော့ လိုအပ်တဲ့အတွက် ဖြစ်ပါတယ်။ အောက်ပါ ဥပမာမှာ personID ဆိုတဲ့ field ကို primary key field အနေနဲ့ ထားတာဖြစ်ပါတယ်။ primary key field တွေဟာ အများအားဖြင့် ID number ဖြစ်ကြပြီးတော့ AUTO_INCREMENT setting ကိုအသုံးပြုထားတာတွေ များပါတယ်။ AUTO_INCREMENT ဆိုတာကတော့ record (column) တစ်ခုကို အသစ်ပေါင်းထည့်တိုင်းမှာ အဲဒီ primary

key အဖြစ်သတ်မှတ်ထားတဲ့ field ထဲကတန်ဖိုးကို အလိုအလျောက် တစ်ပေါင်းပေးတာ ကိုပြောတာပါ။ primary key filed က null မဖြစ်တာသေချာစေဖို့အတွက် ဒီ field အတွက် NOT NULL ဆိုတဲ့ setting ကို ထည့်ပေးထားကြရပါတယ်။

```
$sql = "CREATE TABLE Persons
(
personID int NOT NULL AUTO_INCREMENT,
PRIMARY KEY(personID),
FirstName varchar(15),
LastName varchar(15),
Age int
)";
mysql_query($sql,$con);
```

PHP MySQL Insert Into

INSERT INTO statement ကို table တစ်ခုထဲကို column(record) အသစ်တွေ ထပ်ပေါင်းထည့်ဖို့အတွက် အသုံးပြုပါတယ်။

Syntax

INSERT INTO statement ကို ပုံစံနှစ်မျိုးနဲ့ ရေးနိုင်ပါတယ်။

ပထမ ပုံစံကတော့ data တွေကိုပေါင်းထည့်မယ့် နေရာ column နာမည်တွေကို မဖော်ပြဘဲနဲ့ သူတို့ရဲ့တန်ဖိုးတွေကိုပဲထည့်တဲ့ ပုံစံပါ။

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
```

ဒုတိယပုံစံကတော့ column name ရော ထည့်မည့် တန်ဖိုးရောကို နှစ်ခုစလုံး ဖော်ပြသတ်မှတ်ပေးတဲ့ အောက်ပါပုံစံပါ။

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

PHP ကိုအသုံးပြုပြီးတော့ ဒီ statement တွေကို execute လုပ်နိုင်ဖို့အတွက် mysql_query() function ကိုအသုံးပြုကြရမယ်။ ဒီ function ကို MySQL connection တစ်ခုဆီကို query တစ်ခု ဒါမှမဟုတ် command တစ်ခု ပို့ဖို့အတွက် အသုံးပြုကြရပါတယ်။

ဥပမာ

အရှေ့မှာ ကျနော်တို့ Persons လို့နာမည်ပေးထားတဲ့ table တစ်ခုကို Firstname, Lastname နဲ့ Age ဆိုတဲ့ Column သုံးခုနဲ့ ဖန်တီးခဲ့ကြပါတယ်။ ဒီဥပမာမှာ အဲဒီ table ကိုပဲ ပြန်အသုံးပြုပါမယ်။ ဒီမှာ ဒီ Persons ဆိုတဲ့ table ထဲကိုပဲ record အသစ်နှစ်ခုကို အောက်ပါအတိုင်း ပေါင်းထည့်လိုက်ပါမယ်။

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
```

```

{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Peter', 'Griffin', '35')");
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Glenn', 'Quagmire', '33')");
mysql_close($con);
?>

```

Insert Data From a Form Into a Database

အခုကျနော်တို့ Persons table ထဲကို record အသစ်တွေပေါင်းထည့်ပေးနိုင်မယ့် HTML form တစ်ခုကို ဖန်တီးပါမယ်။
အောက်ကတော့ HTML form ပါ။

```

<html>
<body>
<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body>
</html>

```

user က ဒီဥပမာထဲက HTML form ထဲမှာပါတဲ့ Submit button ကို နှိပ်လိုက်တဲ့အခါ form ထဲမှာထည့်ထားတဲ့ data တွေက insert.php ဆီကို ပို့ပေးမှာပါ။

insert.php ဖိုင်ကနေ database တစ်ခုဆီကို ဆက်သွယ်ပါမယ်။ ပြီးတော့ PHP \$_POST variables တွေကိုသုံးပြီး form ထဲက တန်ဖိုးတွေကို ထုတ်ယူမှာဖြစ်ပါတယ်။ အဲဒီနောက်မှာတော့ mysql_query() function ကနေပြီးတော့ INSERT INTO statement ကိုအလုပ်လုပ်သွားမှာဖြစ်ပြီး Persons table ထဲကို record အသစ်တစ်ခု ထပ်ပေါင်းထည့်ပေးမှာဖြစ်ပါတယ်။

insert.php အတွက် code ပါ။

```

<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES
('$_POST[firstname]', '$_POST[lastname]', '$_POST[age]')";
if (!mysql_query($sql,$con))
{
    die('Error: ' . mysql_error());
}
echo "1 record added";
mysql_close($con)
?>

```


PHP MySQL Select

Select Data From a Database Table

SELECT statement ကို database တစ်ခုထဲကနေ data ကို select လုပ်ဖို့အတွက် အသုံးပြုပါတယ်။

Syntax

```
SELECT column_name(s)
FROM table_name
```

ထုံးစံအတိုင်း PHP နဲ့အလုပ်လုပ်ဖို့အတွက် mysql_query() function ကိုကျနော်တို့ သုံးကြရပါမယ်။

ဥပမာ

အောက်ပါ ဥပမာမှာ Persons table ထဲမှာသိမ်းထားတဲ့ data တွေအားလုံးကို ထုတ်နှုတ်ပြမှာဖြစ်ပါတယ်။ * ဆိုတဲ့ character လေးက table ထဲမှာရှိတဲ့ data တွေအားလုံးကို ရွေးချယ်ထုတ်ပေးပါတယ်။

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}
mysql_close($con);
?>
```

အထက်ပါ ဥပမာမှာ mysql_query() ကနေ return ပြန်လာတဲ့ data တွေကို \$result ဆိုတဲ့ variable ထဲမှာ သိမ်းဆည်းထားပါတယ်။

အဲဒီနောက်မှာ array အနေနဲ့ recordset ထဲက ပထမဆုံး row ကို return ပြန်ဖို့အတွက် mysql_fetch_array () function ကို အသုံးပြုထားပါတယ်။ mysql_fetch_array() ကို ခေါ်သုံးတိုင်းမှာ recordset ထဲမှာ ရှိတဲ့ နောက်ထပ် နောက်ထပ် row တွေကို return ပြန်ပေးပါတယ်။ while loop ကနေ recordset ထဲက record တွေအားလုံးကိုဖြတ်သွားပြီး Looping ယတ်ပါတယ်။ row တစ်ခုစီရဲ့ တန်ဖိုးတွေကို Screen မှာ Print ထုတ်ပြဖို့အတွက် PHP \$row variable (\$row['FirstName'] and \$row['LastName']) ကိုကျနော်တို့ အသုံးပြုထားပါတယ်။

ဒီ code ရဲ့ Output အနေနဲ့ အောက်ပါအတိုင်း ထွက်လာမှာဖြစ်ပါတယ်။

Peter Griffin
Glenn Quagmire

Display the Result in an HTML Table

အောက်ပါ ဥပမာအနေနဲ့ကတော့ အထက်ပါဥပမာအတိုင်းပဲ တူညီတဲ့ data တွေကိုထုတ်နှုတ်မှာ ဖြစ်ပါတယ်။ ဒါပေမယ့် သူကတော့ HTML table တစ်ခုနဲ့ data တွေကိုထုတ်ပြပေးသွားမှာဖြစ်ပါတယ်။

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

mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons");

echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>";
while($row = mysql_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";
    echo "<td>" . $row['LastName'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysql_close($con);
?>
```

အထက်ပါ code တွေရဲ့ output ကတော့ အောက်ပါအတိုင်းဖြစ်ပါလိမ့်မယ်။

Firstname	Lastname
Glenn	Quagmire
Peter	Griffin

PHP MySQL The Where Clause

WHERE clause ကတော့ သတ်မှတ်ထားတဲ့ အခြေအနေတစ်ခုကို ပြည့်စုံစေတဲ့ record တွေကိုပဲ ထုတ်ပြဖို့အတွက် အသုံးပြုကြရပါတယ်။

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value
```

ဥပမာ

အောက်ပါဥပမာမှာ Persons table ထဲက FirstName က Peter ဖြစ်နေတဲ့ row တွေအားလုံးကို ထုတ်နှုတ်ပြပေးပါ လို့ ဖော်ပြထားပါတယ်။

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

mysql_select_db("my_db" , $con);

$result = mysql_query("SELECT * FROM Persons
WHERE FirstName='Peter'");

while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}

?>
```

output အနေနဲ့ကတော့ -

Peter Griffin
ဆိုပြီးတော့ ရပါမယ်ခင်ဗျာ။

PHP MySQL Order By Keyword

ORDER BY keyword ကိုတော့ recordset တစ်ခုထဲက data တွေကို စီထားဖို့အတွက် အသုံးပြုပါတယ်။
ORDER BY keyword လေးက default အနေနဲ့ ascending order နဲ့ record တွေကို စီပေးပါတယ်။
descending order နဲ့ record တွေကို စီမယ်ဆိုရင် DESC keyword ကိုထည့်သွင်း အသုံးပြုနိုင်ပါတယ်။

Syntax

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s) ASC|DESC
```

ဥပမာ

အောက်ပါဥပမာမှာ Persons table ထဲမှာ သိမ်းထားတဲ့ data တွေအားလုံးကို ထုတ်ပြပေးပါတယ်။ ပြတဲ့နေရာမှာ Age column နဲ့ result တွေကို စီထားပေးတာဖြစ်ပါတယ်။

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

mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons ORDER BY age");

while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'];
    echo " " . $row['LastName'];
    echo " " . $row['Age'];
    echo "<br />";
}
mysql_close($con);
?>
```

ဒီ code ရဲ့ output ကတော့

```
Glenn Quagmire 33
Peter Griffin 35
ဖြစ်ပါတယ်။
```

Order by Two Columns

column တစ်ခုမက သုံးပြီးတော့လည်း order by လုပ်နိုင်ပါတယ်။ တစ်ခုထက်ပိုတဲ့ column တွေရဲ့ Ordering လုပ်မယ်ဆိုရင်တော့ ပထမ column ကတန်းပြီးတော့ တူညီနေမှသာ ဒုတိယ column ကိုအသုံးပြုပါတယ်။

```
SELECT column_name(s)
FROM table_name
ORDER BY column1, column2
```

PHP MySQL Update

UPDATE statement ကိုတော့ table တစ်ခုထဲက ရှိနေပြီးသာ record တွေကို ပြင်ဆင်ရန် Update လုပ်ရန်အသုံးပြုပါတယ်။

Syntax

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

ဒီမှာအရေးကြီးတာက WHERE clause ပဲဖြစ်ပါတယ်။ UPDATE syntax ကိုအသုံးပြုတဲ့ခါ WHERE clause အနေနဲ့ ဘယ် record တွေကို ပြင်ဆင်မှာလည်းဆိုတာကို သတ်မှတ်ပေးရတာဖြစ်ပါတယ်။ WHERE မပါဘဲ သုံးမယ်ဆို records တွေ အားလုံး Update တွေ့ဖြစ်ကုန်မှာဖြစ်ပါတယ်။

ဥပမာ

ကျနော်တို့ ဖန်တီးခဲ့တဲ့ Persons ဆိုတဲ့ Table မှာ အောက်ပါအတိုင်းရှိနေတာပါ..။

FirstName	LastName	Age
Peter	Griffin	35
Glenn	Quagmire	33

အောက်ပါ ဥပမာလေးက Persons table ထဲမှာ ရှိတဲ့ data တစ်ချို့ကို update လုပ်ပေးပါတယ်။

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

mysql_query("UPDATE Persons SET Age = '36'
WHERE FirstName = 'Peter' AND LastName = 'Griffin'");
mysql_close($con);
?>
```

Update လုပ်ပြီးတဲ့အခါမှာ Persons table ထဲမှာ အောက်ပါပုံစံအတိုင်းဖြစ်သွားပါတယ်။

FirstName	LastName	Age
Peter	Griffin	36
Glenn	Quagmire	33

PHP MySQL Delete

DELETE statement ကို database table ထဲက records တွေကို ဖြတ်ဖို့အတွက် အသုံးပြုပါတယ်။ FROM ဆိုပြီးတော့ DELETE ရဲ့ နောက်မှာပါပါတယ်။

Syntax

```
DELETE FROM table_name
WHERE some_column = some_value
```

သူ့မှာလည်းအရေးကြီးတာက WHERE clause ပဲဖြစ်ပါတယ်။ WHERE clause အနေနဲ့ ဘယ် record တွေကို ဖြတ်မယ် မှာလည်းဆိုတာကို သတ်မှတ်ပေးရတာဖြစ်ပါတယ်။ WHERE မပါဘဲ သုံးမယ်ဆို records တွေ အားလုံး Delete တွေ့ဖြစ်ကုန်မှာဖြစ်ပါတယ်။

ဥပမာ

အထောက်အကူ Persons table က ဒီလိုရှိပါမယ်။

FirstName	LastName	Age
Peter	Griffin	35
Glenn	Quagmire	33

အောက်က code တွေကိုသုံးပြီးတော့ LastName က Griffin ဖြစ်တဲ့ Persons table ထဲက record တွေကို ဖျက်ပါ လို့ ပြောလိုက်ပါမယ်။

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

mysql_query("DELETE FROM Persons WHERE LastName='Griffin'");
mysql_close($con);
?>
```

Delete လုပ်ပြီးတဲ့အခါမှာတော့ အောက်ပါအတိုင်းဖြစ်သွားပါမယ်။

FirstName	LastName	Age
Glenn	Quagmire	33

PHP Database ODBC

ODBC ဆိုတာကတော့ Microsoft Access Database လိုမျိုး Data Source တစ်ခုကို ဆက်သွယ်ပေးမယ့် Application Programming Interface (API) ဖြစ်ပါတယ်။

Create an ODBC Connection

ODBC connection တစ်ခုနဲ့ ကျနော်တို့အနေနဲ့ ODBC connection တစ်ခု ရနေသရွေ့ ကျနော်တို့ နတ်ဝက် ထဲမှာရှိတဲ့ ဘယ်ကွန်ပျူတာပေါ်မှာပဲဖြစ်ဖြစ်ရှိတဲ့ ဘယ် database နဲ့မဆို ချိတ်ဆက်အသုံးပြုနိုင်ပါတယ်။

အောက်ပါဖော်ပြချက်တော့ MS Access Database တစ်ခုနဲ့ ODBC Connection တစ်ခု ဘယ်လို ဖန်တီးတယ်ဆိုတာပါ။

1. Open the **Administrative Tools** icon in your Control Panel.
2. Double-click on the **Data Sources (ODBC)** icon inside.
3. Choose the **System DSN** tab.

4. Click on **Add** in the System DSN tab.
5. Select the **Microsoft Access Driver**. Click **Finish**.
6. In the next screen, click **Select** to locate the database.
7. Give the database a **Data Source Name (DSN)**.
8. Click **OK**.

ဒီ configuration ကို website ရှိနေတဲ့ ကွန်ပျူတာပေါ်မှာ ပြုလုပ်ရမှာဖြစ်ပါတယ်။ ကျနော်တို့ရဲ့ ကွန်ပျူတာမှာ Internet Information Server ဆိုတဲ့ IIS ကို run ထားမယ်ဆိုရင် ဒီ instruction တွေက အလုပ်လုပ်မှာဖြစ်ပါတယ်။ remote server ပေါ်မှာ ကျနော်တို့ရဲ့ website ရှိတာဆိုရင်တော့ အဲဒီ server ကို Physical access ရှိရမယ် ဒါမှမဟုတ် web host ကို ကျနော်တို့အသုံးပြုရမယ်။ DNS တစ်ခုကို set up လုပ်ပေးဖို့ တောင်းခံရပါမယ်။

Connecting to an ODBC

odbc_connect() function ကို ODBC data source တစ်ခုနဲ့ ချိတ်ဆက်ဖို့အတွက်သုံးပါတယ်။ ဒီ function မှာ parameter လေးခုရှိပါတယ်။ data source နာမည်ရယ်၊ username ရယ်၊ Password ရယ်နဲ့ Optical cursor type ဖြစ်ပါတယ်။ SQL statement တွေကို execute လုပ်ဖို့အတွက် odbc_exec() function ကို အသုံးပြုပါတယ်။

ဥပမာ

အောက်ကဥပမာမှာ northwind လို့ခေါ်တဲ့ DNS တစ်ခုကို connection ဖန်တီးပါတယ်။ username နဲ့ password နဲ့ northwind မှာရှိပါဘူး။ အဲဒီနောက် SQL တစ်ခုကိုဖန်တီးပြီးတော့ execute လုပ်ပါတယ်။

```
$conn=odbc_connect('northwind','','');
$sql="SELECT * FROM customers";
$rs=odbc_exec($conn,$sql);
```

Retrieving Records

odbc_fetch_row() ကိုသုံးပြီးတော့ result-set ကနေ record တွေကို return ပြန်ပါတယ်။ ဒီ function က rows တွေကို return ပြန်နိုင်တယ်ဆိုရင် true လို့ပြောပြီးတော့ ဒါမှမဟုတ်ရင်တော့ false ပေးပါတယ်။

ဒီ function ကတော့ ODBC result identifier နဲ့ Optical row number ဆိုတဲ့ parameter နှစ်ခုကို ယူပါတယ်။

```
odbc_fetch_row($rs)
```

Retrieving Fields from a Record

odbc_result() function က record တစ်ခုကနေ fields တွေကိုဖတ်ဖို့အသုံးပြုပါတယ်။ ဒီ function က ODBC result identifier နဲ့ field number ဒါမှမဟုတ် name တစ်ခု စုစုပေါင်း parameter နှစ်ခုကို ယူပါတယ်။

အောက်က code line မှာတော့ record ထဲက ပထမဆုံး field ရဲ့ တန်ဖိုးကို return ပြန်ပေးပါတယ်။

```
$compname=odbc_result($rs,1);
```

အောက်က code line ကတော့ CompanyName လို့ခေါ်တဲ့ field တစ်ခုရဲ့ တန်ဖိုးကို return ပြန်ပေးပါတယ်။

```
$compname=odbc_result($rs,"CompanyName");
```

Closing an ODBC Connection

odbc_close() function ကိုသုံးပြီး ODBC connection တစ်ခုကိုပိတ်ရပါမယ်။

```
odbc_close($conn);
```

ODBC ဥပမာ

အောက်ပါဥပမာမှာ database connection တစ်ခုကို အရင်ဆုံးဘယ်လိုဖန်တီးတယ် ပြီးတော့ result set တစ်ခုနဲ့ HTML table တစ်ခုထဲမှာ data တွေကိုပြပေးဖို့အတွက် ရေးတာကို ပြထားပါတယ်။

```
<html>
<body>
<?php
$conn=odbc_connect('northwind','','');
if (!$conn)
    {exit("Connection Failed: " . $conn);}
$sql="SELECT * FROM customers";
$rs=odbc_exec($conn,$sql);
if (!$rs)
    {exit("Error in SQL");}
echo "<table><tr>";
echo "<th>Companyname</th>";
echo "<th>Contactname</th></tr>";
while (odbc_fetch_row($rs))
{
    $compname=odbc_result($rs,"CompanyName");
    $conname=odbc_result($rs,"ContactName");
    echo "<tr><td>$compname</td>";
    echo "<td>$conname</td></tr>";
}
odbc_close($conn);
echo "</table>";
?>
</body>
</html>
```


Chapter 9. XML & PHP

ကျနော်တို့အတွက် Web Development နဲ့ပတ်သက်ရင် XML ဆိုတာ မသိမဖြစ်ပဲဖြစ်ပါတယ်။ XML ဆိုတာ Extensible Markup Language ဖြစ်ပါတယ်။ XML ကို data သယ်ဆောင်ဖို့နဲ့ သိမ်းဆည်းဖို့ ဖန်တီးထားတာဖြစ်ပါတယ်။ ဒီ ကျူတိုရီရယ်မှာ XML အခြေခံကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။ HTML နဲ့ XML နဲ့ဘာတွေကွာခြားသလဲဆိုတာမျိုးတွေလည်းရှင်းသွားပါမယ်။ XML ကိုသိဖို့အရေးကြီးပါတယ်၊ သင်ယူဖို့လည်းလွယ်ကူပါတယ်။

Introduction to XML

XML ကို data သယ်ဆောင်ဖို့နဲ့ သိမ်းဆည်းဖို့ တီထွင်ထားတာဖြစ်ပါတယ်။

HTML ကတော့ Data တွေကို ထုတ်ပြဖို့အတွက်ဖြစ်ပါတယ်။ XML က HTML လိုမျိုးပဲအဆောင်အယောင်တူတဲ့ markup language ပါပဲ။ XML tags တွေက ကြိုတင်သတ်မှတ်ထားတဲ့ predefined တွေမဟုတ်ပါဘူး.. ကျနော်တို့ ကိုယ့်ဘာကိုယ် သတ်မှတ်ပေးရပါမယ်။ XML က self-descriptive ပုံစံဖြစ်ပါတယ်။ XML ကို W3C ကနေလည်း ထောက်ခံမှုပေးထားပါတယ်။

XML နဲ့ HTML တို့ရဲ့ခြားနားချက်တွေကိုသိရပါမယ်။ XML က HTML နေရာကိုအစားဝင်တာမဟုတ်ပါဘူး။ သူတို့နှစ်ခုက မတူညီတဲ့ ရည်ရွယ်ချက်တွေနဲ့ ပေါ်လာကြတာပါ။ XML က data တွေကို သယ်ဆောင်ဖို့ နဲ့ သိမ်းဆည်းဖို့ ဖြစ်ပါတယ်.. ဘာအတာ တွေလည်း ဆိုတဲ့အပေါ်မှာ ဦးစည်ပါတယ်။ HTML ကတော့ ဒီဒေတာတွေက ဘယ်လိုပုံစံရှိလဲဆိုတဲ့ အပေါ်မှာ မူတည်ပြီးတော့ data တွေကို ပြသဖို့အတွက်ပါ။ HTML ကအချက်အလက်တွေကို ထုတ်ပြဖို့ဖြစ်ပြီး XML ကတော့ အချက်အလက်တွေ သယ်ဆောင်သိမ်းဆည်းဖို့ဖြစ်တယ်လို့ နားလည်ရပါမယ်။

XML Does not DO Anything

XML က ဘာကိုမှမလုပ်ပေးနိုင်ပါဘူး။ နားလည်ဖို့ခက်ပါမယ်။ တကယ်တော့ XML က တစ်စုံတစ်ရာကို မပြုလုပ်ပါဘူး။ အချက်အလက်တွေကို ဖွဲ့စည်းတည်ဆောက် သိမ်းဆည်း ပြီး သယ်ဆောင်ဖို့ ပဲဖြစ်ပါတယ်။

အောက်ပါ ဥပမာကတော့ XML နဲ့သိမ်းဆည်းထားတဲ့ Jani ကနေ Tove ဆီကိုပေးတဲ့ note တစ်ခုဖြစ်ပါတယ်။

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ကဲ ဒီနုတ်က ကိုယ်နားလည်အောင် ကိုယ့်ဘာကို ရေးထားတာဖြစ်ပါတယ်။ self descriptive သဘောတရားပါ။ သူ့မှာ ပို့တဲ့သူနဲ့ လက်ခံတဲ့သူရဲ့ အချက်အလက်တွေပါပါတယ်။ သူ့မှာ ခေါင်းစဉ်ပါသလို စာကိုယ်လည်းပါဝင်ပါတယ်။ မှန်ပါတယ် ဒါပေမယ့် ဒီ XML document လေးက တစ်စုံတစ်ရာကိုအလုပ်မလုပ်ဘူးလေ.. tags တွေနဲ့ ရေးထားတဲ့ အချက်အလက် ရှင်းရှင်းလေးမျှသာပဲ ဖြစ်ပါတယ်။ တစ်စုံတစ်ယောက်က Software တစ်ခုနဲ့ ချိတ်ဆက်ပြီး ဒီအချက်အလက်တွေကို ပို့ပေးဖို့ လက်ခံဖို့ ပြသဖို့ လုပ်ဆောင်ရမှာဖြစ်ပါတယ်။

XML is Just Plain Text

XML က ဘာမှမထူးခြားပါဘူး။ စာတွေသက်သက်ချည်းပဲဖြစ်ပါတယ်။ plain text တွေကို ကိုင်တွယ်နိုင်တဲ့ Software တွေက XML ကိုလည်း ကိုင်တွယ်နိုင်ပါမယ်။ ဒါပေမယ့် XML အတွက်ဦးစားပေး Application တွေကတော့ XML tag တွေကို ကောင်းကောင်း ကိုင်တွယ်နိုင်ပါတယ်။ ဒီ tags တွေရဲ့ ဘာကိုလုပ်ဆောင်တယ်ဆိုတဲ့ အဓိပ္ပါယ်ကတော့ application ရဲ့ သဘောသဘာဝပေါ်မှာ မှီခိုပါမယ်။

XML ကိုအသုံးပြုပြီးတော့ ကျနော်တို့က ကိုယ်ကြိုက်တဲ့ ကိုယ်ပိုင် tag ကိုဖန်တီးနိုင်ကြပါတယ်။ အပေါ်က ဥပမာက <to> တို့ <from> တို့လိုပေါ့။ အဲဒါတွေဟာ ဘယ် XML standard မှာမှ defined လုပ်ထားတာမရှိပါဘူး။ ဒီ tag တွေကို ဒီ XML document ကိုရေးသားတဲ့ စာရေးဆရာ ကတီထွင်လိုက်တာပဲ။ ဒါဟာ ဘာဖြစ်လို့လဲဆိုတော့ XML language မှာ predefined tags မရှိလို့ပါ။ ရှင်းသွားအောင် HTML နဲ့ယှဉ်ပါမယ်။ HTML မှာသုံးတဲ့ tag တွေကတော့ Predefined ပါ။ HTML documents တွေမှာ HTML standard သတ်မှတ်ထားတဲ့ tags တွေ ဥပမာ <p> <h1> အဲဒါမျိုးတွေကိုပဲ အသုံးပြုနိုင်ပါတယ်။

XML ဆိုတာ HTML ရဲ့အစားထိုးမဟုတ်ဘဲ အတူတကွပူးပေါင်းလုပ်ဆောင်ကြတာသာဖြစ်ပါတယ်။ XML က သတင်းအချက်အလက်တွေကို သယ်ဆောင်ပေးမယ့် software နဲ့ hardware တွေပေါ်မှာ မှီခိုမှုမရှိတဲ့ အရာတစ်ခုဖြစ်ပါတယ်။

XML ကိုရေးသားတဲ့အခါ သစ်ပင်တစ်ပင်ရဲ့ ဖွဲ့စည်းတည်ဆောက်မှုအတိုင်း အမြစ်ကနေစပြီးတော့ သစ်ရွက်တွေအထိ ခွဲဖြာသွားတာ မျိုးဖြစ်ပါတယ်။

XML document ဥပမာ

XML ဖိုင်တစ်ခုကို self-describing ပုံစံနဲ့ရေးပြီးတော့ ရှိရင်းတဲ့ syntax တွေကိုပဲအသုံးပြုကြပါတယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ယထမဆုံးစာကြောင်းကတော့ ဒါ XML ဖိုင်ပါလို့ပြောလိုက်တာပါ။ XML version 1.0 ပါလို့ဖော်ပြပြီးတော့ အသုံးပြုထားတဲ့ encoding ကတော့ (ISO-8859-1 = Latin-1/West European character set) ပါလို့ ကြေငြာလိုက်တာပါ။

နောက်ထပ်တစ်ကြောင်းကတော့ ဒီ document လေးရဲ့ root element ကိုဖော်ပြလိုက်တာပါ။ ဒီ document လေးက note တစ်ခုဖြစ်ပါတယ်လို့ ပြောလိုက်တာပေါ့။

```
<note>
```

နောက်ထပ်ပြီးလာတဲ့ စာကြောင်းလေး ကြောင်းကတော့ to,from,heading နဲ့ body ဆိုတဲ့ root ရဲ့ child elements လေးခုကို ဖော်ပြပေးပါတယ်။

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

နောက်ဆုံးစာကြောင်းမှာတော့ root element ရဲ့အဆုံးသတ်ပါဆိုပြီး ပြန်ပိတ်လိုက်တာပါ။

</note>

ကျနော်တို့အနေနဲ့ ဒီဥပမာကနေ ဒီ XML document ဟာဆိုရင် Jani ကနေ Tove ဆီကို note တစ်ခုပို့ပေးတယ်လို့ နားလည်နိုင်ပါတယ်။

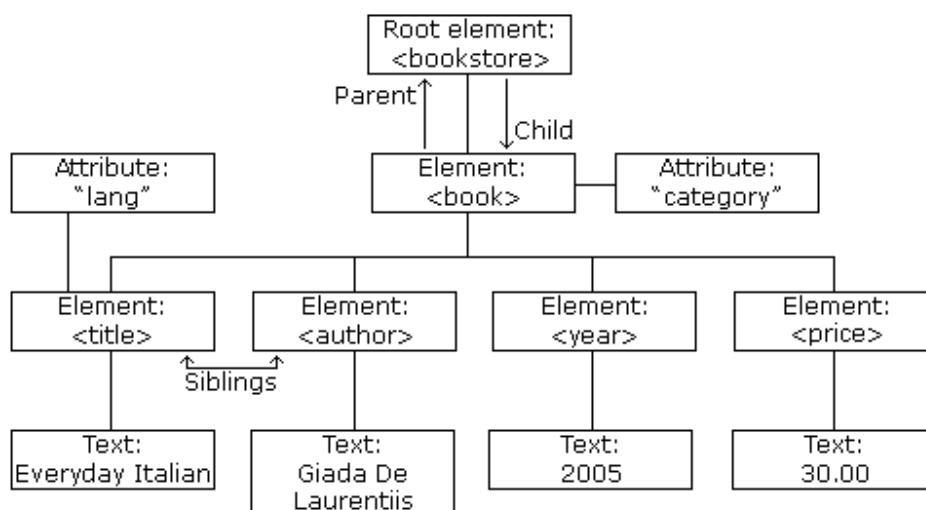
XML documents တွေမှာ root element တစ်ခုပိုင်ကိုပိုင်ရပါမယ်။ ဒီ element က တခြား element တွေအားလုံးရဲ့ မိဘ parent ဖြစ်ပါတယ်။ XML တစ်ခုမှာ ပါတဲ့ elements တွေက document tree တစ်ခုလိုပုံစံဆောင်ပါတယ်။ သစ်ပင်က root ကနေ စပြီးတော့ branches တွေထိ ခွဲဖြာသွားသလိုပေါ့။ elements တွေအားလုံးမှာ ထပ်ခွဲတဲ့ elements (child elements) တွေပါဝင်နိုင်ပါတယ်။

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
```

</root>

parent, child, sibling စတဲ့ ဝေါဟာရတွေကို elements တွေကြားထဲက ဆက်ဆံမှုအဖြစ် ဖော်ပြဖို့ သုံးပါတယ်။ မိဘတွေမှာ ကလေးတွေရှိပါမယ် ၊ level တူကလေးတွေဆိုရင် မောင်နှမ ညီအစ်မ ညီအစ်ကို တွေပေါ့။ elements တွေအားလုံးမှာ HTML မှာလိုပဲ text content နဲ့ attributes တွေပါဝင်နိုင်ပါတယ်။

ဥပမာ



အထက်ပါပုံက စာအုပ်တစ်အုပ်ကို XML နဲ့အောက်ပါအတိုင်းရေးတာကို ကိုယ်စားပြုပါတယ်။

```
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
```

```
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

ဦးပမာထဲမှာ root element က <bookstore> ဖြစ်ပါတယ်။ အားလုံးသော <book> elements တွေက <bookstore> ထဲမှာ ပါဝင်နေပါတယ်။ <book> element ထဲမှာ ကလေး လေးယောက်ရှိပါတယ်။ <title> <author> <year> နဲ့ <price> တို့ဖြစ်ပါတယ်။

XML Syntax Rules

XML Syntax Rules တွေက ရှိရင်းပြီး ယုတ္တိတန်ပါတယ်။ သင်ယူဖို့လွယ်ကူပါတယ်။ အသုံးပြုဖို့အတွက်လွယ်ကူပါတယ်။

အားလုံးသော elements တွေမှာ အပိတ် tag ပါရပါမယ်။

HTML မှာဆိုရင် အပိတ်မပါတဲ့ element တွေကိုခဏခဏမြင်ရပါလိမ့်မယ် အောက်ကအတိုင်းပေါ့။

```
<p>This is a paragraph
<p>This is another paragraph
```

XML မှာကတော့ closing tag ကိုထားခဲ့လို့လုံးဝမရပါဘူး။ အပိတ် tag ရှိရှိရပါမယ်။

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

မှတ်ချက်အနေနဲ့ အပေါ်ကဥပမာမှာ XML declaration စာကြောင်းမှာ closing tag မပါပါဘူး။ အဲဒါမှားတာ မဟုတ်ပါဘူး.. declaration က XML document ရဲ့အစိတ်အပိုင်းတစ်ခုမဟုတ်ပါဘူး ဒါကြောင့် အပိတ်မပါပါဘူး။

XML tags တွေက Case Sensitive ဖြစ်ပါတယ်။ XMLမှာ <letter> က <Letter> နဲ့မတူပါဘူး။ အပိတ်နဲ့အဖွင့် tag တွေကို တူညီတဲ့ case နဲ့ပဲ ရေးသားရပါမယ်။

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

XML elements တွေကို သေသေချာချာ nested လုပ်ရပါမယ်။ HTML မှာဆိုရင် လုပ်ချင်သလို လုပ်လို့ရပါတယ် ဥပမာ bold နဲ့ အရင်ဖွင့်ထား ပြီးမှ italic လိုက် ပိတ်တဲ့အခါ bold ကိုအရင်ပိတ် ပြီးမှ italic ကိုပိတ်သလိုမျိုး။

```
<b><i>This text is bold and italic</b></i>
```

XML မှာကတော့ အဲဒီလိုမဖြစ်ရပါဘူး နောက်မှလာတဲ့ဟာအရင်ပိတ်ရပါမယ်။ XML မှာဆို အောက်ကပုံစံမျိုးနဲ့ပဲ nested လုပ်ကို လုပ်ရပါမယ်။

```
<b><i>This text is bold and italic</i></b>
```

<i> က ရဲ့အတွင်းထဲရောက်မှ ဖွင့်တာဖြစ်တဲ့အတွက် ရဲ့အတွင်းထဲမှာပဲ ပိတ်ကိုပိတ်ရပါမယ် ဆိုတဲ့ သဘောပါ။

XML elements တွေမှာ HTML လိုမျိုးပဲ တန်ဖိုး အမည်နာမ အစုံလိုက်ပါတဲ့ attributes တွေရှိနိုင်ပါတယ်။

attribute value တွေကို XML မှာ quote လုပ်ရပါမယ် အောက်က ပုံစံနှစ်မျိုးကိုကြည့်ပါ ပထမတစ်မျိုးက မှားပြီးတော့ ဒုတိယ လိုရေးမှ မှန်ပါတယ်။

```
<note date=12/11/2007>
<to>Tove</to>
<from>Jani</from>
</note>
```

ဒီပထမဟာမှ " " မထည့်ထားလို့မှားနေပါတယ်။

```
<note date="12/11/2007">
<to>Tove</to>
<from>Jani</from>
</note>
```

ဒီဟာကမှ မှန်ပါတယ်။

Entity References

တစ်ချို့ အက္ခရာတွေမှတော့ XML မှာ သုံးတဲ့အခါ ထူးခြားတဲ့ အဓိပ္ပါယ်တွေရှိကြပါတယ်။ less than "<" character ကို XML element တစ်ခုရဲ့အတွင်းမှာ ထားတဲ့အခါ error ဖြစ်မှာဖြစ်ပါတယ်။ XML parser က ဒီအက္ခရာကို element တစ်ခုရဲ့ အပါအဝင် ဘာသာပြန်တဲ့အတွက်ဖြစ်ပါတယ်။ အောက်ပါစာကြောင်းက error ပြမှာဖြစ်ပါတယ်။

```
<message>if salary < 1000 then</message>
```

ဒီ error ကိုမပေါ်အောင် "<" character ကို entity reference တစ်ခုနဲ့အစားထိုးရပါမယ်။

```
<message>if salary &lt; 1000 then</message>
```

XML မှာ ကြိုတင် သတ်မှတ်ပြီးသား entity references ငါးခုရှိပါတယ်။

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

တကယ်တမ်းမှာတော့ "<" နဲ့ "&" တို့ကသာ XML မှာ illegal ဖြစ်တာပါ.. greater than character ကိုတော့သုံးနိုင်ပါတယ် ဒါပေမယ့် အကောင်းဆုံးကတော့ entity reference နဲ့အစားထိုးသုံးတာပါ။

Comments in XML

XML မှာ comment ရေးတဲ့ syntax ကတော့ HTML မှာနဲ့တူပါတယ်။

```
<!--This is a comment -->
```

What is an XML Element?

XML document တစ်ခုမှာ XML elements တွေပါဝင်ပါတယ်။ XML element ဆိုတာ element ရဲ့ start tag ကနေ element ရဲ့ end tag အထိကိုဆိုလိုပါတယ်။ element တစ်ခုမှာ တခြား elements တွေ ထပ်ပါနိုင်ပါတယ်။ ရိုးရိုးစာတွေလည်းပါနိုင်ပါတယ်။ element တွေမှာ သူတို့ရဲ့ attributes တွေလည်း ပါဝင်နိုင်ပါတယ်။

```
<bookstore>
<book category="CHILDREN">
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title>Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

ဒီဥပမာမှာဆို <bookstore> နဲ့ <book> တို့မှာ element contents တွေဖြစ်ကြပါတယ်။ သူတို့မှာ တခြား elements တွေ ထပ်ပါဝင်နေပါတယ်။ <author> ကတော့ text content ပါ။ သူ့မှာ text တွေပါတယ်။ ဒီဥပမာမှာ <book> မှာ attribute တစ်ခုရှိပါတယ်။ (category="CHILDREN")

XML Naming Rules

XML အနေနဲ့ အောက်ပါအမည်ပေး ဥပဒေတွေကို လိုက်နာရပါမယ်။

နာမည်တွေမှာ letter, numbers တွေနဲ့ တခြား character တွေပါဝင်နိုင်ပါတယ်။ နာမည်တွေဟာ နံပါတ်ဒါမှမဟုတ် punctuation character တစ်ခုနဲ့ မစနိုင်ပါဘူး။ နာမည်တွေဟာ xml, XML, Xml ဆတဲ့ စကားလုံးတွေနဲ့ မစနိုင်ပါဘူး။ နာမည်တွေမှာ spaces တွေ မပါဝင်နိုင်ပါဘူး။ အဲဒါအပြင်တော့ ဘယ်နာမည်ကိုမဆို အသုံးပြုနိုင်ပါတယ်။ ကြိုတင် ဖယ်ရှားထားတဲ့ စကားလုံးမရှိပါဘူး။

XML Elements are Extensible

XML elements တွေကို အချက်အလက်တွေများများပိုသယ်ဆောင်နိုင်ဖို့အတွက် ချဲ့ထွင်ရေးနိုင်ပါတယ်။

အောက်ပါဥပမာကိုကြည့်ပါ။

```
<note>
<to>Tove</to>
<from>Jani</from>
<body>Don't forget me this weekend!</body>
</note>
```

ကျနော်တို့အနေနဲ့ လိုချင်တဲ့ အောက်က Output ကို XML document ရဲ့ <to>, <from> နဲ့ <body> elements တွေကနေ ရယူမယ့် application တစ်ခုကို ဖန်တီးတယ်လို့ ယူဆကြပါစို့။

MESSAGE

To: Tove
From: Jani

Don't forget me this weekend!

အဲဒီအခါမှာ နောက်ထပ် အချက်အလက်အပိုတွေကို ကျနော်တို့အခုလို ထပ်ထည့်မယ်လို့ယူဆမယ်ဆိုရင်

```
<note>
<date>2008-01-10</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ဒီ application က crash ဖြစ်သွားမယ် break ဖြစ်သွားနိုင်ပါသလား။

အဖြေကတော့ No ပါ။ ဒီ application အနေနဲ့ <to>, <from> <body> ထဲ element တွေကို ရှာတွေ့ပြီးတော့ တူညီတဲ့ output ကိုထုတ်ပေးနိုင်မှာပဲဖြစ်ပါတယ်။ XML ရဲ့ကောင်းချက်တစ်ခုကတော့ သူက application ကို ဖျက်ပစ်ခြင်းမရှိဘဲနဲ့ extend ထပ်လုပ်နိုင်တာဖြစ်ပါတယ်။

XML Attributes

XML elements တွေမှာ HTML လိုမျိုးပဲ သူ့ရဲ့ start tag အတွင်းမှာ attribute တွေရှိနိုင်ပါတယ်။ Attribute တွေက element နဲ့ပတ်သတ်တဲ့ အချက်အလက်တွေကိုထပ်ပြီး ပံ့ပိုးပေးပါတယ်။

HTML မှာတုန်းက ကျနော်တို့အနေနဲ့ ဆိုတာမျိုးကို မှတ်မိကြမှာပါ။ src ဆိုတဲ့ attribute က element အတွက်အချက်အလက်တွေကို ထပ်ပြီး ပံ့ပိုးပေးထားတာပါ။

```
ဥပမာ

<a href="demo.asp">
```

XML attribute တွေကို " " Quote အတွင်းမှာရေးရပါတယ်။ single quote ရော double quote ရောကြိုက်တာ သုံးနိုင်ပါတယ်။ လူတစ်ယောက်ရဲ့ sex အမျိုးအစားကိုဖော်ပြဖို့အတွက် person ဆိုတဲ့ tag ကိုဒီလိုရေးသားနိုင်ပါတယ်။

```
<person sex="female">
```

ဒါမှမဟုတ်

```
<person sex='female'>
```

attribute မှာအစကတည်းက double quote ပါနေတဲ့ဟာမျိုးဆို အောက်ကဥပမာလိုမျိုး single quote ကိုအသုံးပြုနိုင်ပါတယ်။

```
<gangster name='George "Shotgun" Ziegler'>
```

ဒီလိုရေးနိုင်ပါတယ် ဒါမှမဟုတ် character entity ကိုသုံးနိုင်ပါတယ်။

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

XML Elements vs. Attributes

အောက်ပါဥပမာနှစ်ခုကို ယှဉ်ကြည့်ပါ။

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

ထမတစ်ခုမှာ sex က attribute ဖြစ်ပါတယ်။ နောက်တစ်ခုတော့ sex က element ဖြစ်ပါတယ်။ ဥပမာ နှစ်ခုစလုံးက တူညီတဲ့အချက်အလက်တွေကို ဖော်ပြပေးပါတယ်။ ဘယ်အချိန်မှာ attribute ကိုသုံးပြီး ဘယ်အချိန်မှာ element ကိုသုံးရမယ်ဆိုတဲ့ ဥပဒေရှိပါတူး။ HTML မှာတော့ attribute တွေကိုအသုံးများပါတယ်။ XML မှာတော့ အဲဒါတွေမသုံးတာ ပိုကောင်းပါတယ်။ element ကိုသာ attribute အစားပိုသုံးသင့်ပါတယ်။

My Favorite Way

အောက်ဖော်ပြပါ XML documents သုံးခုက တူညီတဲ့အချက်အလက်တွေပါဝင်ပါတယ်။

ထမဥပမာမှာ date attribute ကိုအသုံးပြုထားပါတယ်။

```
<note date="10/01/2008">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

အောက်ကဒုတိယဥပမာမှာတော့ date element ကိုသုံးပါတယ်။

```
<note>
<date>10/01/2008</date>
```



```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

date element ကိုမှ ထပ်ပြီး ချဲ့ထွင်ထားတဲ့ ပုံစံကို ဒီတတိယမြောက်မှာတွေ့ရပါမယ်။ ဒါကျနော်အကြိုက်ဆုံး ပုံစံပါ။

```
<note>
<date>
  <day>10</day>
  <month>01</month>
  <year>2008</year>
</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

XML Attributes for Metadata

တစ်စုံတစ်ရာ elements တွေကို ID references တွေထားပေးရပါတယ်။

```
messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

ဒီဥပမာမှာ ပါတဲ့ identifier ကတော့ မတူညီတဲ့ notes တွေကို ခွဲခြားပေးဖို့ပဲဖြစ်ပါတယ်။ note ရဲ့အစိတ်အပိုင်းတစ်ခု ယှဉ်ပါဘူး။ metadata တွေကို attributes အနေနဲ့ သိမ်းသွင်းပါတယ်။ data တွေကိုတော့ elements တွေအဖြစ်သိမ်းသွင်းပါတယ်။

XML Validation

မှန်ကန်တဲ့ syntax တွေနဲ့ရေးထားမဲ့ xml document တစ်ခုကို well formed လို့ခေါ်တွင်ပါတယ်။ သူ့ရဲ့ syntax rule တွေကတော့ အရှေ့မှာလည်း ပြောလာခဲ့ပါပြီ။

-XML document တွေမှာ root element တစ်ခုရှိရမယ်။

-XML tags တွေသည် case sensitive ဖြစ်သည်။

-XML elements တွေကို စနစ်တကျ nested လုပ်ရမယ်။

-XML attribute တန်ဖိုးတွေကို quote ထဲမှာထည့်ရေးရမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Valid XML Documents

valid ဖြစ်တဲ့ XML document တစ်ခုဟာ document type definition (DND) ဥပဒေကို လိုက်နာတဲ့ well form xml documents ကိုဆိုလိုပါတယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

အထက်ပါ ဥပမာမှာပါတဲ့ DOCTYPE ကြေငြာချက်က ပြင်ပ DTD file တစ်ခုကို မှီငြမ်းထားတာဖြစ်ပါတယ်။ အောက်မှာ အဲဒီ DTD file ထဲက content တွေကိုဖော်ပြထားပါတယ်။

XML DTD

DTD ရဲ့ရည်ရွယ်ချက်ကတော့ XML document တစ်ခုရဲ့ ဖွဲ့စည်းတည်ဆောက်ပုံကို define လုပ်ပေးဖို့ပဲဖြစ်ပါတယ်။ legal elements တွေရဲ့ structure ကိုသတ်မှတ်ပေးပါတယ်။

```
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
```

XML Schema

W3C က DTD အစားသုံးနိုင်တဲ့ XML Schema ဆိုတာကိုလည်း support လုပ်ပါသေးတယ်။

```
<xs:element name="note">
<xs:complexType>
  <xs:sequence>
    <xs:element name="to" type="xs:string"/>
    <xs:element name="from" type="xs:string"/>
    <xs:element name="heading" type="xs:string"/>
```

```
<xs:element name="body" type="xs:string"/>
</xs:sequence>
</xs:complexType>

</xs:element>
```

XML Validator

XML document ထဲမှာ error တွေရှိတယ်ဆိုရင် XML application ကိုရပ်လိုက်မှာဖြစ်ပါတယ်။ error တစ်ခုကို တွေ့တာနဲ့ XML document တစ်ခုကို processing လုပ်နေရကနေ အဲဒီ Program ကိုရပ်လိုက်မယ်လို့ W3C XML specification ကဖော်ပြပါတယ်။ အကြောင်းပြချက်ကတော့ XML software က သေးငယ်ရမယ် မြန်ရမယ် compatible ဖြစ်ရမယ်ဆိုတဲ့အတွက် ပါ။

HTML မှာတော့ browsers တွေဟာ error တွေပါတဲ့ document တွေကိုလည်း ပြသကြပါလိမ့်မယ်။ XML မှာတော့ error တွေကို ခွင့်မပြုပါဘူး။

ကျနော်တို့ XML တွေရဲ့ syntax ကိုစစ်ဆေးနိုင်ဖို့အတွက် XML validator တစ်ခုကို ဖန်တီးအသုံးပြုနိုင်ပါတယ် ခင်ဗျာ။

Viewing XML Files

Raw XML file တွေကို ဘယ် browser ဆွဲမှာမဆို ကြည့်နိုင်ပါတယ်။ HTML page တွေလိုမျိုး XML files တွေကိုပြသ မှာတော့ မဟုတ်ပါဘူး။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Displaying XML with CSS

XML document တစ်ခုကို CSS သုံးပြီး ပုံစံချဖို့ ဖြစ်နိုင်ပါတယ်။ အောက်ပါဥပမာမှာ ပြထားပါတယ်။

CD Catalog XML file က ဒီလိုပုံစံရှိပါမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<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>
    .
    .
    .
    .
</CATALOG>

```

CSS file က ဒီလိုရှိပါမယ်။

```

CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}

```

အဲဒါဆိုရင် CSS နဲ့ format လုပ်ထားတဲ့ CD catalog က အောက်ပါပုံစံမျိုးဖြစ်ပါတယ်။

Empire Burlesque Bob Dylan USA Columbia 10.90 1985 Hide your heart Bonnie Tyler UK CBS Records 9.90 1988 Greatest Hits Dolly Parton USA RCA 9.90 1982 Still got the blues Gary Moore UK Virgin records 10.20 1990 Eros Eros Ramazzotti EU BMG 9.90 1997 One night only Bee Gees UK Polydor 10.90 1998 Sylvias Mother Dr.Hook UK CBS 8.10 1973 Maggie May Rod Stewart UK Pickwick 8.50 1990 Romanza Andrea Bocelli EU Polydor 10.80 1996 When a man loves a woman Percy Sledge USA Atlantic 8.70 1987 Black angel Savage Rose EU Mega 10.90 1995 1999 Grammy Nominees Many USA Grammy 10.20 1999 For the good times Kenny Rogers UK Mucik Master 8.70 1995 Big Willie style Will Smith USA Columbia 9.90 1997 Tupelo Honey Van Morrison UK Polydor 8.20 1971 Soulsville Jorn Hoel Norway WEA 7.90 1996 The very best of Cat Stevens UK Island 8.90 1990 Stop Sam Brown UK A and M 8.90 1988 Bridge of Spies T` Pau UK Siren 7.90 1987 Private Dancer Tina Turner UK Capitol 8.90 1983 Midt om natten Kim Larsen EU Medley 7.80 1983 Pavarotti Gala Concert Luciano Pavarotti UK DECCA 9.90 1991 The dock of the bay Otis Redding USA Atlantic 7.90 1987 Picture book Simply Red EU Elektra 7.20 1985 Red The Communards UK London 7.80 1987 Unchain my heart Joe Cocker USA EMI 8.20 1987

XML ကို CSS နဲ့ format လုပ်တာက အရိုးရှင်းဆုံးနည်းလမ်းပါ။ W3C ကတော့ XSLT ကိုသုံးတာကို recommend လုပ်ပါတယ်။
 ရှေ့မှာ ဆက်ကြည့်ကြရအောင်။

Displaying XML with XSLT

XLST ကိုသုံးပြီးတော့ XML document တစ်ခုကို HTML အဖြစ်ကို အသွင်ပြောင်းပေးနိုင်ပါတယ်။

XSLT က XML ရဲ့ recommend ပေးခံရတဲ့ style sheet language ဖြစ်ပါတယ်။ XSLT(eXtensible Stylesheet Language Transformations) က CSS ထက်ပိုပြီးတော့ ဆန်းပြားရှုပ်ထွေးပါတယ်။

XSLT ကိုအသုံးပြုပုံကတော့ အောက်က ဥပမာမှာပြသလို browser နေ မပြခင် XML ကို HTML အဖြစ်ပြောင်းတာဖြစ်ပါတယ်။

ဒါက XML file ပါ။

```
<!-- Edited by XMLSpy® -->

<breakfast_menu>

<food>

<name>Belgian Waffles</name>

<price>$5.95</price>

<description>

two of our famous Belgian Waffles with plenty of real maple syrup

</description>

<calories>650</calories>

</food>

<food>

<name>Strawberry Belgian Waffles</name>
```

```
<price>$7.95</price>

<description>

light Belgian waffles covered with strawberries and whipped cream

</description>

<calories>900</calories>

</food>

<food>

<name>Berry-Berry Belgian Waffles</name>

<price>$8.95</price>

<description>

light Belgian waffles covered with an assortment of fresh berries and whipped
cream

</description>

<calories>900</calories>

</food>

<food>

<name>French Toast</name>

<price>$4.50</price>

<description>

thick slices made from our homemade sourdough bread

</description>

<calories>600</calories>

</food>

<food>

<name>Homestyle Breakfast</name>

<price>$6.95</price>

<description>

two eggs, bacon or sausage, toast, and our ever-popular hash browns
```

```
</description>

<calories>950</calories>

</food>

</breakfast_menu>
```

XSLT Style Sheet က အောက်ပါအတိုင်းပါ။

-
(calories per serving)

ဒီနှစ်ခုပေါင်းထားတဲ့ result ကတော့

Belgian Waffles - \$5.95
two of our famous Belgian Waffles with plenty of real maple syrup650 (calories per serving)
Strawberry Belgian Waffles - \$7.95
light Belgian waffles covered with strawberries and whipped cream900 (calories per serving)
Berry-Berry Belgian Waffles - \$8.95
light Belgian waffles covered with an assortment of fresh berries and whipped cream900 (calories per serving)
French Toast - \$4.50
thick slices made from our homemade sourdough bread600 (calories per serving)
Homestyle Breakfast - \$6.95
two eggs, bacon or sausage, toast, and our ever-popular hash browns950 (calories per serving)

ဖြစ်ပါတယ် ခင်ဗျာ။

XML file ရဲ့အပေါ်ဆုံးမှာ ဒီလိုနှစ်ကြောင်းပါပါမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="simple.xsl"?>
```

ဒုတိယအကြောင်းကတော့ XML file ကို XSLT file နဲ့ ချိတ်ဆက်ပေးတာဖြစ်ပါတယ်။

အခုအချိန်ထိ ကျနော်တို့ XML ရဲ့အခြေခံတွေကို လေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။ အခုက စပြီး PHP နဲ့ XML နဲ့အပိုင်းကို ဆက်လက်လေ့လာပါမယ်။

PHP XML Expat Parser

PHP ရဲ့ Built in Expat parser က PHP ကိုသုံးပြီး XML document တွေကို process လုပ်ဖို့ ဖြစ်နိုင်စေပါတယ်။

What is Expat?

XML document တစ်ခုကို ဖတ်ဖို့ အသစ်ပြုပြင်ဖို့ ဖန်တီးပြီး တွက်ချက်ထုတ်ဖို့အတွက် XML parser တစ်ခုကို လိုအပ်ပါတယ်။

XML parsers အခြေခံအားဖြင့် နှစ်မျိုးရှိပါတယ်။

-Tree-based parser: ဒီ parser ကတော့ XML document တစ်ခုကို tree structure အဖြစ်ကို ပြောင်းလဲပေးပါတယ်။ သူက document တစ်ခုလုံးကို analyze လုပ်ပြီးတော့ tree elements တွေဆီကို access လုပ်ပေးပါတယ်။ ဥပမာ Document Object Model(DOM) လိုမျိုးပါ။

-Event-based parser: XML document တစ်ခုကို events တွေပါဝင်တဲ့ series တစ်ခုအဖြစ် ရှုမြင်ပါတယ်။ specific event တစ်ခု ဖြစ်ပွားတဲ့အခါ သူ့ကို ကိုင်တွယ်ဖို့အတွက် function တစ်ခုကို ခေါ်ယူသုံးပါတယ်။

Expat parser ကတော့ event-based parser ဖြစ်ပါတယ်။

Event-based parsers က XML document ရဲ့ content ပေါ်မှာ မူတည်ပါတယ်။ Structure ပေါ်မှာ မူမတည်ပါဘူး။ ဒါကြောင့် event-based parser တွေက tree-based parses တွေထက် ပိုမြန်မြန် data ကို access လုပ်နိုင်ပါတယ်။

အောက်ပါအပိုင်းလေးကို ကြည့်မယ်ဆိုရင်

```
<from>Jani</from>
```

event သုံးခုပါတဲ့ series တစ်ခုအဖြစ် အထက်ပါ XML ကို event-based parser တစ်ခုက ဖော်ပြပါတယ်။

-Start Element:From

-Start CDATA section, တန်ဖိုးက Jani

-Close element: from

ဒီ XML ဥပမာက well-formed XML ပါဝင်ပါတယ်။ ဒါပေမယ့် valid XML တစ်ခုမဟုတ်ပါဘူး။ သူနဲ့ ဆက်နွှယ်တဲ့ DTD မပါဝင်လို့ပါ။ Expat parser ကိုသုံးရင်တော့ မထူးခြားပါဘူး။ Expat ကလည်း DTDs တွေကို လျစ်လျူရှုတဲ့ non-validating parser ဖြစ်ပါတယ်။

event based အနေနဲ့ non-validating XML parser ဖြစ်တဲ့ Expat ကလည်း PHP web applications တွေအတွက် လျင်မြန်သေးသလှ ပြည့်စုံတဲ့ အတွဲအစပ်တစ်ခုဖြစ်ပါတယ်။ XML document တွေကိုတော့ well-formed ဖြစ်ရမှာဖြစ်ပြီး မဟုတ်ရင်တော့ Expat က error ထုတ်ပြောဖြစ်ပါတယ်။

Installation

XML Expat parser function တွေက PHP core ရဲ့အစိတ်အပိုင်းဖြစ်ပါတယ်။ ဒါကြောင့် ဒီ function ကို သုံးဖို့အတွက် ဘာမှ ထပ်ပြီး install လုပ်စရာမလိုပါဘူး။

ကျနော်တို့ရဲ့ဥပမာမှာ အောက်ပါ XML file ကိုအသုံးပြုကြပါမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```


Initializing the XML Parser

PHP မှာ XML parser ကို အစပြုပါမယ်။ ဆော့ညီတဲ့ XML events တွေအတွက် handlers အချို့ကို သတ်မှတ်ပါမယ်။ ပြီးတဲ့ အခါ XML file ကို parse လုပ်ပါမယ်။

ဥပမာ

```
<?php
//Initialize the XML parser
$parser=xml_parser_create();
//Function to use at the start of an element
function start($parser,$element_name,$element_attrs)
{
    switch($element_name)
    {
        case "NOTE":
            echo "-- Note --<br />";
            break;
        case "TO":
            echo "To: ";
            break;
        case "FROM":
            echo "From: ";
            break;
        case "HEADING":
            echo "Heading: ";
            break;
        case "BODY":
            echo "Message: ";
            break;
    }
}
//Function to use at the end of an element
function stop($parser,$element_name)
{
    echo "<br />";
}
//Function to use when finding character data
function char($parser,$data)
{
    echo $data;
}
//Specify element handler
xml_set_element_handler($parser,"start","stop");
//Specify data handler
xml_set_character_data_handler($parser,"char");
//Open XML file
$fp=fopen("test.xml","r");
//Read data
while ($data=fread($fp,4096))
{
    xml_parse($parser,$data,feof($fp)) or
    die (sprintf("XML Error: %s at line %d",
    xml_error_string(xml_get_error_code($parser)),
    xml_get_current_line_number($parser)));
}
//Free the XML parser
xml_parser_free($parser);
```

?>

ဒီ အထက်ပါ code ရဲ့ output ကတော့ အောက်ပါအတိုင်းပါ။

```
-- Note --
To: Tove
From: Jani
Heading: Reminder
Message: Don't forget me this weekend!
```

သူ့ရဲ့အလုပ်လုပ်ပုံကတော့

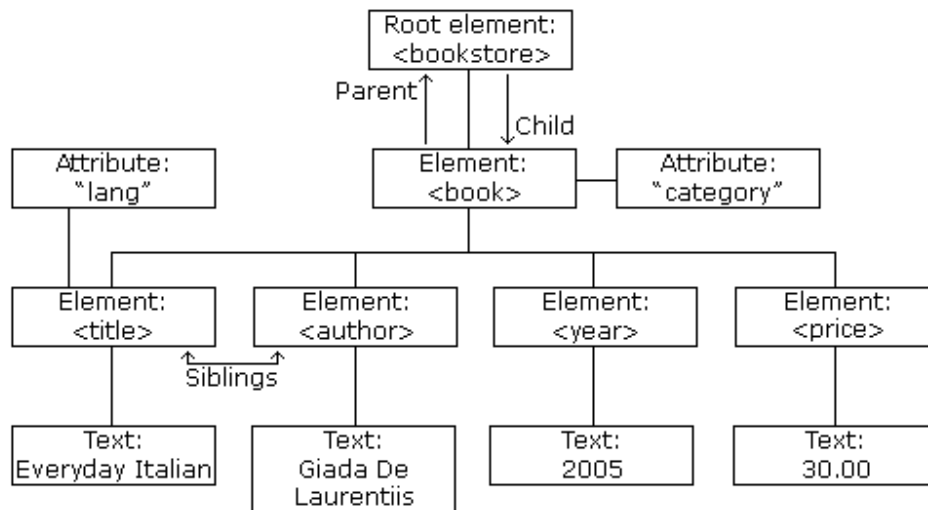
1. XML parser ကိုအစပြုဖို့ xml_parser_create() function ကိုခေါ်သုံးပြီး စလိုက်ပါတယ်။
2. မတူညီတဲ့ event handlers ဆွဲနဲ့အသုံးပြုနိုင်ဖို့ function တွေကို ဖန်တီးပါတယ်။
3. parser က opening နဲ့ closing tag တွေကိုတွေ့တဲ့အခါ ဘယ် function တွေကို အလုပ်လုပ်စေမယ်ဆိုတာကို သတ်မှတ်ပေးဖို့အတွက် xml_set_handler() function ကို ပေါင်းထည့်ပါတယ်။
4. parser က character data ကို တွေ့တဲ့အခါ ဘယ် function ကိုတွက်ချက်မယ်ဆိုတာ သတ်မှတ်ဖို့အတွက် xml_set_character_data_handler() ကိုပေါင်းထည့်ပါတယ်။
5. text.xml ဖိုင်လေးကို parse လုပ်ပါတယ်။ xml_parse () function ကိုသုံးပြီးတော့ပါ။
6. အကယ်လို့ error ဖြစ်တယ်ဆို xml_error_string() function ကိုသုံးပြီး XML error တစ်ခုကို စာသားဖော်ပြချက်တစ်ခု အဖြစ် ပြောင်းပေးပါတယ်။
7. xml_parser_create() function နဲ့ memory allocate လုပ်ထားတာကို ဖယ်ထုတ်ဖို့ xml_parser_free() function ကို ခေါ်လိုက်ပါတယ်။

PHP XML DOM

built-in ပါပြီးသား DOM parser ကိုသုံးပြီး PHP ထဲမှာ XML documents တွေကို Process လုပ်ခွင့်ရစေပါတယ်။

What is DOM?

XML DOM(Document Object Model) က XML documents တွေကို access လုပ်တာရယ် တွက်ချက်တာရယ် အတွက် standard နည်းလမ်းတစ်ခုဖြစ်ပါတယ်။ DOM က XML document ကို tree structure နဲ့ပုံပြပါတယ် element တွေ attribute တွေ နဲ့ nodes တွေအဖြစ် စာသားတွေပါဝင်ပါတယ်။အောက်ပါပုံစံအတိုင်းပါ။



XML နဲ့အလုပ်လုပ်နေတဲ့ သူတိုင်းက XML DOM ကိုသိကိုသိရပါမယ်။

PHP XML DOM ကိုဆက်မကြည့်ခင် DOM အခြေခံလေး နည်းနည်းတော့ လေ့လာပါမယ်။ DOM ဆိုတာ W3C (World Wide Web Consortium) Standard ဖြစ်ပါတယ်။ DOM က XML နဲ့ HTML လို document တွေကို access လုပ်ဖို့အတွက် စံတစ်ခုပဲ ဖြစ်ပါတယ်။ DOM ကို သုံးဆိုင်းခွဲထားပါတယ်။

-Core DOM- ဘယ်လိုဖွဲ့စည်းထားတဲ့ document မဆိုအတွက် စံ model

-XML DOM- XML documents တွေအတွက် စံ model

-HTML DOM – HTML documents တွေအတွက် စံ model

တို့ဖြစ်ပါတယ်။

DOM က document elements အားလုံးနဲ့ object နဲ့ properties တွေကို သတ်မှတ်ပေးပြီးတော့ သူတို့ကို access လုပ်မယ့် methods (interface) တွေကိုလည်း သတ်မှတ်ပေးပါတယ်။ HTML DOM ကိုတော့ မကြည့်တော့ဘဲ XML DOM ကိုဆက်ကြည့် ပါမယ်။

XML DOM ကတော့

-XML အတွက် standard Object Model တစ်ခုဖြစ်ပါတယ်။

-XML အတွက် standing programming interface ဖြစ်ပါတယ်။

-Platform နဲ့ Language မှီခိုမှုမရှိပါဘူး။

-W3C စံသတ်မှတ်ချက်တစ်ခုပါ။

XML DOM ဆိုတာ XML elements တွေကို ဘယ်လို ရယူဖို့ ပြောင်းလဲဖို့ မောင်းထည့်ဖို့ နဲ့ ဖျက်ဆီးဖို့ တို့အတွက် စံ တစ်ခု ဖြစ်တယ်လို့ ပြောရပါမယ်။

XML DOM Nodes

DOM ထဲမှာဆိုရင် XML document ထဲမှာ ရှိတဲ့ အရာရာတိုင်းက node တစ်ခုဖြစ်ကြပါတယ်။

DOM အနေနဲ့ဆိုရင်

-document တစ်ခုလုံးဟာ document node တစ်ခုဖြစ်တယ်။

-XML element တိုင်းဟာ element node တစ်ခုဖြစ်တယ်။

-XML elements ထဲမှာရှိတဲ့ text တွေဟာ text node တွေဖြစ်ကြတယ်။

- attribute တွေတိုင်းဟာ attribute node တစ်ခုဖြစ်တယ်။

-Comments တွေဟာ comment nodes တွေဖြစ်ကြတယ်။

DOM Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="web">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
```

```
<book category="web" cover="paperback">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

ဒီ XML ထဲက root node က <bookstore> ဖြစ်ပါတယ်။ တခြားသော nodes တွေအကုန်လုံးက <bookstore> ထဲမှာ ရှိတာဖြစ်ပါတယ်။

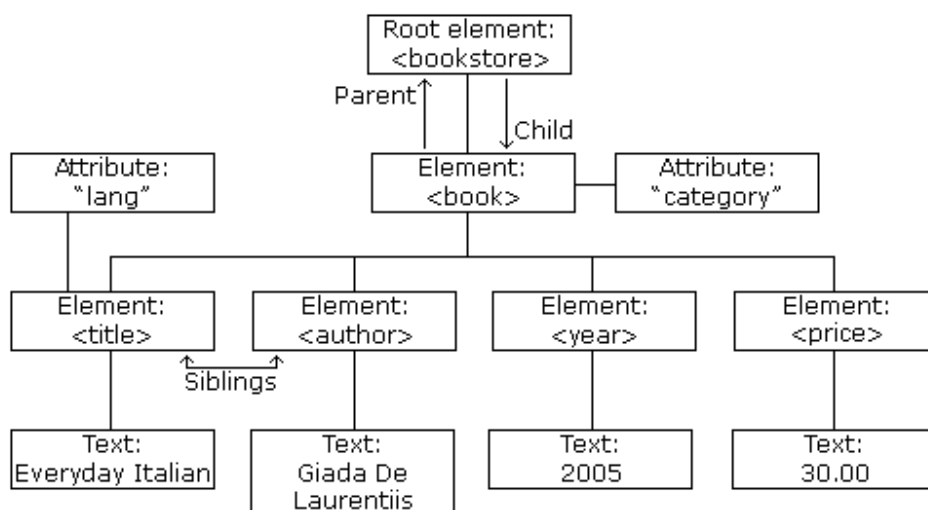
root node <bookstore> ထဲမှာ <book> nodes လေးခုရှိပါတယ်။ ပထမ <book> node မှာ <title>, <author>, <year> နဲ့ <price> ဆိုတဲ့ nodes လေးခုရှိပါတယ်.. တစ်ခုစီမှာ "Everyday Italian", "Giada De Laurentiis", "2005", and "30.00" ဆိုတဲ့ text node တွေပါဝင်ကြပါတယ်။

DOM processing မှာဖြစ်နေကြ error ကတော့ စာသားတွေ ထည့်ဖို့အတွက် element node တစ်ခုကို မှောင်လင့်တာပါပဲ။ ဘယ်လိုပဲဖြစ်ဖြစ် element node တစ်ခုရဲ့ စာသားတွေကို text node တစ်ခုထဲမှာပဲ သိမ်းဆည်းပါတယ်။ ဒီဥပမာမှာဆိုရင် <year>2005</year> မှာဆိုရင် <year> ဆိုတဲ့ element node က 2005 ဆိုတဲ့ တန်ဖိုးရှိတဲ့ text node တစ်ခုကို ထိန်းသိမ်း ပါတယ်။ 2005 ဆိုတာ <year> ဆိုတဲ့ element ရဲ့ တန်ဖိုးမဟုတ်ပါဘူး။

XML DOM Node Tree

XML DOM က XML document တစ်ခုကို node- tree တစ်ခုအနေနဲ့ မြင်ပါတယ်။ tree တစ်ခုမှာရှိတဲ့ nodes တွေအားလုံး မှာ တစ်ခုနဲ့တစ်ခု ဆက်သွယ်ချက်တစ်ခုရှိကြပါတယ်။

nodes တွေအားလုံးကို tree ကိုဖြတ်ပြီး access လုပ်နိုင်ပါတယ်။ သူတို့ရဲ့ content တွေကို ပြင်ဆင်နိုင်ပါတယ်၊ ဖျက်နိုင်ပါတယ်၊ အသစ် element တွေ ထပ်ဖန်တီးနိုင်ပါတယ်။ node tree က nodes တွေ ရဲ့အရာကို ပြသပေးပါတယ်။ ပြီးတော့ သူတို့ ကြားထဲက ဆက်သွယ်ချက်ကိုလည်း ဖော်ပြပါတယ်။ tree က root node ကနေ စပြီးတော့ tree ရဲ့ အနိမ့်ဆုံး အဆင့်အထိ text node တွေဆီကို ခွဲဖြာသွားပါတယ်။



ဒီမှာ တင်ပြထားတဲ့ ပုံက အောက်ပါ books.xml ဆိုတဲ့ ဖိုင်လေးကို ကိုယ်စားပြုပါတယ်။

```
<!-- Edited by XMLSpy® -->

<bookstore>

<book category="cooking">

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="children">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="web">

<title lang="en">XQuery Kick Start</title>

<author>James McGovern</author>

<author>Per Bothner</author>

<author>Kurt Cagle</author>

<author>James Linn</author>

<author>Vaidyanathan Nagarajan</author>

<year>2003</year>

<price>49.99</price>

</book>

<book category="web" cover="paperback">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>
```

```
<year>2003</year>

<price>39.95</price>

</book>

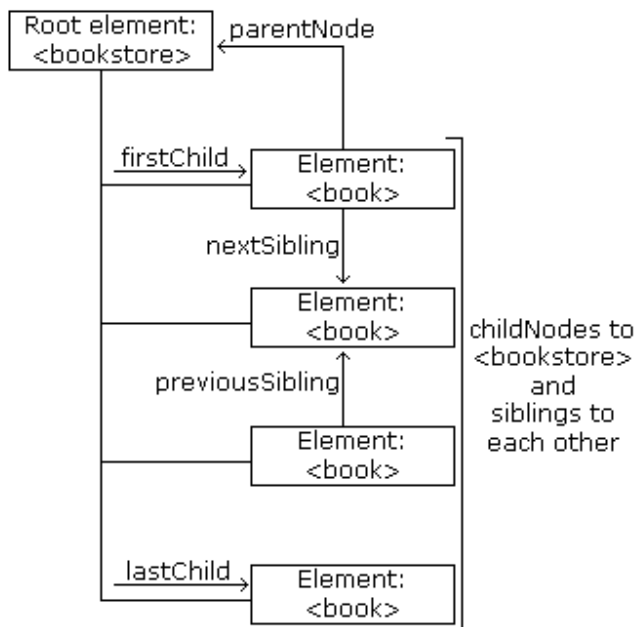
</bookstore>
```

Node Parents, Children, and Siblings

node tree မှာပါတဲ့ nodes တွေမှာ တစ်ခုနဲ့တစ်ခု မျိုးရိုးဆိုင်ရာ ဆက်သွယ်ချက် ရှိကြပါတယ်။ parent, child , sibling ဆိုတဲ့ ဝေါဟာရတွေကို သုံးပြီး သူတို့ရဲ့ ဆက်သွယ်ချက်ကို ဖော်ပြကြပါတယ်။ parent node မှာ children တွေရှိပါတယ်။ level တူ children တွေက siblings တွေဖြစ်ကြပါတယ်။

- node tree တစ်ခုမှာ ထိပ်ဆုံး node ကို root လို့ခေါ်ပါတယ်။
- ပထမဆုံး root ကလွဲပြီး ကျန်တဲ့ node တိုင်းမှာ parent node တစ်ခုပဲရှိပါတယ်။
- node တစ်ခုမှာ ကလေး ဘယ်နှစ်ယောက်ဖြစ်ဖြစ် ရှိနိုင်ပါတယ်။
- leaf ဆိုတာကတော့ ကလေးမရှိတဲ့ node ကိုဆိုလိုပါတယ်။
- sibling တွေက မိဘတူညီတဲ့ nodes တွေဖြစ်ပါတယ်။

အောက်ပါ ပုံက node tree ရဲ့အစိတ်အပိုင်း နဲ့ nodes တွေကြားထဲက ဆက်သွယ်ချက်ကို ဖော်ပြပါတယ်။



First Child - Last Child

အောက်က XML အပိုင်းလေးကို ကြည့်မယ်ဆိုရင်

```
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

<title> element က <book> element ရဲ့ ပထမဆုံး ကလေး ဖြစ်ပါတယ်။ <price> element ကတော့ <book> element ရဲ့ နောက်ဆုံးကလေးဖြစ်ပါတယ်။ ဒါ့အပြင် <book> element က <title>, <author>, <year>, နဲ့ <price> elements တွေရဲ့ parent node ဖြစ်ပါတယ်။

XML Parsing in PHP

DOM အကြောင်း နည်းနည်း လေ့လာပြီးပြီဆိုတော့ PHP ဘက်ပြန်သွားပါမယ်။ အထက်မှာ ပြောခဲ့သလိုပဲ ကျနော်တို့က XML document တစ်ခုကို ဖတ်ဖို့ ပြင်ဆင်ဖို့ ဖန်တီး တွက်ချက်ဖို့အတွက် XML parser ကိုလိုပါတယ်။

XML parser အမျိုးအစားနှစ်ခုရှိပြီးတော့ tree-based နဲ့ event-based ဆိုပြီးဖြစ်ပါတယ်။

နောက် လေ့လာခဲ့တဲ့ Expat parser က event based ဖြစ်ပါတယ်။ အခုလေ့လာမယ့် DOM parser ကတော့ tree-based ဖြစ်ပါတယ်။

အောက်ပါ XML အပိုင်းလေးကို လေ့လာရအောင်

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<from>Jani</from>
```

XML DOM အနေနဲ့ ဒီ XML လေးကို tree structure နဲ့မြင်ပါတယ်။

- Level 1: XML Document
- Level 2: Root element: <from>
- Level 3: Text element: "Jani"

DOM XML parser ကလည်း PHP core ရဲ့အစိတ်အပိုင်းဖြစ်တဲ့အတွက် သူ့ရဲ့ function တွေကိုသုံးဖို့ ထပ်မံ install လုပ်စရာ မလိုပါဘူး။

အောက်ပါ XML file ကို ကျနော်တို့ရဲ့ ဥပမာအတွက် အသုံးပြုပါမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```


အခု XML parser ကို အစပြုပါမယ် ပြီးရင် xml ကို load လုပ်ပြီးတော့ output ပြပါမယ်။

ဒါကတော့ PHP code ပါ။

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");

print $xmlDoc->saveXML();
?>
```

ဒီ code ရဲ့ Output က အောက်ပါအတိုင်းဖြစ်ပါတယ်။

Tove Jani Reminder Don't forget me this weekend!

တကယ်လို့ ကျနော်တို့ browser window ရဲ့ view source ကို ကြည့်မယ်ဆိုရင် အောက်ပါ HTML ကိုမြင်ရပါမယ်။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ဒီဥပမာဟာဆိုရင် DOM Document Object တစ်ခုကို ဖန်တီးပြီးတော့ သူ့အထဲကို note.xml ကနေ XML ကိုလှမ်းယူပြီး load လုပ်ပါတယ်။ အဲဒီတော့ saveXML() function နဲ့ XML document ကို string အဖြစ်ပြောင်းပြီး print ထုတ်လို့ရပါတယ်။

Looping through XML

ဆိုပါစို့ XML parser ကိုစမယ် ပြီးရင် XML ကို load လုပ်မယ် ပြီးတဲ့အခါ <note> element ထဲမှာရှိတဲ့ element တွေအားလုံးကို looping ပတ်မယ်ပေါ့။

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");
$x = $xmlDoc->documentElement;
foreach ($x->childNodes AS $item)
{
    print $item->nodeName . " = " . $item->nodeValue . "<br />";
}
?>
```

ဒီ code ရဲ့ output ကတော့ ဒီလိုပါ။

```
#text =
to = Tove
#text =
from = Jani
#text =
heading = Reminder
#text =
body = Don't forget me this weekend!
```

```
#text =
```

ဒီဥပမာမှာဆိုရင် element တစ်ခုရဲ့ကြားမှာ ဗလာ ဖြစ်နေတဲ့ text nodes တွေရှိတာကိုတွေ့ရပါမယ်။ XML generate လုပ်တဲ့အခါ nodes တွေရဲ့ကြားထဲမှာ white-space တွေပါဝင်လာပါတယ်။ XML DOM parser က အဲဒါတွေကို သာမန် element တွေအဖြစ်မှတ်ယူပါတယ်။ တကယ်လို့ ကျနော်တို့က သတိမထားမိဘူးဆို တခါတလေ ပြဿနာတွေကိုဖြစ်စေမှာပါ။

PHP SimpleXML

SimpleXML ကတော့ PHP5 မှာမှ ပါဝင်လာတာပါ။ သူကလည်း တကယ်လို့ကျနော်တို့က XML document ရဲ့ layout ကိုသိရင် element တစ်ခုရဲ့ attribute နဲ့ text တွေကို ရယူနိုင်ဖို့အတွက် လွယ်ကူတဲ့ နည်းလမ်းတစ်ခုပဲဖြစ်ပါတယ်။

DOM တို့ Expat parser တို့နဲ့ယှဉ်မယ်ဆို SimpleXML က element တစ်ခုကနေ text data တွေဖတ်ဖို့အတွက် code နည်းနည်းပဲ လိုအပ်ပါတယ်။

SimpleXML က အောက်ပါအတိုင်း XML document တစ်ခုကို object တစ်ခုအဖြစ်ပြောင်းလဲပေးပါတယ်။

Elements- တွေကို SimpleXMLElement object ရဲ့ သီးခြား attribute အဖြစ်ပြောင်းပေးပါတယ်။ level တစ်ခုမှာ element တစ်ခုထက်ပိုပြီးရှိနေရင် array တစ်ခုအတွင်းမှာ ထားပေးပါတယ်။

Attributes တွေကိုတော့ associative array ကိုအသုံးပြုပြီး access လုပ်ပါတယ်။ index တစ်ခုက attribute name တစ်ခုကို ဆက်သွယ်နေပါတယ်။

Element data- element တွေက text data တွေကိုတော့ strings တွေအဖြစ်ပြောင်းပေးပါတယ်။ element တစ်ခုမှာ text node တစ်ခုထက်ပိုပြီးရှိနေရင် သူတို့ကို ရှိတဲ့ order အတိုင်း အစီအစဉ်တကျထားပေးပါတယ်။

SimpleXML က အောက်ပါအခြေခံလုပ်ငန်းစဉ်တွေကိုဆောင်ရွက်တဲ့အခါ လွယ်ကူပြီး မြန်ဆန်ပါတယ်။

-XML file တွေကို read လုပ်တာ

-XML string တွေက data တွေကို extract လုပ်တာ

-text nodes တွေဒါမှမဟုတ် attribute တွေကိုတည်းဖြတ်တာ။

ဘယ်လိုပဲဖြစ်ဖြစ် advanced XML မှာအလုပ်လုပ်တဲ့အခါ Expat parser ဒါမှမဟုတ် XML DOM ကိုသုံးတာက ပိုပြီး သင့်လျော်ပါတယ်။

PHP 5.0 မှာ SimpleXML functions တွေက PHP core ထဲမှာပါလာပြီးတော့ ဒီ function တွေကို install လုပ်စရာမလိုဘဲ သုံးနိုင်ပါတယ်။

Using SimpleXML

အောက်ကတော့ XML file တစ်ခုပါ။

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
```

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ကျနော်တို့က အဲဒီ file ထဲကနေ element အမည်နဲ့ data တွေကို Output ပြချင်တယ်ဆိုပါစို့

လုပ်ရမှာတွေကတော့

1. XML file ကို Load လုပ်မယ်။
2. ပထမဆုံး element ရဲ့အမည်ကို ရယူမယ်။
3. child node တစ်ခုစီကို children()function ကိုသုံးပြီးတော့ loop လုပ်မယ်။
4. child node တစ်ခုစီရဲ့ element အမည်နဲ့ data တွေကို ထုတ်ပြမယ်။

ဥပမာ

```
<?php
$xml = simplexml_load_file("test.xml");
echo $xml->getName() . "<br />";
foreach($xml->children() as $child)
{
    echo $child->getName() . ": " . $child . "<br />";
}
?>
```

ဒီ code ကို run ရင် ဒီလို Output မျိုးရပါပြီ။

```
note
to: Tove
from: Jani
heading: Reminder
body: Don't forget me this weekend!
```

Chapter 10. Exception Handling & Filter in PHP

Exception ဆိုတာဘာလဲ

PHP 5 နဲ့အတူ error တွေပတ်သတ်ပြီးကိုင်တွယ်ဖို့အတွက် Object Oriented နည်းလမ်းအသစ်တစ်ခု ထွက်ပေါ်လာပါတယ်။

Exception handling ဆိုတာ error အခြေအနေတစ်ခုခုဖြစ်လာပြီဆိုရင် code ကို execute လုပ်နေတဲ့ သူ့ရဲ့ flow ကို ပြောင်းလဲ ပစ်လိုက်ဖို့အသုံးပြုတာဖြစ်ပါတယ်။ ဒီအခြေအနေကို exception တစ်ခုလို့ခေါ်တွင်ပါတယ်။

အောက်ပါအခြေအနေတွေကတော့ exception တစ်ခုဖြစ်ပွားသွားပြီဆိုရင် ပုံမှန်ဖြစ်ပျက်သွားမယ့် ကိစ္စတွေပေါ့။

- လက်ရှိ code ရဲ့အခြေအနေကို သိမ်းဆည်းထား(save) ထားပေးမယ်။
- code execution က (custom) ကြိုတင် သတ်မှတ်ပေးထားတဲ့ exception handler function တစ်ခု ဆီကို ကူးပြောင်း သွားပေးမယ်။
- အခြေအနေပေါ်မှာမူတည်ပြီးတော့ handler က save လုပ်ပေးထားတဲ့ code အခြေအနေကနေ ဆက်ပြီးတော့ run တာကို resume လုပ်ပေးရင်လုပ်ပေးမယ်။ ဒါမှမဟုတ် script execution ကို အဆုံးသတ်ပစ်လိုက်မယ်။ ဒါမှမဟုတ် code ရဲ့တခြားတစ်နေရာကနေ script ကိုဆက်ပြီးတော့ run သွားပေးမယ်။

အောက်ပါError handling methods အမျိုးမျိုးကို လေ့လာကြပါမယ်။

- Basic use of Exceptions
- Creating a custom exception handler
- Multiple exceptions
- Re-throwing an exception
- Setting a top level exception handler

မှတ်ချက်- Exception များကို error အခြေအနေများတွင်သာ အသုံးပြုသင့်သည်။ code ထဲတွက်နေရာတစ်ခု မှ တခြားနေရာ တစ်ခုသို့ နှစ်ကူးရန်အတွက် အသုံးပြုသင့်ပါ။

Basic Use of Exceptions

exception တစ်ခုဖြစ်ပွားသွားပြီဆိုရင် code တွေကိုဆက်ပြီး execute လုပ်မှာမဟုတ်တော့ပါဘူး။ PHP က catch ဆိုတဲ့ block ကို ရှာဖွေဖို့ ကြိုးစားပါမယ်။ exception တစ်ခုကို caught မလုပ်ထားဘူးဆိုရင် Uncaught Exception ဆိုပြီး error ဖြစ်ပွားသွား မှာပါ။ catch မလုပ်ဘဲနဲ့ exception တစ်ခုကို ဖန်တီးကြရအောင်။

```
<?php
//create function with an exception
function checkNum($number)
{
    if($number>1)
    {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}
```

```
//trigger exception
checkNum(2);
?>
```

ကျနော်တို့မှာ အခုလို error ပြပါတယ်။

Fatal error: Uncaught exception 'Exception' with message 'Value must be 1 or below' in C:\xampp\htdocs\php\1.php:7 Stack trace: #0 C:\xampp\htdocs\php\1.php(13): checkNum() #1 {main} thrown in C:\xampp\htdocs\php\1.php on line 7

Try, throw and catch

Try, throw နဲ့ catch တို့ကတော့ exception တွေမှာ သုံးတဲ့ ဝေါဟာရတွေပါ။ အထက်လိုမျိုး error တွေမဖြစ်ရအောင် exception တစ်ခုကို ကိုင်တွယ်ဖြေရှင်းဖို့အတွက် code သေချာဖန်တီးဖို့လိုပါတယ်။

Proper exception code ပေါ့၊ စနစ်တကျရှိတဲ့ ကိုင်တွယ်နိုင်မယ် code တွေမှာ အောက်ပါအချက်တွေ ပါဝင်သင့်ပါတယ်။

1. Try- exception တစ်ခု ကိုအသုံးပြုမယ့် function က try block တစ်ခု ထဲမှာရှိနေသင့်ပါတယ်။ exception တစ်ခု ဖြစ်ပွားသွားဘူးဆိုရင် ဒီ code တွေက ပုံမှန်အတိုင်းပဲ ဆက် run သွားမှာပါ။ မဟုတ်ဘဲ exception တစ်ခု ဖြစ်ပွားသွားမယ် ဆိုမှသာ exception တစ်ခု thrown လုပ်ပေးမှာပေါ့။

2. Throw- ဒါကတော့ exception တစ်ခုကို ကျနော်တို့ ဘယ်လို ဖြေရှင်းမယ်ဆိုတာကိုပေါ့။ throw တိုင်းမှာ အနည်းဆုံးတော့ catch တစ်ခုရှိရပါမယ်။

3. Catch- catch block တစ်ခုက exception တစ်ခုကို လက်ခံရယူတယ်၊ ပြီးတဲ့နောက် exception information တွေပါဝင်တဲ့ object တစ်ခုကို ဖန်တီးပေးပါတယ်။

valid code တွေ exception တစ်ခုကို ဖြေရှင်းကြည့်ရအောင်

```
<?php
//create function with an exception
function checkNum($number)
{
    if($number>1)
    {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}

//trigger exception in a "try" block
try
{
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e)
```

```
{
    echo 'Message: ' . $e->getMessage();
}
?>
```

ဒီ ဖော်ပြပါ code က အောက်ပါအတိုင်း error ကိုကောင်းကောင်း မွန်မွန် output ထုတ်ပေးပါတယ်။

Message: Value must be 1 or below

ဒီဖော်ပြပါ ဥပမာက exception တစ်ခုကို throw ဖြစ်စေပါတယ် ပြီးတဲ့နောက်မှာ ပြန်ပြီး catch လုပ်လိုက်ပါတယ်။

1. checkNum() function ကိုဖန်တီးပါတယ်။ သူက နံပါတ်တစ်ခုကို တစ်ထက် ကြီးလား စစ်တာကို လုပ်ပါတယ်။ တကယ်လို့ ကြီးတယ်ဆိုရင် exception တစ်ခုကို ဖြစ်စေပါတယ်။
2. try block ထဲမှာ checkNum() function ကိုခေါ်လိုက်ပါတယ်။
3. checkNum() function ထဲမှာ ရှိတဲ့ exception ကြီးဖြစ်ပွား(throw) သွားပါတယ်။
4. catch block က exception ကိုရယူပြီးတော့ (\$e) object တစ်ခုကိုဖန်တီးပါတယ်။ ဒီ object မှာ exception information တွေပါဝင်ပါတယ်။
5. exception အတွက် error message ကို \$e->getMessage() ဆိုပြီးတော့ exception object ဆီကနေ လှမ်းခေါ်ပြီး echo နဲ့ထုတ်ပြပါတယ်။

Creating a Custom Exception Class

ကိုယ့်ဘာသာကိုယ် custom exception class တစ်ခုဖန်တီးတာကလည်း ရိုးရှင်းပါတယ်။ PHP မှာဖြစ်ပွားမယ့် exception တစ်ခု ပေါ်ပေါက်လာရင် ခေါ်သုံးလို့ရမယ့် function တွေပါမယ့် class တစ်ခုကို ဖန်တီးရုံပါ။ ဒီ class ကတော့ exception class ရဲ့ extension ဖြစ်ရပါမယ်။

ကိုယ့်ဘာသာကိုဖန်တီးမယ့် custom exception class က PHP ရဲ့ exception class ဆီက ဂုဏ်သတ္တိတွေကို မျိုးရိုးဆက်ခံရမှာ ဖြစ်ပြီးတော့ custom function တွေကို ထပ်ပေါင်းထည့်ပေးနိုင်ပါတယ်။

exception class တစ်ခုကို ဖန်တီးကြရအောင်

```
<?php
class customException extends Exception
{
    public function errorMessage()
    {
        //error message
        $errorMsg = 'Error on line '.$this->getLine().' in '.$this->getFile()
        .': <b>'.$this->getMessage().</b> is not a valid E-Mail address';
        return $errorMsg;
    }
}
$email = "someone@example...com";
try
{
    //check if
    if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE)
    {
        //throw exception if email is not valid
        throw new customException($email);
    }
}
```

```

    }
}
catch (customException $e)
{
    //display custom message
    echo $e->errorMessage();
}
?>

```

class အသစ်က old exception class ရဲ့ ကော်ပီ တစ်ခုဖြစ်ပါတယ်။ သူ့မှာ အပိုအနေနဲ့ errorMessage() ဆိုတဲ့ function တစ်ခု ပါလာပါတယ်။ old class ရဲ့ copy ဖြစ်တဲ့အတွက် properties နဲ့ method တွေကို old class ကနေ မျိုးရိုးလိုက်ပါတယ်။ ကျနော်တို့အနေနဲ့ getLine(), getFile() နဲ့ getMessage() လိုမျိုး exception class methods တွေကို အသုံးပြုနိုင်ပါတယ်။

အထက်ကဥပမာကို ရှင်းရရင်

ဒီဥပမာမှာ exception တစ်ခုကို ဖြစ်ပွားစေပြီးတော့ custom exception class တစ်ခုနဲ့ ပြန်ဖမ်းထားပါတယ်။

1. customException() class တစ်ခုကို Old exception class ရဲ့ extension တစ်ခုအနေနဲ့ ဖန်တီးပါတယ်။ ဒါကြောင့် old exception class ကနေ methods နဲ့ properties တွေအားလုံးကို မျိုးရိုးဆက်ခံပါတယ်။
2. errorMessage() ကိုဖန်တီးပါတယ်။ email address က invalid ဖြစ်နေမယ်ဆိုရင် ဒီ function က error message တစ်ခုကို ထုတ်ပေးတာပါ။
3. \$email variable ကို valid မဖြစ်တဲ့ email လိပ်စာတစ်ခုအဖြစ် ထားပေးလိုက်တဲ့အခါ။
4. try block ကို execute လုပ်လိုက်တော့ email လိပ်စာက valid မဖြစ်တဲ့အတွက် exception တစ်ခု thrown သွားပါတယ်။
5. catch block က exception ကို ဖမ်းထားလိုက်ပြီးတော့ error message ကိုထုတ်ပြပေးပါတယ်။

Multiple Exceptions

အခြေအနေတွေ အများကြီးကို စစ်ဆေးဖို့အတွက် exception တွေအများကြီးကို အသုံးပြုဖို့ script တစ်ခုတည်းကို သုံးနိုင်ပါတယ်။ if else block တွေ switch တို့ nested multiple exception တွေအများကြီးကို အသုံးပြုလို့ရပါတယ်။ ဒီ exception တွေကနေ မတူညီတဲ့ exception class တွေကိုအသုံးပြုပြီးတော့ မတူညီတဲ့ error message တွေကို ထုတ်ပေးပါလိမ့်မယ်။

```

<?php
class customException extends Exception
{
    public function errorMessage()
    {
        //error message
        $errorMsg = 'Error on line '.$this->getLine().' in '.$this->getFile()
        .': <b>'.$this->getMessage().'</b> is not a valid E-Mail address';
        return $errorMsg;
    }
}

$email = "someone@example.com";

try
{

```

```
//check if
if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE)
{
    //throw exception if email is not valid
    throw new customException($email);
}
//check for "example" in mail address
if(strpos($email, "example") !== FALSE)
{
    throw new Exception("$email is an example e-mail");
}
}

catch (customException $e)
{
    echo $e->errorMessage();
}
catch(Exception $e)
{
    echo $e->getMessage();
}
?>
```

run လိုက်မယ် ဆိုရင် someone@example.com is an example e-mail ဆိုပြီးတော့ ရလာပါလိမ့်မယ်။

code တွေကိုပြန်ရှင်းရရင်တော့

1. customException() class တစ်ခုကို Old exception class ရဲ့ extension တစ်ခုအနေနဲ့ ဖန်တီးပါတယ်။ ဒါကြောင့် old exception class ကနေ methods နဲ့ properties တွေအားလုံးကို မျိုးရိုးဆက်ခံပါတယ်။
2. errorMessage() ကိုဖန်တီးပါတယ်။ email address က invalid ဖြစ်နေမယ်ဆိုရင် ဒီ function က error message တစ်ခုကို ထုတ်ပေးတာပါ။
3. \$email variable ကို valid ဖြစ်တဲ့ email လိပ်စာတစ်ခုအဖြစ် ထားပေးထားပါတယ်။ ဒါပေမယ့် example ဆိုတဲ့ string တစ်ခုပါနေပါတယ်။
4. try block ကို execute လုပ်ပါတယ်တယ် ယထာဝရ အခြေအနေမှာ exception တစ်ခုမဖြစ်ပွားပါဘူး။
5. ဒုတိယအခြေအနေကြမှာသာ example ဆိုတဲ့ စာသားကြောင့် exception တစ်ခုဖြစ်ပွားသွားပါတယ်။
6. catch ဘလောက်က exception ကိုဖမ်းလိုက်ပါတယ် ပြီးတော့ မှန်ကန်တဲ့ error message ကိုထုတ်ပြပေးပါတယ်။

Re-throwing Exceptions

တစ်ခါတလေ exception တစ်ခုကို thrown လုပ်ပြီးတဲ့နောက် ပုံမှန်နည်းလမ်းနဲ့မဟုတ်ဘဲ တခြားနည်းနဲ့ ကိုင်တွယ်ချင်ပါလိမ့်မယ်။ ဒုတိယအကြိမ် catch block တစ်ခုနဲ့ exception တစ်ခုကို throw လုပ်လိုရပါတယ်။

script တစ်ခုရဲ့ ဂုဏ်သတ္တိကောင်း တစ်ခုအနေနဲ့ user တွေကို system error တွေမမြင်ရအောင် လုပ်ဆောင်ပေးနိုင်ရပါမယ်။ System error တွေက coder တွေအတွက်အရေးကြီးပါတယ် ဒါပေမယ့် user တွေအတွက် ဘာမှ စိတ်ဝင်စားစရာ မရှိဘူးလေ။ user တွေအတွက် အရာရာ လွယ်ကူနေစေဖို့ user friendly message တစ်ခုနဲ့ exception တွေကို re-throw လုပ်နိုင်ပါတယ်။

```
<?php
class customException extends Exception
{
```



```

public function errorMessage()
{
    //error message
    $errorMsg = $this->getMessage().' is not a valid E-Mail address.';
    return $errorMsg;
}
}
$email = "someone@example.com";
try
{
    try
    {
        //check for "example" in mail address
        if(strpos($email, "example") !== FALSE)
        {
            //throw exception if email is not valid
            throw new Exception($email);
        }
    }
    catch(Exception $e)
    {
        //re-throw exception
        throw new customException($email);
    }
}
catch (customException $e)
{
    //display custom message
    echo $e->errorMessage();
}
?>

```

Output ကတော့

someone@example.com is not a valid E-Mail address. ဖြစ်ပါတယ်။

1. customException() class တစ်ခုကို Old exception class ရဲ့ extension တစ်ခုအနေနဲ့ ဖန်တီးပါတယ်။ ဒါကြောင့် old exception class ကနေ methods နဲ့ properties တွေအားလုံးကို မျိုးရိုးဆက်ခံပါတယ်။
2. errorMessage() ကိုဖန်တီးပါတယ်။ email address က invalid ဖြစ်နေမယ်ဆိုရင် ဒီ function က error message တစ်ခုကို ထုတ်ပေးတာပါ။
3. \$email variable ကို valid ဖြစ်တဲ့ email လိပ်စာတစ်ခုအဖြစ် ထားပေးထားပါတယ်။ ဒါပေမယ့် example ဆိုတဲ့ string တစ်ခုပါနေပါတယ်။
4. try block ထဲမှာ exception ကို re-throw လုပ်နိုင်စေဖို့အတွက် နောက်ထပ် try block တစ်ခုရှိနေပါတယ်။
5. email မှာ example ဆိုတဲ့ string ပါနေလို့ exception ဖြစ်ပွားသွားပါတယ်။
6. catch block က exception ကိုဖမ်းပြီးတော့ customException တစ်ခုကို re-throws လုပ်လိုက်ပါတယ်။
7. customException ကို ဖမ်းလိုက်ပြီးတော့ error message တစ်ခုထုတ်ပြပေးပါတယ်။

Set a Top Level Exception Handler

set_exception_handler() function ကနေ ဖမ်းလိုက်ရတဲ့ exception တွေအားလုံးကို ကိုင်တွယ်ဖို့အတွက် user-defined function တစ်ခုကို set လုပ်ပေးပါတယ်။

```
<?php
function myException($exception)
{
    echo "<b>Exception:</b> " , $exception->getMessage();
}
set_exception_handler('myException');
throw new Exception('Uncaught Exception occurred');
?>
```

output ကတော့

Exception: Uncaught Exception occurred

ဒီ code မှာ catch block မပါဝင်ပါဘူး အဲဒီအစား top level exception handler ကိုသုံးပါတယ်။ ဒီ function ကို uncaught exception တွေအတွက် အသုံးပြုသင့်ပါတယ်။

Exception အတွက် ဥပဒေများ

1. Code တွေကို try block ထဲမှာ ထားသင့်ပါတယ်။
2. try block ဒါမှမဟုတ် throw တိုင်းမှာ အနည်းဆုံးတော့ သတ်ဆိုင်တဲ့ catch block တစ်ခုရှိရပါမယ်။
3. မတူညီတဲ့ exception class တွေအတွက် catch block တွေအများကြီး အသုံးပြုနိုင်ပါတယ်။
4. try block တစ်ခုအတွင်းမှာ ရှိတဲ့ catch block တစ်ခုအတွင်းမှာ exception တွေကို thrown (or rethrown) လုပ်နိုင်ပါတယ်။
5. နောက်ထပ် ရှင်းလင်းတဲ့ ဥပမာကတော့ တစ်ခုခုကို ပစ်ထုတ်(throw) လိုက်ပြီဆိုရင် အဲဒါကို ပြန်ဖမ်း (catch) ရပါမယ်။

PHP Filter

PHP Filter ဆိုတာဘာလဲ ?

PHP filter ကို စိတ်မချရတဲ့ နေရာတွေကနေလာမဲ့ ဥပမာ user တွေက သွင်းမယ့် input လိုဟာမျိုးက လာမယ့် အော့ တွေကို valid ဖြစ်မဖြစ် နဲ့ filter လုပ်ပေးဖို့အတွက် အသုံးပြုရပါတယ်။

user imputer တွေကို စစ်ဆေးမယ် validate လုပ်မယ် filter လုပ်မယ် ဒါတွေက ဘယ် web application မှာမဆို အရေးကြီးတဲ့ အစိတ်အပိုင်းပဲဖြစ်ပါတယ်။

PHP filter ကို လွယ်လွယ်ကူကူ မြန်မြန်ဆန်ဆန် filter(စစ်ချ) လုပ်နိုင်ဖို့အတွက် ဖန်တီးထားပြီးသားပါ။

ဘာလို့ Filter ကိုသုံးတာလဲ?

web application တွေ အားလုံးနီးပါးဟာ ပြင်ပကလာတဲ့ Input တွေပေါ်မှာ မှီခိုနေကြပါတယ်။ အများအားဖြင့် အဲဒါတွေက user တစ်ယောက်ယောက်ဆီက ဒါမှမဟုတ် တခြား web service လိုမျိုး application တစ်ခုဆီကနေပြီးတော့ပါ။ filter ကို သုံးလိုက်မယ်ဆိုရင်.. ကျနော်တို့ရဲ့ application က မှန်ကန်တဲ့ input အမျိုးအစားတွေကို ရရှိမယ်လို့ သေချာစေမှာ ဖြစ်ပါတယ်။

ကျနော်တို့ အပြင်ကလာတဲ့ data မှန်သမျှကို အမြဲတမ်း filter လုပ်သင့်ပါတယ်။ input filtering က application တွေရဲ့ security နဲ့ပတ်သက်ပြီးတော့ အရေးအကြီးဆုံးအချက်တစ်ချက်ပဲဖြစ်ပါတယ်။

external data ဆိုတာဘာတွေလည်းဆိုတော့

- form တစ်ခုကနေ ထည့်ပေးကြတဲ့ Input data တွေ
- Cookies တွေ
- Web services တွေကနေလာတဲ့ data တွေ
- server variable တွေ နဲ့
- Database query result တွေဖြစ်ပါတယ်။

Functions and Filters

variable တစ်ခုကို filter လုပ်ဖို့အတွက် အောက်ပါ filter function တွေထဲက တစ်ခုကို အသုံးပြုပါတယ်။

- filter_var()- တိကျသေချာတဲ့ filter တစ်ခုကိုသုံးပြီးတော့ variable တစ်လုံးကို filter လုပ်တာ
- filter_var_array()- တူညီတဲ့ ဒါမှမဟုတ် မတူညီတဲ့ filter တွေကိုသုံးပြီးတော့ variable တွေအများကြီးကို filter လုပ်တာ။
- filter_input- variable input တစ်ခုတည်းကို ရယူပြီးတော့ သူ့ကိုပဲ filter လုပ်တာနဲ့
- filter_input_array- input variable တွေအများကြီးကို ရယူထားပြီးတော့ တူညီတဲ့ ဒါမှမဟုတ် မတူညီတဲ့ filter တွေနဲ့ filter လုပ်တာပဲ ဖြစ်ပါတယ်။

အောက်ပါ ဥပမာမှာ filter_var() function ကိုအသုံးပြုပြီးတော့ integer တစ်လုံးကို valid ဖြစ်မဖြစ် စစ်ပါတယ်။

```
<?php
$int = 123;
if(!filter_var($int, FILTER_VALIDATE_INT))
{
    echo("Integer is not valid");
}
else
{
    echo("Integer is valid");
}
?>
```

ဒီအထက်ပါ code က variable ကို filter လုပ်ဖို့အတွက် "FILTER_VALIDATE_INT" filter ကို အသုံးပြုထားပါတယ်။ Integer က valid ဖြစ်တဲ့အတွက် output ကတော့ " Integer is valid". လို့ရပါလိမ့်မယ်။

integer မဟုတ်တဲ့ variable ဥပမာ 123abc လို့စမ်းကြည့်မယ်ဆိုရင် "Integer is not valid" ဆိုပြီးတော့ ထုတ်ပေးမှာ ဖြစ်ပါတယ်။

Validating and Sanitizing

filter နှစ်မျိုးရှိပါတယ်။

Validating filters နဲ့ Sanitizing filters တို့ပါ။

Validating filters တွေကို

- user input တွေကို validate လုပ်ဖို့
- format rule တွေကိုတိတိကျကျ သတ်မှတ်ပေးဖို့ (ဥပမာ URL နဲ့ EMAIL)
- အောင်မြင်တဲ့အခါ မျှော်လင့်ထားတဲ့ အမျိုးအစားကိုထုတ်ပေးပြီးတော့ validate မဖြစ်တဲ့အခါ FALSE ကို ပြန်ထုတ်ပေးဖို့ အသုံးပြုပါတယ်။

Sanitizing filters တွေကိုတော့

- string တစ်ခုထဲမှာရှိတဲ့ character တစ်ခုခုတွေကို ခွင့်ပြုမလား၊ ခွင့်မပြုဘူးလားဆိုတာအတွက်သုံးပါတယ်။
- data format rules မရှိပါဘူး
- အမြဲတမ်း String ကိုသာ return ပြန်ပေးပါတယ်။

Options and Flags

options နဲ့ flags တို့ကို ရွေးချယ်ထားတဲ့ filters တွေအတွက် အပို filter options တွေထပ်ပေါင်းထည့်ဖို့အသုံးပြုပါတယ်။ မတူညီတဲ့ filterတွေမှာ မတူညီတဲ့ Options နဲ့ flags တွေရှိကြပါတယ်။

အောက်ပါဥပမာမှာ filter_var() function ရယ် min_range နဲ့ max_range option တွေကိုသုံးပြီး integer တစ်ခုကို ကျေနပ်တဲ့ validate လုပ်ပါတယ်။

```
<?php
$var=300;
$int_options = array(
    "options"=>array
    (
        "min_range"=>0,
        "max_range"=>256
    )
);
if(!filter_var($var, FILTER_VALIDATE_INT, $int_options))
{
    echo("Integer is not valid");
}
else
{
    echo("Integer is valid");
}
?>
```

ဒီ code မှာ options တွေကို option လို့အမည်ပေးပြီးတော့ associative array တစ်ခုထဲမှာ ထည့်ထားရပါတယ်။ flag ကိုအသုံးပြုတဲ့အခါမှာတော့ သူက array ထဲမှာရှိနေဖို့မလိုအပ်ပါဘူး။

integer က 300 ဖြစ်တဲ့အတွက် သူက သတ်မှတ်ထားတဲ့ အတိုင်းအတာ range ထဲမှာ မရှိနေပါဘူး။ ဒါကြောင့် ဒီ code ကို run ရင် "Integer is not valid" လို့ပြန်ပါမယ်။

Validate Input

form တစ်ခုက input form တစ်ခုကို validate လုပ်ဖို့ ကြိုးစားကြရအောင်။

ထမဆုံးလုပ်ဖို့လိုအပ်တာက ကျနော်တို့ ရှာဖွေနေတဲ့ input data ရှိဖို့ သေချာစေဖို့ဖြစ်ပါတယ်။ ပြီးတဲ့အခါ filter_input () function ကိုသုံးပြီးတော့ input data ကို filter လုပ်ပါမယ်။
 အောက်ပါဥပမာမှာ input variable "email" ကို PHP page ကိုပို့ပေးပါတယ်။

```
<?php
if(!filter_has_var(INPUT_GET, "email"))
{
    echo("Input type does not exist");
}
else
{
    if (!filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL))
    {
        echo "E-Mail is not valid";
    }
    else
    {
        echo "E-Mail is valid";
    }
}
?>
```

ဒီဥပမာမှာ input ဖြစ်တဲ့ (URL) ကို "POST" Method သုံးပြီးတော့ ပို့ပေးထားပါတယ်။

1. POST type ရဲ့ url input က ရှိမရှိ စစ်ပါတယ်။
2. ရှိတယ်ဆိုရင် အဲဒါကို sanitize(valid မဖြစ်တဲ့ character) တွေကိုဖယ်ထုတ်ပြီးတော့ \$url variable ထဲမှာ သိမ်းပေးပါတယ်။

တကယ်လို့ Input string က <http://www.moemyintsøhein.com/> ဖြစ်နေမယ်ဆိုရင် sanitizing လုပ်ပြီးတဲ့အခါမှာတော့ <http://www.moemyintsøhein.com> လို့ဖြစ်သွားမှာပါ။

Filter Multiple Inputs

Form တစ်ခုအနေနဲ့ အမြဲတမ်း input field တစ်ခုထက်မက ရှိပါတယ်။ filter_var ဒါမှမဟုတ် filter_input function တွေကို တစ်ပြိုင်တစ်ခါ ခေါ်ရတာမျိုးမဖြစ်စေဖို့ ကျနော်တို့က filter_var_array ဒါမှမဟုတ် filter_input_array စတဲ့ function တွေကို အသုံးပြုကြပါတယ်။

ဒီဥပမာမှာ ကျနော်တို့က GET variable သုံးခုကို filter လုပ်ဖို့အတွက် filter_input_array() function ကိုအသုံးပြုပါတယ်။ လက်ခံရရှိတဲ့ GET variable က နာမည်တစ်ခု အသက်တစ်ခု နဲ့ အီးမေးလ်တစ်ခုတို့ဖြစ်ပါတယ်။

```
<?php
$filters = array
(
    "name" => array
    (
```

```

    "filter"=>FILTER_SANITIZE_STRING
  ),
  "age" => array
  (
    "filter"=>FILTER_VALIDATE_INT,
    "options"=>array
    (
      "min_range"=>1,
      "max_range"=>120
    )
  ),
  "email"=> FILTER_VALIDATE_EMAIL,
);
$result = filter_input_array(INPUT_GET, $filters);
if (!$result["age"])
{
  echo("Age must be a number between 1 and 120.<br />");
}
elseif(!$result["email"])
{
  echo("E-Mail is not valid.<br />");
}
else
{
  echo("User input is valid");
}
?>

```

ဒီဥပမာမှာ (name,age,email) ဆိုတဲ့ Input သုံးခုရှိပါတယ် ပြီးတော့ GET method နဲ့ပို့ထားပါတယ်။

1. input variable တွေရဲ့အမည်ပါဝင်တဲ့ array တစ်ခုကို သတ်မှတ်ပါတယ် ပြီးတော့ input variables တွေအတွက် filter တွေကိုလည်း သတ်မှတ်ပါတယ်။
2. filter_input_array() function ကို GET input variable တွေနဲ့ ကျနော်တို့ ဖန်တီးထားတဲ့ array တို့နဲ့ ခေါ်လိုက်ပါတယ်။
3. valid input တွေရရှိအတွက် \$result variable အတွင်းမှာ age, email variable တွေကို စစ်ပါတယ်။ ဒီ input တွေထဲက တစ်ခုခု invalid ဖြစ်တယ်ဆိုရင် filter_input_array() function ပြီးတဲ့အခါမှာ input variable က FALSE ဖြစ်မှာဖြစ်ပါတယ်။

filter_input_array() function ရဲ့ ဒုတိယ parameter က array လည်းဖြစ်နိုင်သလို single filter ID တစ်ခုလည်း ဖြစ်နိုင်ပါတယ်။

တကယ်လို့ parameter က single filter ID ဆိုရင် input array ရဲ့တန်ဖိုးတွေအားလုံးက သတ်မှတ်ထားတဲ့ filter အားဖြင့် filter လုပ်သွားမှာဖြစ်ပါတယ်။

parameter က array ဖြစ်တယ်ဆိုရင် အောက်ပါ ဥပဒေတွေကိုလိုက်နာရပါမယ်။

- (age input variable လိုမျိုး) array key တစ်ခုအနေနဲ့ input variable တစ်ခုပါဝင်တဲ့ associative array တစ်ခု ဖြစ်ရပါမယ်။
- array တန်ဖိုးက filter ID ဒါမှမဟုတ် filter, flag နဲ့ option တွေကို သတ်မှတ်ပေးထားတဲ့ array တစ်ခုဖြစ်ရပါမယ်။

Using Filter Callback

user defined function တစ်ခုကို ခေါ်သုံးပြီးတော့ FILTER_CALLBACK filter ကိုသုံးပြီး filter လုပ်ဖို့ ဖြစ်နိုင်ပါတယ်။ ဒီနည်းလမ်း နဲ့ ကျနော်တို့ data filtering ကို အပြည့်အဝထိန်းချုပ်နိုင်ပါတယ်။ ကျနော်တို့အနေနဲ့ ကိုယ်ပိုင် User defined function ကိုဖန်တီးနိုင်သလို ရှိပြီးသား PHP function တစ်ခုကို သုံးနိုင်ပါတယ်။ အောက်ပါဥပမာမှာ ကျနော်တို့က "_" တွေကို whitespaces တွေအဖြစ်ပြောင်းပေးတဲ့ user created function တစ်ခုကို အသုံးပြုပါတယ်။

```
<?php
function convertSpace($string)
{
    return str_replace("_", " ", $string);
}

$string = "Peter_is_a_great_guy!";

echo filter_var($string, FILTER_CALLBACK,
    array("options"=>"convertSpace"));
?>
```

output ကတော့

Peter is a great guy!

ဖြစ်ပါတယ်။

အလုပ်လုပ်ပုံကတော့

1. "_" တွေကို whitespaces တွေအဖြစ်ပြောင်းပေးမယ့် function တစ်ခုကို ဖန်တီးပါတယ်။
2. FILTER_CALLBACK filter ရယ် ကျနော်တို့ function ပါတဲ့ array တစ်ခုရယ်နဲ့အတူ filter_var() function ကိုခေါ်သုံးပါတယ်။

Conclusion

ကျနော်ရဲ့စာအုပ်ကို ဖတ်ပြီး လေ့လာကြသူအားလုံးကိုကျေးဇူးတင်ပါတယ်ခင်ဗျာ။ ကျနော် အတတ်နိုင်ဆုံး ကြိုးစားထားပါတယ်။ လိုအပ်တာများရှိရင် ဝေဖန်အကြံပြုနိုင်ကြပါတယ်။ ဒီနေရာမှာ ရပ်မထားဘဲ ဆက်လက်လေ့လာသွားနိုင်ဖို့ အတွက် English လိုရေးထားတဲ့ PHP စာအုပ်တွေကို ကျနော့်ဘလော့မှာ တင်ပေးထားပါမယ်။ ဒေါင်းလုပ်ဆွဲပြီး ဆက်လက်လေ့လာသွားကြပါ။ အားလုံး ကိုယ်ဖြစ်ချင်တာတွေကို အောင်မြင်အောင် လုပ်ဆောင်နိုင်ကြပါစေ။

Moe Myint Shein

Email: moemyintsheinster@gmail.com

Web: www.moemyintshein.com