PHP & Web programing

Training Text

GWI(Guyana Water Inc.)

Kenichi Tezuka

JICA (Japan International Cooperation Agency)

Senior Volunteer (ICT)

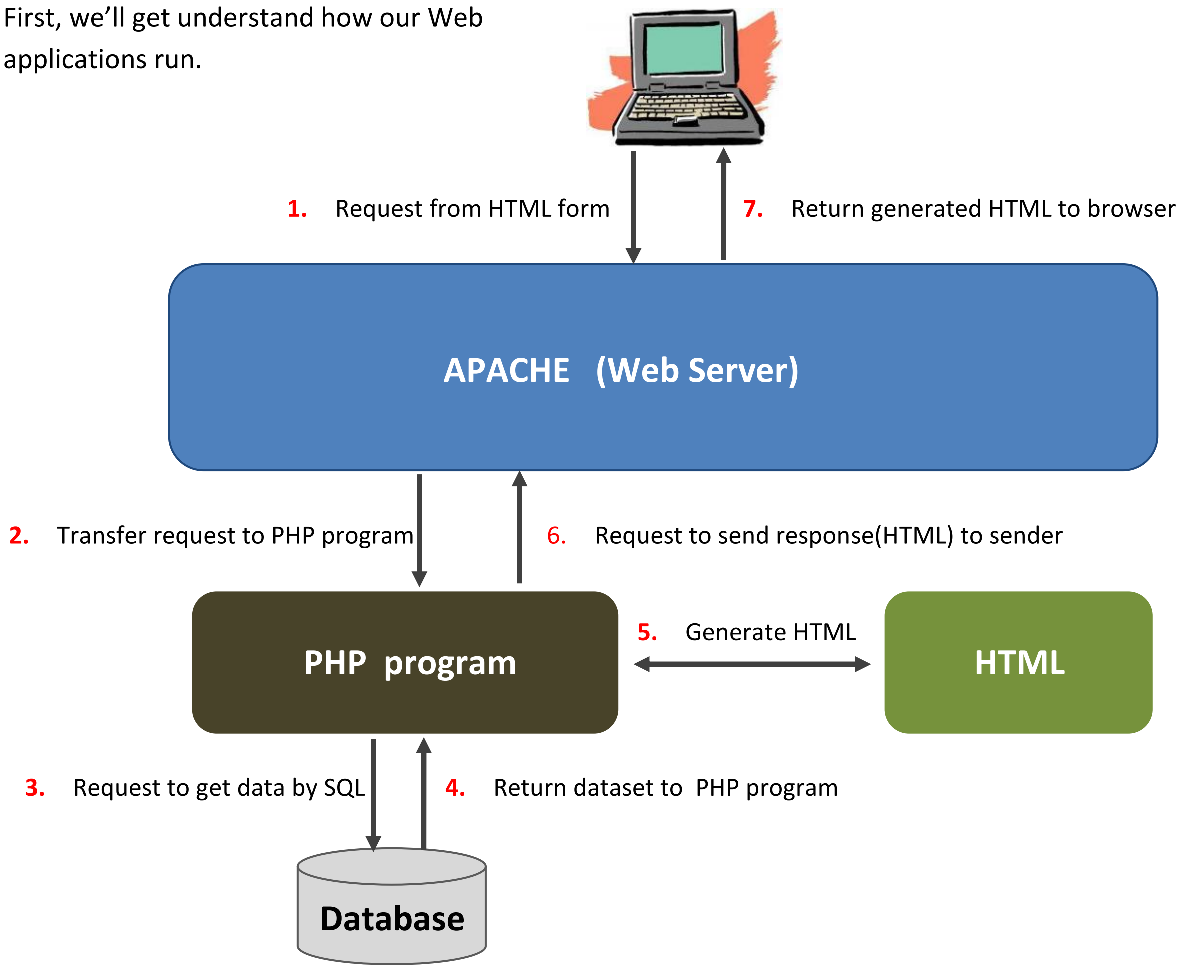
2015.9

Ver. 1.0

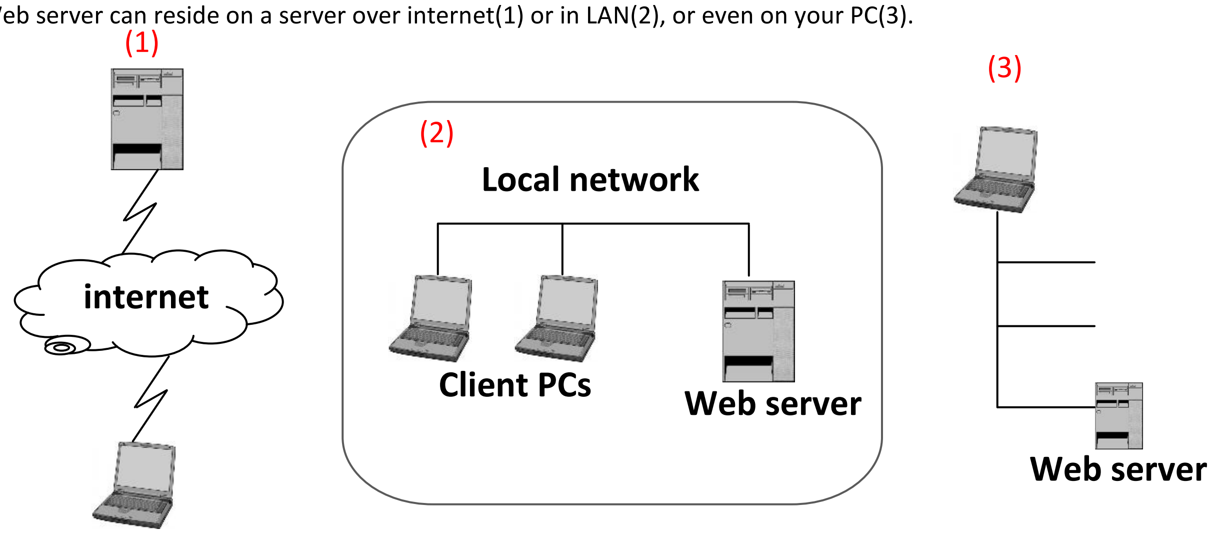
INDEX

This training aims for getting knowledge on how to make Web applications by using PHP as programming language.

First, we'll get understand how our Web applications run.


1. Request from HTML form
7. Return generated HTML to browser
APACHE (Web Server)
2. Transfer request to PHP program
6. Request to send response(HTML) to sender
PHP program
5. Generate HTML
HTML
3. Request to get data by SQL
4. Return dataset to PHP program
Database

Web server can reside on a server over internet(1) or in LAN(2), or even on your PC(3).



XAMPP

What's XAMPP?

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.
(by Wikipedia)

XAMPP provides

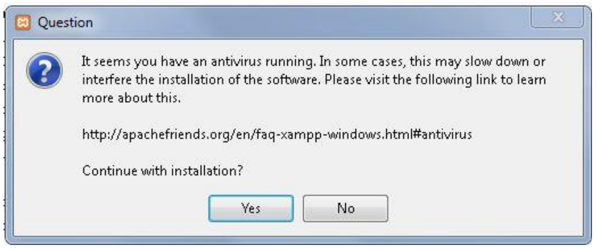
Web server ---------------------------------------	APACHE
Programing environment ------------------------	PHP, Perl
Database --	MySQL
File transfer protocol (FTP) server -----------	FileZilla
Mail server --	Mercury Mail
Web server for JAVA environment ----------	Tomcat

and so on.

Step 1. XAMPP installation

We have now XAMPP installer named 'xampp-win32-5.6.11-0-VC11-installer.exe'.

Executing this program, then you'll see a screen below;

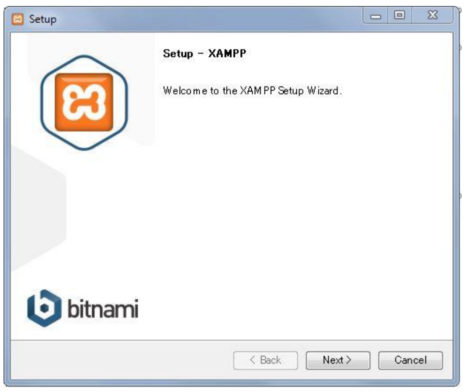


It seems you have an antivirus running. In some cases, this may slow down or interfere the installation of the software. Please visit the following link to learn more about this.

http://apachefriends.org/en/faq-xampp-windows.html#antivirus

Continue with installation?

For successful installation, you'd better stop your anti-virus software temporally.
After responding 'Yes' to this, you'll see a screen below;

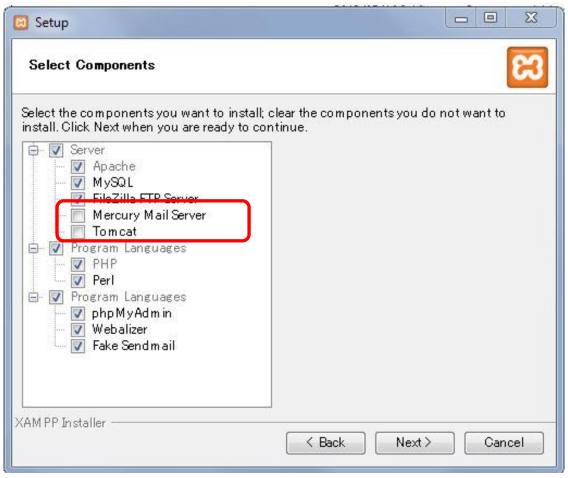


Setup – XAMPP

Welcome to the XAMPP Setup Wizard.

Simply press 'Next'.

Next, you'll see 'Component selection' window. Here, you'd better check off 'Mercury' and 'Tomcat', which are not necessary now.

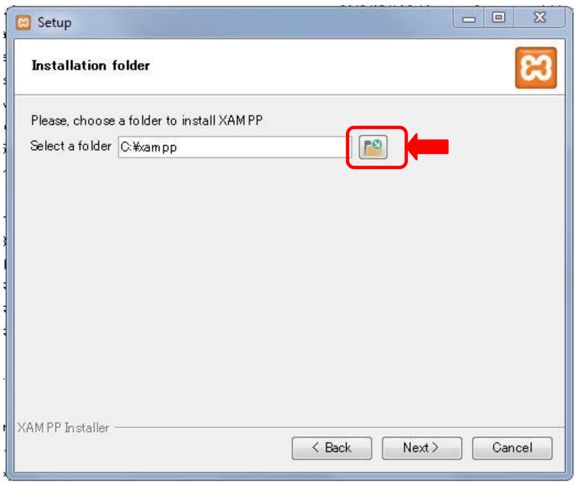


Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [✓] Server
  - [✓] Apache
  - [✓] MySQL
  - [✓] FileZilla FTP Server
  - [ ] Mercury Mail Server
  - [ ] Tomcat
- [✓] Program Languages
  - [✓] PHP
  - [✓] Perl
- [✓] Program Languages
  - [✓] phpMyAdmin
  - [✓] Webalizer
  - [✓] Fake Sendmail

Then, next step is to define where(= under which folder) we install XAMPP.
XAMPP recommend not to install XAMPP on top level folder.

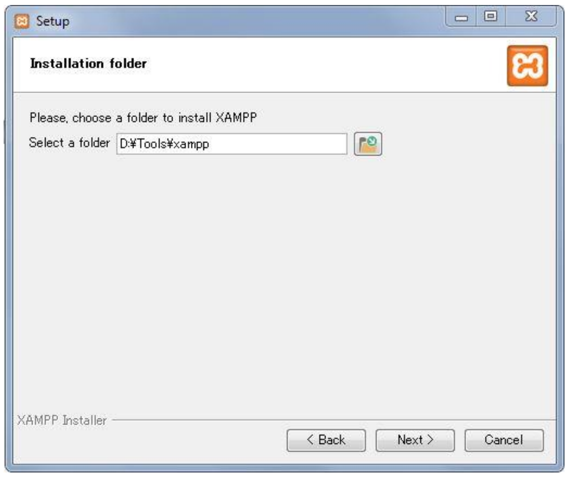


Installation folder

Please, choose a folder to install XAMPP

Select a folder C:¥xampp

You'd better change the default('C:\xampp') to some new folder under other higher level folder.

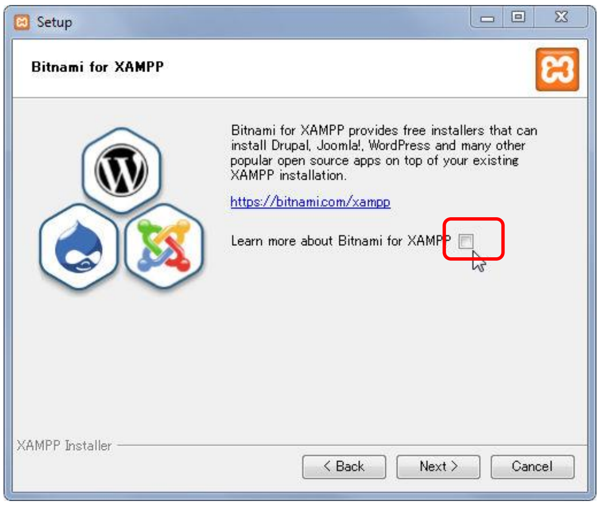


Installation folder

Please, choose a folder to install XAMPP

Select a folder D:¥Tools¥xampp

Then you'll see a screen below. You'd better clear the check, otherwise you'll see some bothersome advertisement later.



Setup

Bitnami for XAMPP

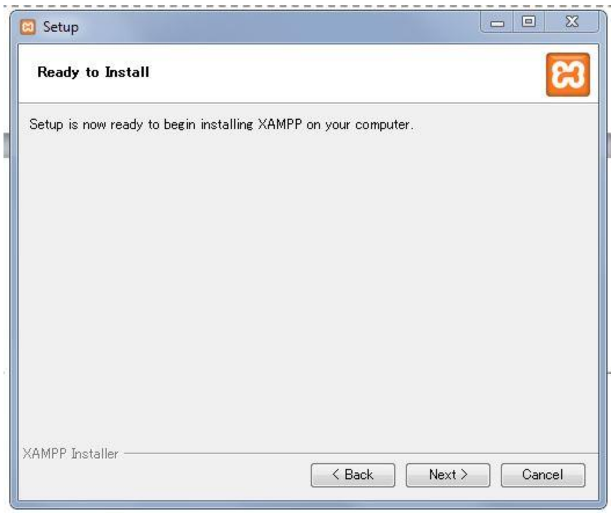
Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.

https://bitnami.com/xampp

Learn more about Bitnami for XAMPP ☐

XAMPP Installer

Finally, we complete preparation for installation. Press 'Next', then installation process will start.

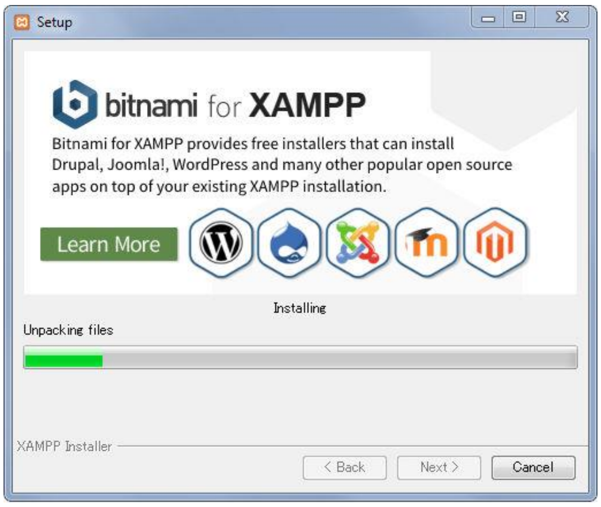


Setup

Ready to Install

Setup is now ready to begin installing XAMPP on your computer.

XAMPP Installer



Setup

bitnami for XAMPP

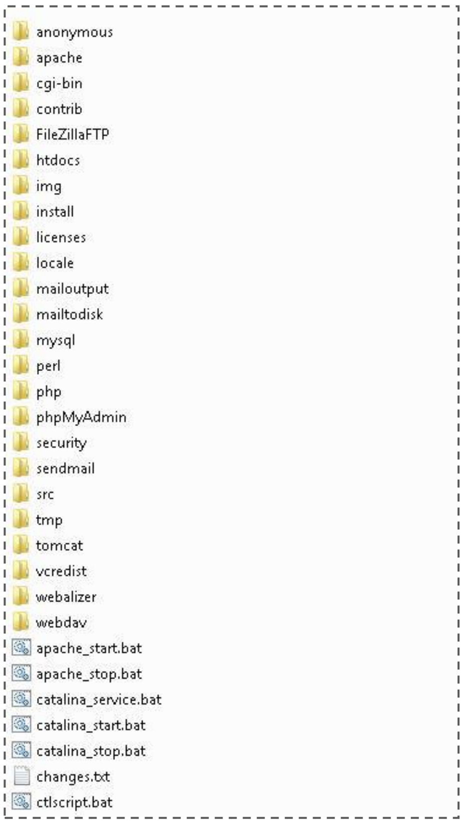
Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.

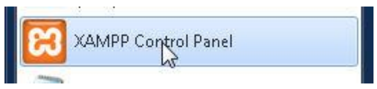
Installing

Unpacking files

XAMPP Installer

After installation, check folder where XAMPP resides. And you can see XAMPP controller button on 'Start Menu' for Windows 7. For Windows 8, XAMPP controller button may be on tiles.


anonymous
apache
cgi-bin
contrib
FileZillaFTP
htdocs
img
install
licenses
locale
mailoutput
mailtodisk
mysql
perl
php
phpMyAdmin
security
sendmail
src
tmp
tomcat
vcredist
webalizer
webdav
apache_start.bat
apache_stop.bat
catalina_service.bat
catalina_start.bat
catalina_stop.bat
changes.txt
ctlscript.bat



XAMPP controller button on Start Menu

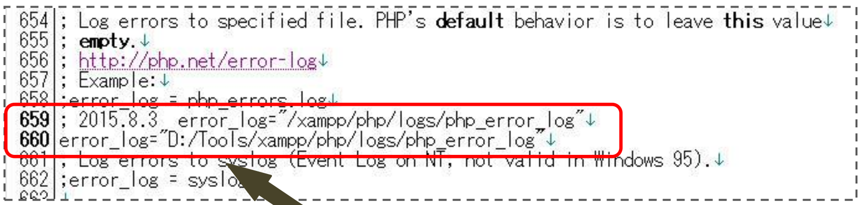
Step 2. XAMPP configuration

php.ini (PHP configuration   xampp/php/php.ini)

php.ini is a text file to configure PHP interpreter.
php.ini resides under ‘php’ folder in xampp.

Which part of php.ini should we modify before starting to use PHP?
(Example shown below is those of my environment. Drive and folder name should be arranged to your environment.)

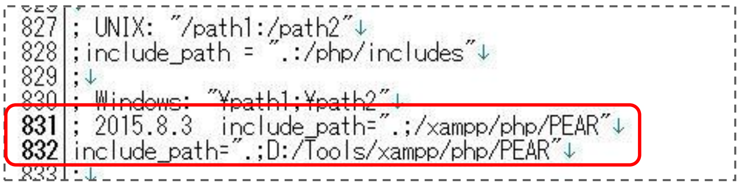
1. Path for error log



```
; Log errors to specified file. PHP's default behavior is to leave this value
; empty.
; http://php.net/error-log
; Example:
;error_log = php_errors.log
; 2015.8.3  error_log="/xampp/php/logs/php_error_log"
error_log="D:/Tools/xampp/php/logs/php_error_log"
; Log errors to syslog (Event Log on NT, not valid in Windows 95).
;error_log = syslog
```

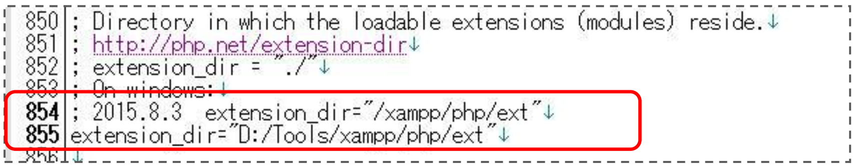
You’d better copy the target line to be modified and make one as comment line with date modified. (Don’t delete original line)
Then you could easily find where you had modified the original.

2. Path for Add-on tools’ folder



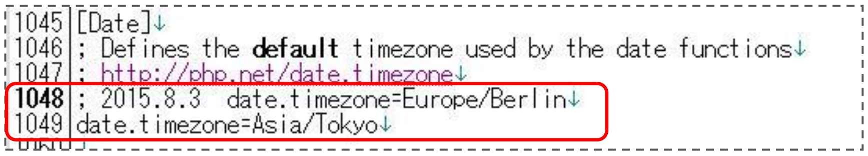
```
; UNIX: "/path1:/path2"
;include_path = ".:/php/includes"
;
; Windows: "¥path1;¥path2"
; 2015.8.3  include_path=".;/xampp/php/PEAR"
include_path=".;D:/Tools/xampp/php/PEAR"
;
```

3. Path for Extension (= external library) folder



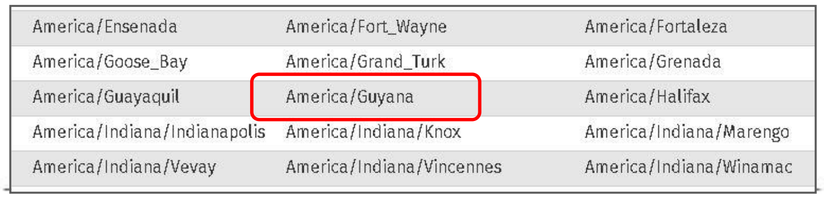
```
; Directory in which the loadable extensions (modules) reside.
; http://php.net/extension-dir
; extension_dir = "./"
; On windows:
; 2015.8.3  extension_dir="/xampp/php/ext"
extension_dir="D:/Tools/xampp/php/ext"
;
```

4. Set Time-zone



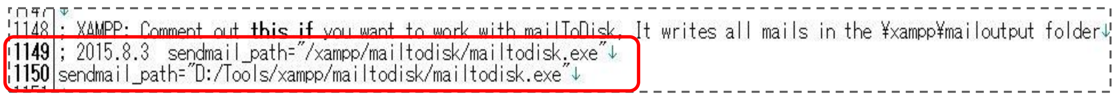
```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
; 2015.8.3  date.timezone=Europe/Berlin
date.timezone=Asia/Tokyo
```

You can get Time zone data in http://php.net/manual/en/timezones.america.php



America/Ensenada	America/Fort_Wayne	America/Fortaleza
America/Goose_Bay	America/Grand_Turk	America/Grenada
America/Guayaquil	America/Guyana	America/Halifax
America/Indiana/Indianapolis	America/Indiana/Knox	America/Indiana/Marengo
America/Indiana/Vevay	America/Indiana/Vincennes	America/Indiana/Winamac

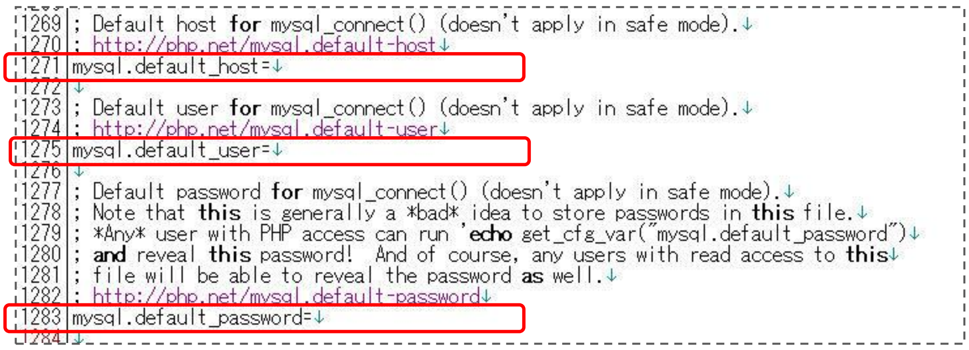
5. For debug or test mode, mails from PHP programs should be forwarded to (= saved in) local disk.



```
; XAMPP: Comment out this if you want to work with mailToDisk, It writes all mails in the ¥xampp¥mailoutput folder
; 2015.8.3  sendmail_path="/xampp/mailtodisk/mailtodisk.exe"
sendmail_path="D:/Tools/xampp/mailtodisk/mailtodisk.exe"
```

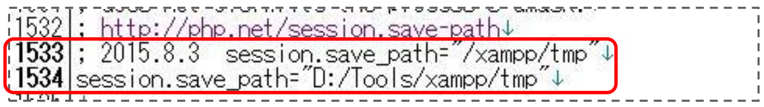
6. Environment information for MySQL database

   You can keep them as they are. We can give them later in PHP application programs.



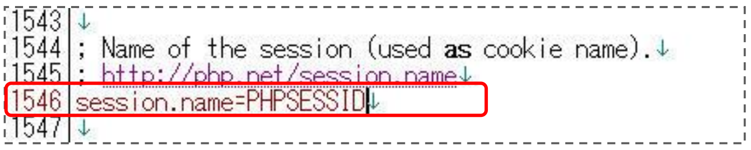
```
; Default host for mysql_connect() (doesn't apply in safe mode).
; http://php.net/mysql.default-host
mysql.default_host=

; Default user for mysql_connect() (doesn't apply in safe mode).
; http://php.net/mysql.default-user
mysql.default_user=

; Default password for mysql_connect() (doesn't apply in safe mode).
; Note that this is generally a *bad* idea to store passwords in this file.
; *Any* user with PHP access can run 'echo get_cfg_var("mysql.default_password")
; and reveal this password!  And of course, any users with read access to this
; file will be able to reveal the password as well.
; http://php.net/mysql.default-password
mysql.default_password=
```

7. Path for file to save Session data



```
; http://php.net/session.save-path
; 2015.8.3  session.save_path="/xampp/tmp"
session.save_path="D:/Tools/xampp/tmp"
```

8. Change Session ID if you want.



```

; Name of the session (used as cookie name).
; http://php.net/session.name
session.name=PHPSESSID

```

You can change Session ID, but you need to remember it as you often use it in programs.

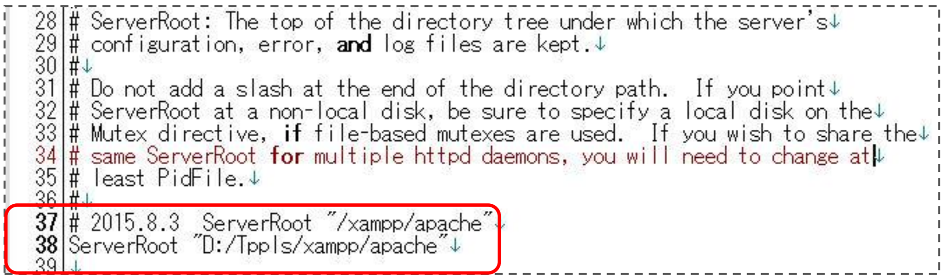
httpd.conf (Apache configuration   xampp/apache/conf/httpd.conf)

httpd.conf is a configuration file for Apache web server.

Httpd.conf resides apache/conf folder under XAMPP folder.

Which part of httpd.conf should we modify before starting to use Apache web server?
(Example shown below is those of my environment. Drive and folder name should be arranged to your environment.)

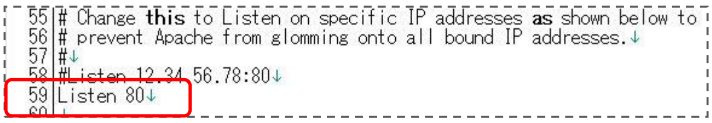
1. ServerRoot Directive
   It shows where apache system resides.



```
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
# 2015.8.3  ServerRoot "/xampp/apache"
ServerRoot "D:/Tppls/xampp/apache"
```

In XAMPP, it indicates apache top folder under XAMPP folder.

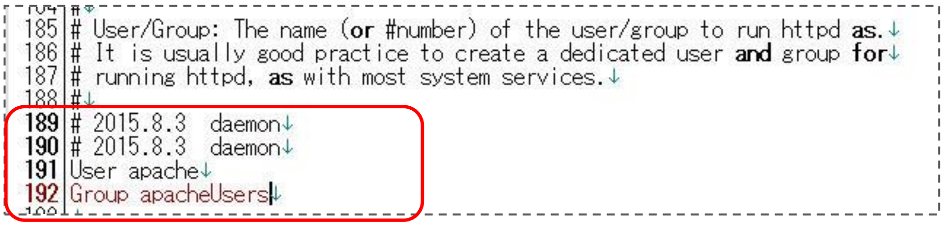
2. Port to listen



```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

   If your PC uses Port:80 for other use, you need to change port for apache web server.
   Skype program will use port 80, IIS too.   If you run Skype or IIS on your PC, then you need to change port number for apache.

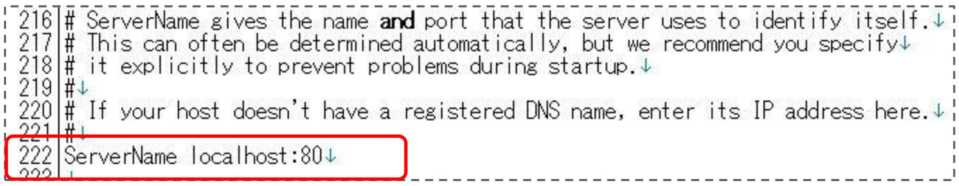
3. Use /Group for apache(httpd).
   If you don't define 'daemon' as User/Group, you need to define User 'daemon' and Group 'daemon'.
   In my case, I defined User 'apache' and Group 'apacheUsers'.



```
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
# 2015.8.3  daemon
# 2015.8.3  daemon
User apache
Group apacheUsers
```

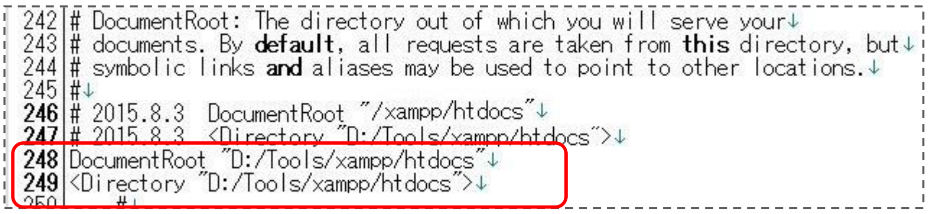
   Notes : This user and group must have read/write privilege on 'apache' folder and all folders under 'apache'.

4. ServerName : We don't need to change this ServerName, but if you change Port to other than 80, you need to change port number here.



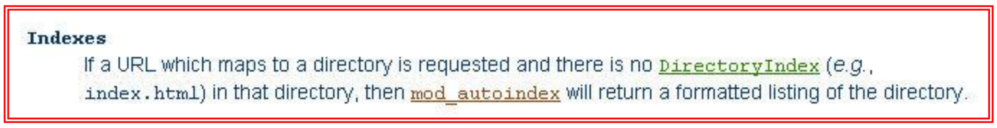
```
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80
```

5. 'DocumentRoot' is a folder that is stored on your host's servers and holds most of web files. Suppose you access to google web site by the URL:http://www.google.com, then you will access to 'DocumentRoot' folder in Google site.

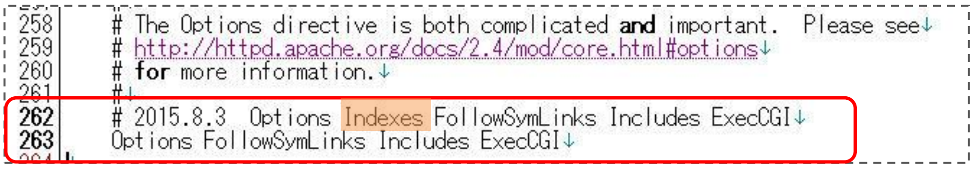


```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
# 2015.8.3  DocumentRoot "/xampp/htdocs"
# 2015.8.3  <Directory "D:/Tools/xampp/htdocs">
DocumentRoot "D:/Tools/xampp/htdocs"
<Directory "D:/Tools/xampp/htdocs">
```

6. From 'Options' directives, 'Indexes' option must be deleted. With 'Indexes' option, if someone access to your site with some file path and no file matches to it, then apache will return file name list of the folder to the visitor. This means that you may give all information of the folder, what are the files that resides in the folder and so on, to hackers.



Indexes
If a URL which maps to a directory is requested and there is no `DirectoryIndex` (*e.g.*, `index.html`) in that directory, then `mod_autoindex` will return a formatted listing of the directory.



```
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    # 2015.8.3  Options Indexes FollowSymLinks Includes ExecCGI
    Options FollowSymLinks Includes ExecCGI
```

Other .conf files to be checked if apache can't start normally.

in apache/conf/extra folder

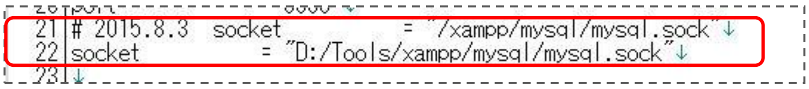
httpd-autoindex.conf
httpd-dav.conf
httpd-manual.conf
httpd-multilang-errordoc.conf
httpd-xampp.conf

These configuration files have path to some folder in their definition. If XAMPP installation couldn't complete normally, some of them couldn't have been updated normally.

my.ini (MySQL configuration xampp/mysql/bin/my.ini)

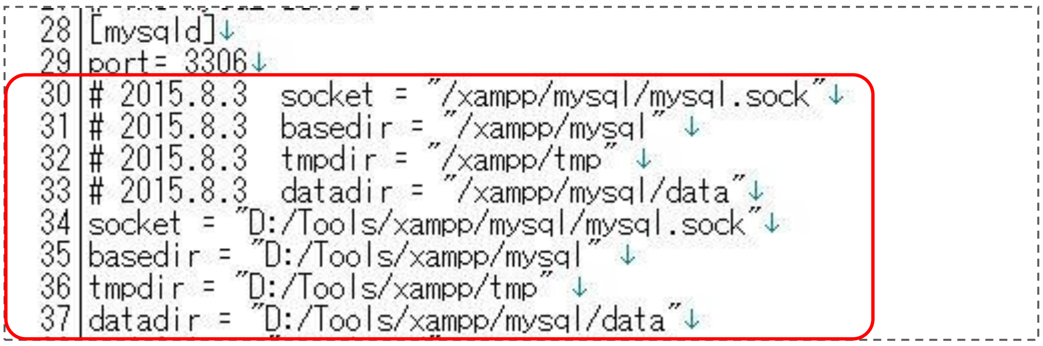
These configuration shown below are not necessary if XAMPP installation has been completed normally.
If not or if you want modify port number for MySQL or modify MySQL data folder, then follow these steps shown below.

1. Socket file



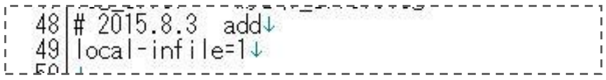
```
# 2015.8.3  socket          = "/xampp/mysql/mysql.sock"
socket          = "D:/Tools/xampp/mysql/mysql.sock"
```

2. Folders' definition



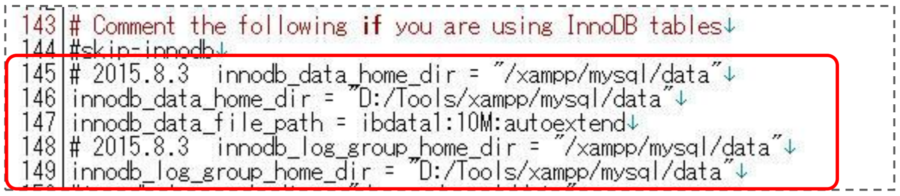
```
[mysqld]
port= 3306
# 2015.8.3  socket = "/xampp/mysql/mysql.sock"
# 2015.8.3  basedir = "/xampp/mysql"
# 2015.8.3  tmpdir = "/xampp/tmp"
# 2015.8.3  datadir = "/xampp/mysql/data"
socket = "D:/Tools/xampp/mysql/mysql.sock"
basedir = "D:/Tools/xampp/mysql"
tmpdir = "D:/Tools/xampp/tmp"
datadir = "D:/Tools/xampp/mysql/data"
```

3. If you load data from external CSV file into MySQL table, add lines shown below in my.ini.



```
# 2015.8.3  add
local-infile=1
```

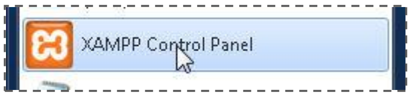
4. InnoDB settings



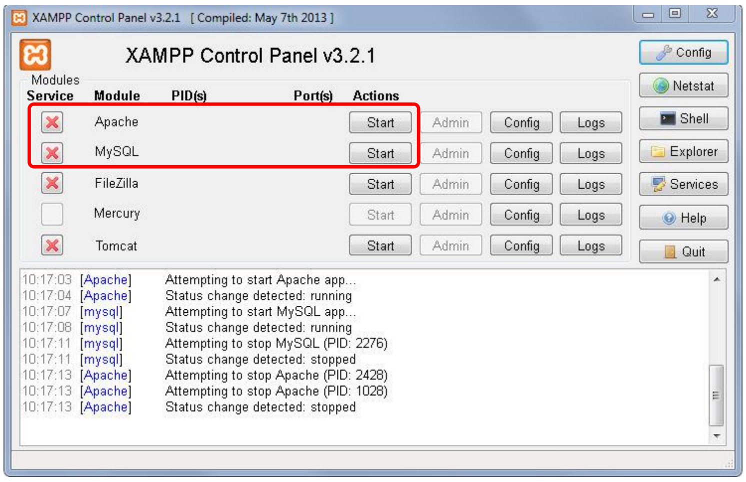
```
# Comment the following if you are using InnoDB tables
#skip-innodb
# 2015.8.3  innodb_data_home_dir = "/xampp/mysql/data"
innodb_data_home_dir = "D:/Tools/xampp/mysql/data"
innodb_data_file_path = ibdata1:10M:autoextend
# 2015.8.3  innodb_log_group_home_dir = "/xampp/mysql/data"
innodb_log_group_home_dir = "D:/Tools/xampp/mysql/data"
```

Step 3. Start XAMPP from XAMPP control panel

1. After normal installation, you'll find XAMPP control panel icon on your Start menu(for Windows7).



2. Click XAMPP control panel button, then you'll see XAMPP control panel on your screen.

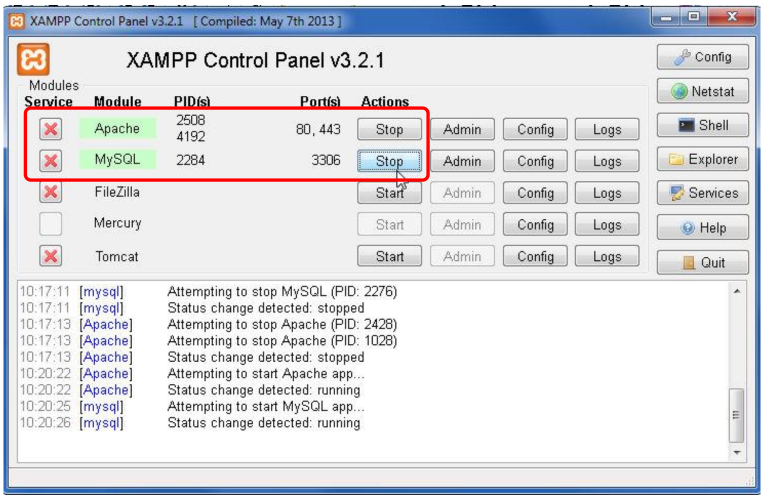

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Config
Netstat
Shell
Explorer
Services
Help
Quit

Modules

Service	Module	PID(s)	Port(s)	Actions			
✗	Apache			Start	Admin	Config	Logs
✗	MySQL			Start	Admin	Config	Logs
✗	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
✗	Tomcat			Start	Admin	Config	Logs

```
10:17:03  [Apache]    Attempting to start Apache app...
10:17:04  [Apache]    Status change detected: running
10:17:07  [mysql]     Attempting to start MySQL app...
10:17:08  [mysql]     Status change detected: running
10:17:11  [mysql]     Attempting to stop MySQL (PID: 2276)
10:17:11  [mysql]     Status change detected: stopped
10:17:13  [Apache]    Attempting to stop Apache (PID: 2428)
10:17:13  [Apache]    Attempting to stop Apache (PID: 1028)
10:17:13  [Apache]    Status change detected: stopped
```

3. Start Apache and MySQL.


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Config
Netstat
Shell
Explorer
Services
Help
Quit

Modules

Service	Module	PID(s)	Port(s)	Actions			
✗	Apache	2508 4192	80, 443	Stop	Admin	Config	Logs
✗	MySQL	2284	3306	Stop	Admin	Config	Logs
✗	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
✗	Tomcat			Start	Admin	Config	Logs

```
10:17:11  [mysql]     Attempting to stop MySQL (PID: 2276)
10:17:11  [mysql]     Status change detected: stopped
10:17:13  [Apache]    Attempting to stop Apache (PID: 2428)
10:17:13  [Apache]    Attempting to stop Apache (PID: 1028)
10:17:13  [Apache]    Status change detected: stopped
10:20:22  [Apache]    Attempting to start Apache app...
10:20:22  [Apache]    Status change detected: running
10:20:25  [mysql]     Attempting to start MySQL app...
10:20:26  [mysql]     Status change detected: running
```

While development stage, you'd better start Apache and MySQL manually.
In production level, you should set these application in the service and start them automatically when the Windows starts.

PHP basic (1)

1. **Basic syntax**

   A PHP script starts with **<?php** and ends with **?>**

```
<?php
  //  PHP code goes here
?>
```

The default file extension for PHP files is ".php".
A PHP file normally contains HTML tags, and some PHP scripting code.

[Practice 01] Our first PHP program.

Open a text editor like NotePad.

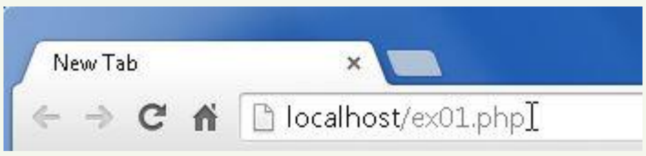
Write PHP program shown below;

```
<!DOCTYPE html>
<html>
 <body>

  <?php
   echo "My first PHP script!";
  ?>

 </body>
</html>
```

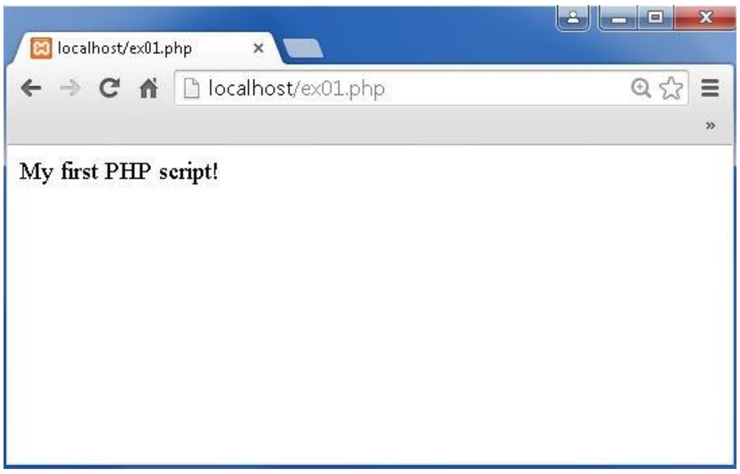
Save this under the 'DocRoot' folder. ('DocRoot' : See 'httpd.conf' and search 'DocumentRoot')

Then,

1. Start 'Apache' from XAMPP control panel.
2. Open a browser.
3. Access to 'localhost/ex01.php'



You'll see a browser window shown below;



My first PHP script!

2. Case sensitivity

	Case-sensitive	Not Case-sensitive
Key words		○
Class name		○
Function name		○
User-defined function name		○
Variable name	○	

[Practice 02] Program to check case-sensitivity no.1.

Open a text editor like NotePad.

Write PHP program shown below;

```
<!DOCTYPE html>
<html>
 <body>

  <?php
   echo "My second PHP script!";
   ECHO "My second PHP script!";
   EchO "My second PHP script!";
  ?>

 </body>
</html>
```

Save this in the same way as Practice01 with name 'ex02.php' and access.

[Practice 03] Program to check case-sensitivity no.2.

Open a text editor like NotePad.

Write PHP program shown below;

```
<!DOCTYPE html>
<html>
 <body>

   <?php
     $myVar  =  “12345”;
     echo "My Variable is " . $myVar . “<br />”;
     echo " My Variable is " . $MYVAR . “<br />”;
     echo " My Variable is " . $myVAR . “<br />”;
   ?>

 </body>
</html>
```

Save this in the same way as Practice01 with name ‘ex03.php’ and access to this program.
What comes out on your browser?

3. Comment

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment

/*
This is a multiple-lines comment block
that spans over multiple
lines
*/

// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>

</body>
</html>
```

4. Variables

Variables are "containers" for storing information.

Rules;

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

5. Data types

PHP is a loosely typed language, while C, C++, and Java, the programmer must declare the name and type of the variable before using it.

PHP automatically converts the variable to the correct data type, depending on its value.

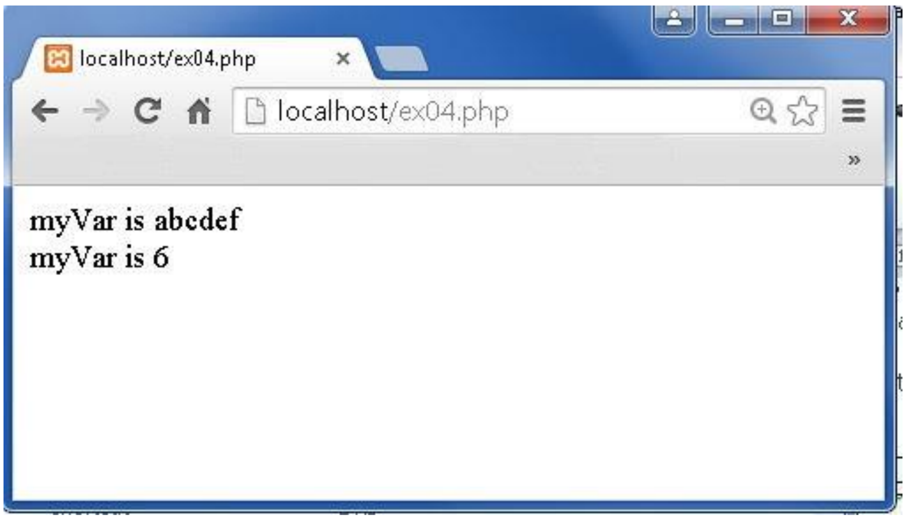
[Practice 04] Program to check PHP data types.

Open a text editor like NotePad.

Write PHP program shown below;

```
<!DOCTYPE html>
<html>
 <body>

   <?php
     $myVar  = "abcdef";
     echo "myVar is " . $myVar . "<br />";
     $myVar  =   1 + 2 + 3;
     echo "myVar is " . $myVar . "<br />";
   ?>

 </body>
</html>
```

Save this in the same way as Practice01 with name ‘ex04.php’ and access to this program.
What comes out on your browser?



```
myVar is abcdef
myVar is 6
```

In the first case, $myVar has string data, but in the second, it has integer value.

6. Strings

A string is a series of characters, which means a string is an array of characters.

6-1 Single quoted strings

If characters are enclosed by single quotes, it's a literal string.

ex)

```
<?php
   echo 'This is an example of simple string';
?>
```

6-2 Double quoted strings

If characters are enclosed by double quotes,

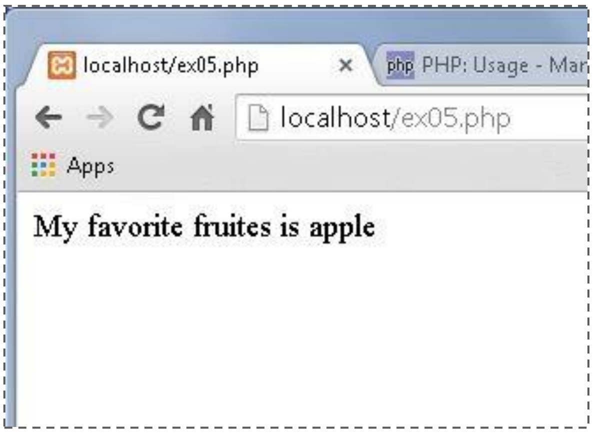
1) PHP will interpret escape sequences for special characters
   (See 6-3 Escape sequences)
2) Variable names will be expanded.

[Practice 05] Double quoted string

Open a text editor like NotePad and write PHP program shown below;

```
<?php
    $var  =  'apple';
    echo "My favorite fruites is $var";
?>
```

Save it with name 'ex05.php' in 'DOCROOT' folder and access via browser.



My favorite fruites is apple

6-3 Escape sequences

Sequence	Meaning
\n	Linefeed (0x0A)
\r	Carriage return (oxoD)
\t	Horizontal tab (0x09)
\v	Vertical tab
\e	Escape (ox1B)
\f	Form feed (0x0C)
\\	Backslash
\$	Dollar sign
\”	Double-quote

6-4 Heredoc

<<< : Heredoc operator. After this operator, an identifier and newline are provided.
Then some strings follow.
After finishing strings, at the top of newline, the same identifier comes to show the end of Heredoc.

```
<?php
   $var = <<< EOD
This is a sample of Heredoc.
We can put here some strings.
EOD;
?>
```

6-5 String as an array of characters

String is an array of characters. See an example below;

```
<?php
   $var = “This is a string.”;

   echo  “First character of the string is “ .  $var[0];
?>

   -→  This program will display a character ‘T’.
```

7. Constant

A constant is an identifier name for a simple value.
The value can’t be changed during the execution.
By naming convention, constant name is always upper cases.

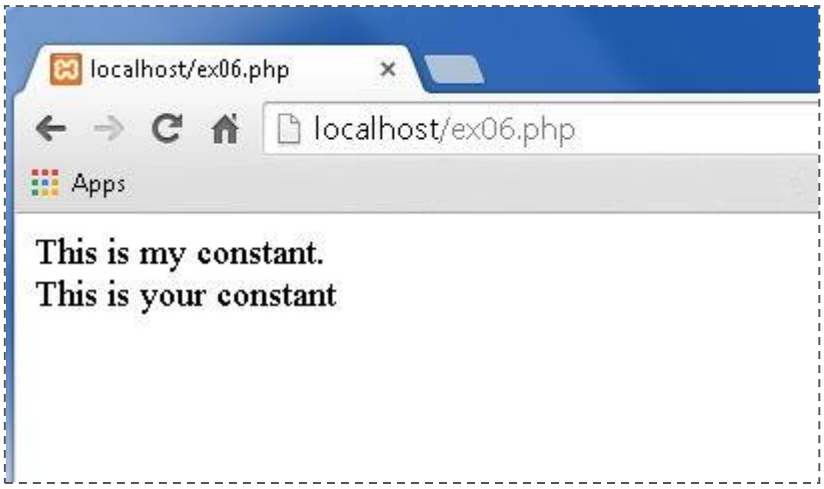
To define constants, ‘define’ function is used, in class ‘const’ keyword is used.
(See examples shown below)
.

[Practice 06] Define constant

Open a text editor like NotePad and write PHP program shown below;

```
<?php
   define( “MYCONST”,  “This is my constant.” );

   Class MyConst {
      const  YOURCONST = “This is your constant.”;

      Public static function getYourConst() {
         return  self::YOURCONST;
      }
   }

   echo  MYCONST . “<br />”;

   echo  MyConst::getYourConst() . “<br />”;

?>
```

Save it with name ‘ex06.php’ in ‘DOCROOT’ folder and access via browser.



```
This is my constant.
This is your constant
```

PHP basic (2)

7. Operators

PHP has several kinds of operators;

- Arithmetic operators
- Comparison operators
- Assignment operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators

7-1 Arithmetic operators

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power (Introduced in PHP 5.6)

[Practice 07] Arithmetic operators

Open a text editor like NotePad and write PHP program shown below;

```
<?php
    $x  =  15,  $y  =  4;
    echo “x + y = ”  .  ( $x + $y );
    echo “x - y = ”  .  ( $x - $y );
    echo “x * y = ”  .  $x * $y;
    echo “x / y = ”  .  $x / $y;
    echo “x % y = ”  .  $x % $y;
    echo “x ** y = ”  .  $x ** $y;
?>
```

Save it with name ‘ex07.php’ in ‘DOCROOT’ folder and access via browser.

7-2 Comparison operators

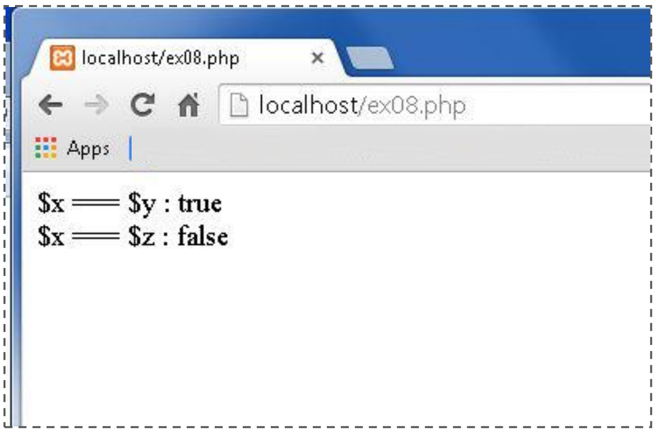
Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

[Practice 08] Comparison operator ===

Open a text editor like NotePad and write PHP program shown below;

```
<?php
    $x  =  array( 1, 2, 3 );
    $y  =  array( 1, 2, 3 );
    $z  =  array( 1, 2, 3, 4 );
     if( $x === $y ) {
        echo ‘#x === $y  :  true<br />’;
     } else {
        echo ‘#x === $y  :  false<br />’;
     }
     if( $x === $z ) {
        echo ‘#x === $z  :  true<br />’;
     } else {
        echo ‘#x === $z  :  false<br />’;
     }
?>
```

Save it with name ‘ex08.php’ in ‘DOCROOT’ folder and access via browser.



```
$x === $y : true
$x === $z : false
```

7-3 Assignment operators

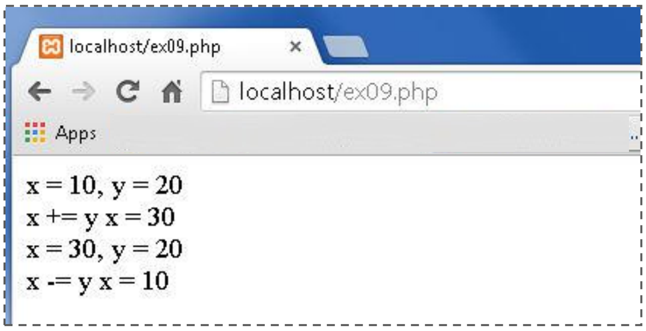
Assignment	Same as ...	description
x = y	x = y	Left operand gets value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

[Practice 09] Assignment operator +=, -=

Open a text editor like NotePad and write PHP program shown below;

```
<?php
   $x  =  10;
   $y  =  20;
   echo “x = $x,  y = $y<br />”;
   $x += $y;  echo “ x += y   x = $x<br />”;
   echo “x = $x,  y = $y<br />”;
   $x -= $y;  echo “ x -= y   x = $x”;

?>
```

Save it with name ‘ex09.php’ in ‘DOCROOT’ folder and access via browser.


localhost/ex09.php

localhost/ex09.php

```
x = 10, y = 20
x += y x = 30
x = 30, y = 20
x -= y x = 10
```

7-4 Increment/Decrement operators

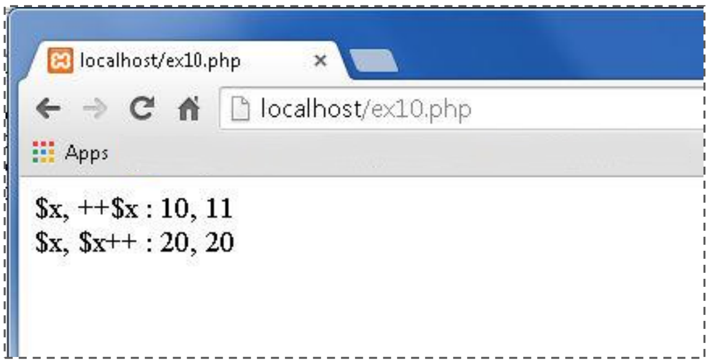
Operator	Name	description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

[Practice 10] Increment/Decrement operator

Open a text editor like NotePad and write PHP program shown below;

```
<?php
   $x  =  10;
   echo '$x, ++$x  : '  . $x . ',   ' .  ++$x . '<br />';
   $x  =  20;
   echo '$x, $x++  : '  . $x . ',   ' .  $x++ . '<br />';
?>
```

Save it with name 'ex10.php' in 'DOCROOT' folder and access via browser.


localhost/ex10.php

localhost/ex10.php

```
$x, ++$x : 10, 11
$x, $x++ : 20, 20
```

7-5 Logical operators

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x|| $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

7-6 String operators

Operator	Name	Example	Result
. (dot)	Concatenation	$x . $y	Concatenation of $x and $y
.=	Concatenation assignment	$x .= $y	Appends $y to $x

7-7 Array operators

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y
==	Equality	$x == $y	True if $x and $y have the same key/value pairs
===	Identity	$x === $y	True if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	True if $x is not equal to $y
<>	inequality	$x <> $y	True if $x is not equal to $y
!==	Non-identity	$x !== $y	True if $x is not identical to $y

8. if … else

8-1 if statement

Syntax :

```
if (condition) {
    code to be executed if condition is true;
}
```

Example:

```
<?php
  $t = date('H');

  If( $t < '20' ) {
    echo 'Have a nice day';
  }

?>
```

8-2 if … else statement

Syntax :

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

Example:

```
<?php
  $t = date('H');

  If( $t < '20' ) {
    echo 'Have a nice day';
  } else {
    echo 'Have a good night';
  }

?>
```

8-3 if … elseif … else statement

Syntax :

```
if (condition) {
    code to be executed if condition is true;
} elseif (condition) {
    code to be executed if condition is false;
} else {
    code to be executed if condition is false;
}
```

Example:

```
<?php
  $t = date('H');

  If( $t < '10' ) {
     echo 'Good morning';
  } elseif ( $t < '20' ){
     echo 'Have a nice day';
  } else {
     echo 'Have a good night';
  }

?>
```

9. switch Statement

The switch statement is used to perform different actions based on different conditions.

Syntax :

Attention!
Don’t forget ‘:’(colon) !
It’s colon, not semi-colon.

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

Attention!
Don’t forget ‘break’!

Example:

```
<?php
  $myBirth = “Feb”;

  switch ( $myBirth ) {
    case ‘Jan’:
      echo ‘Your birthday is in Jan’;
      break;
    case ‘Feb’:
      echo ‘Your birthday is in Feb’;
      break;
    case ‘Mar’:
      echo ‘Your birthday is in Mar’;
      break;
        .
        .

    default’:
      echo ‘When were you born?’;
  }
?>
```

10. Loop control

10-1 While loop

While loop runs a block of codes while the specified condition is true.

Syntax :

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

Example :

```
<?php
  $loopCnt = 1:
  $sum = 0;

  while ( $loopCnt <= 100 ) {
    $sum += $loopCnt++;
  }
  echo "Adding integer from 1 to 100 makes $sum";

?>
```

10-2 Do … while loop

Do … while loop checks specified conditions after executing a block of codes.
The difference between While loop and Do … while loop is ;
"Do … while loop will execute block of codes at least once.
While loop may not execute block of codes in case where the condition is true before entering into the loop."

Syntax :

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

Example :

```
<?php
  $loopCnt = 1:
  $sum = 0;

  do {
    $sum += $loopCnt++;
  } while ( $loopCnt <= 100 );
  echo "Adding integer from 1 to 100 makes $sum";

?>
```

[Practice 11] While / Do … while loop

Make a program named ‘ex11.php’ which adds all even numbers from 1 to 100 and show the sum;

Save it with name ‘ex11.php’ in ‘DOCROOT’ folder and access via browser.

10-3 For loop

For loop runs a block of codes while the specified condition is true.

Syntax :

```
for (init counter; test counter; increment counter) {
    code to be executed;
}
```

parameters :

init counter: Initialize the loop counter value

test counter: Evaluated for each loop iteration.
If it evaluates to TRUE, the loop continues.
If it evaluates to FALSE, the loop ends.

increment counter: Increases the loop counter value

Example :

```
<?php
   for ( $loopCnt = 1, $sum = 0; $loopCnt <= 100; $loopCnt++ ) {
      $sum += $loopCnt;
   }
   echo “Adding integer from 1 to 100 makes $sum”;

?>
```

[Practice 12] For loop

Make a program named ‘ex12.php’ which adds all odd numbers from 1 to 100 using for loop and show the sum;

Save it with name ‘ex12.php’ in ‘DOCROOT’ folder and access via browser.

10-4 Foreach loop

Foreach loop works only on an array or on an object.
Foreach loop assigns value or pair of key&value or object instance to variables.

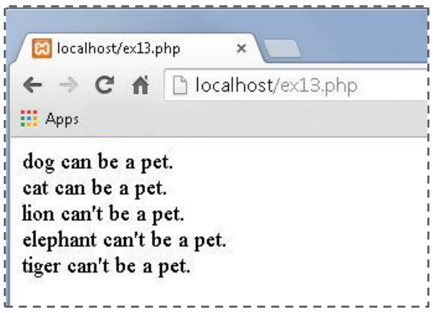
Syntax :

```
foreach ($array as $value) {
    code to be executed;
}

foreach ($array as $key => $value) {
    code to be executed;
}

foreach ($object as $instance) {
    code to be executed;
}
```

Example :

```
<?php
  $animals = array( “dog”, “cat”, “lion”, “elephant”, “tiger” );
  foreach ( $animals as $animal ) {
    echo “$animal <br />”;
  }
?>
```

```
<?php
  $animals = array( “dog” => TRUE,
                    “cat” => TRUE,
                    “lion” => FALSE,
                    “elephant” => FALSE,
                    “tiger”  => FALSE );
  foreach ( $animals as $animal => $ableToBePet) {
    if( $ableToBePet  ) {
      echo “$animal can be a pet.<br />”;
    } else {
      echo “$animal can’t be a pet.<br />”;
    }
  }
?>
```



```
dog can be a pet.
cat can be a pet.
lion can't be a pet.
elephant can't be a pet.
tiger can't be a pet.
```

[Practice 13] Loop

Make a program named ‘ex13.php’ which has a group of data with a pair of ‘nation name’ and ‘capital city’ such as USA and Washington.
Program will scan the data and show all nation names and capital city.

Save it with name ‘ex13.php’ in ‘DOCROOT’ folder and access via browser.

PHP basic (3)

11. functions

11-1 User defined functions

A function is a block of statements that can be used repeatedly in a program.

A function will not execute immediately when a page loads.

A function will be executed by a call to the function.

Syntax :

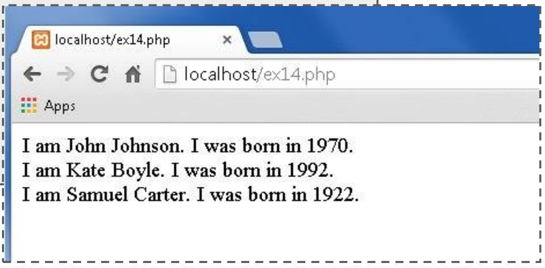
```
function functionName(argument1, argument2, … ) {
    code to be executed;
}
```

A function name can start with a letter or underscore (not a number).
Function names are NOT case-sensitive.

Example :

```
<?php
   function showMessage( ) {
      echo “Hello World!”;
   }

   showMessage();

?>
```

```
<?php
   function showMessage( $name, $birthYear ) {
      echo “I am $name. I was born in $birthYear.”;
   }

   showMessage( “John Johnson”, 1970);
   showMessage( “Kate Boyle”, 1992);
   showMessage( “Samuel Carter”, 1922);

?>
```



```
I am John Johnson. I was born in 1970.
I am Kate Boyle. I was born in 1992.
I am Samuel Carter. I was born in 1922.
```

11-2 Default argument value

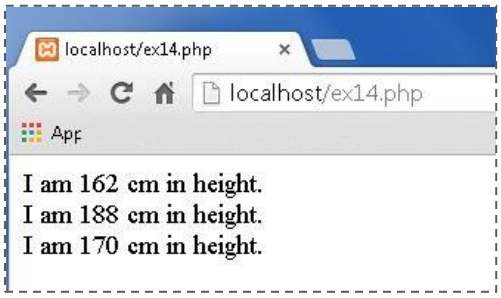
Arguments of a function can have a default value.

Syntax :

```
function functionName(argument1 = defaultValue,  … ) {
}
```

Example :

```
<?php
   function setHeight( $height = 170 ) {
      echo "I am $height cm in height.";
   }

   setHeight( 162 );
   setHeight( 188 );
   setHeight( );      // use default value

?>
```



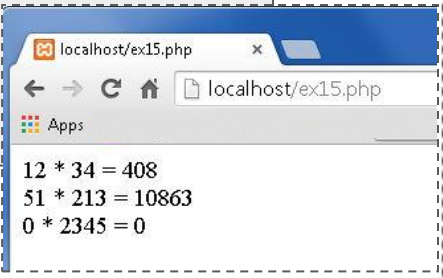
```
I am 162 cm in height.
I am 188 cm in height.
I am 170 cm in height.
```

11-3 Returning values

Using return statement, a function can return a value.

Example :

```
<?php
   function calc( $arg1, $arg2 ) {
      return $arg1 * $arg2;
   }

   echo "12 * 34 = " . calc( 12, 34) . <br />;
   echo "51 * 213 = " . calc( 51, 213) . <br />;
   echo "0 * 2345 = " . calc( 0, 2345) . <br />;
?>
```



```
12 * 34 = 408
51 * 213 = 10863
0 * 2345 = 0
```

12. Arrays

There are three kinds of array;

1) Indexed array
2) Associated array
3) Mutidimensional array

12-1 Indexed array

An array indexed by number stating by 0.

Example :

```
$months = array( “Jan”, “Feb”, “Mar”, “Apr”, “May”, “Jun”, “Jul”, “Aug”, “Sep”,
                 “Oct”, “Nov”, “Dec” );
```

In an array shown above,

```
$months[0]   has value ‘Jan’,
$months[10] has value ‘Nov’,    and so on
```

12-2 Associated array

An array indexed by ‘name’.

Example :

```
$daysInMonth = array( ‘Jan’ => 31, ‘Feb’ => 28, ‘Mar’ => 31, ….. );
or
$daysInMonth[ ‘Jan’ ] = 31; $daysInMonth[ ‘Feb’ ] = 28; ….
```

12-3 Multidimensional array

When a value of an array may have an array, the array is multidimensional.

In 12-1 and 12-2, the array contains value of number of days of each month.
We can define month name and numbers of days in each month in multidimensional array ;

```
$months = array( ( ‘Jan’, 31 ), ( ‘Feb’, 28 ), ( ‘Mar’, 31 ), ( ‘Apr’, 30 ), ….


$month[0][0] → ‘Jan’,   $month[1][1] → 28 ….
```

12-4 Size of array

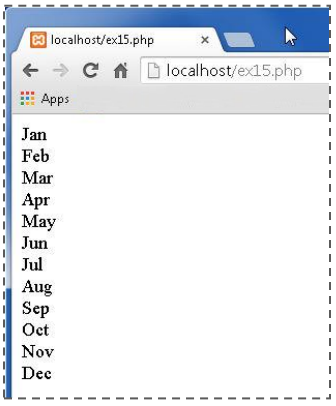
Size of array means the numbers of elements in a array.
we can get it by ‘count’ function.

```
$months = array( “Jan”, “Feb”, “Mar”, “Apr”, “May”, “Jun”, “Jul”, “Aug”, “Sep”,
                 “Oct”, “Nov”, “Dec” );
echo count( $months );     -→  12
```

12-5 Loop with array (examples)

With indexed array ;

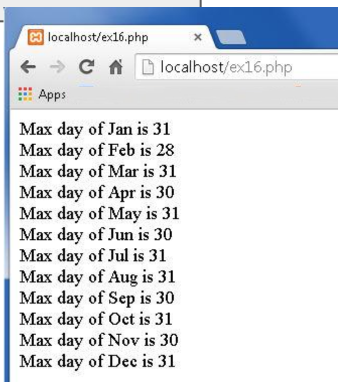
```
<?php
  $months = array( “Jan”, “Feb”, “Mar”, “Apr”,
                   “May”, “Jun”, “Jul”, “Aug”,
                   “Sep”, “Oct”, “Nov”, “Dec” );

  For( $cnt = 0; $cnt < count( $months ); $cnt++ ) {
     echo $months[$cnt] . “<br />”;
  }

?>
```



```
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
```

With associated array ;

```
<?php
  $months = array( “Jan” =>31, “Feb” => 28, “Mar” => 31,
                   “Apr” => 30, “May” => 31, “Jun” => 30,
                   “Jul” => 31,  “Aug” => 31, “Sep” => 30,
                  “Oct” => 31, “Nov” => 30, “Dec” => 31 );

  Foreach( $months as $name => $maxDay) {
     echo “Max day of $name is $maxDay “<br />”;
  }

?>
```



```
Max day of Jan is 31
Max day of Feb is 28
Max day of Mar is 31
Max day of Apr is 30
Max day of May is 31
Max day of Jun is 30
Max day of Jul is 31
Max day of Aug is 31
Max day of Sep is 30
Max day of Oct is 31
Max day of Nov is 30
Max day of Dec is 31
```

13. Superglobals

Superglobals are predefined variables in PHP.
They can be accessed from anywhere in PHP programs.

Name	Explanation
$GLOBALS	Array which stores all global variables
$_SERVER	Array which holds information about headers, paths, and script locations.
$_REQUEST	Array which collects data from an HTML form
$_POST	Array which collects data from an HTML form when the form sent by ‘method=”POST”’
$_GET	Array which collects data in the URL from an HTML form when the form sent by ‘method=”GET”’
$_FILES	Array which contains items uploaded to the current script via the HTTP POST method.
$_ENV	Array which holds a list of defined environment variables passed to the current script
$_COOKIE	Array which holds a list passed to the current script via HTTP Cookies.
$_SESSION	Array which hold session variables available to the current script

Examples :
$_SERVER

$_SERVER['PHP_SELF'] : file name of the current script(PHP program)
$_SERVER['SERVER_ADDR'] : the IP address of the host server
$_SERVER['REQUEST_METHOD'] : the request method used to access the page (such as POST)

In details, visit URL below;
http://php.net/manual/en/reserved.variables.server.php

[Practice 14] Training to find algorithm and express it in PHP program

Make a PHP program to draw a diamond shape on your browser shown below;

```
     X
    XXX
   XXXXX
  XXXXXXX
 XXXXXXXXX
XXXXXXXXXXX
 XXXXXXXXX
  XXXXXXX
   XXXXX
    XXX
     X
```

[Practice 15] Training to find algorithm and express it in PHP program

Make more than 2 PHP programs with different algorithm to show 2 digit number from 0 to 99 (from 0 to 9 numbers are shown like 00,01,02,…,09).

You will show numbers shown below;

00 01 02 03 04 05 06 07 08 09 10 11 12 ….. 98 99 (ascending ordered)

or

13 63 28 07 16 …… 55 72 (non-ordered)

PHP Basic (4)

14. PHP date & time

14-1 PHP date() function

Syntax

```
date(format,timestamp)
```

Format parameter strings

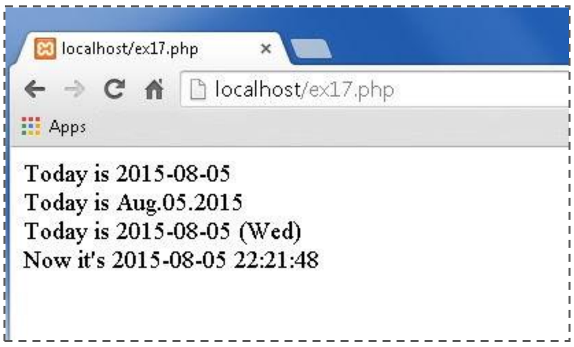
	parameter	Description	Example
day	d	Day of the month, 2 digits leading zeros	01 to 31
	D	Day of a week in 3 letters	Mon to Sun
	j	Day of the month, without leading zeros	1 to 31
	l(lower L)	Day of a week in full text	Monday…
	w	Day of a week in numerics	0 for Sunday,…
	z	Day of the year (starting from 0)	0 to 365(366)
week	W	Week number of year	30:30th week of year
month	F	Month in full text	January,…
	m	Month in numeric with leading zeros	01,02,…,12
	M	Month in short text	Jan, Feb,…,Dec
	n	Month in numeric without leading zeros	1,2,…,12
	t	Number of days in given month	28 to 31
year	Y	Year in 4 digits	2015
	y	Year in 2 digits	15, 00, 99 ….
time	a	Lower case of AM, PM	am, pm
	A	Upper case of AM, PM	AM, PM
	g	Hour:12-hour format without leading zero	1 - 12
	G	Hour:24-hour format without leading zero	1 - 24
	h	Hour:12-hour format with leading zero	01 - 12

Format parameter strings (continued)

	parameter	Description	Example
	H	Hour:24-hour format with leading zero	01 - 24
	i(lower I)	Minutes with leading zero	00 - 59
	s	Seconds with leading zero	00 - 59
Full	c	Date in ISO8601 ex) 2015-08-03T09:00:15-02:00	
	r	Date in RFC2822 ex) Mon,03,Aug 2015 09:00:15-02:00	

```
<?php
   echo “Today is “ . date( “Y-m-d”) . “<br />”;
   echo “Today is “ . date( “M.d.Y”) . “<br />”;
   echo “Today is “ . date( “Y-m-d (D)”) . “<br />”;

   echo “Now it’s “ .  date( “Y-m-d H:i:s” . “<br />”;

?>
```

Examples;



```
Today is 2015-08-05
Today is Aug.05.2015
Today is 2015-08-05 (Wed)
Now it's 2015-08-05 22:21:48
```

14-2 mktime() function

Get a UNIX timestamp for a date given in arguments.

Syntax

int mktime *([int $hour = date("H") [, int $minute = date("i") [, int $second = date("s") [,int $month = date("n") [, int $day = date("j") [, int $year = date("Y") [, int $is_dst = -1]]]]]]])*

Complicated symtax!!
You can neglect it and check examples on next page.

Example:

1) Get yesterday;
   date(“Y-m-d”, mktime(0,0,0, date(“n”), date(”d”)-1, date(“Y”)));

2) Get last day of this month;
   date(“Y-m-d”, mktime(0,0,0, date(“n”)+1, 0, date(“Y”)));
   ‘month’ argument = 0 : the function returns the last day of the last month of ‘given month’.
   in the case above, ‘given month’ is ‘date(“n”)+1’ = next month, then ‘last month of given month’ means ‘this month’.

3) Get date after 100 days from today
   date("Y-m-d", mktime(0,0,0, date("n"), date("d")+100, date("Y")))

14-3 strtotime() function

Get UNIX timestamp by formatted date

Syntax :

```
int strtotime ( string $time [, int $now = time() ] );
```

$now : if supplied, it should be timestamp value of a certain date.
if not supplied, it will be the timestamp of current date & time.

$time : A date/time string.
Valid formats are explained in http://php.net/manual/en/datetime.formats.php

Example:

```
<?php
  echo strtotime(“now”)  . “<br />”;
                -→  show timestamp value of current time
  echo date( “Y-m-d”, strtotime(“now”) ) . “<br />”;
                -→ show current date in ‘yyyy-mm-dd’ format
  echo date( “Y-m-d”, strtotime(“+1 day”) ) . “<br />”;
                -→ show tomorrow in ‘yyyy-mm-dd’ format
?>
```

Here you’ll see some examples often used in *$time* argument;

now	timestamp of now
today	time stamp if today
tomorrow	timestamp of tomorrow
yesterday	timestamp of yesterday
+1 day	timestamp of tomorrow
+1 week	timestamp of the day one week from today

+1 month	timestamp of the day one month from today
+1 year	timestamp of the day one year from today
+1 hour	timestamp of one hour later from now
+1 minute	timestamp of one minute later from now
+1 second	timestamp of one second later from now

Number value in '+1' can be changed. Minus value is available.

[Practice 16] date/time related functions → ex16.php

1) Show date of 5 days after today in 'yyyy-mm-dd' format
2) Show the last day of current month in 'yyyy-mm-dd' format
3) Show how many days there are between today and the last day of the current month.

15. Coding standard & Naming convention

15-1 Coding standard (by PEAR project)

[Indenting & line length]

1) Use an indent of 4 spaces, with no tabs.

2) It is recommended to keep lines at 75 – 85 characters for better readability.

[control structure]

3) Control statements should have one space between keyword and opening braces.
You'd better use curly braces even when they are technically optional.

```
<?php
    switch ( condition ) {
        case 1:
            action1;
            break;
        case 2:
            action2;
            break;
        default:
            defaultaction;
            break;
    }
?>
```

or

```
<?php
    switch ( condition ) {
    case 1:
        action1;
        break;
    case 2:
        action2;
        break;
    default:
        defaultaction;
        break;
    }
?>
```

4) Split long if statements onto several lines.
Logical operators($$, || etc) should be at the beginning of the line
for better readability and to comment easily.

```
<?php
    if ( (   $condition1     //  some comment
          || $condition2  ) //  some comment
        && $condition3      //  some comment
        && $condition4
       ) {
            //  code here
    }
?>
```

5) When if clause is really long to split, it might be better to simplify.
You could express conditions as variables. See example below;

```
<?php
  if ( (   $condition1
        || $condition2   )
     &&
      (    $condition3
        && $condition4  )
    ) {
        //  code here
  }
?>
```

⬇

```
<?php
  $isBlock1 = ( $condition1 || $condition2 );
  $isBlock2 = ( $condition3 && $condition4 );
  if ( $isBlock1 && $isBlock2 ) {
     //   code here
  }
?>
```

[function calls]

6) Functions should be called with no spaces between the function name, the opening parenthesis, and the first parameter; spaces between commas and each parameter, and no space between the last parameter, the closing parenthesis, and the semicolon.

```
<?php

  $var = $someFunction($arg1,  $arg2, $arg3);

?>
```

To promote readability, a block of related assignments can be aligned

```
<?php

  $return1       = $function1($arg1,  $arg2);
  $longReturn   = $function2($arg3);

?>
```

[Class definition]

7) Class definitions have their opening brace on a new line.

```
<?php
   class SomeClass
   {
        // code here
   }
?>
```

15-2 Naming convention (by PEAR project)

1) Global variables

Names of global variables should start with single underscore followed by the package name and another underscore.

ex) $_PACKAGE1_some_global_var

2) Global functions

CamelCase(Camel caps) is the practice of writing compound words or phrases such that each word or abbreviation begins with a capital letter.
In PHP, camel case starts with lower case letter.
In addition, they should have the package name as a prefix.

ex) PACKAGE1_someFunction()

3) Class name
Class name always begin with uppercase and should be given descriptive names.
Class name would better reflect class hierarchy separated with a single underscore.

ex) GWI_DB_Access GWI_Customer

4) Class variables, methods
Class variables and methods should be named using 'Camel Case' style.

ex) connect() getRecordByName() $recordCount

5) Private variables, methods
Private variables and methods in Class or in function should be preceded by a single underscore.

ex) _status _sort()

6) Constant
Constants should always be all-uppercase.

ex) DB_CONNECTION_STRINGS HTML_ERROR_404

16. Scope of variables

16-1 local and global scope

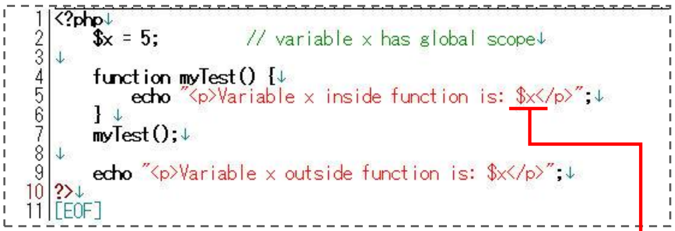
scope of variables;
local
global
static

A variable declared outside function has global scope.
A variable declared inside function has local scope.

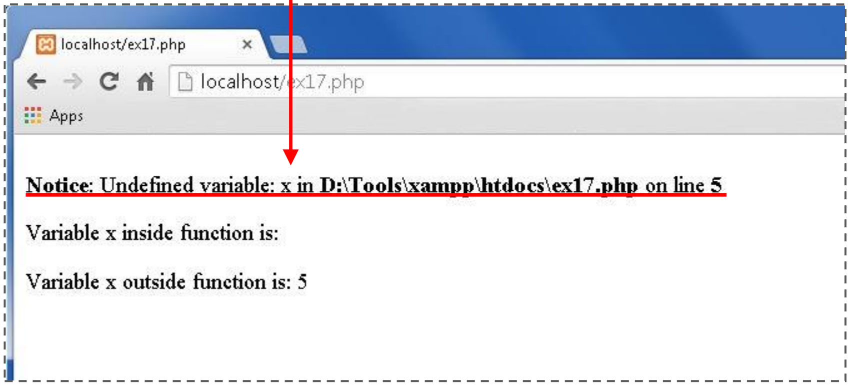
[Practice 17] local and global scope → ex17.php

Make a program shown below with name 'ex17.php' and execute it.
Check the output and understand how the variable scope works.

```
<?php
   $x = 10;                // variable x has global scope

   function myTest() {
      echo "Variable x inside function is: $x<br />";
   }

   myTest();

   echo "Variable x outside function is: $x<br />";
?>
```



```
<?php
    $x = 5;          // variable x has global scope

    function myTest() {
        echo "<p>Variable x inside function is: $x</p>";
    }
    myTest();

    echo "<p>Variable x outside function is: $x</p>";
?>
[EOF]
```

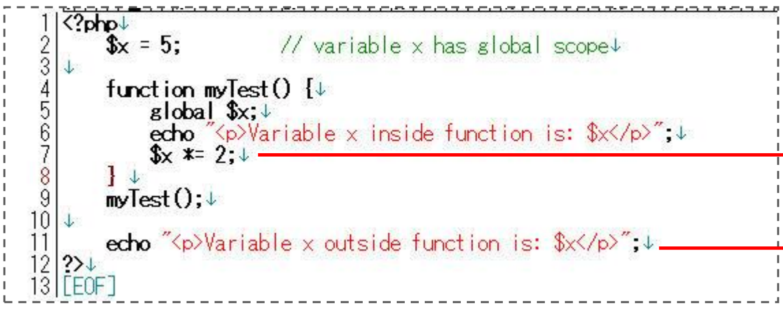


```
Notice: Undefined variable: x in D:\Tools\xampp\htdocs\ex17.php on line 5

Variable x inside function is:

Variable x outside function is: 5
```

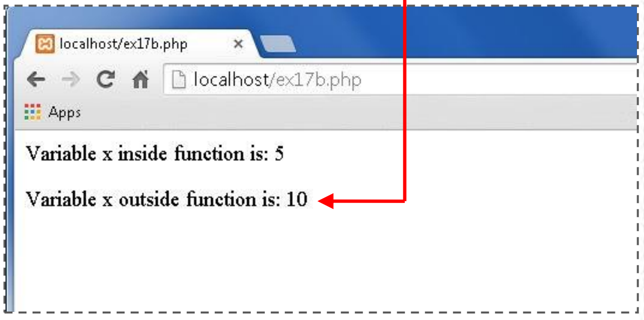
In the previous practice, how can we refer to variable x defined outside function from inside the function myTest?

We need to ‘global declaration’. Sample is shown below;

```
<?php
   $x = 10;               // variable x has global scope

   function myTest() {
      global $x;
      echo "Variable x inside function is: $x<br />";
      $x *= 2;            //  try to modify the value of x
   }

   myTest();

   echo "Variable x outside function is: $x<br />";
?>
```



```
<?php
    $x = 5;          // variable x has global scope

    function myTest() {
        global $x;
        echo "<p>Variable x inside function is: $x</p>";
        $x *= 2;
    }
    myTest();

    echo "<p>Variable x outside function is: $x</p>";
?>
[EOF]
```



localhost/ex17b.php

Variable x inside function is: 5

Variable x outside function is: 10

16-2 static variable

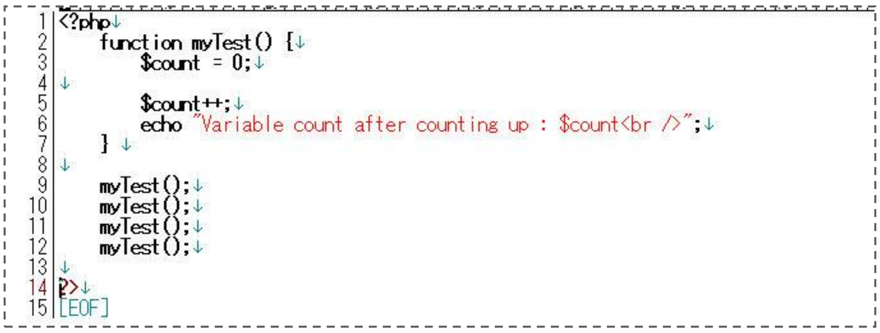
[Practice 18] without static scope → ex18.php

Make a program shown below with name ‘ex18.php’ and execute it.
Check the output.

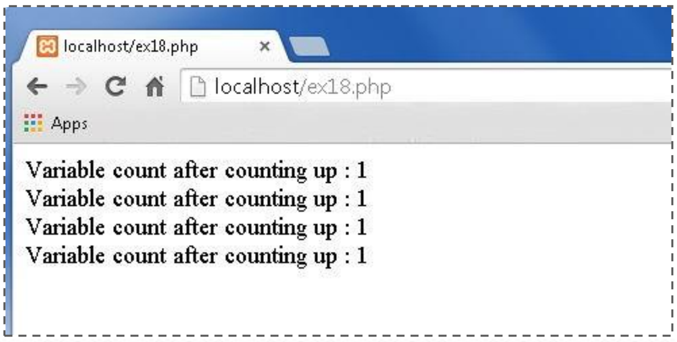
```
<?php
  function myTest() {
    $count  =  0;
    $count++;
    echo "Variable count after count up: $count<br />";
  }

  myTest();
  myTest();
  myTest();
  myTest();

?>
```

We expect $count will be counted up by 1 each time we call the function muTest, so we expect incremented number like 1,2,3,4 shown on the browser.

How is the result?



```
<?php
    function myTest() {
        $count  = 0;

        $count++;
        echo "Variable count after counting up : $count<br />";
    }

    myTest();
    myTest();
    myTest();
    myTest();

?>
[EOF]
```

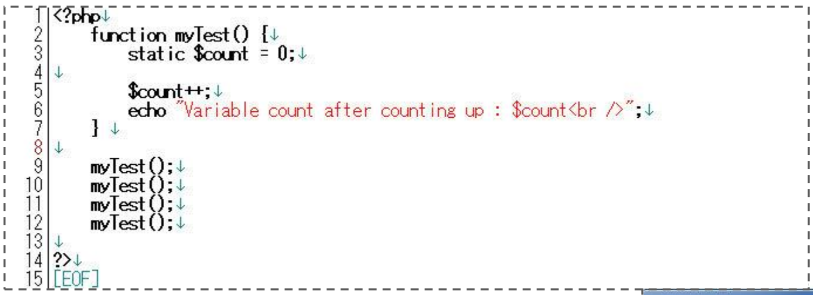


```
Variable count after counting up : 1
Variable count after counting up : 1
Variable count after counting up : 1
Variable count after counting up : 1
```

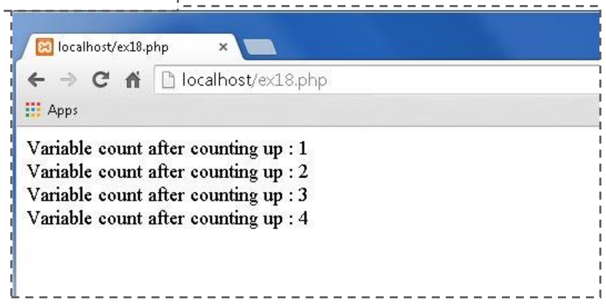
How can we keep the value of $count in the function myTest?

It's the keyword 'static'. Let's modify the ex18.php like this;

```
<?php
   function myTest() {
      static $count   =  0;
      $count++;
      echo "Variable count after count up: $count<br />";
   }

   myTest();
   myTest();
   myTest();
   myTest();

?>
```



```
<?php
    function myTest() {
        static $count = 0;

        $count++;
        echo "Variable count after counting up : $count<br />";
    }

    myTest();
    myTest();
    myTest();
    myTest();

?>
[EOF]
```



```
Variable count after counting up : 1
Variable count after counting up : 2
Variable count after counting up : 3
Variable count after counting up : 4
```

Static variable can keep value as it is.
In the ex18.php, from the second time when the function myTest is called,
variable $count is never initialized, will keep the value as it is during the program life. This ivariables.s the static scope of

17. include / require statements

In PHP, we can insert PHP programs into another PHP program by using 'insert' statement or 'require' statement.

Two statement have the same function to insert the contents of an external file into the position where these statement is written. The difference between the two is ;

in case of failure of the statement,

'require' statement will produce fatal error and stop the script

'include' statement will produce only Warning and continue the script

So, for security reason, we often use 'require' statement.

Syntax :

```
include   'file-name';
include_once   'file-name';
require   'file-name';
require_once   'file-name';
```

'include_once','require_once' has the same function as 'include','require'.
PHP will check whether the requested file is already included/required or not,
and if already included/required, then PHP will never include/require again.

Examples above don't use (and), because both 'include' and 'require' are not functions but basic statement in PHP. But it's accepted(allowed) to use (and)
like examples below;

```
include('prog1.php');
include_once('progb,php');
require('prog1.php');
require_once('progb,php');
```

[Practice 19] example of 'include' → ex19.php, ex19b,php

Make two programs shown below with name 'ex19.php','ex19b.php' and access 'ex19,php' from browser.
Check result carefully from the point of 'variable scope'.

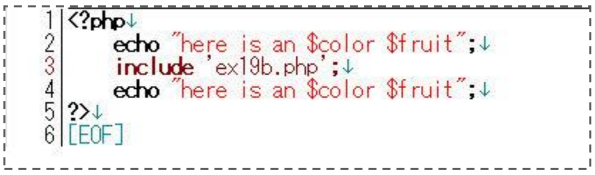
ex19.php

```
<?php
  echo "Here is an $color $fruit";
  include 'ex19b.php';
  echo "Here is an $color $fruit";
?>
```

ex19b.php

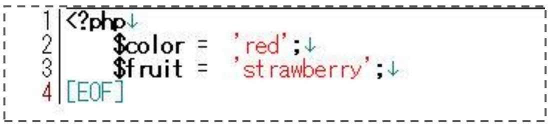
```
<?php
  $color  =  'red';
  $fruit   =  'strawberry';
?>
```

ex19.php

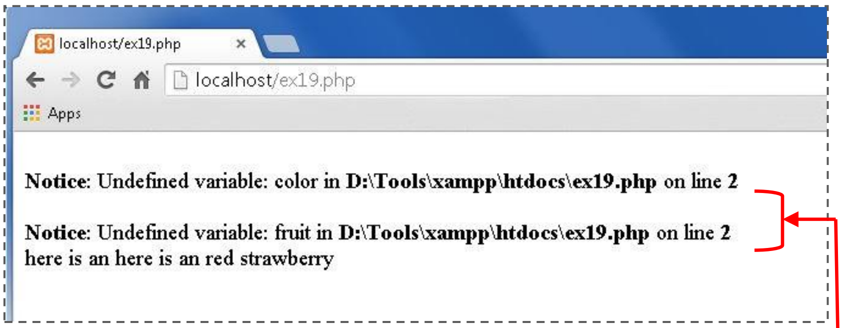


```
<?php
    echo "here is an $color $fruit";
    include 'ex19b.php';
    echo "here is an $color $fruit";
?>
[EOF]
```

ex19b.php



```
<?php
    $color =  'red';
    $fruit =  'strawberry';
[EOF]
```



```
Notice: Undefined variable: color in D:\Tools\xampp\htdocs\ex19.php on line 2

Notice: Undefined variable: fruit in D:\Tools\xampp\htdocs\ex19.php on line 2
here is an here is an red strawberry
```

After execution of 'include' statement, this program will be like follows;

```
<?php
   echo "Here is an $color $fruit";
   $color  =  'red';
   $fruit   =  'strawberry';
   echo "Here is an $color $fruit";
?>
```

Then you can easily find the reason of error message.

HTML overview

As you know, HTML is a mark-up text for web pages.

HTML syntax rules are somehow loose and browser tools especially IE(Internet Explorer) accepts, so called, ‘bad-formed’ HTML.
For future use, you’d better follow XHTML syntax rules.
XHTML is HTML redesigned as XML, has syntax rules stricter than HTML.

Some XHTML rules different from HTML

1. XTML DOCTYPE is mandatory.
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 //EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<!DOCTYPE html>
```
2. The xmlns(XML name space) attributes in <html> is mandatory
```
<html xmlns="http://www.w3.org/1999/xhtml">
```
3. <html>, <head>, <title> and <body> are mandatory

4. XHTML elements must be properly nested.
```
bad example:   <p>This is <b>emphasized letters</p></b>
                        -→ <p>this is <b> emphasized letters</b></p>
```
5. XHTML elements must always be closed,
```
bad example:   <p>New paragraph             -→ <p>New paragraph</p>
               we’ll go to next line.<br>  -→ We’ll go to next line.<br />
```
6. XHTML elements must be in lowercase.
```
bad example:   <P>New paragraph</P>  -→ <p>New paragraph</p>
```
7. XHTML documents must have one root element.

8. Attribute names must be in lower case.
```
bad example:   <img  SRC=….. />   -→ <img src=….. />
```
9. Attribute values must be quoted.
```
bad example:     <img width=400px   …   />
                         → <img width=”400px”  …   />
```
10. Attribute minimization is prohibited.
```
bad example:     <input type=”checkbox”  ….  checked>
                        → <input type=”checkbox”  ….  checked=”checked”>
```

HTML files in XHTML have always 3 top lines shown below;

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

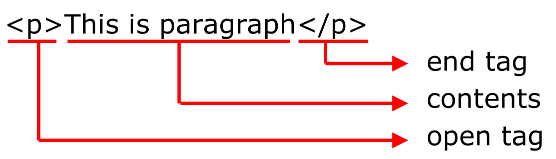
First line shows 'this document follows xml syntax. 'DOCTYPE' shows HTML version.
Third line shows 'this document follows syntaxes defined in the shown name space'.

HTML elements, attributes

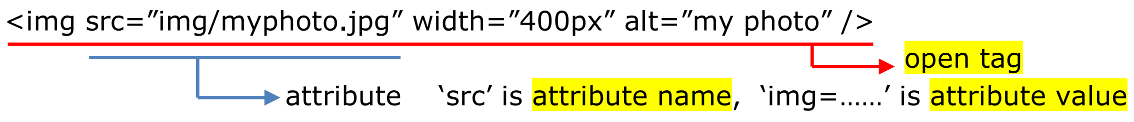
HTML document is made up by a set of HTML elements.
HTML element has ;
- open tag
- contents
- end tag (closing tag)

    some elements have only start tag and don't have contents nor end tag.

Examples :



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



```
<img src="img/myphoto.jpg" width="400px" alt="my photo" />
```

<img> tag doesn't have contents nor end tag.

Table layout model & Box layout model

1) Table layout model
   To design/layout web page by usingrows and columns of a table.

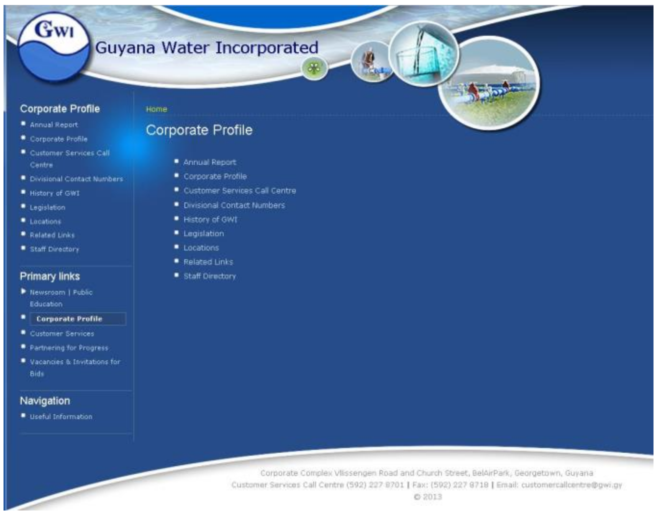
2) Box layout model
   All HTML element can be considered as boxes.
   In CSS, the term 'box model' is used when talking about design and layout.

3) HTML4 specification said;
   "Tables should not be used purely as a means to layout document content, because this may cause problems when rendering to non-visual media."

4) HTML5 specification said;
   "Tables should not be used as layout aids."

You'd better follow 'Box layout model'.
In 'Box layout model', CSS (Cascade Style Sheet) has an important part, so you'd better learn CSS syntax and how to use it.

Let's check GWI web site. How are the web pages in GWI site designed?
We'll check one of the web pages, 'GWI corporate profile' page shown below.


Guyana Water Incorporated
Corporate Profile
Annual Report
Corporate Profile
Customer Services Call Centre
Divisional Contact Numbers
History of GWI
Legislation
Locations
Related Links
Staff Directory
Primary links
Newsroom | Public Education
Corporate Profile
Customer Services
Partnering for Progress
Vacancies & Invitations for Bids
Navigation
Useful Information
Home
Corporate Profile
Annual Report
Corporate Profile
Customer Services Call Centre
Divisional Contact Numbers
History of GWI
Legislation
Locations
Related Links
Staff Directory
Corporate Complex Vlissengen Road and Church Street, BelAirPark, Georgetown, Guyana
Customer Services Call Centre (592) 227 8701 | Fax: (592) 227 8718 | Email: customercallcentre@gwi.gy
© 2013

Here's a list of all tags used in 'body block' of 'GWI corporate profile' page.

<div>, <a>, <img>, <h1>, <ul>, <li>, <table>, <tr>, <td>
<h2>, <br>

Not so many. You need to understand these tags and attributes.

See additional articles 'Analysis of GWI web page structure' and HTML source.

Role of HTML & CSS

The role of CSS is to define design and layout details in HTML.

The role of HTML is to keep contents of web page and layout them by CSS support.

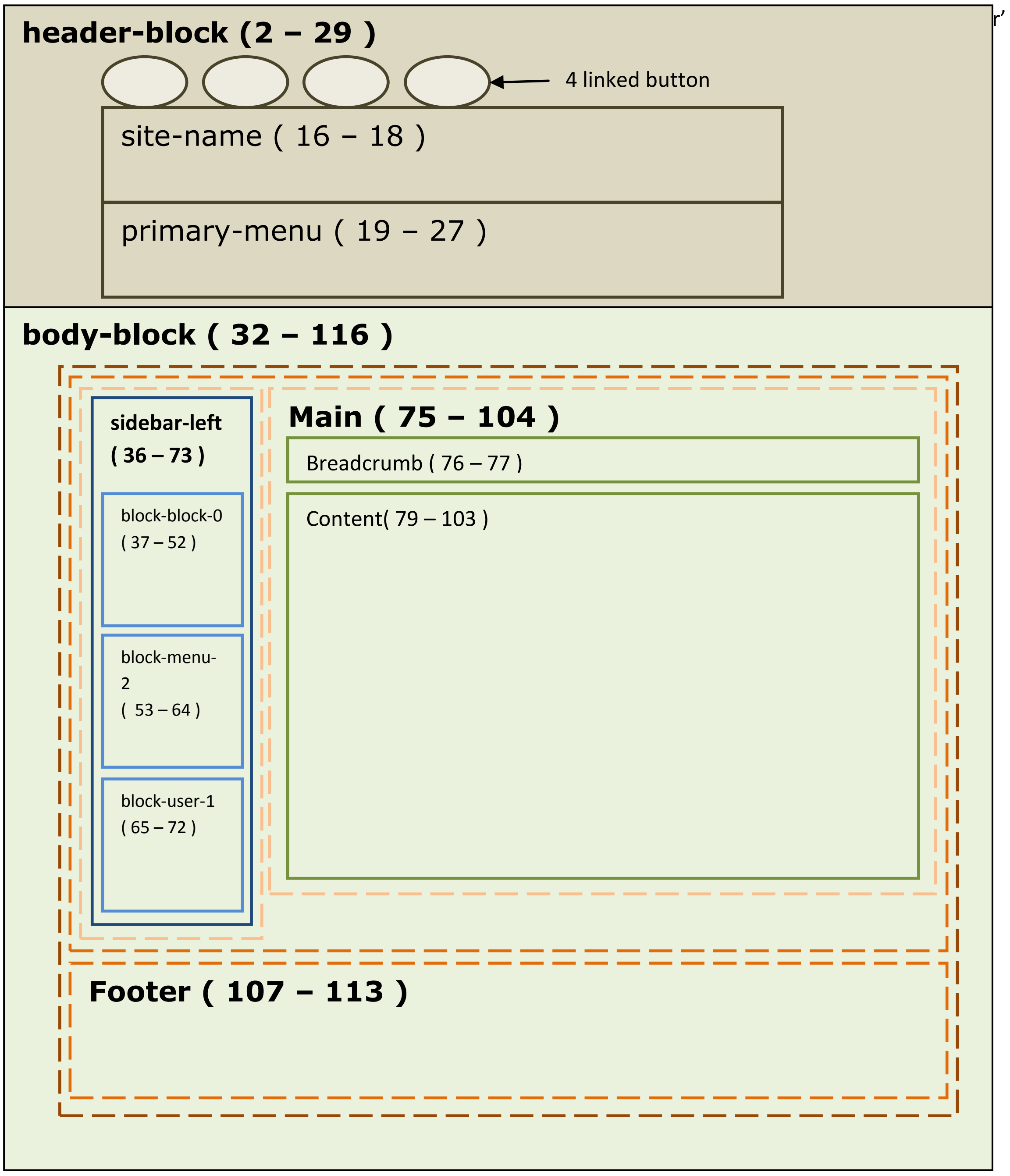
Ideally, HTML might have only contents and doesn't have layout/design factors like 'wigth','length' and so on, these layout/design factors should be in CSS.

Analysis of GWI web page structure (Corporate Profile page)

Shapes enclosed by straight line are blocks defined by ‘DIV’.
Shapes enclosed by dotted line are defined by ‘TABLE’.

We can describe this web-page ;

Blocks in top level are designed by ‘BOX laout’., ‘header-block’ and ‘bofy-block’.
But in ‘body-block’, the main structure is designed in ‘TABLE layout’. Why???



Boxes with dotted rectangle are designed by Table Layout.

```
<body>
  <div id="header-block">
    <a href="http://www.gwiguyana.com/?q=node/32" id="circle-a" onmouseover="showPopup('water conservation knowledge base')" onMouseout="hidePopup()">
      <img src="/themes/gwi-liquid-blue/images/circle-a.png" />
    </a><!-- water conservation knowledge base-->
    <a href="http://www.gwiguyana.com/?q=node/30" id="circle-b" onmouseover="showPopup('water distribution operations')" onMouseout="hidePopup()">
      <img src="/themes/gwi-liquid-blue/images/circle-b.png" />
    </a><!-- water distribution operations -->
    <a href="http://www.gwiguyana.com/?q=announcements" id="circle-c" onmouseover="showPopup('customer service; application, connection, billing')" onMouseout="hidePopup()">
      <img src="/themes/gwi-liquid-blue/images/circle-c.png" />
    </a><!-- customer service; application, connection, billing,.. -->
    <a href="http://www.gwiguyana.com/?q=node/31" id="circle-d" onmouseover="showPopup('water resource and treatment')" onMouseout="hidePopup()">
      <img src="/themes/gwi-liquid-blue/images/circle-d.png" />
    </a><!-- water resource and treatment -->

    <div id="site-name">
      <h1><a href="/" title="Guyana Water Incorporated"><span>Guyana Water Incorporated</span></a></h1>
    </div>
    <div id="primary-menu">
      <ul class="menu"><li class="first menu-1-1-2"><a href="/?q=node/27" class="menu-1-1-2">Newsroom | Public Education</a></li>
        <li class="menu-1-2-2-active"><a href="/?q=about" class="menu-1-2-2-active active">Corporate Profile</a></li>
        <li class="menu-1-3-2"><a href="/?q=announcements" title="Contact us on 227-8701, 227-8703 or 227-8704 " class="menu-1-3-2">Customer Services</a></li>
        <li class="menu-1-4-2"><a href="/?q=information" class="menu-1-4-2">Partnering for Progress</a></li>
        <li class="last menu-1-5-2"><a href="/?q=node/43" class="menu-1-5-2">Vacancies &amp; Invitations for Bids</a></li>
      </ul>
      <!--                          -->
    </div>

  </div><!--header-block-->


  <div id="body-block">
    <table cellpadding="0" cellspacing="0">
      <tr>
        <td valign="top" width="180">
          <div id="sidebar-left" class="sidebar">
            <div id="block-book-0" class="clear-block block block-book">
              <h2>Corporate Profile</h2>
            <div class="content">
              <ul class="menu">
                <li class="leaf"><a href="/?q=node/6">Annual Report</a></li>
                <li class="leaf"><a href="/?q=node/5">Corporate Profile</a></li>
                <li class="leaf"><a href="/?q=node/90">Customer Services Call Centre</a></li>
                <li class="leaf"><a href="/?q=node/89">Divisional Contact Numbers</a></li>
                <li class="leaf"><a href="/?q=node/7">History of GWI</a></li>
                <li class="leaf"><a href="/?q=node/29">Legislation</a></li>
                <li class="leaf"><a href="/?q=node/9">Locations</a></li>
                <li class="leaf"><a href="/?q=node/10">Related Links</a></li>
                <li class="leaf"><a href="/?q=node/8">Staff Directory</a></li>
              </ul>
            </div>
        </div>
        <div id="block-menu-2" class="clear-block block block-menu">
          <h2>Primary links</h2>
          <div class="content">
            <ul class="menu">
              <li class="collapsed"><a href="/?q=node/27">Newsroom | Public Education</a></li>
              <li class="leaf"><a href="/?q=about" class="active">Corporate Profile</a></li>
              <li class="leaf"><a href="/?q=announcements" title="Contact us on 227-8701, 227-8703 or 227-8704 ">Customer Services</a></li>
              <li class="leaf"><a href="/?q=information">Partnering for Progress</a></li>
              <li class="leaf"><a href="/?q=node/43">Vacancies &amp; Invitations for Bids</a></li>
            </ul>
          </div>
        </div>
        <div id="block-user-1" class="clear-block block block-user">
          <h2>Navigation</h2>
          <div class="content">
            <ul class="menu">
              <li class="leaf"><a href="/?q=book">Useful Information</a></li>
            </ul>
          </div>
        </div>
      </div>
    </td>
    <td valign="top" id="main">
      <div class="breadcrumb"><a href="/">Home</a>
```

```
        </div>
        <h2 class="title">Corporate Profile</h2>
        <div id="content">
          <div id="node-20" class="node">
            <div class="content">
              <div class="book-navigation">
                <ul class="menu">
                  <li class="leaf"><a href="/?q=node/6">Annual Report</a></li>
                  <li class="leaf"><a href="/?q=node/5">Corporate Profile</a></li>
                  <li class="leaf"><a href="/?q=node/90">Customer Services Call Centre</a></li>
                  <li class="leaf"><a href="/?q=node/89">Divisional Contact Numbers</a></li>
                  <li class="leaf"><a href="/?q=node/7">History of GWI</a></li>
                  <li class="leaf"><a href="/?q=node/29">Legislation</a></li>
                  <li class="leaf"><a href="/?q=node/9">Locations</a></li>
                  <li class="leaf"><a href="/?q=node/10">Related Links</a></li>
                  <li class="leaf"><a href="/?q=node/8">Staff Directory</a></li>
                </ul>
                <div class="page-links clear-block"><a href="/?q=node/6" class="page-next" title="Go to next page">Annual Report ›</a>
                </div>
              </div>
            </div>
            <div class="clear-block clear">
              <div class="meta">
              </div>
            </div>
          </div>
        </content>
      </td>
    </tr>
    <tr>
      <td colspan="3" id="footer">
       Corporate Complex Vllissengen Road and Church Street, BelAirPark, Georgetown, Guyana
       <br>
       Customer Services Call Centre (592) 227 8701 <strong>|</strong> Fax: (592) 227 8718  <strong>|</strong> Email: customercallcentre@gwi.gy
       <br>
       &copy; 2013
      </td>
    </tr>
```

```
        </table>
    </div><!--body-block-->
</body>
```

HTML form

HTML form is used to collect user input data from a web page.

Let’s make one very simple page with form and one very simple response page.

[Practice 20] HTML form → ex20.php, ex20b,php

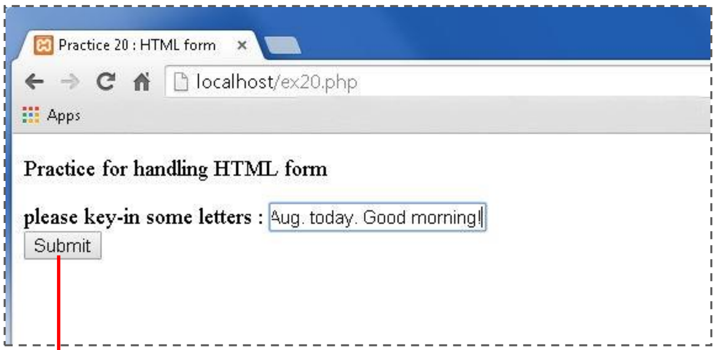
Make two programs shown below with name ‘ex20.php’,’ex20b.php’ and
access ‘ex20,php’ from browser.
input some letters in the input box and press submit button,

ex20.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Practice 20 : HTML form </title>
</head>
<body>
   <p>Practice for handling HTML form</p>
   <form action="ex20b.php" method="POST">
      please key-in some letters  :  <input type="text" name="inputField" /><br />
      <input type="submit" name="submit" />
   </form>
</body>
</html>
```

ex20b.php

```
?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Practice 20 : HTML form (response)</title>
</head>
<body>
   <p>Practice for handling HTML form (response to ex20.php)</p>
   You typed in and sent "<?php echo $_POST["inputField"]; ?> "<br />
</body>
</html>
```


Practice 20 : HTML form

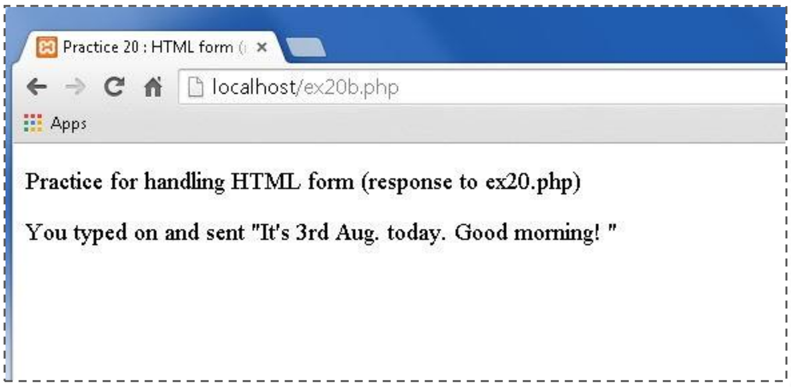
localhost/ex20.php

Apps

Practice for handling HTML form

please key-in some letters : Aug. today. Good morning!

Submit


Practice 20 : HTML form (

localhost/ex20b.php

Apps

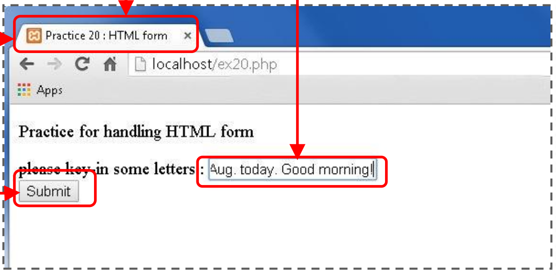
Practice for handling HTML form (response to ex20.php)

You typed on and sent "It's 3rd Aug. today. Good morning! "

Program analysis (ex20.php)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html>
3  <html xmlns="http://www.w3.org/1999/xhtml">
4  <head>
5    <title>Practice 20 : HTML form </title>
6  </head>
7  <body>
8    <p>Practice for handling HTML form</p>
9    <form action="ex20b.php" method="POST">
10       please key-in some letters  :  <input type="text" name="inputField" /><br />
11       <input type="submit" name="submit" />
12   </form>
13 </body>
14 </html>
```

1 – 3 3 top lines of set expression
4 – 6 HEAD tag
5 TITLE tag
7 – 13 BODY tag
8 P tag
9 - 12 FORM tag
‘action’ : program/document to be called
‘method’ : POST or GET


Practice 20 : HTML form

localhost/ex20.php

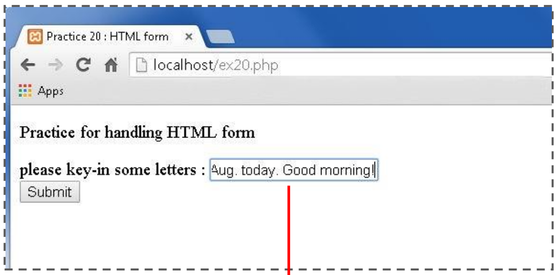
Apps

Practice for handling HTML form

please key-in some letters : Aug. today. Good morning!

Submit

10 INPUT tag
  ‘type’ : ‘text’ will show a text box
  ‘name’ : name of text box
    This name may be used in
    next PHP program.
11 INPUT tag
  ‘type’ : ‘submit’ will show a button
  ‘name’ : name of button

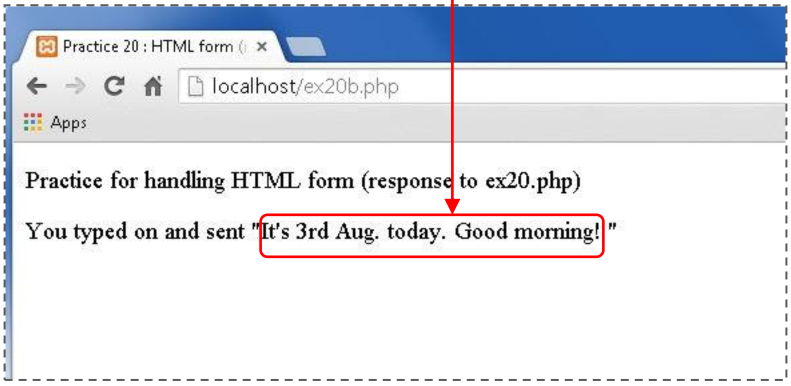


Practice for handling HTML form

please key-in some letters : Aug. today. Good morning!

Program analysis (ex20b.php)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html>
3  <html xmlns="http://www.w3.org/1999/xhtml">
4  <head>
5      <title>Practice 20 : HTML form (response)</title>
6  </head>
7  <body>
8      <p>Practice for handling HTML form (response to ex20.php)</p>
9      You typed in and sent "<?php echo $_POST['inputField"]; ?> "<br />
10 </body>
11 </html>
```



Practice for handling HTML form (response to ex20.php)

You typed on and sent "It's 3rd Aug. today. Good morning! "

Line 9: $_POST : one of super globals.
It ‘s an associative array of variables passes to the current script via HTTP POST method.

‘echo’ statement of this line will show the value of “inputField” in ‘$_POST’, “inputField” in ‘$_POST’ has the value of what you have input into the field in the previous screen ‘ex20.php’.

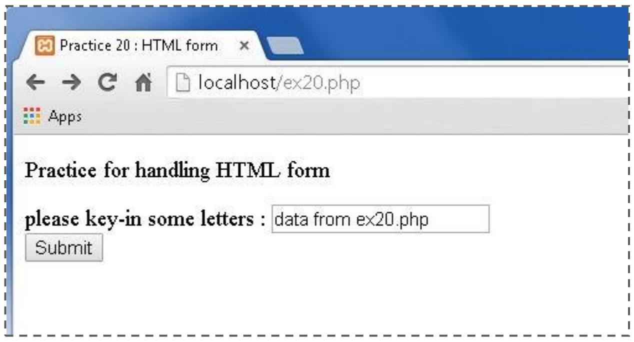
GET and POST

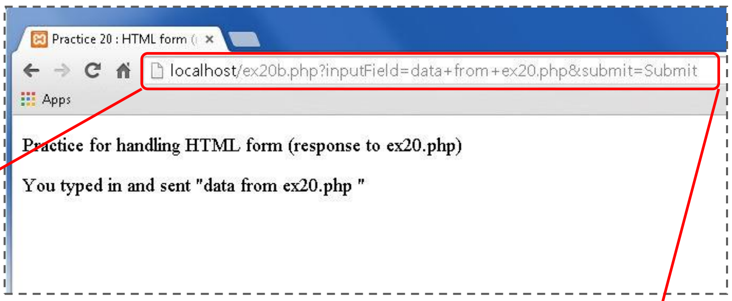
In the form defined in ‘ex20.php’, you see ‘POST’ ‘method’ attribute.
What’s POST? And what’s GET? What are the differences?

There are two types of how a browser can send data to a web server.
GET method and POST method.

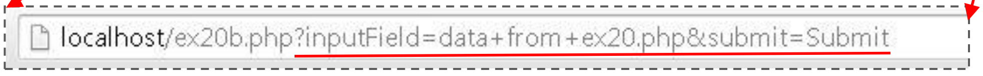
GET method will send the data from a form appended to the page request (URL strings). The page request and the data is separated by ‘?’.

ex : if you send data in ‘ex20.php’ using GET method, you will see the window shown below:


Practice 20 : HTML form
localhost/ex20.php
Apps
Practice for handling HTML form
please key-in some letters : data from ex20.php
Submit


Practice 20 : HTML form (
localhost/ex20b.php?inputField=data+from+ex20.php&submit=Submit
Apps
Practice for handling HTML form (response to ex20.php)
You typed in and sent "data from ex20.php "

page request (URL strings)


localhost/ex20b.php?inputField=data+from+ex20.php&submit=Submit

Changes in source files : ex20.php

```
( Lines from 1 t0 8, they are the same as that for POST method )
9    <form action="ex20b.php" method="GET">
10       please key-in some letters  :  <input type="text" name="inputField" /><br />
11      <input type="submit" name="submit" />
…….
```

ex20b.php

```
( Lines from 1 t0 8, they are the same as that for POST method )
9     You typed in and sent "<?php echo $_GET["inputField"]; ?> "<br />
10 </body>
11 </html>
```

Using GET method, all data given in the form may be shown in URL strings.
It sometime causes security problem. Password given in the form can be seen in URL strings.
So, in most of Web applications, we use POST method.
In POST method, data in the form are given in $_POST and not shown in URL strings.

Here you may do one practice where you will see how dangerous to use GET method.

[Practice 21]

Make a program shown below with name 'ex21.php' and 'ex21b.php'.
'ex21.php' will show one form to input ID and password. (For password field, 'type' attribute is defined as 'password', which hide data in the field.)
'ex21b.php' will show ID and password you have given and sent. (Usually, the password must not be shown, this is an example)

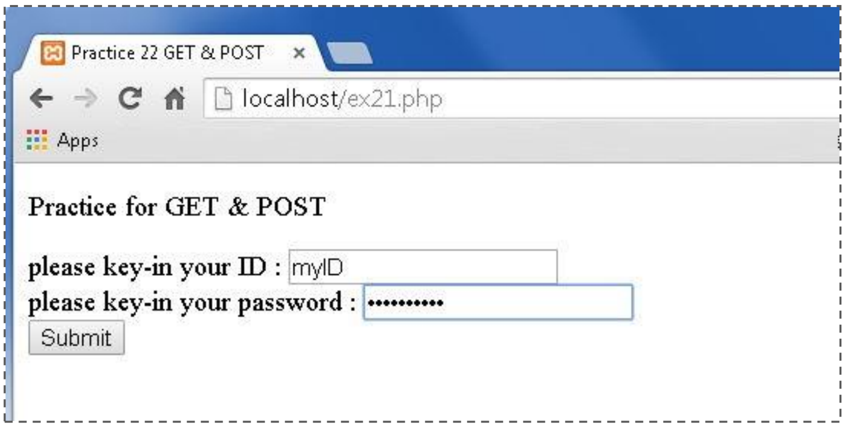
ex21.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 21  GET & POST</title>
</head>
<body>
  <p>Practice for GET & POST</p>
  <form action="ex21b.php" method="GET">
    please key-in your ID  :  <input type="text" name="yourID" /><br />
    please key-in your password  :  <input type="password" name="yourPSWD" /><br />
    <input type="submit" name="submit" />
  </form>
</body>
</html>
```

ex21b.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 21 : HTML form (response)</title>
</head>
<body>
  <p>Practice for handling HTML form (response to ex21.php)</p>
  You typed in and sent "<?php echo $_GET["yourID"]; ?> "<br />
  Your password is : "<?php echo $_GET["yourPSWD"]; ?> "
</body>
</html>
```

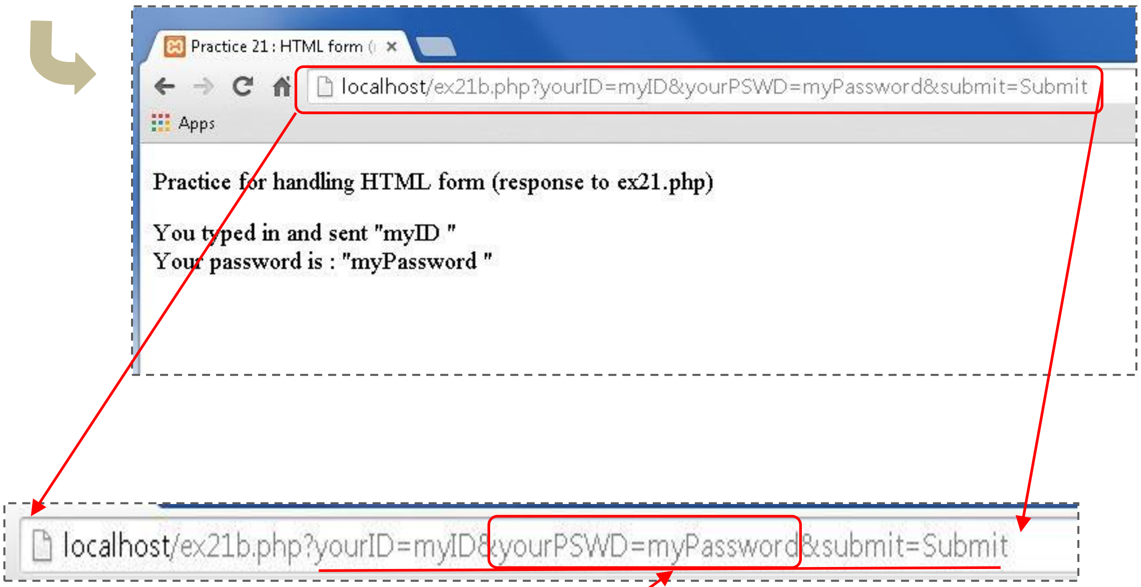
You will see screens shown below;


Practice 22 GET & POST

localhost/ex21.php

Practice for GET & POST

please key-in your ID : myID
please key-in your password : ••••••••••
Submit


Practice 21 : HTML form (

localhost/ex21b.php?yourID=myID&yourPSWD=myPassword&submit=Submit

Practice for handling HTML form (response to ex21.php)

You typed in and sent "myID "
Your password is : "myPassword "

localhost/ex21b.php?yourID=myID&yourPSWD=myPassword&submit=Submit

Page request (URL strings) shows your password in plain text!

You may understand how dangerous it is to send sensitive data from a HTML form with GET method.

input elements

1.Text input

2.Radio button input

3.Submit button

select element (drop-down list)

textarea element

button element

datalist element (new element in HTML5)

keygen element (new element in HTML5)

In details, see
http://www.w3schools.com/html/html_form_elements.asp

XSS (Cross Site Scripting) vulnerability

What's XSS attacks?

XSS attacks are type of injection, where malicious scripts are injected into a web page.

We'll try some example of XSS attack here:

Notes: Some browser like Google Chrome has anti-XSS filter itself.
As for Chrome, to disable this filter, you can start Chrome from command prompt with the argument shown below;

```
chrome.exe –disable-xss-auditor
```

(two hyphens + 'disable-xss-auditor')

[Practice 22] XSS attack example

Make a program shown below with name 'ex22.php' and 'ex22b.php'.

ex22.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 22  GET & POST</title>
</head>
<body>
  <p>Practice for GET & POST</p>
  <form action="ex22b.php" method="POST">
    please key-in your ID  :  <input type="text" name="yourID" /><br />
    <input type="submit" name="submit" />
  </form>
</body>
</html>
```

ex22b.php

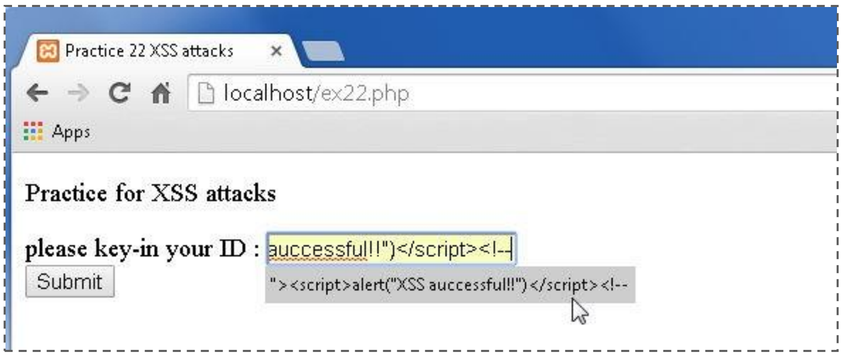
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 22: HTML form (response)</title>
</head>
<body>
  <p>Practice for handling HTML form (response to ex21.php)</p>
  You typed in and sent "<?php echo $_POST["yourID"]; ?> "<br />
</body>
</html>
```

Here is an example for XSS attack which is only showing one alert window on the browser.

After accessing to 'localhost/ex22.php', type in following letters into input box.

"><script>alert('XSS successful!!!');</script><!--

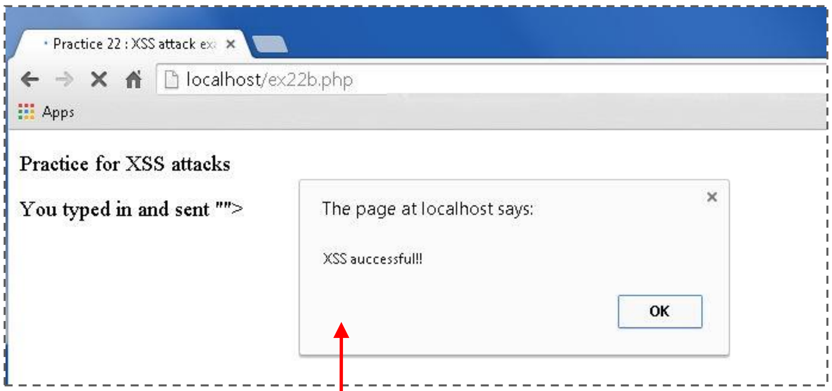
What comes out on your browser?



Practice for XSS attacks

please key-in your ID : auccessful!!")</script><!--

"><script>alert("XSS auccessful!!")</script><!--



Practice for XSS attacks

You typed in and sent "">

The page at localhost says:

XSS successful!!

An alert window, which is not defined in 'ex22b,php', is shown on your browser.

In actual cases, attackers will show some malicious HTML with some buttons which might lead you to some malicious site.
This is a very simple example of XSS attack.

Don't do it on internet Web pages, it may result in committing a crime.
But you must check Web pages in GWI site before it might be taken over by hackers.

(If you use Google Chrome, it has anti XSS filter as default. To bypass this filter on Chrome, you can open Chrome from the command prompt window as follows;)

```
>cd  c:/program files (x86)/Google/Chrome/Application
>chrome  --args  --disable-xss-auditor
```

How to guard your web page from XSS attacks

General concept: validation for input data, sanitizing output data

1. Input validation & sanitizing

First step to prevent XSS attacks is 'Input Validation'.
For example;
Input fields for mail address must only have valid letters for mail address.
Input fields for height must only numerics, comma or period.

In PHP, we have one function shown below:
filter_var() function
see http://php.net/manual/en/function.filter-var.php

```
ex)   $ip =”http://www.gwi.gy”;
      if( ! filter_var( $ip, FILTER_VALIDATE_URL ) === false ) {
          //  some codes in case of valid URL
      }
```

Sanitizing : to convert/remove illegal letters in input data
see http://php.net/manual/en/filter.filters.sanitize.php

```
ex)   $ip = “http://www.gwi+%#.gy”;
      $url = filter_var( $ip, FILTER_SANITIZE_URL );
```

[Practice 23] Validation check by using filter_var()

Make a program shown below with name 'ex23.php' and 'ex23b.php'.
ex23.php

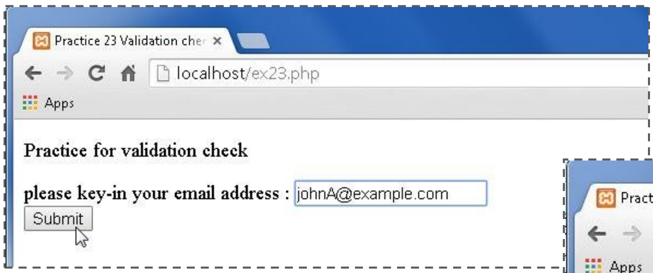
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 23  Validation check</title>
</head>
<body>
  <p>Practice for validation check</p>
  <form action="ex23b.php" method="POST">
    please key-in your email address  : <input type="text" name="yourEmail" /><br />
    <input type="submit" name="submit" />
  </form>
</body>
</html>
```

Try to input illegal characters(% # ! and so on).

[Practice 23] (continued)

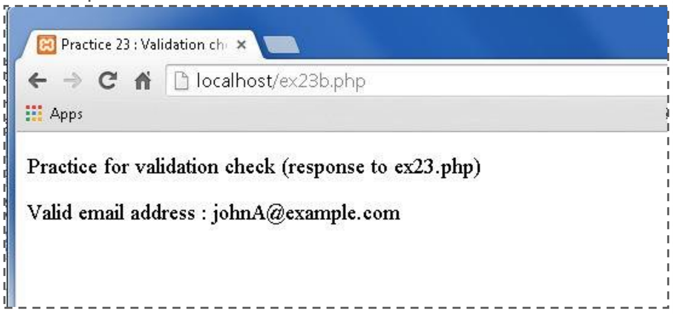
ex23b.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Practice 23 : Validation check</title></head>
<body>
  <p>Practice for validation check (response to ex23.php)</p>
  <? php
    $email = filter_var($_POST[“yourEmail”], FILTER_SANITIZE_EMAIL );  // sanitizing
    if (   ( $email == ($_POST[“yourEmail”] )  )
            &&
            ( ! filter_var( $email, FILTER_VALIDATE_EMAIL ) === false )
       ) {        // validating
           echo “Valid email address :  $email<br />”;
    } else {
            echo “Invalid email address : “ . htmlspecialchars($_POST[“yourEmail”]   ) . “<br />”;
    }
  ?>
```



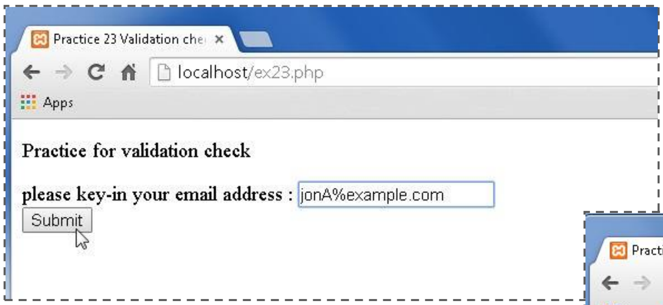
Practice for validation check

please key-in your email address : johnA@example.com



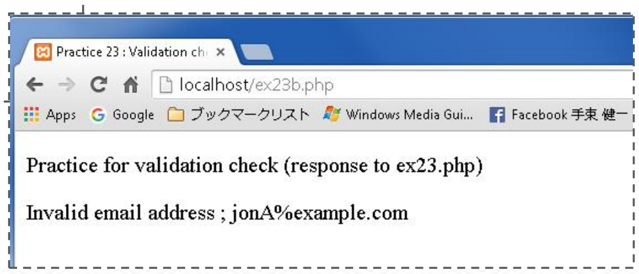
Practice for validation check (response to ex23.php)

Valid email address : johnA@example.com



Practice for validation check

please key-in your email address : jonA%example.com


Practice 23 : Validation ch
localhost/ex23b.php
Apps Google ブックマークリスト Windows Media Gui... Facebook 手束 健一

Practice for validation check (response to ex23.php)

Invalid email address ; jonA%example.com

2.Escaping

[Practice 24] escaping in PHP

Make a program shown below with name ‘ex24.php’ and access to it on browser

ex24.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Practice 24 Escaping in PHP</title>
</head>
<body>
   <p>Practice for escaping in PHP</p>
   <?php  $var  =  ‘my mother’s book’;
              print $var;
   ?>
</body>
</html>
```

You may have got an error message on your browser.

What did PHP tell you in the error message?

```
syntax error, unexpected 's' (T_STRING)
```

It means “an apostrophe after ‘mother’ is recognized as closing apostrophe, where “my mother” may make a text block, and after one text block, PHP expects a dot(.) or a semicolon(;), but PHP found letter ‘s’ instead.

We need to let PHP recognize this apostrophe after ‘mother’ as a letter, not as a closing sign of a text block. For this purpose, we need ‘Escaping mechanism’ in PHP.

Back slash(\) is used for this purpose.
Modify line 9 as shown below;

```
<?php  $var  =  ‘my mother\’s book’;
```

Try to access ex24.php again.

3.htmlspecialchars() function

htmlspecialchars() function converts some predefined characters to HTML entities.

Predefined characters and their HTML entities;

	explanation	HTML entities
&	ampersand	&amp;
"	double quote	&quot;
'	single quote	&#039;
<	less than	&lt;
>	greater than	&gt;

Syntax

htmlspecialchars(*string*, *flag*, *character-set*, *double-encode*)

string : target string to convert

flag : specifies how to handle quotes, invalid encoding and so on.
See in details in
http://php.net/manual/en/function.htmlspecialchars.php

character-set ; default value depends on *default_charset* in php.ini

double-encode : true : encode existing html entities (default)
false : not encode existing html entities

XSS attacks generally plant some harmful java scripts into web sites by using non-guarded HTML definition, and in most cases, they send tags which include '<' , '>' or quotes, so it's basically important to convert these harmful characters into HTML entities when responding HTML code to browser.

htmlspecialchars() should be used for output data on HTML just before sending response to browser.

[Practice 25] htmlspecialchars()

Make a program shown below with name ‘ex25.php’, ‘ex25b.php’ and key-in including special characters ‘<’,’>’,’&’ and quotes.

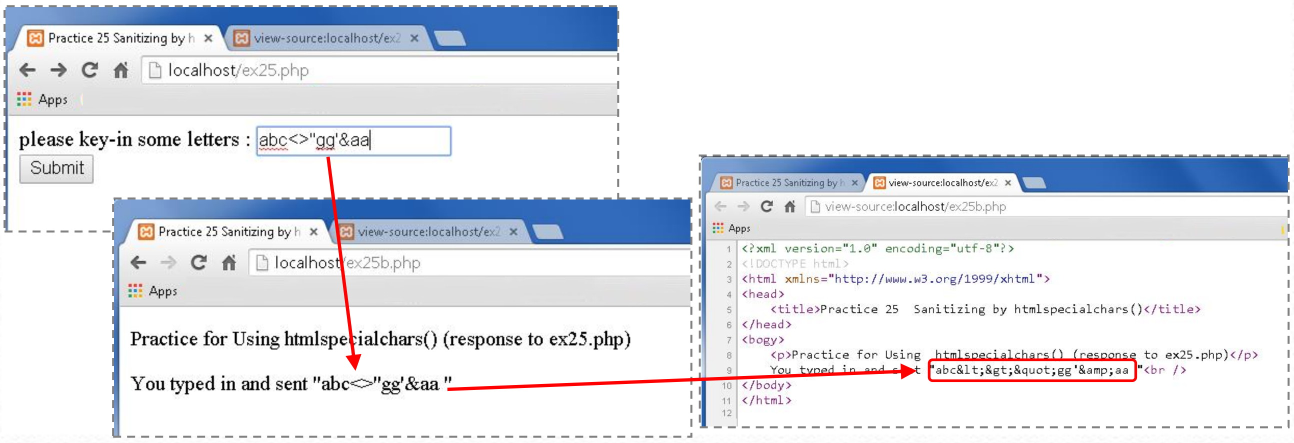
ex25.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Practice 25  Sanitizing by htmlspecialchars()</title>
</head>
<body>
   <form action="ex25b.php" method="POST">
      please key-in some letters  :  <input type="text" name="yourInput" /><br />
      <input type="submit" name="submit" />
   </form>
</body>
</html>
```

ex25b.php

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Practice 25:Using  htmlspecialchars() (response)</title>
</head>
<body>
   <p>Practice for Using  htmlspecialchars() (response to ex25.php)</p>
   You typed in and sent "<?php echo htmlspecialchars( $_POST["yourInput"] ); ?> "<br />
</body>
</html>
```

After getting response on your browser, you can check HTML source to confirm how htmlspecialchars() works, how HTML entities are used.


Practice 25 Sanitizing by h
view-source:localhost/ex2
localhost/ex25.php
Apps
Submit
localhost/ex25b.php
view-source:localhost/ex25b.php

please key-in some letters : abc<>"gg'&aa

Practice for Using htmlspecialchars() (response to ex25.php)

You typed in and sent "abc<>"gg'&aa "

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Practice 25  Sanitizing by htmlspecialchars()</title>
</head>
<bogy>
    <p>Practice for Using  htmlspecialchars() (response to ex25.php)</p>
    You typed in and sent "abc&lt;&gt;&quot;gg'&amp;aa "<br />
</body>
</html>
```

PHP file handling

When we need to handle files, file handling functions will help us.
There are so many functions. We'll select some functions assuming 5 cases shown below;

1. We read whole contents in a file into some variable and process them.
2. We read a file line-by-line.
3. We create a file, update a file.
4. We read INI format file.
   Using INI format, we'll set up application environment.
5. We upload some images to the web server.

1. Reading whole contents

1-1 file() : This function reads whole of a file into an array.

Each array element has a line from the file.
New line code will still in the element.

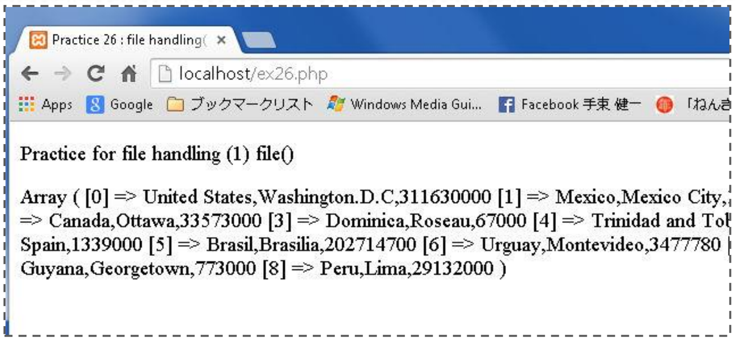
[Practice 26] File handling (1) file() function

Make one text file named 'nations.txt' with data shown below;

```
United States,Washington.D.C,311630000
Mexico,Mexico City,112322767
Canada,Ottawa,33573000
Dominica,Roseau,67000
Trinidad and Tobago,Port of Spain,1339000
Brasil,Brasilia,202714700
Urguay,Montevideo,3477780
Peru,Lima,29132000
```

ex26.php

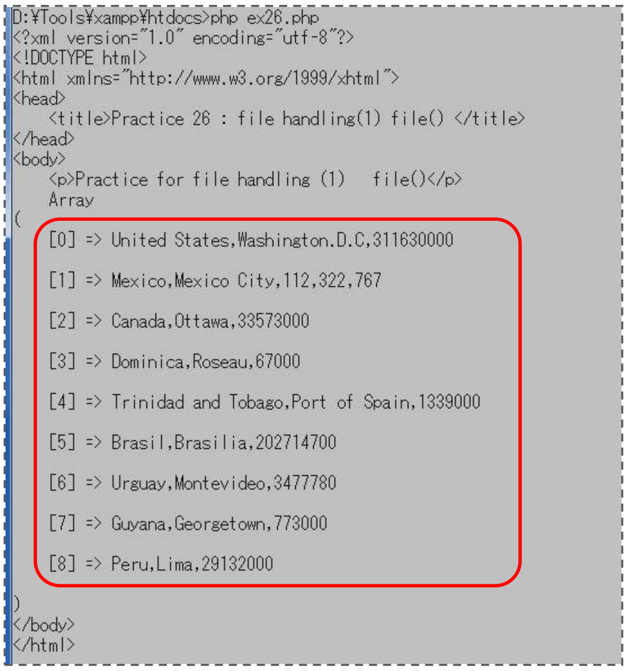
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Practice 26 : file handling(1) file() </title>
</head>
<body>
  <p>Practice for file handling (1) file()</p>
  <?php print_r( file( 'nations.txt' ) ); ?>
</body>
</html>
```



```
Practice 26 : file handling(  ×
localhost/ex26.php
Apps  Google  ブックマークリスト  Windows Media Gui...  Facebook 手束 健一  「ねんき

Practice for file handling (1) file()

Array ( [0] => United States,Washington.D.C,311630000 [1] => Mexico,Mexico City,
=> Canada,Ottawa,33573000 [3] => Dominica,Roseau,67000 [4] => Trinidad and To
Spain,1339000 [5] => Brasil,Brasilia,202714700 [6] => Urguay,Montevideo,3477780
Guyana,Georgetown,773000 [8] => Peru,Lima,29132000 )
```

You may see such a list shown above, in spite that file() function keeps newline in result array. Why?

It's because browser will neglect newline(x0A0D) code.
When we want to put newline in the list on browser, we must follow HTML manner, which is <br> HTML tag, or we print out the list not through browser, but through another media like a prompt window.



```
D:¥Tools¥xampp¥htdocs>php ex26.php
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Practice 26 : file handling(1) file() </title>
</head>
<body>
    <p>Practice for file handling (1)  file()</p>
    Array
(
    [0] => United States,Washington.D.C,311630000

    [1] => Mexico,Mexico City,112,322,767

    [2] => Canada,Ottawa,33573000

    [3] => Dominica,Roseau,67000

    [4] => Trinidad and Tobago,Port of Spain,1339000

    [5] => Brasil,Brasilia,202714700

    [6] => Urguay,Montevideo,3477780

    [7] => Guyana,Georgetown,773000

    [8] => Peru,Lima,29132000

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

ex26.php executed on a prompt window screen

1-2 fopen() – fread() – fclose()

fopen() function will open a file.
fread() function wll get whole contents of the file.
fclose() function will close the file.

fopen()

syntax :

file_handle = fopen(*file_path*, *mode*) [or die(*message*)];

mode : See a table on next page. optional

Mode definition for fopen()

mode	description	file pointer
r	Open a file for read only.	at the beginning
w	Open a file for write only. Erase the contents if the file exists.	at the beginning
a	Open a file for write only. Existing data is preserved. If the file doesn't exist, new file will be created.	at the end
x	Open a file for write only. If the file already exists, it will fail.	at the beginning
r+	Open a file for read/write.	at the beginning
w+	Open a file for read/write. Erase the contents if the file exists.	at the beginning
a+	Open a file for read/write. Existing data is preserved. If the file doesn't exist, new file will be created.	at the end
x+	Open a file for read/write. If the file already exists, it will fail.	at the begining
b	Open binary file	

ex) fopen('text.txt', 'a+');

If a file 'text.txt' exists in current folder,

open 'text.txt' and set file pointer at the end of file to add data,

else if it doesn't exist,

create 'text.txt' in current folder and set file pointer at the beginning.

ex) fopen('test.jpg', 'rb') or die("Couldn't open test.jpg");

Open a jpeg file in binary mode. If failed, program will stop with the message given in 'die' option.

fclose()

syntax :

fclose(*file_handle*);

```
ex)  $fileHandle  =  fopen(  'example.txt', 'r' );
fclose ( $fileHandle );
```

fread() (this function is more suitable for processing binary files) syntax :

fread(*file_handle, length*);

length : maximum number of bytes to be read

Note: fread() will read maximum 8 KB at one time.
To read a file with more than 8KB, we need to repeat to read.

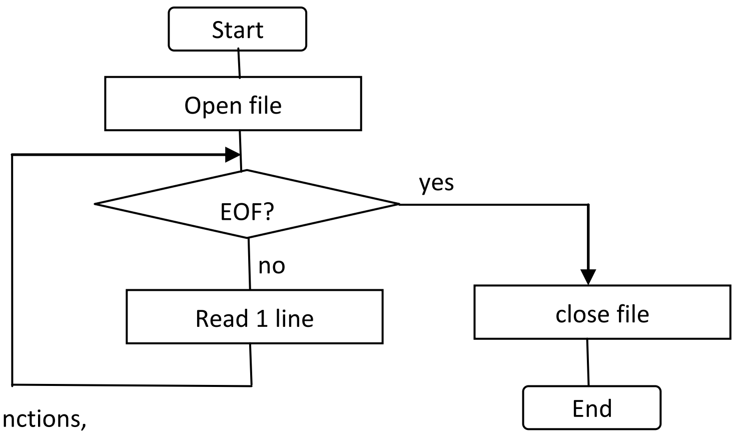
ex)

```
$filename    = 'img/xampp.ico';
$filehandle  = fopen( $filename, 'rb' ); // open binary file
$contents    = fread( $filehandle, filesize( $filename ) );
```

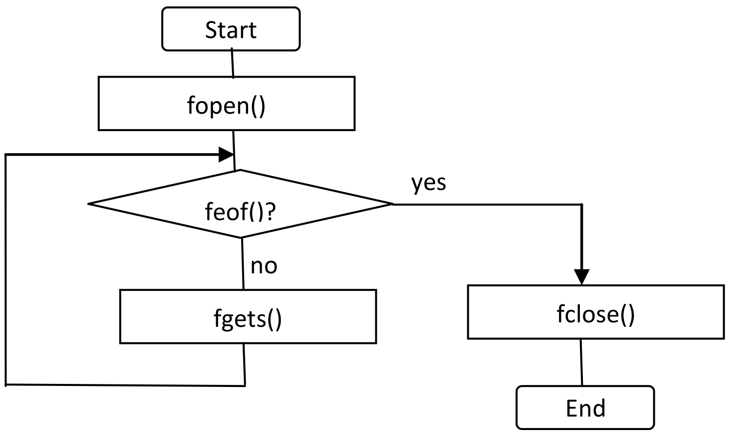
read a file 'xampp.ico' in icon format from 'img' folder in binary mode

2. Reading file line-by-line

For text files, it's often better to process line-by-line.
For such cases, fgets() function is used, and process is shown below;



In PHP functions,



2-1 feof() returns true if file pointer reaches end of file.

syntax :

```
$boolean = feof( file_handle );
```

2-2 fgets() get one line from text file

Syntax :

```
$string/$boolean = fgets( file_handle, length );
```

file_handle : required
file handle assigned to the file when opening the file

length : optional
if omitted, one line text or text up to EOF.
Default size is 1024 bytes.

[Practice 27] File handling (2) fopen()/fgets()/fclose()

Using 'nations.txt' (see Practice26), make a program named 'ex27.php' which process the text file line-by-line and show the content on a browser.

3. Creating a file, updating a file

fopen() gives 'w' or 'x' option to create a file.
After creating a new file, fwrite() function will write to the file.
For update a file, we need to read whole lines from the file, modify target lines to be updated and make a new file with whole lines again.

3-1 fwrite()

fwrite() is a function to write some data to file.

Syntax :

```
fwrite( file_handle, data );
```

[Practice 28] File handling (3) add some records in a file

Using 'nations.txt', make a program named 'ex28.txt' to meet a request shown below;

add a line(record) 'Guyana,Georgetown,780000'

[Practice 29] File handling (4) modify some records in a file

Using 'nations.txt', make a program named 'ex29.txt' to meet a request shown below;

Modify a line(record) 'Dominica,Roseau,67000'
to 'Dominican Republic,Santo Domingo,10090000'

After modification, show nations list on your browser.

4. INI format

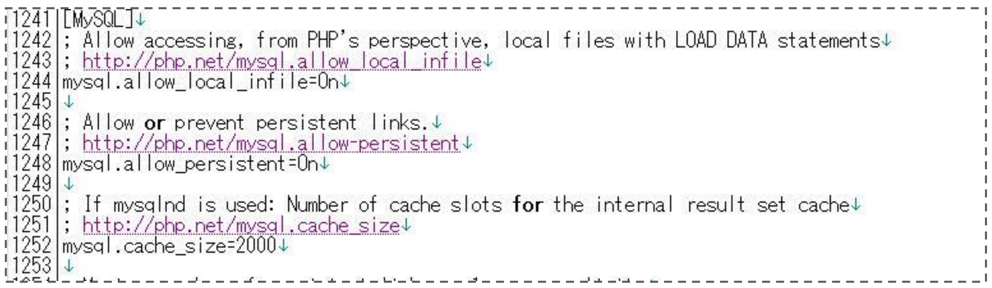
INI file format is a simple text file to use as a configuration for some software.

It is composed of 'sections', 'properties' and 'values'.

Syntax :

[*section_name*]
; comments
property_name = *property_value*

Here's one example, php.ini (PHP configuration file);



```
[MySQL]
; Allow accessing, from PHP's perspective, local files with LOAD DATA statements
; http://php.net/mysql.allow_local_infile
mysql.allow_local_infile=On

; Allow or prevent persistent links.
; http://php.net/mysql.allow-persistent
mysql.allow_persistent=On

; If mysqlnd is used: Number of cache slots for the internal result set cache
; http://php.net/mysql.cache_size
mysql.cache_size=2000

```

example shown above is a MySQL part of php.ini.

4-1 parse_ini_file()

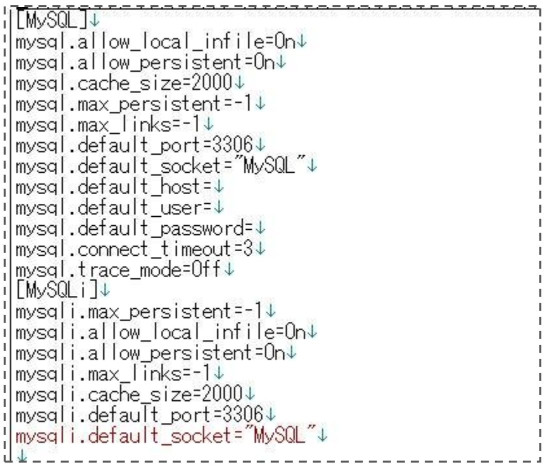
parse_ini_file() reads ini format file into array.

Syntax :

parse_ini_file(*file_path, process_section*)

file_path	:	path to ini file
process_section	:	true or false. If true, sections are included in array. If false, setions are not included. Default is false.

ex) here's a part of php.ini ;



```
[MySQL]↓
mysql.allow_local_infile=On↓
mysql.allow_persistent=On↓
mysql.cache_size=2000↓
mysql.max_persistent=-1↓
mysql.max_links=-1↓
mysql.default_port=3306↓
mysql.default_socket="MySQL"↓
mysql.default_host=↓
mysql.default_user=↓
mysql.default_password=↓
mysql.connect_timeout=3↓
mysql.trace_mode=Off↓
[MySQLi]↓
mysqli.max_persistent=-1↓
mysqli.allow_local_infile=On↓
mysqli.allow_persistent=On↓
mysqli.max_links=-1↓
mysqli.cache_size=2000↓
mysqli.default_port=3306↓
mysqli.default_socket="MySQL"↓
↓
```

Parse_ini_file('php.ini', true) will make an array shown below;

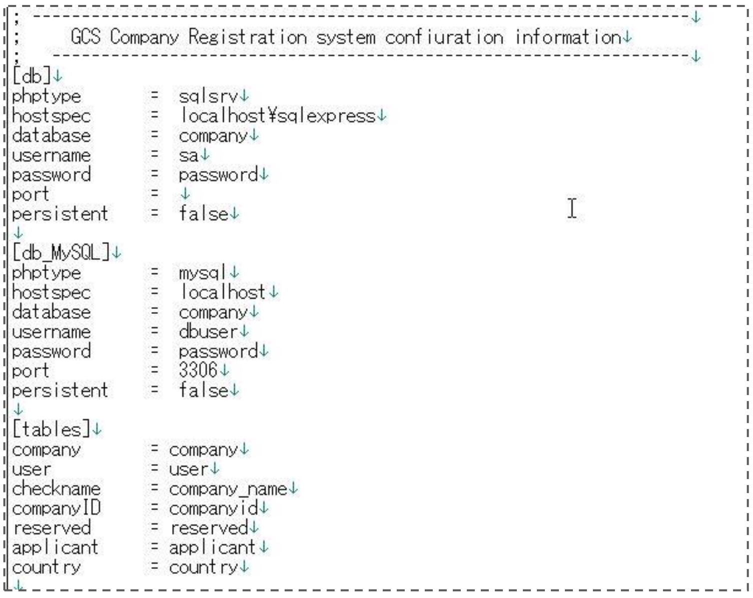
```
Array {
        [MySQL] => Array {
                        [mysql.allow_local_inifile] => On
                        [mysql.allow_persistant] => On
                        [mysql.cache_size] => 2000
                        [mysql.max?persistant] => -1
                        [mysql.default_port] => 3306
                        [mysql. default_socket] => MySQL
                }                   ……
        [MySQLi] => Array {
                        [mysql.max?persistant] => -1
                        [mysql.allow_local_inifile] => On
                        [mysql.allow_persistant] => On
                        [mysql.max_links] => -1
                        [mysql.cache_size] => 2000
                                    …….
                }
}
```

Parse_ini_file(‘php.ini’, false) will make an array shown below;

```
Array {
      [mysql.allow_local_inifile] => On
      [mysql.allow_persistant] => On
      [mysql.cache_size] => 2000
      [mysql.max?persistant] => -1
      [mysql.default_port] => 3306
      [mysql. default_socket] => MySQL
                    …….
      [mysql.max?persistant] => -1
      [mysql.allow_local_inifile] => On
      [mysql.allow_persistant] => On
      [mysql.max_links] => -1
      [mysql.cache_size] => 2000
                    …….
}
```

As you see, as for parse_ini_file(…., false) may cause some problem with same property names in different sections.

ex) here’s one example I’ve made for one organization ;



```
; -------------------------------------------------------------------
;     GCS Company Registration system confiuration information
; -------------------------------------------------------------------
[db]
phptype        =  sqlsrv
hostspec       =  localhost¥sqlexpress
database       =  company
username       =  sa
password       =  password
port           =
persistent     =  false

[db_MySQL]
phptype        =  mysql
hostspec       =  localhost
database       =  company
username       =  dbuser
password       =  password
port           =  3306
persistent     =  false

[tables]
company        = company
user           = user
checkname      = company_name
companyID      = companyid
reserved       = reserved
applicant      = applicant
country        = country
```

[db] section has environment values for MS SQL Server.
[db_MySQL] section has environment values for MySQL.
[tables] section has table names for both Database.

[Practice 30] File handling (5) use of INI format file

What are advantages to define and use such INI format file shown above in an application system? Discuss it with each other.

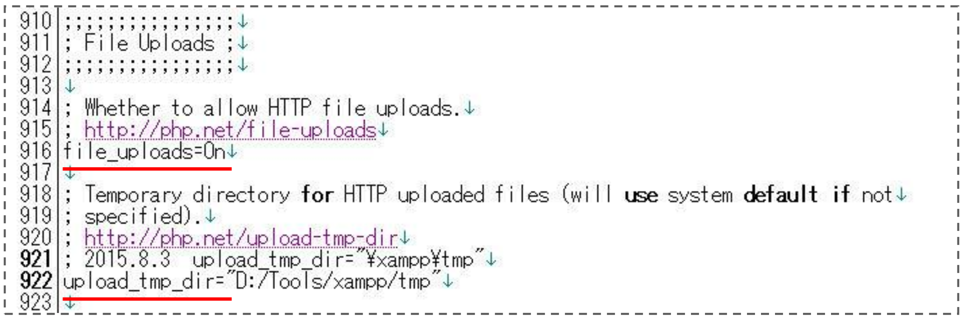
5. Upload files to web server

There are 3 points to check for uploading files to web server.

1) Upload configuration defined in php.ini
2) Definition of a form to select target files
3) PHP process to receive target files on the server

5-1 Configuration in php.ini

In php.ini, there’s a definition for uploading files configuration shown below;



```
;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
; http://php.net/file-uploads
file_uploads=On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
; http://php.net/upload-tmp-dir
; 2015.8.3  upload_tmp_dir="¥xampp¥tmp"
upload_tmp_dir="D:/Tools/xampp/tmp"

```

1) “file_uploads” property : it’s value should be ‘on’

2) “upload_tmp_dir” property ; it indicates temporary folder.
Usually we use ‘tmp’ folder under ‘xampp’.

5-2 Form to select target files

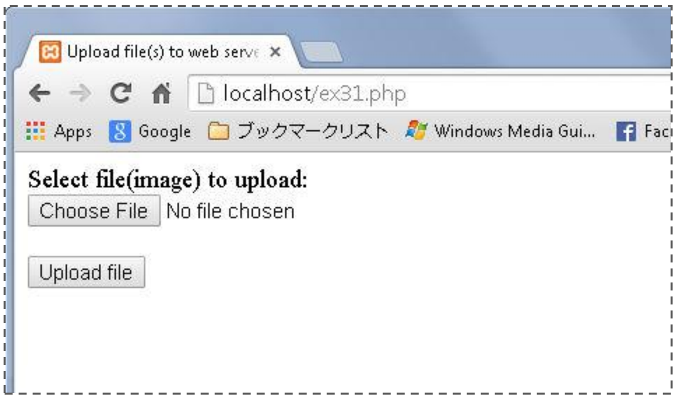
Sample :

[Practice 31] Upload files (1) form to select target

Make a program shown below with name ‘ex31.php’

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Upload file(s) to web server</title>
</head>
<body>
  <form action="ex31b.php" method="post" enctype="multipart/form-data">
    Select file(image) to upload:<br />
    <input type="file" name="targetFile"  /><br /><br />
    <input type="submit" value="Upload file" name="submit" />
  </form>
</body>
</html>
```

When you access to this program, you'll see a screen shown below;



Upload file(s) to web serv

localhost/ex31.php

Select file(image) to upload:

Choose File No file chosen

Upload file

In "ex31.php", these parts with red under line shown above are new factors. "enctype" attribute and 'type="file"' attribute value.

"enctype" attribute :

```
<form action="ex31b.php" method="post" enctype="multipart/form-data">
```

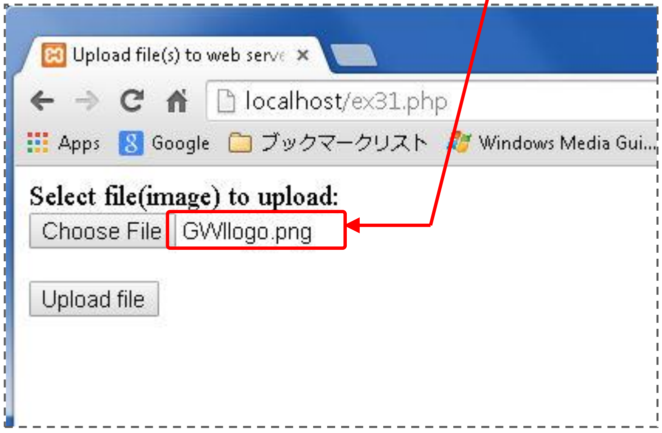
When we upload files to web server, this 'enctype' attribute must have value 'multipart/form-data' and 'method' must be 'post'.

"type" attribute :

```
<input type="file" name="targetFile" />
```

To upload files, 'input' attribute value for files must be 'file'.
"input='type'" attribute will show a button and some message.
(see screen shot shown above)
Pushing 'Choose File' button will pop up 'File Select control' and you can select files to upload.

After you selected a file to upload, you'll see the file name on the screen.



Upload file(s) to web serv

localhost/ex31.php

Select file(image) to upload:

Choose File GWIlogo.png

Upload file

5-3 Receiving target file(s) on the server

Last point to check for uploading files is the process after pressing the ‘submit’ button on the screen in Practice 31.

As you see in ‘php31.php’, it’s a process in ‘php31b.php’.

[Practice 32] Upload files (2) uploading files

Make a program shown below with name ‘ex31b.php’

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Upload file(s) to web server</title>
</head>
<body>
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["targetFile"]["name"]);
if(isset($_POST["submit"])) {
   if (move_uploaded_file($_FILES["targetFile"]["tmp_name"], $target_file)) {
      echo "The file ". basename( $_FILES["targetFile"]["name"]). " has been uploaded.<br />";
   } else {
      echo "Sorry, there was an error uploading your file.<br />";
   }
}
?>
</body>
</html>
```

Notes on above source :

```
$target_dir = "uploads/";
```

“uploads/” : it means ‘relative folder from ‘DocRoot’ (= ‘<xamppRoot>/htdocs/uploads’, if you don’t change ‘DocRoot’ in ‘httpd.cong’.

```
$_FILES["targetFile"]["name"]
```

After you select file(s) in ‘File Select control’ and press ‘submit’ in ‘Practice 31’, all information of target file are set in global variable ‘$_FILE’.
“targetFile” is the attribute value of ‘name’ attribute. (See ex31.php)

$_FILE[“targetFile”][“name”] will have file name of the target file(s).

```
basename
```

basename() is a function to get file name without folder path.

basename(“c:/folderA/folder/fileC.exe”) will return “fileC.exe”.

```
isset($_POST["submit"])
```

$_POST keeps all data submitted from HTML form.
If ‘submit’ button was pressed, then the name and value of the button will be set in $_POST as an array such as “submit” => “Upload”. (See ex31.php)

So, in this case, $_POST[“submit”] has value “Upload” (means ‘not false’).

```
move_uploaded_file($_FILES["targetFile"]["tmp_name"], $target_file)
```

Mechanism to upload file(s) from client PC to web server is ;

1) Select target file(s) on client PC
2) Send target file(s) to temporary folder
3) From temporary folder, copy the file to target folder

In the process 2), target file(s) will be named with some special names, which we can access with the name $_FILES[“*<name_attribute>*”][“tmo_name”]

We don’t need to know where the temporary folder is. PHP knows it.
We don’t need to know what’s the temporary file name of the target file(s). PHP knows it.

move_uploaded_file(*temporary_file_name, target_file_path*)

This function will move target file(s) in temporary folder to the target folder with specified name.
If this function return true, target file(s) has been successfully uploaded.

[Practice 33] Upload files (3) uploading files

Using ‘ex31.php’ and ‘ex31b.php’, upload some file on your PC to your ‘xampp’ web server.
Don’t forget to make ‘uploads’ folder under ‘xampp/htdocs’ folder.

In these practices of this chapter, we don't care anything about target files.
How can we solve such request shown below?

1) Client wants multiple(more than one) files to upload at one time.
2) Client (or you can say a hacker) tries fake upload.
   (with fake extension, with too big size and so on)
3) Client doesn't want overwrite files if they already exist.
4) Client want to search only image files with extension .jpg, .jpeg, .gif, .png, doesn't want to include text files and other types in the list.

5-4 additional checks before upload

1) multiple upload

ex31.php

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Upload file(s) to web server</title>
</head>
<body>
   <form action="ex31b.php" method="post" enctype="multipart/form-data">
      Select file(image) to upload:<br />
      <input type="file" name="targetFile[]" multiple=”multiple” /><br /><br />
      <input type="submit" value="Upload file" name="submit" />
   </form>
</body>
</html>
```

[Practice 34] Upload files (4) uploading multiple files

Modify 'ex31.php' and 'ex31b.php' to upload multiple files on your PC to your 'xampp' web server.

2) check extension to prevent faked file injection

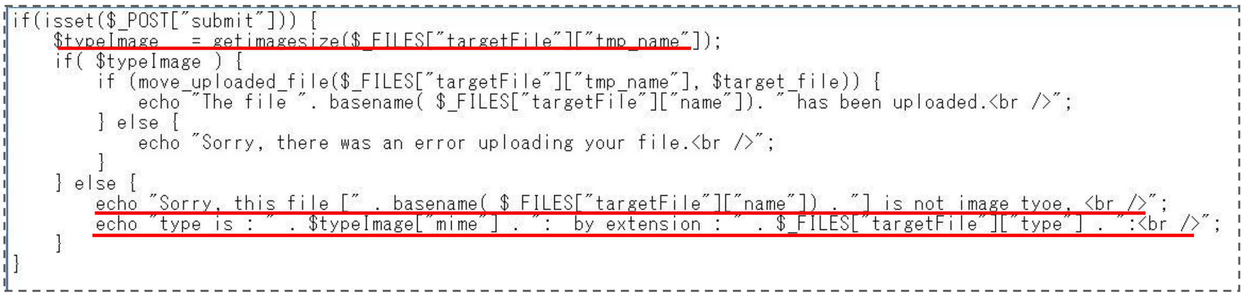
ex31b.php

```
.....
<body>
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["targetFile"]["name"]);
if(isset($_POST["submit"])) {
  $typeImage  = getimagesize($_FILES["targetFile"]["tmp_name"]);
  if( $typeImage ) {
    if (move_uploaded_file($_FILES["targetFile"]["tmp_name"], $target_file)) {
      echo "The file ". basename( $_FILES["targetFile"]["name"]). " has been uploaded.<br />";
    } else {
      echo "Sorry, there was an error uploading your file.<br />";
    }
  } else {
    echo "Sorry, this file is not image tyoe, <br />";
  }
}
?>
.......
```

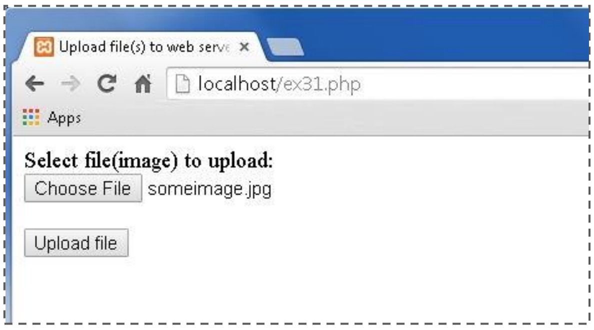
getimagesize

getimagesize(*file_path*)is a function to check image file (.jpg, .jpeg, .png. .gif and so on).

This function return an array with image's information (width, height, type and so on) if target file is an image type file, and return FALSE if not.

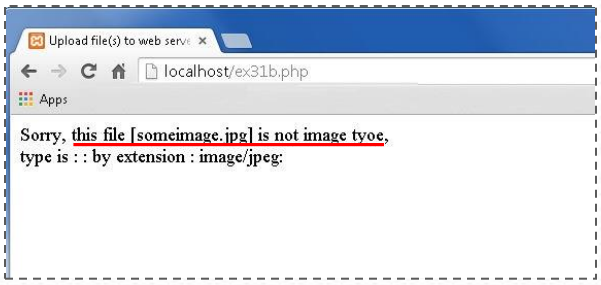


```
if(isset($_POST["submit"])) {
    $typeImage   = getimagesize($_FILES["targetFile"]["tmp_name"]);
    if( $typeImage ) {
        if (move_uploaded_file($_FILES["targetFile"]["tmp_name"], $target_file)) {
            echo "The file ". basename( $_FILES["targetFile"]["name"]). " has been uploaded.<br />";
        } else {
            echo "Sorry, there was an error uploading your file.<br />";
        }
    } else {
        echo "Sorry, this file [" . basename( $_FILES["targetFile"]["name"]) . "] is not image tyoe, <br />";
        echo "type is : " . $typeImage["mime"] . ":  by extension : " . $_FILES["targetFile"]["type"] . ":<br />";
    }
}
```



Select file(image) to upload:
Choose File someimage.jpg

Upload file



Sorry, this file [someimage.jpg] is not image tyoe,
type is : : by extension : image/jpeg:

In the example above, a file ‘someimage.jpg’ is a text file.
It was rejected to upload because ‘getimagesize’ function returned FALSE.
‘$_FILES[<faile_name_tag>][“type”] indicates that this file is ‘image/jpeg’ because it checks only the extension of the file name.
Don’t trust $_FILES[<faile_name_tag>][“type”]. It can be faked.

Remember “Extension of a file can be changed into any extension name”.

3) prevent to overwrite existing files

```
if( file_exists( $target_file ) ) {
   //  some codes in case where the target file already exists
}
```

```
file_exists()
```

file_exists() is a function to check whether specified file or folder exists or not.

Syntax :

file_exists(*$name*)

$name : path to the file or the folder

return value	:	true if the file or the folder exists
		false if the file or the folder doesn’t exists

4) search files limited by file types

when we puch ‘Choose file’ button, ‘File Select control’ will be shown.
If you want to list only limited file types (ex. only image files), how can you do it?
You can add some options to ‘input’ tag shown below;

```
<input type=”file” name=”targetFile” accept=”image/*” />
```

Example shown above is a case to list only image type files(.jpg etc).
“accept” attribute has syntax below;

```
<input accept=”file_extension|image/*|audio/*|video/*|media_type” />
```

You can give extension’s list like this;

<input …… accept=”.html, .htm” … />

This will list only files with extension “html”,”htm”.

[Practice 35] Upload files (5) to prevent to upload faked file

1) Make a text file with some text and modify extension to ‘jpg’ manually
2) Using ‘ex31.php’, upload the file to your web server.
3) Confirm that you can upload a text file with extension ‘.jpg’.
4) Delete uploaded this text file from your web server.
5) Add codes to prevent to upload faked file type by using ‘getimagesize()’ function.
6) Confirm that you can’t upload the file.

[Practice 36] Upload files (6) to prevent to overwrite file

1) Make a text file with some text on your PC
2) Using ‘ex31.php’, upload the file to your web server.
3) Modify the contents of the text file on your PC
4) Again using ‘ex31.php’, upload the modified file to your web server.
5) Confirm that a file with same name is uploaded and overwritten without any caution. This is default.
6) Add code to prevent to overwrite by using ‘file_exists()’ function.
7) Confirm that you can prevent to overwrite a file with same name.

PHP session

Sometime it is said that “Web application is stateless one”.
What does it mean? What’s the meaning of the word ‘stateless’?

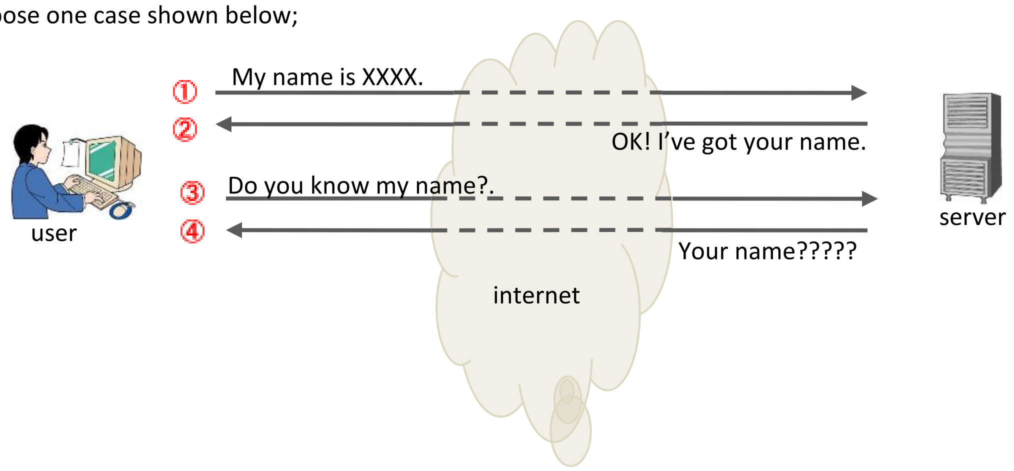
1) Stateful and stateless

Stateful protocol keeps data between connection.
Stateless protocol doesn’t.

HTTP protocol is one of stateless protocol.
Our web applications run with HTTP protocol, so we can’t keep data.

What does it mean?

Suppose one case shown below;



In the picture shown above, communication between ‘user’ and ‘server’ will show you what the stateless protocol is.
In the stateless protocol, data can be kept only in a transaction, and a transaction in HTTP protocol means a period between①and②, or ③and④ , that is, from the time when ‘user’ send request to ‘server’ until the time ‘server’ return response to ‘user’.

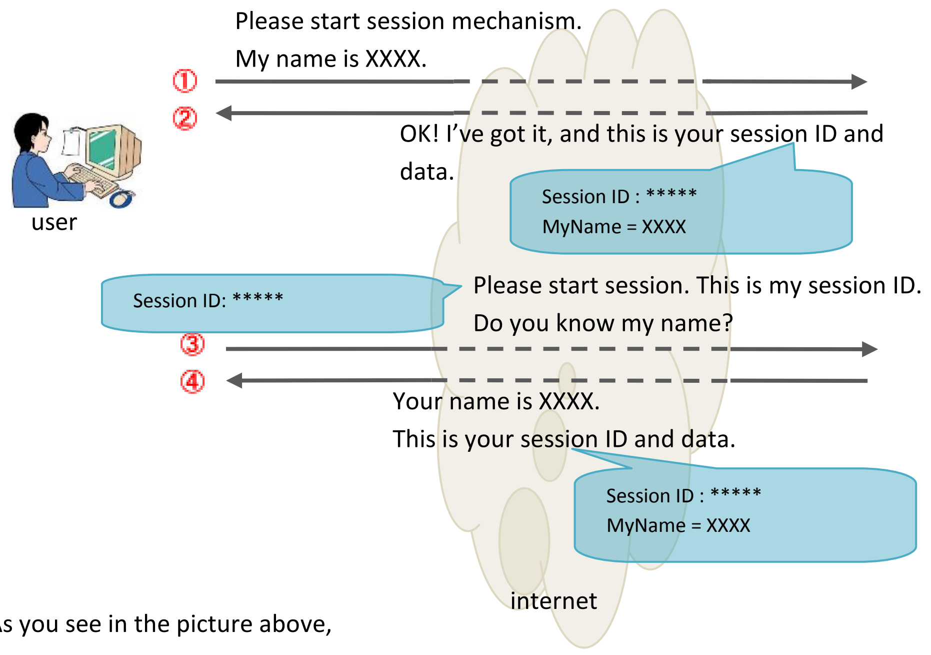
A period from ①to②, which means one transaction, and from ③to ④ it is another transaction. The ‘server’ can’t keep ‘My name’ over a transaction, so ‘server’ can’t answer when it is asked ‘My name’ in the next transaction.

We can say ‘In web application, we can’t keep data across web pages’.

But we need to keep data across web pages otherwise we need to enter our ID or name every time we change web page on an application.
PHP session is a mechanism to keep data across pages.

2) PHP session

When PHP session mechanism is working, how does it works on web apps.


Please start session mechanism.
My name is XXXX.
①
②
OK! I've got it, and this is your session ID and data.
Session ID : *****
MyName = XXXX
user
server
Session ID: *****
Please start session. This is my session ID.
Do you know my name?
③
④
Your name is XXXX.
This is your session ID and data.
Session ID : *****
MyName = XXXX
internet

As you see in the picture above,

a. Session data is generated on 'server' and kept in 'user' side.
The path where session data is stored is defined in 'php.ini'.
You can find by the property name 'session.save_path'.

b. Every time 'user' send request to 'server', session data is sent to 'server', too.

c. Session data is updated on the 'server' and returned to 'user'.

d. Session data will be deleted when 'user' log out from the application or 'user' quit the browser or 'user' leave session data
as it is for over specified seconds which is defined in 'php.ini'.

3) Functions and variables for session data handling

```
<?php session_start(); ?>
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Session sample </title>
</head>
<body>
  <p>Session started</p>
  My name is S.John<br />
  <?php $_SESSION["myName"] = "S.John"; ?>
</body>
</html>
```

In the example above,

session_start() : Function to declare starting session.
There must be no output before this function.
Don't put even one space before session_start(), which means before HTML output like '<html>…', session_start() function should be executed.

$_SESSION : One of superglobal variables. We can generate, update session data by this superglobal.
In the example above, one session data with key 'myName' is created, and the value 'K.Tezuka' is set.

Other functions related to session;

session_unset() : Function to clear all keys and values in $_SESSION.

session_destroy() : Function to delete session file.

[Practice 37] Session data handling

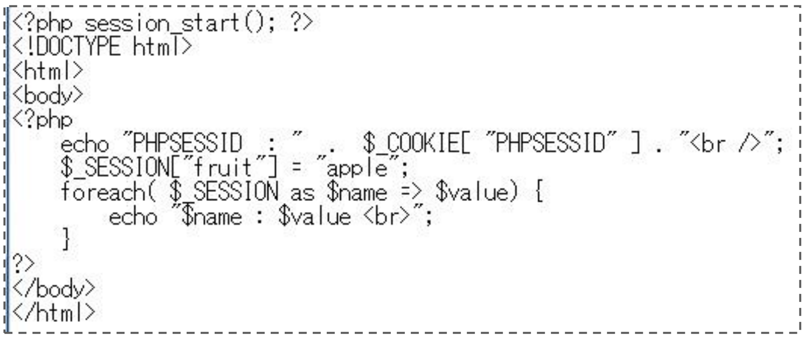
Make a program 'ex37.php' shown below, and access to it on your browser;

```
<?php session_start(); ?>
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Session data handling</title>
</head>
<body>
  <?php
    echo "PHPSESSID  : " . $_COOKIE["PHPSESSID"] . "<br />";
    $_SESSION["fruit"] = "apple";
    foreach ($_SESSION as $name => $value) {
      echo "$name : $value <br>"
    }
  ?>
</body>
</html>
```

After accessing to 'ex37,php', check the folder '<xampp root>/tmp'.
Check whether session file exists or not, and if exists, check the file name and 'PHPSESSID' value shown on your browser.

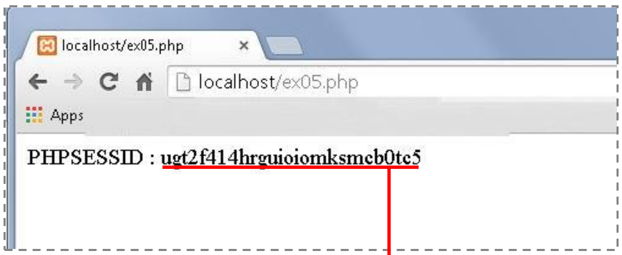
Confirm that session mechanism makes session file on your local folder.

ex37.php



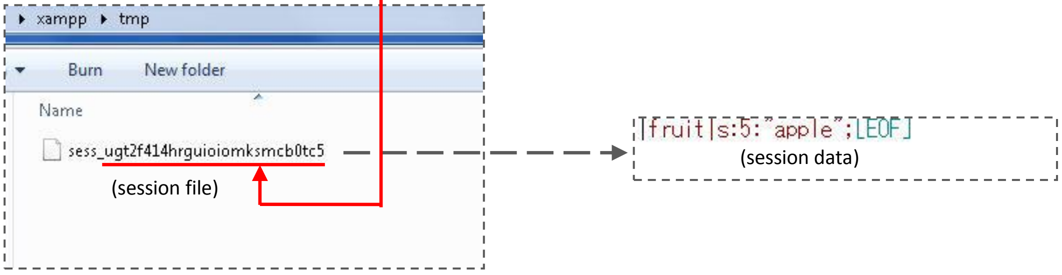
```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<body>
<?php
    echo "PHPSESSID  : "  .  $_COOKIE[ "PHPSESSID" ] . "<br />";
    $_SESSION["fruit"] = "apple";
    foreach( $_SESSION as $name => $value) {
        echo "$name : $value <br>";
    }
?>
</body>
</html>
```

output from 'ex37.php' on a browser:



PHPSESSID : ugt2f414hrguioiomksmcb0tc5

after accessing, session file is on '<xampp root>/tmp' folder



```
fruit|s:5:"apple";[EOF]
```

[Practice 38] Session data handling

Make a program 'ex38.php' shown below, and access to it on your browser;

```
<?php session_start(); ?>
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Session data handling</title>
</head>
<body>
  <?php
    $_SESSION["fruit"] = "banana"      //  update session data
    $_SESSION["color"] = "yellow";     //  add new session data
    foreach ($_SESSION as $name => $value) {
      echo "$name : $value <br>"
    }
  ?>
</body>
</html>
```

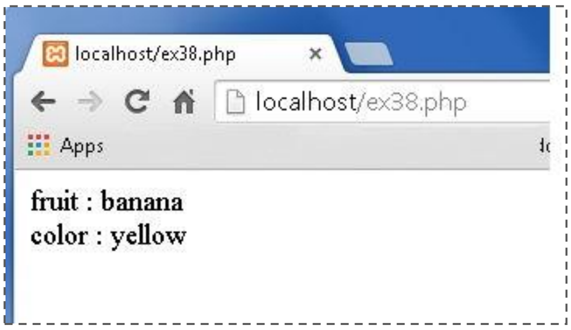
[Practice 39] Session data handling (clear session, delete session)

You can understand how to add/update session data by $_SESSION superglobal variable.

Make a program 'ex39.php' shown below, and access to it on your browser;

```
<?php session_start(); ?>
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Session data handling</title>
</head>
<body>
  <?php
    foreach ($_SESSION as $name => $value) {
      echo "$name : $value <br>"
    }
    session_unset();
  ?>
</body>
</html>
```

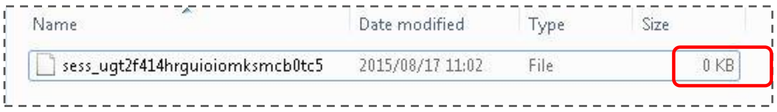
session data before executing session_unset();



```
fruit : banana
color : yellow
```

```
1|fruit|s:6:"banana";color|s:6:"yellow";[EOF]
```

session data after executing session_unset();



Name	Date modified	Type	Size
sess_ugt2f414hrguioiomksmcb0tc5	2015/08/17 11:02	File	0 KB

There's nothing in the session file.

If you put 'session_destroy();' instead of 'session_unset()' in 'ex39.php', then this session file 'sess_ugt2・・・・' will be removed. Try it.

4) cookie

'cookie' is also one mechanism in windows to keep some data across states in web applications.

'cookie' has the same structure as 'session', it's an array with key and value.

'cookie' is a browser mechanism, which means if you use IE, then you will use IE's 'cookie'.

In case of Google Chrome, 'cookie' is a binary file stored in a folder below:

c:/document and settings/<user>/AppData/local/Google/Chrome/User Data/default

Functions related to 'cookie' management:

setcookie() : Function to define a new cookie.

Syntax : setcookie(*$name, $value, $expire, $path, $domain, $secure, $http*);

$name	Name of a cookie. (required)
$value	Value of a cookie. (optional)
$expire	define when this cookie expires in seconds.(optional) ex) 'time() + 86400 * 7' means to expire 1 week later ex) 0 : expire after browser close Default is 0.
$path	Path(directory) of the cookie on the server. (optional) '/' means the cookie can reside in entire domain. Default is the current directory.
$domain	Specify the name of domain (optional)
$secure	If set TRUE, cookie will be sent only in HTTPS secure protocol. Default is FALSE.
$http	If set TRUE, only HTTP protocol can be tranmit cookies, and any other scripting language can't access cookies. Better for avoiding XSS attacks. Default is FALSE.

$_COOKIE[] : Superglobal for cookies.

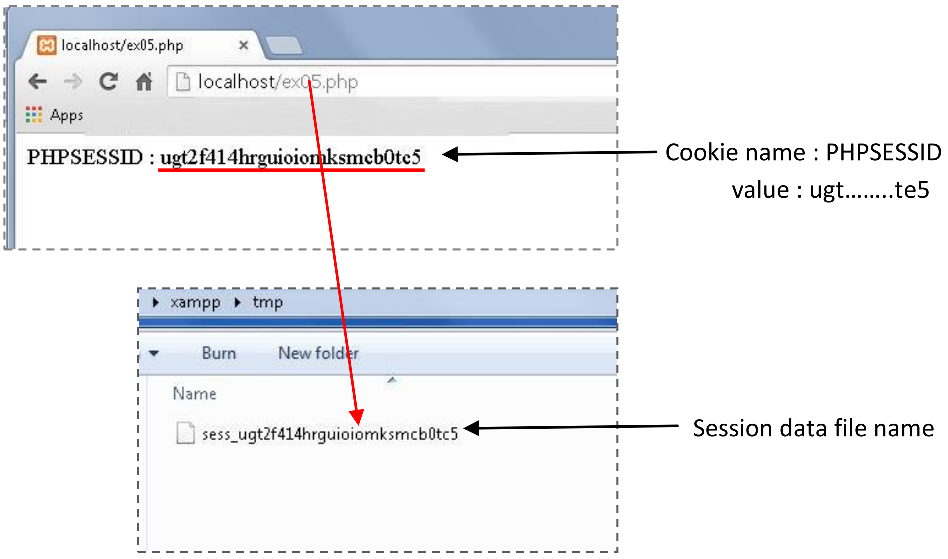
5) Difference between session and cookie

'session' is a mechanism which is controlled by web applications on web server.

'cookie' is a mechanism which is controlled by browser on user's PC.

'session' data are usually deleted 'log-off' operation or after specified period, 'cookies' are deleted when user do 'delete' operation on a browser.

In "Practice 37", we check one superglobal '$_COOKIE'.
Session mechanism uses Cookie to save 'Session ID' with Cookie name 'PHPSESSID' whose value shows random value of session data file name.



PHPSESSID : ugt2f414hrguioiomksmcb0tc5

sess_ugt2f414hrguioiomksmcb0tc5

PHP filters

In web applications, we should validate input data from application users.

PHP filters support our validation process in our PHP programs.

Functions related to PHP filters :

filter_var() : function to validate and sanitize input data.

Syntax :

filter_var(*$variable*, *$filter*)

$variable	Value to filter
$filter	Filter ID. Filter ID is a integer value assigned to each filter. See a list below.

Return value : if successful, filtered value wwill be returned
if not (= there's unmatched characters in $varuable),
then return FALSE.

filter_input() : function to validate data in external variable GET, POST, SESSION, COOKIE or ENV.

Syntax :

filter_input(*$type*, *$variable*, *$filter, $option*)

$type	Dara type to check. One of the following; INPUT_GET. INPUT_POST, INPUT_SESSION, INPUT_COOKIE, INPUT_ENV
$variable	Variable name to check
$filter	Filter ID. Filter ID is a integer value assigned to each filter. See a list below.
$option	some *$filters* have option (see example 2) in next page)

Some filter's definition :
(in details, see http://php.net/manual/en/filter.filters.php)

Filter Constant	ID value	description
FILTER_VALIDATE_ENAIL	274	Validate an e-mail address
FILTER_VALIDATE_IP	275	Validate an IP address
FILTER_VALIDATE_URL	273	Validate a URL
FILTER_SANITIZE_ENAIL	517	Remove all illegal characters from an e-mail address

Examples:

1) Validating e-mail address

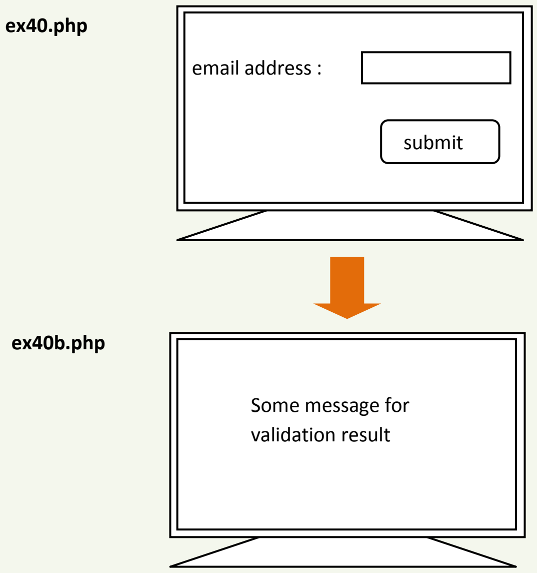
```
<?php
   $email  =  “scot.baily@example.com”;
   if( filter_var( $email, FILTER_VALIDATE_EMAIL ) === FALSE ) {
      echo( “$email is invalid email address”);
   } else {
      echo( “$email is valid email address”);
   }
?>
```

2) Validating IP address

```
<?php
   $ip  =  “192.168.101.18”;
   if( filter_var( $ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4 ) === FALSE ) {
      echo( “$ip is invalid ip address for ipV4”);
   } else {
      echo( “$ip is valid ip address for ipV4”);
   }
?>
```

[Practice 40] Filtering

Make a program ‘ex40.php’ which has a form with input field for email address and send it to ‘ex40b.php’ which validate received email address. Validation result should be displayed with some message on the browser.



PHP error handling

There are several ways for error handling in PHP;

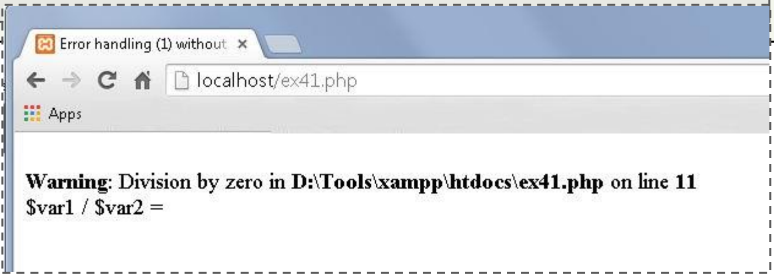
1) Using die() function, stop PHP script
2) Define custom error handling function
3) "try-catch" exception handling (PHP5 and later)

Without error handling, you will get some message from PHP;

[Practice 41] Error handling : without error handling

Make a program 'ex41.php' ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Error handling (1) without any error handling</title>
</head>
<body>
  <?php
    $var1  =  100;
    $var2  =  0;
    $result  =  $var1  /  $var2;
    echo('$var1 / $var2  = " . $result . '<br />' );
  ?>
</body>
</html>
```

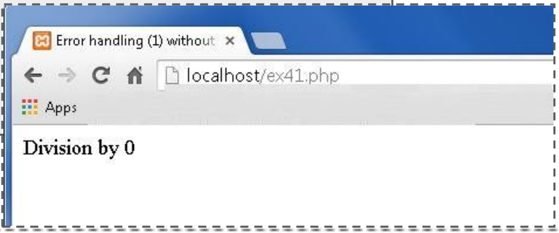


```
Warning: Division by zero in D:\Tools\xampp\htdocs\ex41.php on line 11
$var1 / $var2 =
```

1. Using die() function

In the example above, we can do validation check for calculation and add very simple error handling. Modify example above to codes shown below and try it;

```
<?php
  $var1  =  100;
  $var2  =  0;
  if( $var2 <> 0 ) {
    $result  =  $var1  /  $var2;
    echo('$var1 / $var2  = " . $result . '<br />' );
  } else {
    die( 'Division by 0' );
  }
?>
```



```
Division by 0
```

2. Custom error handling function

Sometimes we don’t want to stop script and continue it depending on error level.
PHP defines several error level and such errors as ‘fatal error’ level can’t continue a script any more but other errors can continue.

Table below shows PHP error levels;

value	constant	description
1	E_ERROR	Fatal run-time error.
2	E-WARNING	Run-time warning. Not fatal.
4	E_PARSE	Compile-time parse error. Not fatal.
8	E_NOTICE	Run-time notice. Not fatal.
16	E_CORE_ERROR	Fatal error at PHP start up.
32	E_CORE_WARNING	Warning error at PHP start up. Not fatal.
64	E_COMPILE_ERROR	Compile-time error. Fatal.
128	E_COMPILE_ WARNING	Compile-time warning error. Not fatal.
256	E_USER_ERROR	User-defined fatal error.
512	E_USER_ WARNING	User-defined warning error. Not fatal.
1024	E_USER_NOTICE	User-defined notice. Not fatal.
2048	E_STRICT	PHP suggest changes to your codes for forward compatibility
4096	E_RECOVERABLE_ERROR	Catchable fatal error
8192	E_DEPRECATED	Run-time notice. Code will not work in future version.
16384	E_USER_DEPRECATED	User-defined E_DEPRECATED notice.
32767	E_ALL	Enable all PHP errors and warning

error_reporting() : define which levels of error are reported.

Syntax : error_reporting(*level*);

level : (optional) specify error report level for current script.

ex)

```
error_reporting( E_ERROR | E_WARNING | E_PARSE );
                        //  report runtime errors
```

Custom error handling function may have such definition ;

```
function_name( error_level,      // error level.   Required.
               error_message,    //  error message.  Required.
               error_file,       //  file name where error occurred.
               error_line,       //  line number where error occurred
               error_context );  //  array of variables and values
```

set_error_handler() : function to set user-defined error handler function.

Syntax : set_error_handler(*errorhandler*, *errorlevel*);

errorhandler : name of user-defined error handling function. Required.

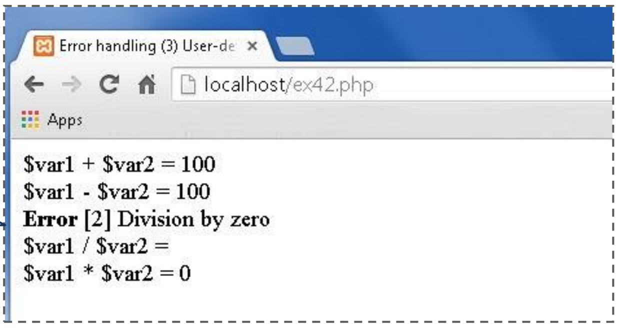
errorlevel : which error level the function will show. Optional. Default is E_ALL

[Practice 42] Error handling : user-defined error handler

Make a program 'ex42.php' ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Error handling (1) without any error handling</title>
</head>
<body>
  <?php
    function customErrorHandle ( $errlvl,  $errmsg ){
      eho("<b>Error</b> [$errlvl]  $errmsg <br />");
    }
    set_error_handler( "customErrorHandle" )
    $var1  =  100;
    $var2  =  0;
    echo('$var1 + $var2 =' . ( $var1 + $var2 ) . '<br />');
    echo('$var1 - $var2 =' . ( $var1 - $var2 ) . '<br />');
    echo('$var1 / $var2 =' . ( $var1 / $var2 ) . '<br />');
    echo('$var1 * $var2 =' . ( $var1 * $var2 ) . '<br />');
  ?>
</body>
</html>
```

'Division by zero' error occurred. User-defined function 'customErrorHandle' is called and given message is shown with error level, which is 'E_WARNING', not fatal, so process didn't stopped and continue until end.



```
$var1 + $var2 = 100
$var1 - $var2 = 100
Error [2] Division by zero
$var1 / $var2 =
$var1 * $var2 = 0
```

3. try-throw-catch exception handling

'try-throw-catch exception handling' is new error handling in PHP5.

Here's one simple example;

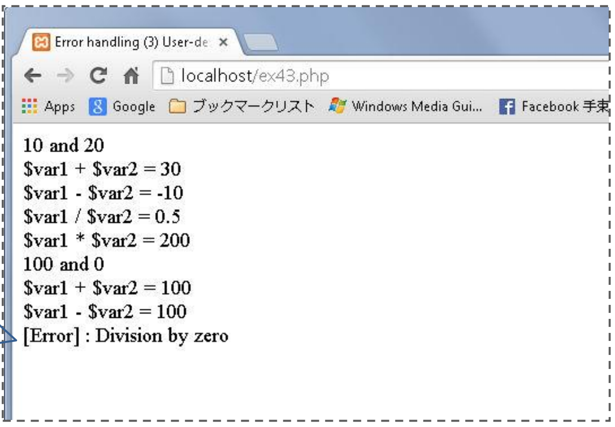
[Practice 43] PHP exception handling

Make a program 'ex43.php' ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exception handling (1)</title>
</head>
<body>
  <?php
    function calc( $var1, $var2 ) {
      echo "$var1 and $var2<br />";
      echo('$var1 + $var2 =' . ( $var1 + $var2 ) . '<br />');
      echo('$var1 - $var2 =' . ( $var1 - $var2 ) . '<br />');
      if( $var2 <> 0 ) {
        echo('$var1 / $var2 =' . ( $var1 / $var2 ) . '<br />');
      } else {
        throw new Exception( "Divide by zero" );
      }
      echo('$var1 * $var2 =' . ( $var1 * $var2 ) . '<br />');
    }

    try {
      calc( 10, 20 );
      calc( 100, 0 );
    } catch ( Exception $e ){
      echo "[Error] : $e->getMessage() ";
    }
  ?>
</body>
```

After execution of 'catch' block, process terminated.

To continue process after exception block, we need to define custom error handling such as codes shown in 'ex42.php'.

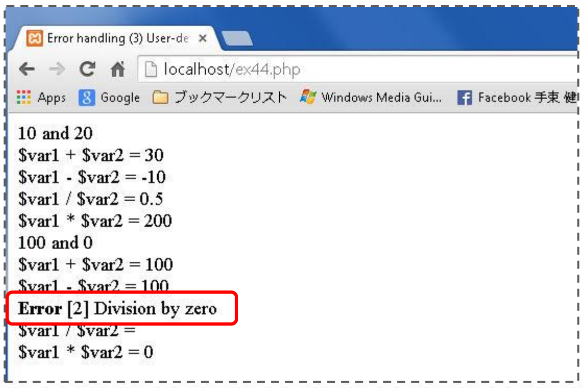

Error handling (3) User-de
localhost/ex43.php
Apps Google ブックマークリスト Windows Media Gui... Facebook 手束

```
10 and 20
$var1 + $var2 = 30
$var1 - $var2 = -10
$var1 / $var2 = 0.5
$var1 * $var2 = 200
100 and 0
$var1 + $var2 = 100
$var1 - $var2 = 100
[Error] : Division by zero
```

[Practice 44] PHP exception handling

Modify 'ex43.php' to continue process after error ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exception handling (1)</title>
</head>
<body>
  <?php
    function customErrorHandle($errlvl,  $errmsg ){
      eho("<b>Error</b> [$errlvl]  $errmsg <br />");
      if( $errlvl === E_ERROR ) {    // if fatal error, exit process
        exit();
      }
    }
    function calc( $var1, $var2 ) {
      echo "$var1 and $var2<br />";
      echo('$var1 + $var2 =' . ( $var1 + $var2 ) . '<br />');
      echo('$var1 - $var2 =' . ( $var1 - $var2 ) . '<br />');
      if( $var2 <> 0 ) {
        echo('$var1 / $var2 =' . ( $var1 / $var2 ) . '<br />');
      } else {
        throw new Exception( "Divide by zero" );
      }
      echo('$var1 * $var2 =' . ( $var1 * $var2 ) . '<br />');
    }

    try {
      set_error_handler( "customErrorHandle" );
      calc( 10, 20 );
      calc( 100, 0 );
    } catch ( Exception $e ){
      echo "[Error] : $e->getMessage() ";
    }
  ?>
</body>
</html>
```



```
10 and 20
$var1 + $var2 = 30
$var1 - $var2 = -10
$var1 / $var2 = 0.5
$var1 * $var2 = 200
100 and 0
$var1 + $var2 = 100
$var1 - $var2 = 100
Error [2] Division by zero
$var1 / $var2 =
$var1 * $var2 = 0
```

MySQL database

MySQL is a database system which is ;

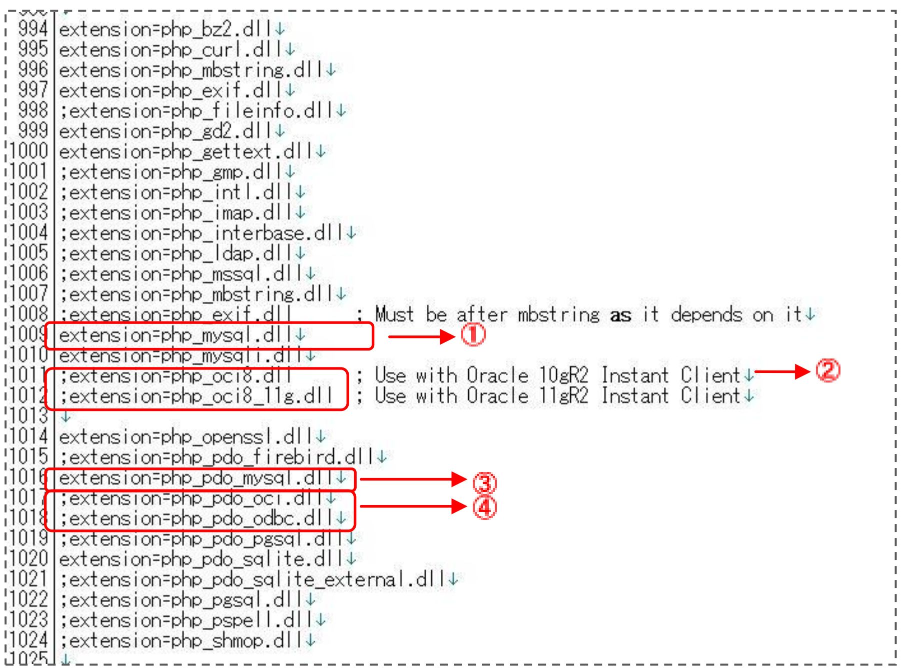
1) free to use
2) using standard SQL
3) easy to use with PHP

When we use MySQL, we need to configure and prepare its operating environment.
Operating environment is defined in 'php.ini' and 'my.ini'.
('my.ini' is a name only for windows, for UNIX, Linux and Mac it is 'my.cnf'.)

1. 'php.ini'
   Information in 'php.ini' related to MySQL resides in;

   'extension' property in [php] section
   'mysql' property in [MySQL] section

Here, you can see 'extension' properties in [php] section;



```
extension=php_bz2.dll
extension=php_curl.dll
extension=php_mbstring.dll
extension=php_exif.dll
;extension=php_fileinfo.dll
extension=php_gd2.dll
extension=php_gettext.dll
;extension=php_gmp.dll
;extension=php_intl.dll
;extension=php_imap.dll
;extension=php_interbase.dll
;extension=php_ldap.dll
;extension=php_mssql.dll
;extension=php_mbstring.dll
;extension=php_exif.dll       ; Must be after mbstring as it depends on it
extension=php_mysql.dll
extension=php_mysqli.dll
;extension=php_oci8.dll       ; Use with Oracle 10gR2 Instant Client
;extension=php_oci8_11g.dll   ; Use with Oracle 11gR2 Instant Client

extension=php_openssl.dll
;extension=php_pdo_firebird.dll
extension=php_pdo_mysql.dll
;extension=php_pdo_oci.dll
;extension=php_pdo_odbc.dll
;extension=php_pdo_pgsql.dll
extension=php_pdo_sqlite.dll
;extension=php_pdo_sqlite_external.dll
;extension=php_pgsql.dll
;extension=php_pspell.dll
;extension=php_shmop.dll
```

① ③ when you use MySQL, uncomment this line

② ④ when you use Oracle, uncomment this line

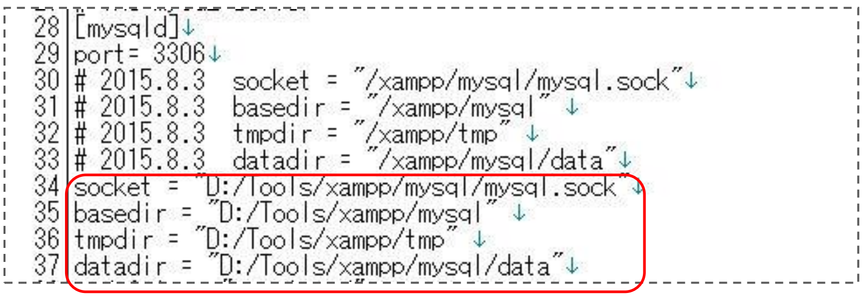
As for [MySQL] section in 'php.ini', we can keep it as it is.

2 'my.ini'

2-1 Folder path used in MySQL.

'tmpdir' and 'datadir' can be set anywhere on your PC, but you must give these folder at least 'read/write oermission'.

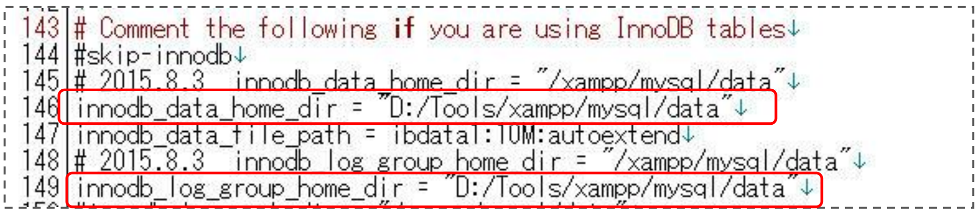
'socket' and 'basedir' will be under <mysql> folder.



```
[mysqld]
port= 3306
# 2015.8.3  socket = "/xampp/mysql/mysql.sock"
# 2015.8.3  basedir = "/xampp/mysql"
# 2015.8.3  tmpdir = "/xampp/tmp"
# 2015.8.3  datadir = "/xampp/mysql/data"
socket = "D:/Tools/xampp/mysql/mysql.sock"
basedir = "D:/Tools/xampp/mysql"
tmpdir = "D:/Tools/xampp/tmp"
datadir = "D:/Tools/xampp/mysql/data"
```

2-2 'innodb'

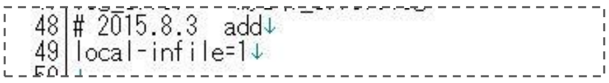
'innodb' is one of MySQL database engine. We'd better use it for better efficiency.



```
# Comment the following if you are using InnoDB tables
#skip-innodb
# 2015.8.3  innodb_data_home_dir = "/xampp/mysql/data"
innodb_data_home_dir = "D:/Tools/xampp/mysql/data"
innodb_data_file_path = ibdata1:10M:autoextend
# 2015.8.3  innodb_log_group_home_dir = "/xampp/mysql/data"
innodb_log_group_home_dir = "D:/Tools/xampp/mysql/data"
```

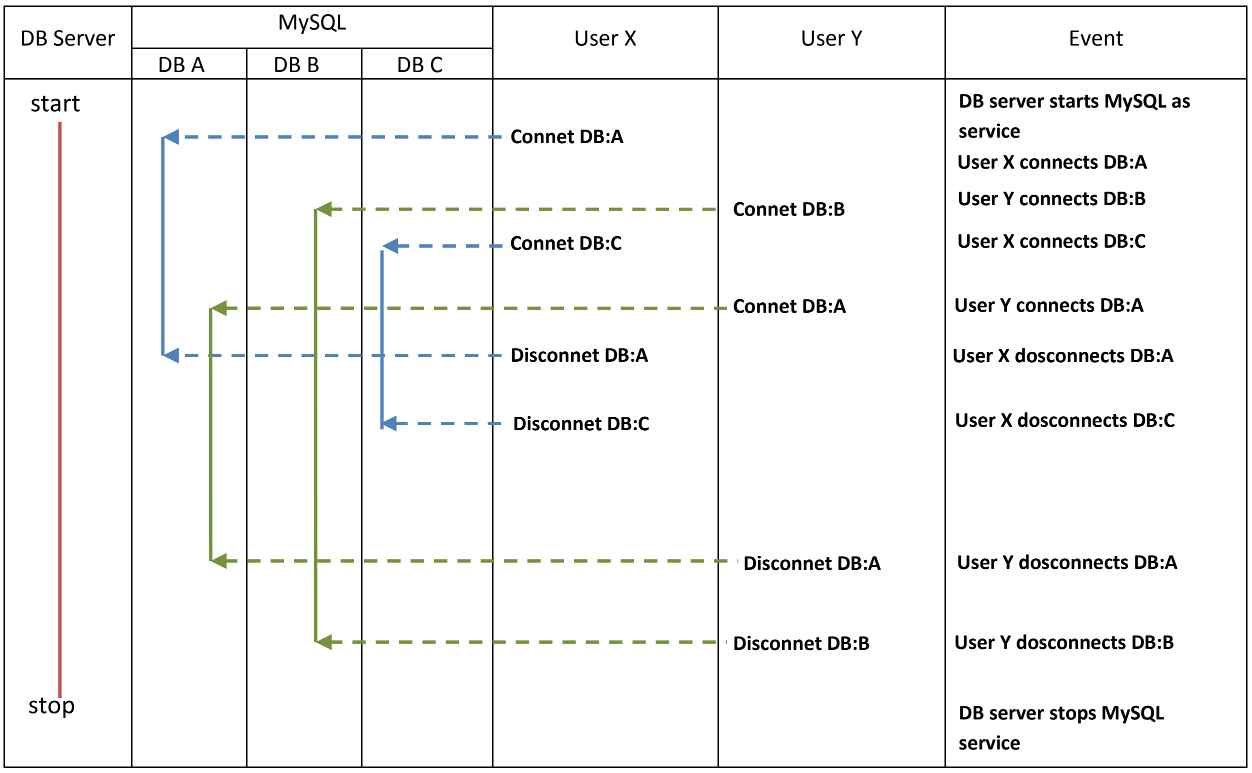
2-3 Import data from an external file

To enable to import data from an external file, add one line shown below;



```
# 2015.8.3  add
local-infile=1
```

3 State transition in MySQL



Start/Stop MySQL service is server side event/operation.

Connect/Disconnect to/from MySQL each database is user/client side event/operation, but executed on the server.

Operations on individual database are also use/client operations, but executed on the server.

Each database can be accessed by multi clients. Tables in a database also can be accessed concurrently by multi clients except when a client open table in exclusive mode.

(in case where XAMPP is installed in user PC and run, all events occur on the same user PC, but in actual service environment, 'start/stop MySQL service' is executed on the server(DB server))

4 Start & stop MySQL

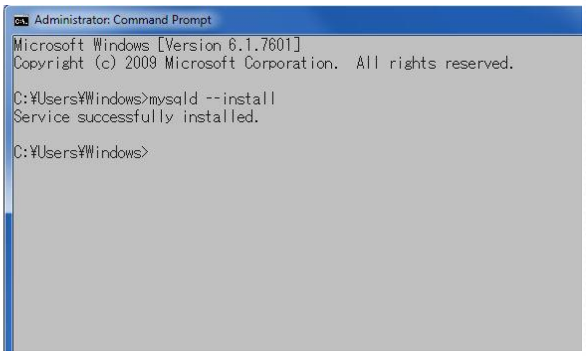
4-1 Register MySQL as service

4-1-1 Using command line

Key in command shown below;

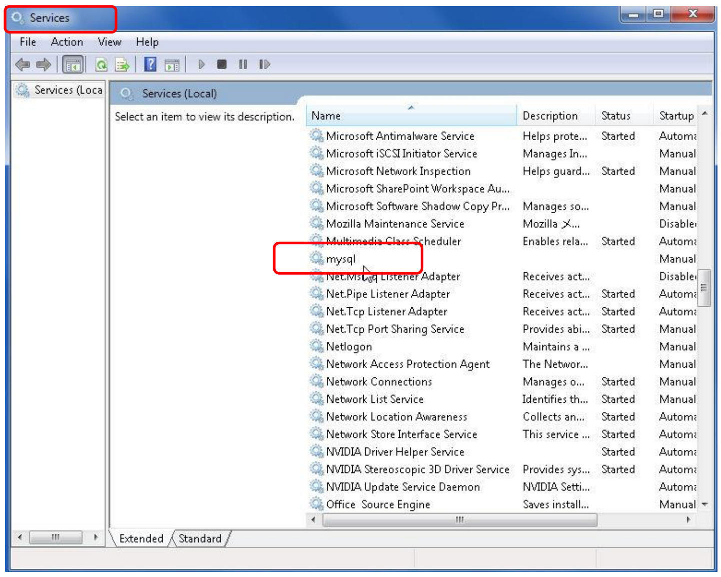
mysqld --install

If you see a message shown on the right, MySQL is successfully installed as service.



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:¥Users¥Windows>mysqld --install
Service successfully installed.

C:¥Users¥Windows>
```

After registering successfully, you can check it as follows;
<control panel> - <administration tool> - <service>



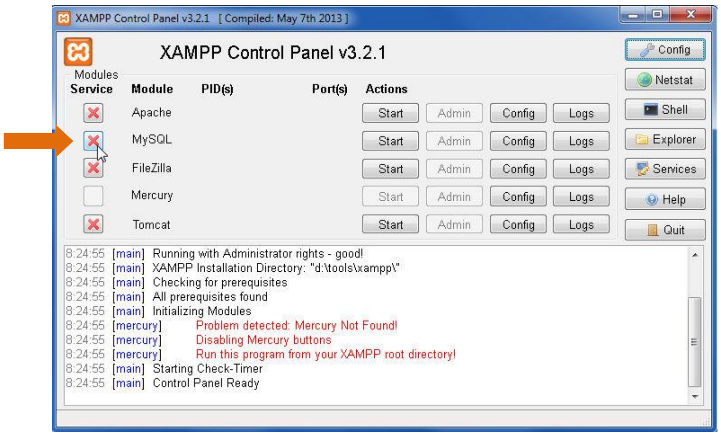
Services (Local)

Select an item to view its description.

Name	Description	Status	Startup
Microsoft Antimalware Service	Helps prote...	Started	Automa
Microsoft iSCSI Initiator Service	Manages In...		Manual
Microsoft Network Inspection	Helps guard...	Started	Manual
Microsoft SharePoint Workspace Au...			Manual
Microsoft Software Shadow Copy Pr...	Manages so...		Manual
Mozilla Maintenance Service	Mozilla メ...		Disable
Multimedia Class Scheduler	Enables rela...	Started	Automa
mysql			Manual
Net.Msmq Listener Adapter	Receives act...		Disable
Net.Pipe Listener Adapter	Receives act...	Started	Automa
Net.Tcp Listener Adapter	Receives act...	Started	Automa
Net.Tcp Port Sharing Service	Provides abi...	Started	Manual
Netlogon	Maintains a ...		Manual
Network Access Protection Agent	The Networ...		Manual
Network Connections	Manages o...	Started	Manual
Network List Service	Identifies th...	Started	Manual
Network Location Awareness	Collects an...	Started	Automa
Network Store Interface Service	This service ...	Started	Automa
NVIDIA Driver Helper Service		Started	Automa
NVIDIA Stereoscopic 3D Driver Service	Provides sys...	Started	Automa
NVIDIA Update Service Daemon	NVIDIA Setti...		Automa
Office Source Engine	Saves install...		Manual

4-1-2 Using XAMPP control panel

On XAMPP control panel, you will see 'service' checkbox on the left. Check it, then MySQL is registered as service.



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

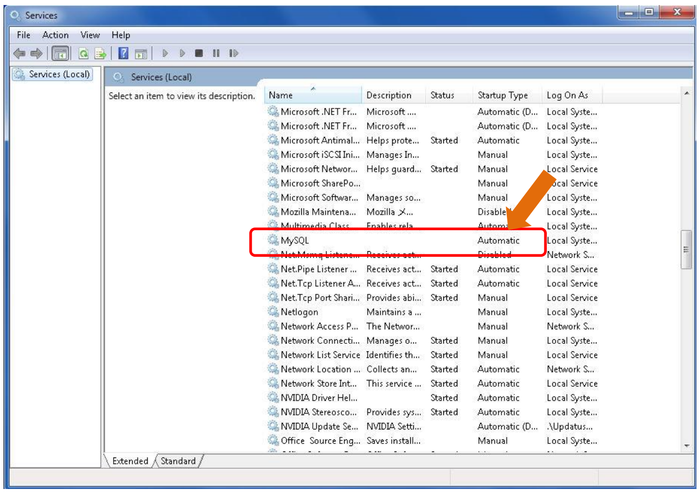
Service	Module	PID(s)	Port(s)	Actions
✗	Apache			Start Admin Config Logs
✗	MySQL			Start Admin Config Logs
✗	FileZilla			Start Admin Config Logs
	Mercury			Start Admin Config Logs
✗	Tomcat			Start Admin Config Logs

```
8:24:55  [main]     Running with Administrator rights - good!
8:24:55  [main]     XAMPP Installation Directory: "d:\tools\xampp\"
8:24:55  [main]     Checking for prerequisites
8:24:55  [main]     All prerequisites found
8:24:55  [main]     Initializing Modules
8:24:55  [mercury]      Problem detected: Mercury Not Found!
8:24:55  [mercury]      Disabling Mercury buttons
8:24:55  [mercury]      Run this program from your XAMPP root directory!
8:24:55  [main]     Starting Check-Timer
8:24:55  [main]     Control Panel Ready
```

4-2 Start MySQL

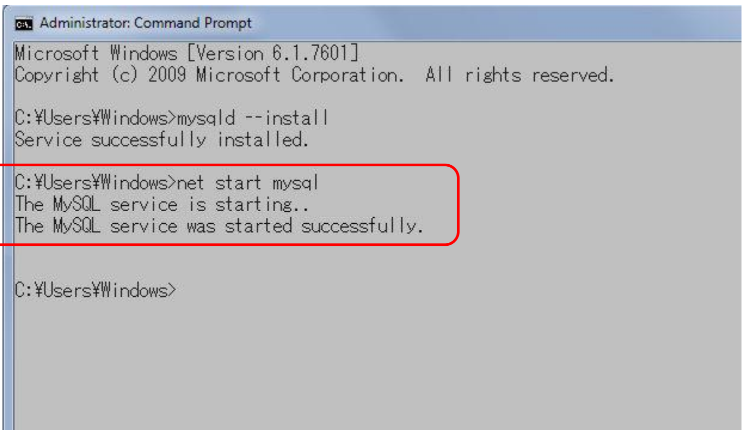
4-2-1 from command line

After registering MySQL as a service and 'startup type' is 'automatic', you don't need to start MySQL manually.



Name	Description	Status	Startup Type	Log On As
Microsoft .NET Fr...	Microsoft		Automatic (D...	Local Syste...
Microsoft .NET Fr...	Microsoft		Automatic (D...	Local Syste...
Microsoft Antimal...	Helps prote...	Started	Automatic	Local Syste...
Microsoft iSCSI Ini...	Manages In...		Manual	Local Syste...
Microsoft Networ...	Helps guard...	Started	Manual	Local Service
Microsoft SharePo...			Manual	Local Service
Microsoft Softwar...	Manages so...		Manual	Local Syste...
Mozilla Maintena...	Mozilla メ...		Disabled	Local Syste...
Multimedia Class ...	Enables rela...		Automatic	Local Syste...
MySQL			Automatic	Local Syste...
Net.Msmq Listene...	Receives act...		Disabled	Network S...
Net.Pipe Listener ...	Receives act...	Started	Automatic	Local Service
Net.Tcp Listener A...	Receives act...	Started	Automatic	Local Service
Net.Tcp Port Shari...	Provides abi...	Started	Manual	Local Service
Netlogon	Maintains a ...		Manual	Local Syste...
Network Access P...	The Networ...		Manual	Network S...
Network Connecti...	Manages o...	Started	Manual	Local Syste...
Network List Service	Identifies th...	Started	Manual	Local Service
Network Location ...	Collects an...	Started	Automatic	Network S...
Network Store Int...	This service ...	Started	Automatic	Local Service
NVIDIA Driver Hel...		Started	Automatic	Local Syste...
NVIDIA Stereosco...	Provides sys...	Started	Automatic	Local Syste...
NVIDIA Update Se...	NVIDIA Setti...		Automatic (D...	.\Updatus...
Office Source Eng...	Saves install...		Manual	Local Syste...

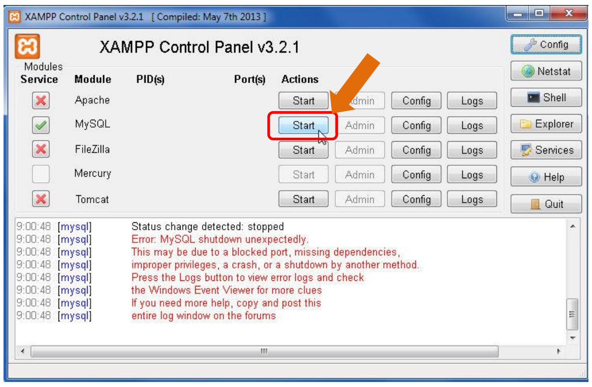
But just after registering as a service, once you need to start MySQL manually. See a picture below; 'net start mysql' is to start MySQL as service.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:¥Users¥Windows>mysqld --install
Service successfully installed.

C:¥Users¥Windows>net start mysql
The MySQL service is starting..
The MySQL service was started successfully.


C:¥Users¥Windows>
```

4-2-2 from XAMPP control panel

Very simple. You need only to press 'start' button for MySQL.

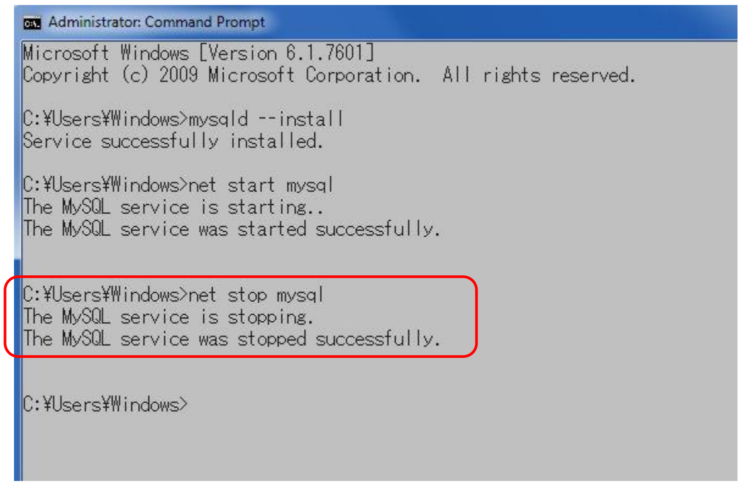


```
9:00:48 [mysql]   Status change detected: stopped
9:00:48 [mysql]   Error: MySQL shutdown unexpectedly.
9:00:48 [mysql]   This may be due to a blocked port, missing dependencies,
9:00:48 [mysql]   improper privileges, a crash, or a shutdown by another method.
9:00:48 [mysql]   Press the Logs button to view error logs and check
9:00:48 [mysql]   the Windows Event Viewer for more clues
9:00:48 [mysql]   If you need more help, copy and post this
9:00:48 [mysql]   entire log window on the forums
```

4-3 Stop MySQL

4-3-1 from command line

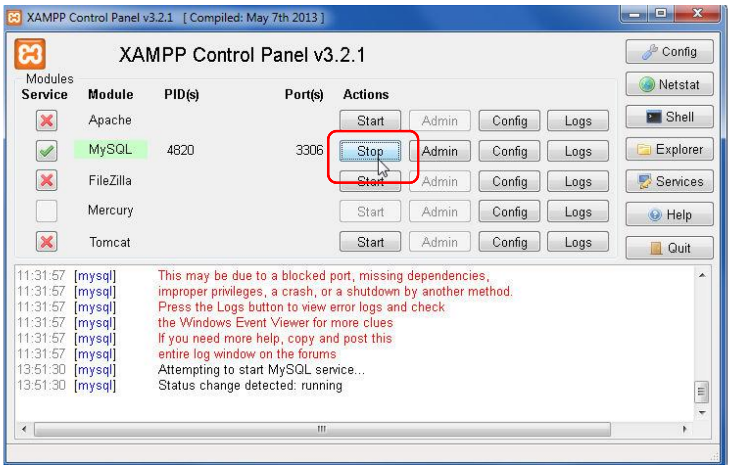
'net stop mysql' is to stop MySQL service.



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:¥Users¥Windows>mysqld --install
Service successfully installed.

C:¥Users¥Windows>net start mysql
The MySQL service is starting..
The MySQL service was started successfully.


C:¥Users¥Windows>net stop mysql
The MySQL service is stopping.
The MySQL service was stopped successfully.


C:¥Users¥Windows>
```

4-3-2 from XAMPP control panel

Press 'Stop' button of MySQL shown below;



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Config
Netstat
Shell
Explorer
Services
Help
Quit

```
Modules
Service  Module     PID(s)   Port(s)   Actions
         Apache                        Start  Admin  Config  Logs
         MySQL      4820     3306      Stop   Admin  Config  Logs
         FileZilla                     Start  Admin  Config  Logs
         Mercury                       Start  Admin  Config  Logs
         Tomcat                        Start  Admin  Config  Logs

11:31:57  [mysql]    This may be due to a blocked port, missing dependencies,
11:31:57  [mysql]    improper privileges, a crash, or a shutdown by another method.
11:31:57  [mysql]    Press the Logs button to view error logs and check
11:31:57  [mysql]    the Windows Event Viewer for more clues
11:31:57  [mysql]    If you need more help, copy and post this
11:31:57  [mysql]    entire log window on the forums
13:51:30  [mysql]    Attempting to start MySQL service...
13:51:30  [mysql]    Status change detected: running
```

4-4 Using MySQL command-line tool

In this training, we use MySQL database from PHP programs, so you don't have a chance to use MySQL command-line tool.

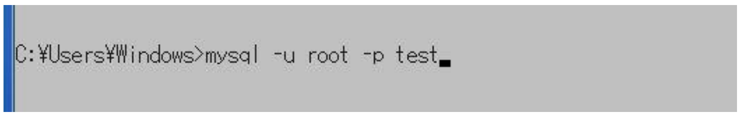
When you don't have any GUI tool for MySQL like XAMPP, phpMyAdmin, you can use them.

As we don't have enough time to study MySQL command-line tool, only two commands are shown here.

MySQL command-line tool will run on 'MySQL shell, which can be started by 'mysql' command shown below;

mysql -u *user-name* -p*password* *database-name*

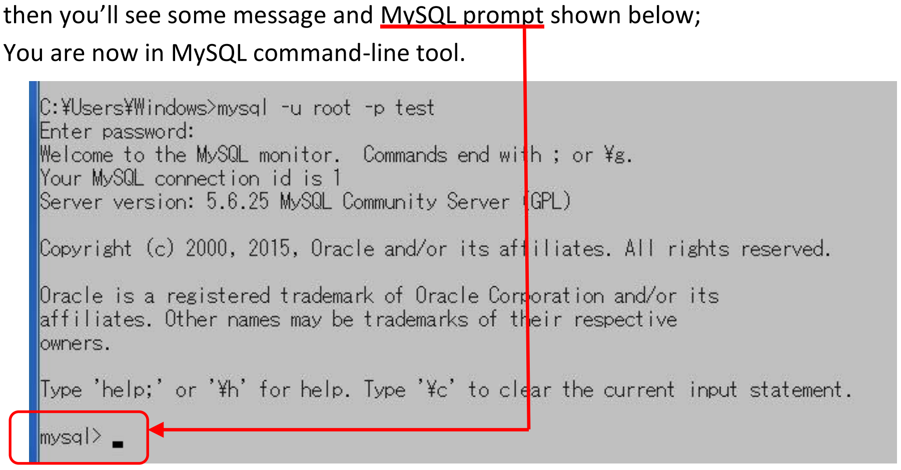
user-name	:	valid user registered on MySQL Default user 'root' can be used without password even though very dangerous.
password	:	valid password. Just after installing MySQL, user 'root' doesn't have any password. Attention! Between '-p' and *password* there's no blank.
database-name	:	valid database name.



```
C:¥Users¥Windows>mysql -u root -p test
```

Above example shows to connect MySQL database 'test' by the user 'root'. A prompt to ask password will come out.

then you'll see some message and MySQL prompt shown below;
You are now in MySQL command-line tool.



```
C:¥Users¥Windows>mysql -u root -p test
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or ¥g.
Your MySQL connection id is 1
Server version: 5.6.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '¥h' for help. Type '¥c' to clear the current input statement.

mysql>
```

You can give SQL statements or MySQL commands.

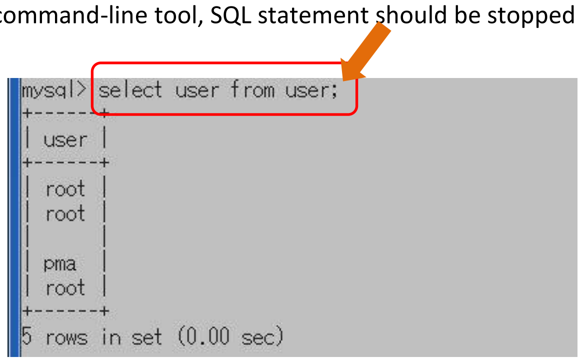
4-4-1 open one database

use *database-name*;

You can specify database name in 'mysql' command shown in the previous page.

In the command-line tool, you can open other database with 'use' command.

Attention! : In command-line tool, SQL statement should be stopped by ';'(semi-colon).



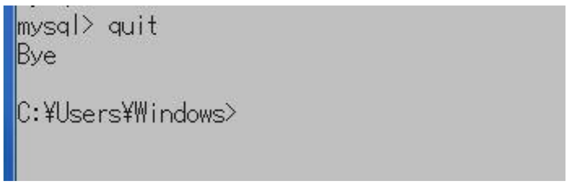
```
mysql> select user from user;
+------+
| user |
+------+
| root |
| root |
|      |
| pma  |
| root |
+------+
5 rows in set (0.00 sec)
```

4-4-2 close database and quit the command-line tool

quit;

exit;

'quit' or 'exit' command close current database and quit/exit command-line tool, and go back to Windows/DOS prompt.



```
mysql> quit
Bye

C:¥Users¥Windows>
```

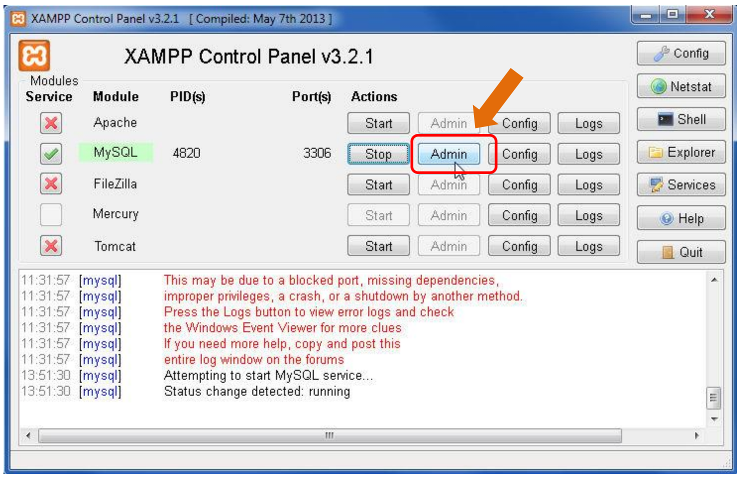
For MySQL command-line tool in detail, visit the site below ;

http://dev.mysql.com/doc/refman/5.6/en/mysql.html

5. phpMyAdmin

'phpMyAdmin' is a GUI tool for MySQL included in XAMPP and has been installed when you installed XAMPP.

You can start phpMyAdmin from XAMPP control panel, by pressing 'Admin' button on 'MySQL' line.

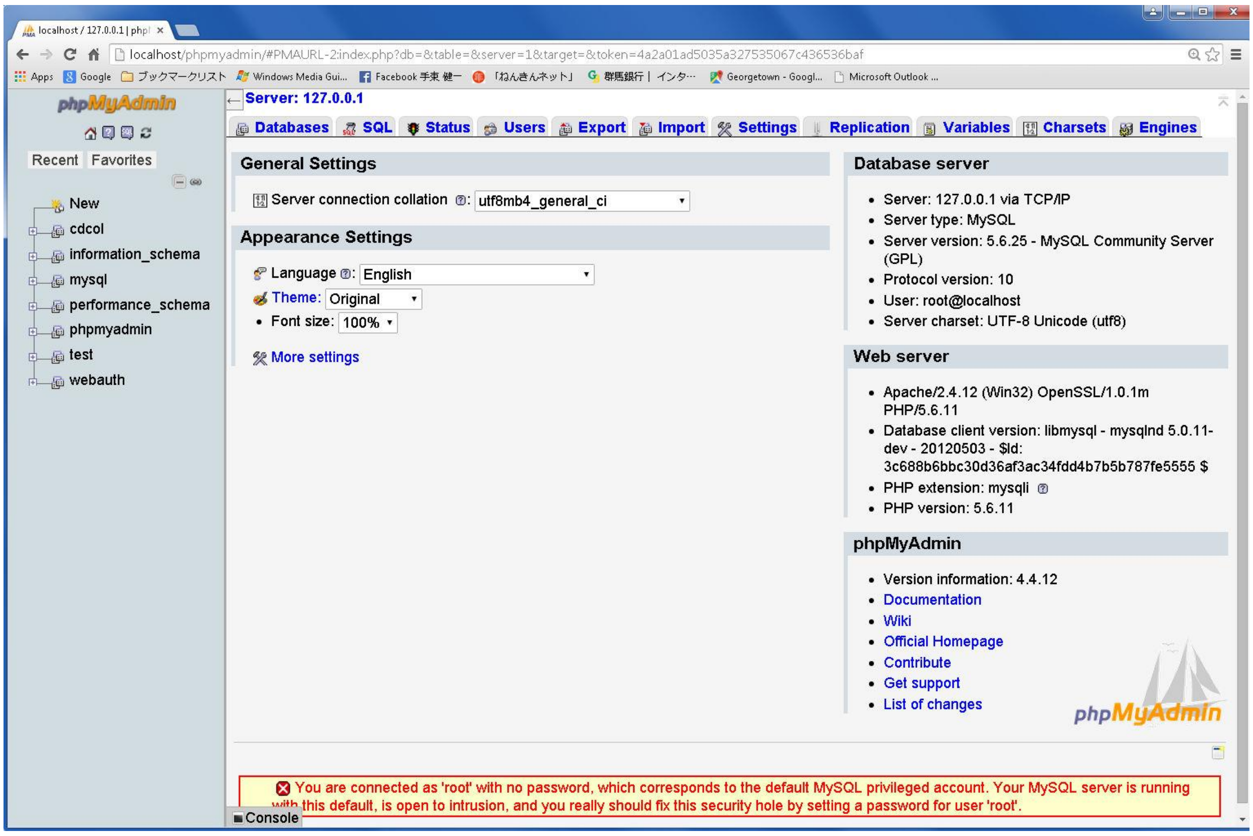

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

Service	Module	PID(s)	Port(s)	Actions			
✗	Apache			Start	Admin	Config	Logs
✓	MySQL	4820	3306	Stop	Admin	Config	Logs
✗	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
✗	Tomcat			Start	Admin	Config	Logs

```
11:31:57  [mysql]   This may be due to a blocked port, missing dependencies,
11:31:57  [mysql]   improper privileges, a crash, or a shutdown by another method.
11:31:57  [mysql]   Press the Logs button to view error logs and check
11:31:57  [mysql]   the Windows Event Viewer for more clues
11:31:57  [mysql]   If you need more help, copy and post this
11:31:57  [mysql]   entire log window on the forums
13:51:30  [mysql]   Attempting to start MySQL service...
13:51:30  [mysql]   Status change detected: running
```

Screen of phpMyAdmin is shown below;


localhost/phpmyadmin/#PMAURL-2:index.php?db=&table=&server=1&target=&token=4a2a01ad5035a327535067c436536baf

Server: 127.0.0.1

Databases | SQL | Status | Users | Export | Import | Settings | Replication | Variables | Charsets | Engines

Recent | Favorites

- New
- cdcol
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test
- webauth

General Settings

Server connection collation: utf8mb4_general_ci

Appearance Settings

- Language: English
- Theme: Original
- Font size: 100%

More settings

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MySQL
- Server version: 5.6.25 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.12 (Win32) OpenSSL/1.0.1m PHP/5.6.11
- Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - $Id: 3c688b6bbc30d36af3ac34fdd4b7b5b787fe5555 $
- PHP extension: mysqli
- PHP version: 5.6.11

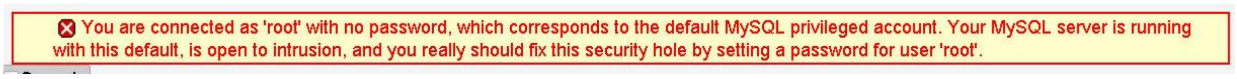
phpMyAdmin

- Version information: 4.4.12
- Documentation
- Wiki
- Official Homepage
- Contribute
- Get support
- List of changes

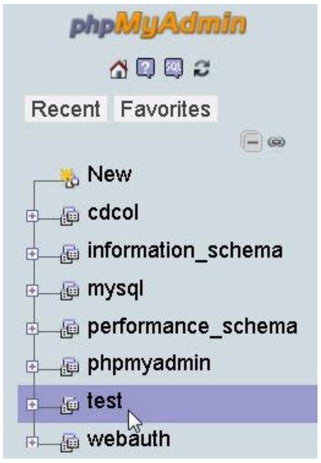
You are connected as 'root' with no password, which corresponds to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole by setting a password for user 'root'.

Console

If you see some message on a lower part, you need to fix each problem.
Such message as shown below indicates that administrative user 'root' doesn't have password. You'd better set password to 'root'.


You are connected as 'root' with no password, which corresponds to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole by setting a password for user 'root'.

To use phpMyAdmin, first you have to select database from the database list shown below or create database.



You can select database from the list or create new database.

After you select one database, then you can manipulate database, tables in the tool. You will see it later.

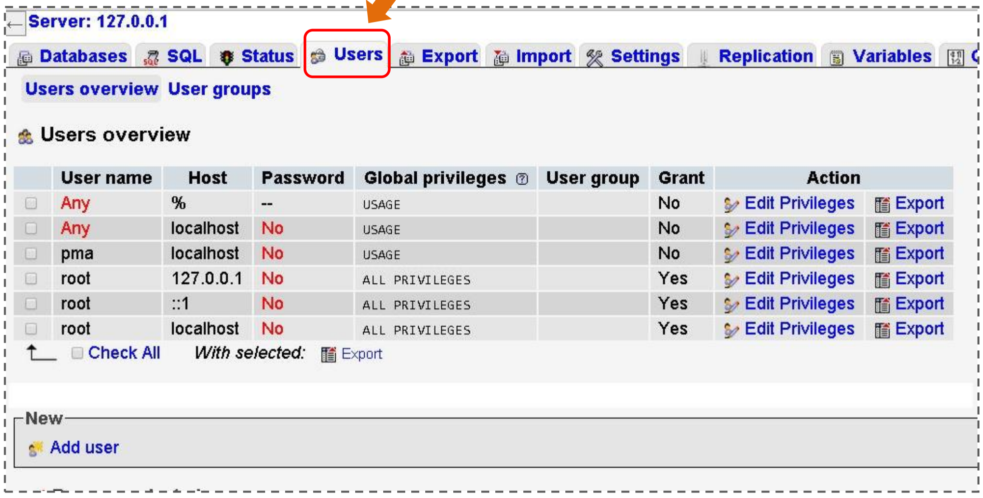
6 Making new user

In our training, we'll not use 'root' as a MySQL user because it is recommended not to use 'root' in web applications.

Our new MySQL user is;

user name(ID)	:	DBuser
user password	:	password
host	:	localhost

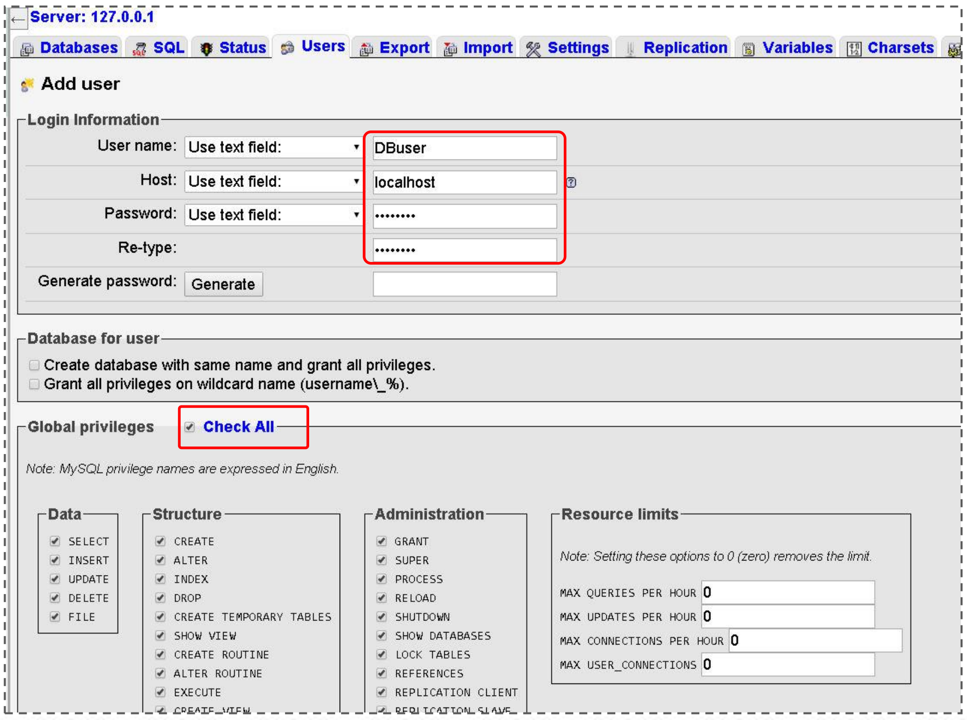
Open phpMyAdmin and select 'Users' on the menu;


Server: 127.0.0.1
Databases SQL Status Users Export Import Settings Replication Variables
Users overview User groups

Users overview

	User name	Host	Password	Global privileges	User group	Grant	Action	
☐	Any	%	--	USAGE		No	Edit Privileges	Export
☐	Any	localhost	No	USAGE		No	Edit Privileges	Export
☐	pma	localhost	No	USAGE		No	Edit Privileges	Export
☐	root	127.0.0.1	No	ALL PRIVILEGES		Yes	Edit Privileges	Export
☐	root	::1	No	ALL PRIVILEGES		Yes	Edit Privileges	Export
☐	root	localhost	No	ALL PRIVILEGES		Yes	Edit Privileges	Export

Input 4 fields; 'User name','Host','Password' and 'Re-type'.
Check a check-box 'Global Privileges – Check All' check box.
Press 'Go' button at the lower part of the screen.


Server: 127.0.0.1
Databases SQL Status Users Export Import Settings Replication Variables Charsets

Add user

Login Information

User name:	Use text field:	DBuser
Host:	Use text field:	localhost
Password:	Use text field:	••••••••
Re-type:		••••••••
Generate password:	Generate	

Database for user

☐ Create database with same name and grant all privileges.
☐ Grant all privileges on wildcard name (username_%).

Global privileges ☑ Check All

Note: MySQL privilege names are expressed in English.

Data
☑ SELECT
☑ INSERT
☑ UPDATE
☑ DELETE
☑ FILE

Structure
☑ CREATE
☑ ALTER
☑ INDEX
☑ DROP
☑ CREATE TEMPORARY TABLES
☑ SHOW VIEW
☑ CREATE ROUTINE
☑ ALTER ROUTINE
☑ EXECUTE
☑ CREATE VIEW

Administration
☑ GRANT
☑ SUPER
☑ PROCESS
☑ RELOAD
☑ SHUTDOWN
☑ SHOW DATABASES
☑ LOCK TABLES
☑ REFERENCES
☑ REPLICATION CLIENT
☑ REPLICATION SLAVE

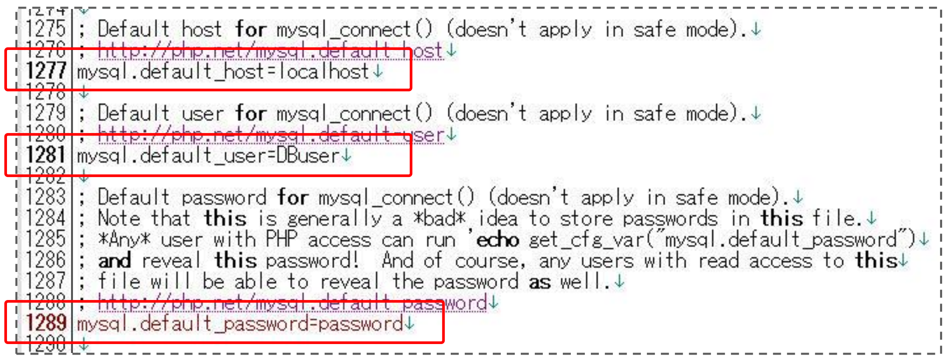
Resource limits

Note: Setting these options to 0 (zero) removes the limit.

MAX QUERIES PER HOUR	0
MAX UPDATES PER HOUR	0
MAX CONNECTIONS PER HOUR	0
MAX USER_CONNECTIONS	0

After making new user 'DBuser', we'd better update some parts related to MySQL in 'php.ini' shown below;

php.ini



```
; Default host for mysql_connect() (doesn't apply in safe mode).
; http://php.net/mysql.default-host
mysql.default_host=localhost

; Default user for mysql_connect() (doesn't apply in safe mode).
; http://php.net/mysql.default-user
mysql.default_user=DBuser

; Default password for mysql_connect() (doesn't apply in safe mode).
; Note that this is generally a *bad* idea to store passwords in this file.
; *Any* user with PHP access can run 'echo get_cfg_var("mysql.default_password")
; and reveal this password!  And of course, any users with read access to this
; file will be able to reveal the password as well.
; http://php.net/mysql.default-password
mysql.default_password=password

```

7 SQL syntax

We'll see main SQL sentences here, not all.

In this text, we use PDO, which stands for 'PHP Data Objects' and provides us same user interface for different databases like MySQL, Oracle, SQL Server and so on.

PDO is object oriented tool, so you need to get familiar with handling class and objects.

7-1 Connect MySQL

Syntax :

<connection_instance>
= new PDO(<*connection_strings*>,<*user_name*>,<*password*>)

<*connection_strings*>
"mysql:host=<*host_name*>;dbname=<*db_name*>"
<*host_name*> : In our case, it's 'localhost'
<*db_name*> : In our case, it's 'test'

<*user_name*> : user name for MySQL. In our case, it's 'DBuser'.

<*password*> : user password. In our case, it's 'password'.

<*connection_instance*> : object variable for DB. ex) $conn

For example :

```
$conn = new PDO( "mysql:host=localhost;dbname=test", 'DBuser', 'password' );
```

[Practice 45] Connecting to MySQL

Make a program 'ex45.php' ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exception handling (1)</title>
</head>
<body>
  <?php
    $host   = 'localhost';
    $user   = 'DBuser';
    $pass   = 'password';
    $db     = 'test';
    try {
      $conn   = new PDO( "mysql:host=$host;dbname=$db",$user,$pass);
      $conn->setAttribute(PDO::ATTR_ERRMODE,
                          PDO::ERRMODE_EXCEPTION );
      echo "Connected successfully<br />";
    } catch( PDOexception $e ) {
      echo "Connection failed  : " . $e->getMessage() . "<br />";
    }
  ?>
</body>
</html>
```

In the above example, you can try some exceptional cases; for example, you can try to connect database 'testA' non-existing DB name, to use 'passwordX' for password to cause password failure, and so on.

7-2 CREATE database

SQL Syntax :

```
CREATE DATABASE <database_name>
```

Using PDO, codes for creating DB will be as follows;

```
$conn  = new PDO( "mysql:host=$host;dbname=$db", $user, $pass);
       .
       .
$sql  = "CREATE DATABASE myDB";
$conn->exec( $sql );
```

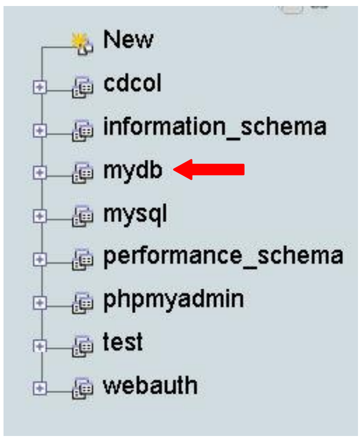
Values of $host, $db, $user and $pass should be given beforehand.
$host : host name(localhost)
$db : database name(myDB)
$user : user name(DBuser)
$pass : user password(password)

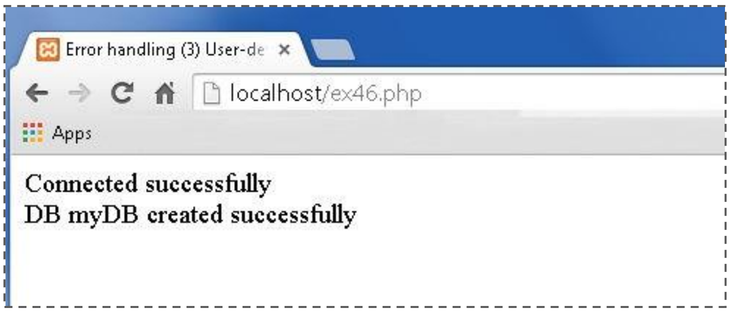
'exec()' is one method in PDO class and execute SQL statement given in argument. 'exec()' method is used when SQL will not return data set.

[Practice 46] Creating MySQL database

Make a program ‘ex46.php’ ;

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>Exception handling (1)</title>
</head>
<body>
   <?php
      $host   = ‘localhost’;
      $user   = ‘DBuser’;
      $pass   = ‘password’;
      $db     = ‘test’;
      $newDB = ‘myDB’;
      try {
         $conn   = new PDO( “mysql:host=$host;dbname=$db”,$user,$pass);
         $conn->setAttribute(PDO::ATTR_ERRMODE,
                                PDO::ERRMODE_EXCEPTION );
         echo “Connected successfully<br />”;
         $sql    = “CREATE DATABASE $newDB”;
         $result  = $conn->exec( $sql );
         if( $result === FALSE ) {
            echo “Failed to CREATE DB  : “ . $sql . “<br />”;
         } else {
            echo “DB $newDB created successfully<br />”;
         }
      } catch( PDOexception $e ) {
        echo “DB Failed  : “ . $e->getMessage() . “<br />”;
      }
   ?>
</body>
</html>
```





```
Connected successfully
DB myDB created successfully
```

7-3 CREATE table

SQL Syntax :

```
CREATE TABLE <table_name> (
        <column_name>  <data_type>  <attribute> … <attribute>,
        <column_name>  <data_type>  <attribute> … <attribute>
            …
        < column_name>  <data_type>  <attribute> … <attribute>
)
```

<table_name> : name of the table

<column_name> : name of the column

<data_type> : data type of the column
some 'data types' are shown below;

Numeric types	Text types	Date&time types
INT(size)	CHAR	DATE()
SMALLINT(size)	VARCHAR(size)	DATATIME()
BIGINT(size)	TINYTEXT	TIMESTAMP()
FLOAT(size,d)	TEXT	TIME
DOUBLE(size,d)	BLOB	YEAR()
DECIMAL(size,d)	LONGTEXT	

s
size : maximum numbers of characters, digits
d : numbers of digits to the right of decimal point

attribute	description
NOT NULL	Not allowed to have NULL as value
DEFAULT	Set default value
UNSIGNED	Stored data must have positive numbers or zero
AUTO INCREMENT	MySQL increases the value one by one each time a new record is added
PRIMARY KEY	The field is defined as unique key in the table

in detail, seSee http://www.w3schools.com/sql/sql_datatypes.asp

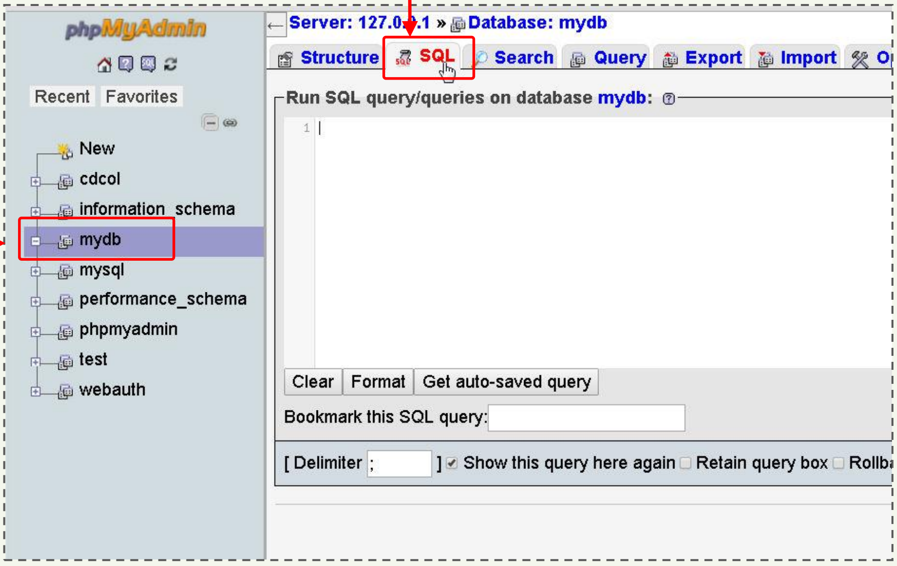
[Practice 47] Creating MySQL table

Using phpMyAdmin, make a SQL statement to create a table, table name is ‘country’ and columns are defined as follows;

Column name	Data type	size	attribute	remarks
id	int		PRIMARY KEY AUTO INCREMENT	
name	varchar	128		Country name
capitalcity	varchar	128		Capital city
population	Int		UNSIGNED	population

To make SQL statement, follow steps below;

1) open phpMyAdmin from XAMPP control panel
2) select database ‘myDB’
3) select SQL tab



7-4 Insert data into MySQL table

SQL Syntax :

```
INSERT INTO TABLE <table_name> ( <column_1>, <column_2>,... )
                VALUE( <value_1>, <value_2>, ... )
```

<column_1>, <column_2>, … : list of column name

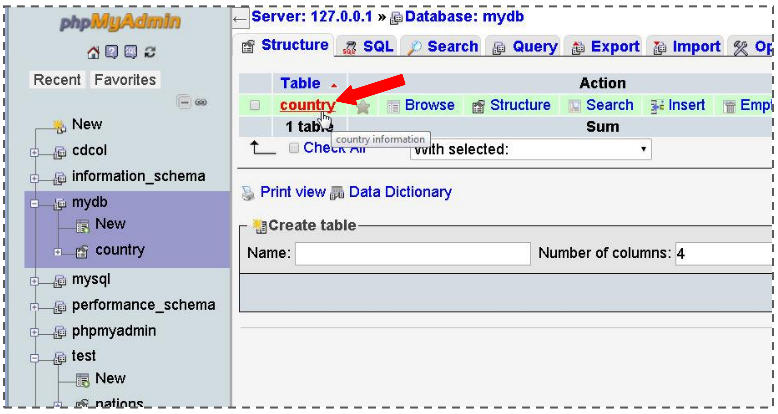
<value_1>, <value_2>, … : value list corresponding with column

There are several ways to insert rows;

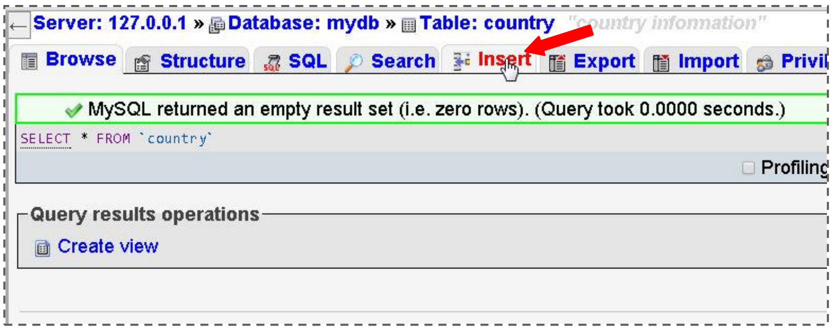
1) using 'insert' tab on phpMyAdmin, insert rows one by one
2) type in Insert SQL statement to MySQL prompt on command prompt window
3) type in Insert SQL statement on phpMyAdmin
4) make a text file with Insert SQL statement and import it with LOAD DATA INFILE statement (on prompt window or on phpMyAdmin)

7-4-1 Insert a row one by one on phpMyAdmin

1) open phpMyAdmin and select 'myDB' database. You'll see one table 'country' which we've made in the previous chapter 7-3.
Select table 'country' (press table name link)



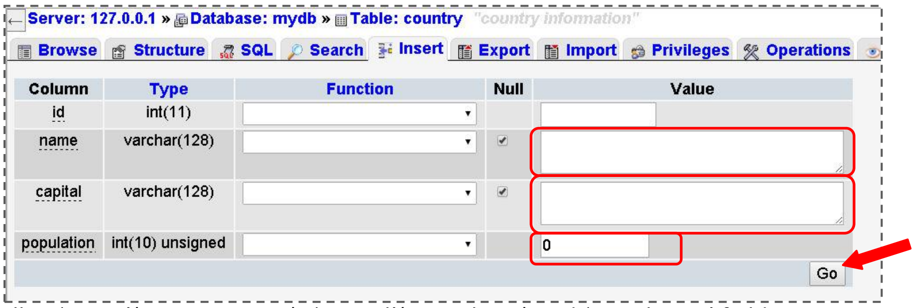
2) Select 'insert' tab



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0000 seconds.)

```
SELECT * FROM `country`
```

3) You'll see a window to insert a row one by one.
You don't need to fill 'id' field because it is defined as 'AUTO INCREMENT' field, which means value of 'id' is automatically set by MySQL.



Server: 127.0.0.1 » Database: mydb » Table: country "country information"
Browse Structure SQL Search Insert Export Import Privileges Operations
Go

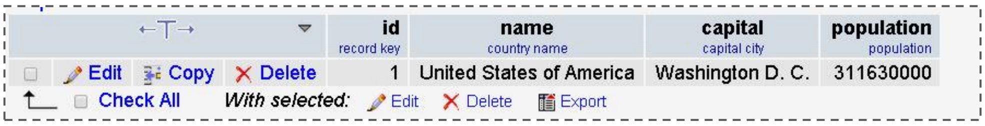
Column	Type	Function	Null	Value
id	int(11)			
name	varchar(128)		✓	
capital	varchar(128)		✓	
population	int(10) unsigned			0

Fill in 'name'(country name), 'capital'(capital city) and 'population' fields and press 'Go' button.

[Practice 48] Insert record one by one via phpMyAdmin

Insert a record shown below using window shown above.

Country name	:	United States of America
Capital city	:	Washington D. C.
Population	:	311630000

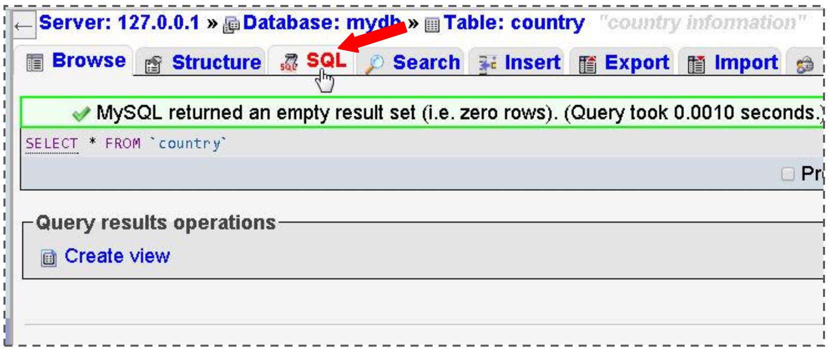


id record key	name country name	capital capital city	population population
1	United States of America	Washington D. C.	311630000

7-4-2 Insert records by Insert SQL statement on phpMyAdmin

1) On phpMyAdmin, selet 'country' table (same as 7-4-1 1))

2) Select 'SQL' tab

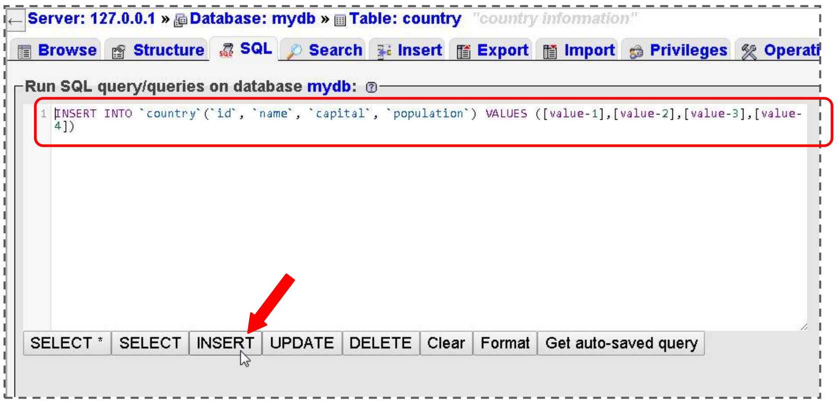


Server: 127.0.0.1 » Database: mydb » Table: country "country information"
Browse Structure SQL Search Insert Export Import
Query results operations
Create view

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.
SELECT * FROM `country`
```

3) SQL tab window is shown below. You can find 'INSERT' button at the bottom. Press it, then you'll find a template of INSERT SQL statement.Now,

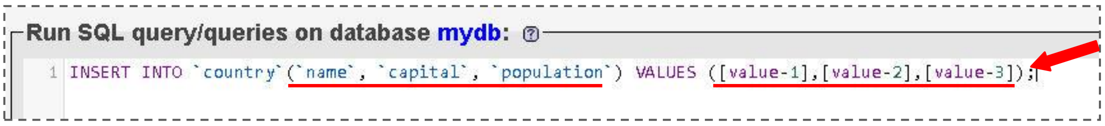
we don't need to input 'id' field, so you can erase 'id' field and one value element.


Server: 127.0.0.1 » Database: mydb » Table: country "country information"
Browse Structure SQL Search Insert Export Import Privileges Operati
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

Run SQL query/queries on database mydb:

```
INSERT INTO `country`(`id`, `name`, `capital`, `population`) VALUES ([value-1],[value-2],[value-3],[value-4])
```

4) Now, the template will be just like shown below;
   Don't forget to put ';' at the end.



Run SQL query/queries on database mydb:

```
INSERT INTO `country`(`name`, `capital`, `population`) VALUES ([value-1],[value-2],[value-3]);
```

You can copy this template as many as you need.

5) After you make SQL statement, press 'Go' button at thebottom.

6) To insert multiple rows in one SQL statement, repeat
   value list enclosed by bracket separated by ','(comma).

```
INSERT INTO <table_name> ( <column_1>, <column_2>,… )
  VALUES
  (  (<value_1_1>,<value_1_2>,… ),
     (<value_2_1>,<value_2_2>,… ),
         ……
     (<value_n_1>,<value_n_2>,… )
  );
```

[Practice 49] Insert record by INSERT SQL statement on phpMyAdmin

Insert two records shown below by INSERT SQL statement on phpMyAdmin

Country name	:	Mexico
Capital city	:	Mexico City
Population	:	112322800
Country name	:	Canada
Capital city	:	Ottawa
Population	:	33573000

7-4-3 Insert records by LOAD DATA INFILE statement

LOAD DATA INFILE statement will insert rows into a table from external text file. We need to make a text file with INSERT SQL statement.

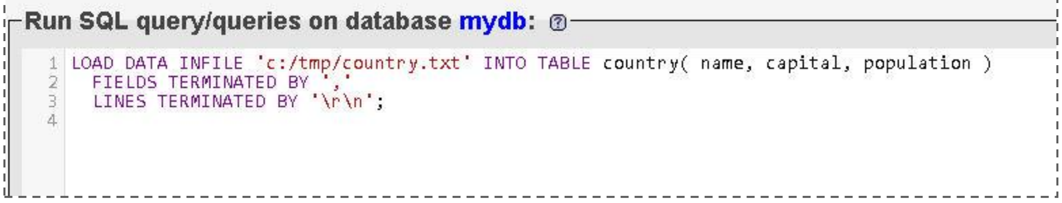
1) Make a text file named 'country.txt' shown below;

country.txt

```
Brasil,Brasilia, 202714700
Peru,Lima, 29132000
```

2) Save it in some folder like 'C:/tmp'.

3) Open SQL tab on table 'country' on phpMyAdmin just like 7-4-2, and type in LOAD DATA INFILE statement as below;

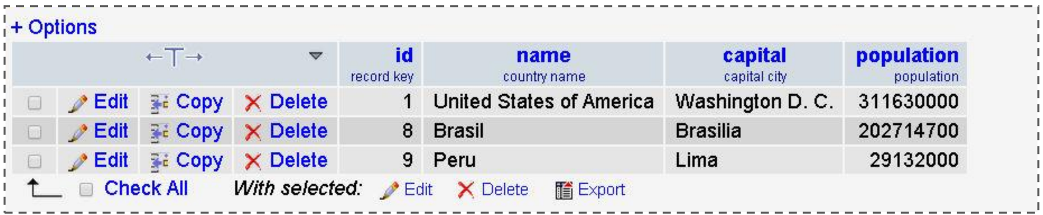


```
LOAD DATA INFILE 'c:/tmp/country.txt' INTO TABLE country( name, capital, population )
  FIELDS TERMINATED BY ','
  LINES TERMINATED BY '\r\n';
```

4) Press 'Go' button

5) If you see some error message, you must follow the instruction shown in message.

6) You can get data list as shown below;



	id record key	name country name	capital capital city	population population
Edit Copy Delete	1	United States of America	Washington D. C.	311630000
Edit Copy Delete	8	Brasil	Brasilia	202714700
Edit Copy Delete	9	Peru	Lima	29132000

[Practice 50] Insert record by LOAD DATA INFILE statement on phpMyAdmin

Make a text file 'country2.txt' with data below;

Trinidad and Tobago,Port of Spain,1339000
Urguay,Montevideo,3477780
Guyana,Georgetown,773000

Insert these 3 records into 'country' table using LOAD DATA INFILE statement.

7-5 SELECT statement

SELECT statement will search data from our database and tables.

Syntax:

SELECT <*column_1*>,<*column_2*>,…,<*column_n*> FROM <*table_name*>;

SELECT * FROM <*table_name*>;

SELECT DISTINCT <*column_1*>,…,<*column_n*> FROM <*table_name*>;

SELECT DISTINCT * FROM <*table_name*>;

SELECT <*column_1*>,<*column_2*>,…,<*column_n*> FROM <*table_name*>
WHERE <*column_1*> <*operator*> <*value*> (AND|OR <*condition*> …);

<column_1>…	:	column name in a table
<table_name>	:	table name
<operator>	:	comparison operator. See 7-5-x in details
<condition>	:	condition consists of column name, operator and value
‘*’	:	‘*’ indicates that all fields in a table will be selected

ex) SELECT name from country;
select column ‘name’ from table ‘country’.

ex) SELECT * from country;
select all columns from table ‘country’.

ex) SELECT * from country WHERE population > 100000000;
select all columns from table ‘country’
on condition where value of population is greater than 100000000.

ex) SELECT * from country WHERE population > 100000000
ORDER BY name;
select all columns from table ‘country’
on condition where value of population is greater than 100000000
and the result are sorted by name in alphabetic ascending order.
(in ORDER clause, ASC is default)

ex) SELECT * from country WHERE population > 1000000
ORDER BY population DESC;
select all columns from table 'country'
on condition where value of population is greater than 1000000
and the result are sorted by population in descending order.

7-5-1 SELECT with JOIN keyword

Suppose that here are 3 tables 'sales','product' and 'customer'

'sales' able

date	orderID	product	qty	customer
8/24	0824001	100	4	C002
8/25	0825001	102	2	A010
8/25	0825003	103	12	C002

'product' table

ID	name	price
100	A	250
101	B	830
102	C	145
103	D	53

'customer' table

ID	name
A010	John
B005	Kate
C002	Mary
C004	Steve

1) Suppose that your boss wants to know which product wasn't sold at all.
How you can get from transaction table 'sales' and master table 'product'?

In such cases, we can use JOIN keyword in SELECT statement.

```
SELECT product.id, product.name, sales.qty FROM product
  LEFT JOIN sales ON product.ID = sales.product
  ORDER BY product.ID
```

This SQL will make such list below;

result set

product.ID	product.name	sales.qty
100	A	4
101	B	null
102	C	2
103	D	12

'sales.qty = null' means that 'this product wasn't sold at all.
Then the SQL statement which your boss wants is;

```
SELECT product.id, product.name, sales.qty FROM product
   LEFT JOIN sales ON product.ID = sales.product
   WHERE sales.qty IS NULL
   ORDER BY product.ID
```

[Practice 51] SELECT statement with JOIN

Make SQL statement to know 'Who didn't buy anything during these days?'

7-6 CREATE VIEW statement

'VIEW' is a virtual table produced by some SQL statement.

For example, in tables shown in 7-5-1, we can calculate amount of each order shown below;

```
SELECT sales.orderID, sales.qty * product.price as amount
   FROM sales, product
   WHERE sales.product = product.ID;
```

Then we'll get a result set shown below;

result set

sales.orderID	amount
0824001	1000
0825001	290
0825002	636

This result set is not an actual table, but we can treat the result set just like a real table.

If we need to use same view in other cases, then we can register this SQL as a VIEW. It isn't a table, it's only a SQL statement on MySQL database. Just like this;

```
CREATE VIEW vSalesAmount AS
   SELECT sales.orderID, sales.qty * product.price as amount
      FROM sales, product
      WHERE sales.product = product.ID;
```

Then we can reuse this VIEW in the following simple SQL;

```
SELECT * FROM vSalesAmount;
```

7-6 UPDATE statement

UPDATE statement is used to update rows in a table.

Syntax:

```
UPDATE <table_name>
   SET <column_name> = <value> (, <column_name = <value>, … )
   WHERE <column_name> <operator> <value>;
```

ex) UPDATE product SET price = 300 WHERE ID = 100;

This UPDATE statement will change the 'price' of product 'A'('ID' = 100) to 300.

ex) UPDATE sales SET qty = 15, customer = 'B005' WHERE orderID = '0835001';

This UPDATE statement will change the 'qty' and 'customer' of sales
where 'oerderID' is equal to '0825001'.

Attention ; In UPDATE statement, don't forget to give WHERE condition, otherwise
you will get result where all rows are updated and can't be recovered.

7-7 DELETE statement

DELETE statement is used to dekete rows in a table.

Syntax:

```
DELETE FROM <table_name>
   WHERE <column_name> <operator> <value>;
```

Attention ; Don't forget to give WHERE condition,
otherwise you will have lost all rows in the table and there's no way or method to recover deleted rows.

There are so many SQL statement in MySQL. We can't follow all statements here, so you can visit the following URL to check SQL statements;

https://dev.mysql.com/doc/refman/5.0/en/sql-syntax.html
for data definition statement:
https://dev.mysql.com/doc/refman/5.0/en/sql-syntax-data-definition.html
for data manipulation statement:
https://dev.mysql.com/doc/refman/5.0/en/sql-syntax-data-manipulation.html

SQL Injection

What is SQL Injection?

Wikipedia says;

> SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.

Web application is one of 'data-driven application'.
As we studied already, HTML form with input fields and buttons will drive some program on the server.

We'll develop very simple web application here and try to check how SQL injection is done on a web application and how we can guard our apps from injection attacks.

Our simple example is :
A HTML form for log in into our application.
A table with user information including password field.
Using these forms and table, we'll try to log in.
If user ID and password are valid, welcome message will be displayed.

1. Preparation

Before entering into this topic, we'll make some tables and HTML forms.

1-1 'user' table

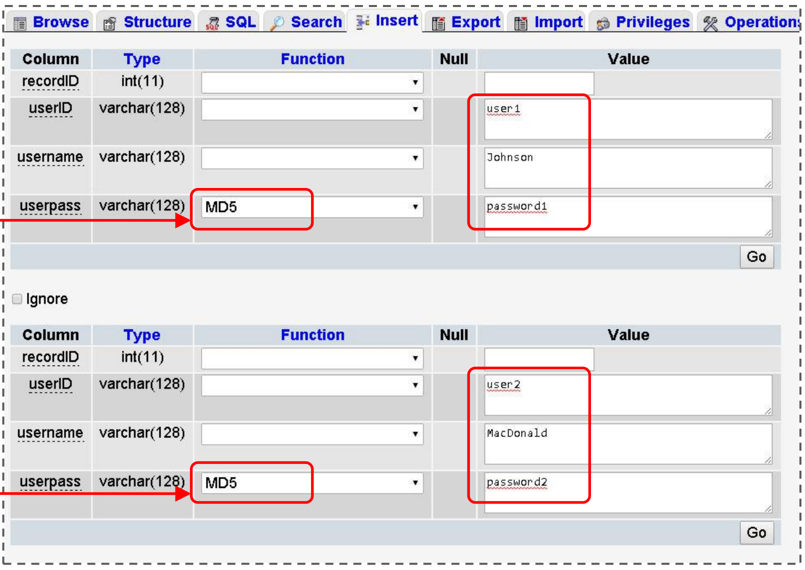
Table structure

Column name	Data type	size	Attribute	remarks
recordID	int		PRIMARY KEY AUTO INCREMENT	Record key
userID	varchar	20	UNIQUE	User ID
username	varchar	128		User name
userpass	varchar	128	PASSWORD	password

data list

recordID	userID	username	userpass
1	user1	Johnson	password1
2	user2	MacDonald	password2

After creating table 'user', add 2 rows shown above;



Column	Type	Function	Null	Value
recordID	int(11)			
userID	varchar(128)			user1
username	varchar(128)			Johnson
userpass	varchar(128)	MD5		password1

Go

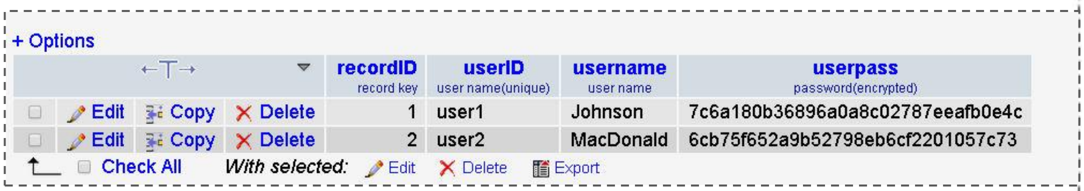
Ignore

Column	Type	Function	Null	Value
recordID	int(11)			
userID	varchar(128)			user2
username	varchar(128)			MacDonald
userpass	varchar(128)	MD5		password2

Go

'password' column should be saved encrypted.
We'll use 'MD5' encryption.
Select 'MD5' from dropdown list.

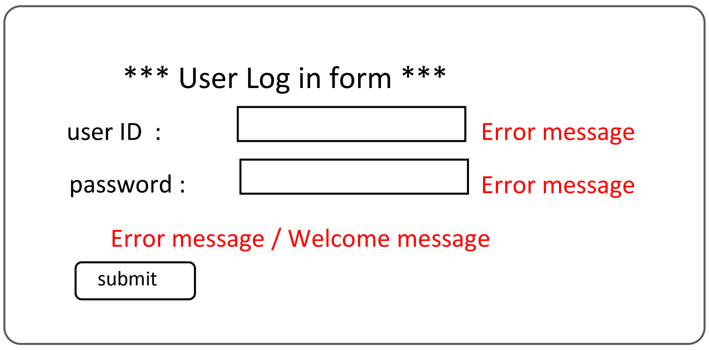
After inserting two rows, contents of the table is show below;



	recordID record key	userID user name(unique)	username user name	userpass password(encrypted)
Edit Copy Delete	1	user1	Johnson	7c6a180b36896a0a8c02787eeafb0e4c
Edit Copy Delete	2	user2	MacDonald	6cb75f652a9b52798eb6cf2201057c73

'password' column is encrypted.
Anyone can't decrypt the 'password'.

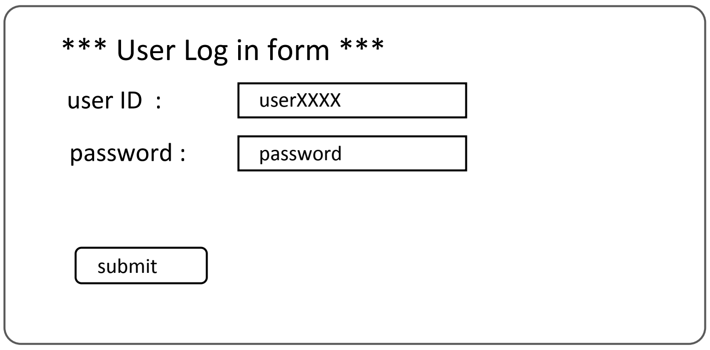
1-2 Login form (ex52.php)



As for source code for this form(ex52.php), see page 137 and further.

2 Trial

2-1 give a wrong 'userID' intentionally



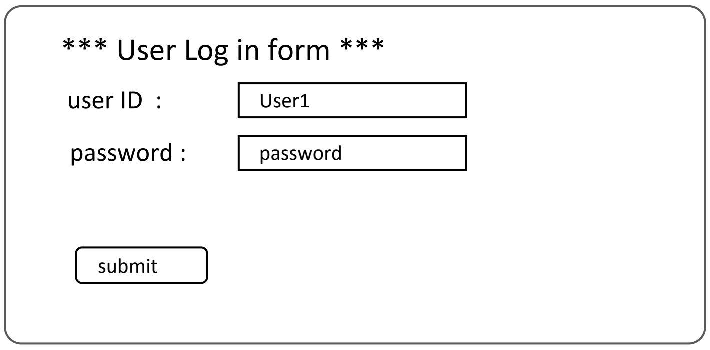
*** User Log in form ***

user ID :	userXXXX
password :	password

submit

You'll see some error message on the screen. It's normal.

2-2 give a correct 'userID' and 'password'



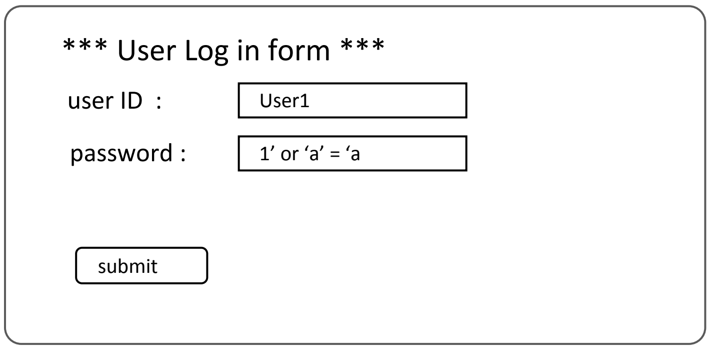
*** User Log in form ***

user ID :	User1
password :	password

submit

You'll see welcome message with user's name. It's normal, too.

2-3 give a correct 'userID', but for 'password', type in as shown bellow;



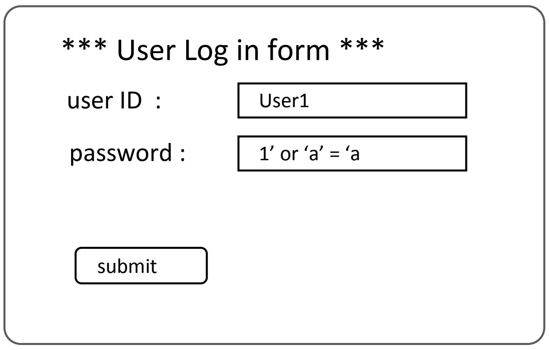
*** User Log in form ***

user ID :	User1
password :	1' or 'a' = 'a

submit

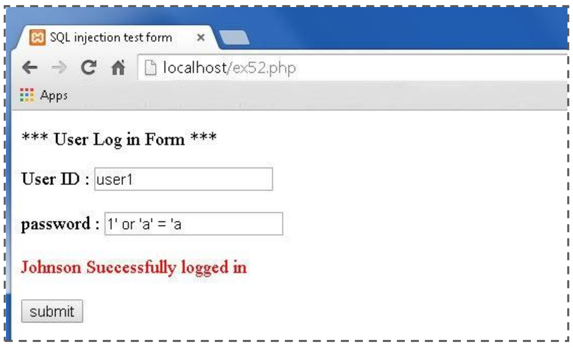
What happened? Did you see welcome message though you entered wrong password? Why could you log in with wrong password?
This is one very simple example of SQL injection.
An example of web application without any guard against SQ injection.

3 Why did welcome message appear even though you typed a wrong password?


*** User Log in form ***

user ID :	User1
password :	1’ or ‘a’ = ‘a

submit


SQL injection test form

localhost/ex52.php

*** User Log in Form ***

User ID : user1

password : 1' or 'a' = 'a

Johnson Successfully logged in

submit

Let’s check the SQL statement which ‘ex52.php’ execute to authenticate.

```
$query      = $conn->prepare("SELECT * FROM user WHERE userID = '" . $userID . "' AND userpass = '" . $password . "'");
```

Our program will assign values of ‘user ID’ and ‘password’ from the form to ‘$userID’ and ‘$password’ in the SQL statement above.
Then the SQL statement will be ;

SELECT * FROM user WHERE userID = ‘user1’ and userpass = ‘1’ or ‘a’ = ‘a’;

How is the WHERE condition evaluated? There are 3 conditions;

```
userID = ‘user1’   ------ condition 1
userpass = ‘1’     ------  condition 2
‘a’ = ‘a’          ------  condition 3
```

<condition 1> is true or false(it depends on a record)
<condition 2> is false (any record doesn’t have such password)
<condition 3> is true (it’s always true)

Then
“<condition 1> and <condition 2>” will be always false,
but “(<condition 1> and <condition 2>) or <condition 3>” will be always true because <condition 3> is always true.

This is because the SQL statement above will return a record set with all rows in a ‘user’ table, because all rows will meet the WHERE condition.

Finally, anyone can log in without password information.
This is a typical example of SQL injection.

In the next page, we’ll study how we can guard our web apps from SQL injection attacks.

4 How to guard our web apps from SQL injection attacks

4-1 Encript ‘password’ field on MySQL using md5() function or sha1() function

4-2 Prevent single and double quotes used in ‘password’ field

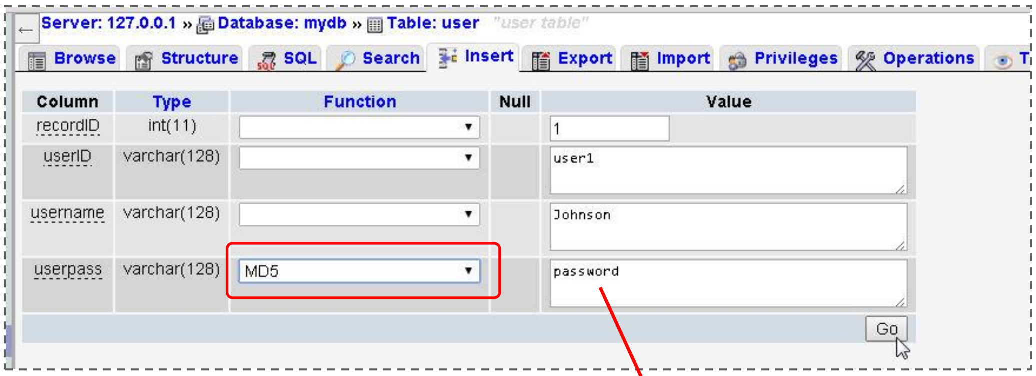
4-3 Use PDO or MySQLi to access MySQL DB in PHP programs

4-4 Use ‘prepared statements’(you can say ‘parameter query’)

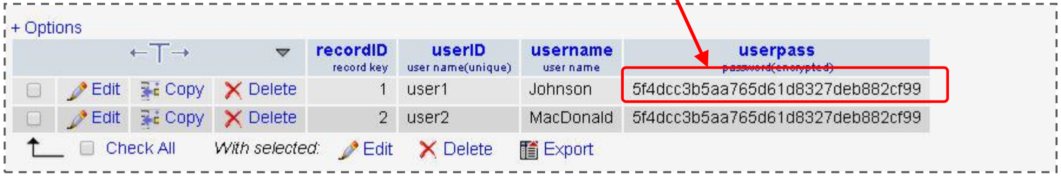
Here’s one example for the previous case ;

```
function getUserRec( $userID, $password, &$erMessage, &$passwordError ) {
   $host   = 'localhost';
   $user   = 'DBuser';
   $pass   = 'password';
   $db     = 'myDB';

   try {
      $conn        = new PDO( "mysql:host=$host;dbname=$db",$user,$pass);
      $conn->setAttribute(PDO::ATTR_ERRMODE,
                         PDO::ERRMODE_EXCEPTION );
      // definition of parameter query
      $query       = $conn->prepare("SELECT * FROM user WHERE userID = ? AND userpass = md5( ? )");
      // parameter query execution
      $query->execute( array( $userID, $password ) );
      $resultSet    = $query->fetchall();
      if( is_array( $resultSet ) and count( $resultSet ) > 0 ) {
         $returnValue        = $resultSet;
      } else {
         $erMessage          = "ID or password not matched";
         $returnValue        = FALSE;
      }
   } catch( PDOexception $e ) {
      $erMessage    = "Error :" . $e->getMessage();
      $returnValue   =  FALSE;
   }
   unset( $query );
   unset( $conn );
   return $returnValue;
}

function checkUserID( &$userID, &$password, &$userIDerror, &$passwordError, &$resultSet) {
   $result              = true;
   // check for userID
   if( empty( $_POST[ "userID" ] ) ) {
      $userIDerror    = "userID is required";
      $result          = false;
   }
   if( ! preg_match( "/^[a-zA-Z0-9 ]*$/", $_POST[ "userID" ] ) ) {
      $userIDerror      = "illegal letters included";
      $result          = false;
   }
   if( empty( $_POST[ "password" ] ) ) {
      $passwordError    = "password is required";
      $result          = false;
   }
   // avoid single, double quote in password field
   if( ! preg_match( "/^[a-zA-Z0-9 #$%&+*?=]*$/", $_POST[ "password" ] ) ) {
      $passwordError    = "illegal letters included";
      $result          = false;
   }
   // escaping input data for XSS attack
   $userID             = escapeForInput( $_POST[ "userID" ] );
   $password          = escapeForInput( $_POST[ "password" ] );
   $password          = $_POST[ "password" ];
   if( $result ) {
      // get user record from database
      $resultSet       = getUserRec( $userID, $password, $userIDerror, $passwordError );
      if( $resultSet === FALSE ) {
         $result        = false;
      }
   }
   return $result;
}
```

We encrypt ‘password’ field in ‘user’ table using md5() function;



Column	Type	Function	Null	Value
recordID	int(11)			1
userID	varchar(128)			user1
username	varchar(128)			Johnson
userpass	varchar(128)	MD5		password



	recordID record key	userID user name(unique)	username user name	userpass password(encrypted)
Edit Copy Delete	1	user1	Johnson	5f4dcc3b5aa765d61d8327deb882cf99
Edit Copy Delete	2	user2	MacDonald	5f4dcc3b5aa765d61d8327deb882cf99

We prevent single and double quotes in ‘password’ fields;

```
if( ! preg_match( "/^[a-zA-Z0-9 #$%&+*?=]*$/", $_POST[ "password" ] ) ) {
    $passwordError    = "illegal letters included";
    $result           = false;
}
```

preg_match() function searches character/characters pattern in given strings (in 2nd argument) in a pattern (in 1st argument).

In the example show above;

1st argument ”/^[a-zA-Z0-9 #$%&+?=]*$/”

‘/’(slash) delimiter for preg_match() function

^(caret) start point of string

[]	Matches a single character that is contained within the brackets.
[]*	Matches the preceding element zero or more times.
a-z	lower case letters
A-Z	upper case letters
0 -9	digits
$	end point of string

^[a-zA-Z0-9 #$%&+?=]*$ means;

from start point to end point of string, repetition of

one of (lower case alphabets) or (upper case of alphabets)

or (digits from 0 to 9) or (one of special letters ‘ #$%&+?=’)

preg_match() function works on 'regular expression' mechanism.
We need to know 'reglar expression' mechanism, but we don't have enough time.
Visit and see the following web pages for further learning;
  for preg_match() function :
    http://php.net/manual/en/function.preg-match.php
  for regular expression;
    https://en.wikipedia.org/wiki/Regular_expression
    http://www.regular-expressions.info/php.html

Finally, we define a parameter query;

```
// definition of parameter query
$query      = $conn->prepare("SELECT * FROM user WHERE userID = ? AND userpass = md5( ? )");
// parameter query execution
$query->execute( array( $userID, $password ) );
$resultSet    = $query->fetchall();
```

In SELECT statement, you'll see two '?'(question marks). These are called 'Placeholder'.
$query is a connection object which PDO has generated (see PHP source in page.5) and '$query->execute' is an PDO execute method with two parameters here ('array($userID, $password)').

These two parameters will be assigned to two placeholders in SELECT statement respectively

$query->fetch or $query->fetchall will get rows which meet the 'WHERE condition' and have been extracted from table. 'fetch' will get rows one by one, and 'fetchall' will get all rows at one time. The result will be set into an array which is return value, if there's no row, then the return value will be 'FALSE'.

'prepared statement' (or 'parameter query') in PDO has a mechanism to guard from SQL injection attacks in itself, so we can give raw value of '$_POST' (or $_GET) variable to parameters.

The important point is ;
  "When we need to give input data from a web form to SQL statement,
  don't forget to use 'prepared statements(parameter query)'".

‘ex52.php’ without any guard against SQL injection attacks

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>SQL injection test form</title>
   <style>
     .error {color: #FF0000;}
   </style>
</head>
<body>
   <?php
      function escapeForInput( $data ) {
         $data        = trim($data);
         $data        = stripslashes($data);
         $data        = htmlspecialchars($data);
         return $data;
      }
      function getUserRec( $userID, $password, &$erMessage, &$passwordError ) {
         $host       =  'localhost';
         $user       =  'DBuser';
         $pass       =  'password';
         $db         =  'myDB';

         try {
            $conn          =  new PDO( "mysql:host=$host;dbname=$db",$user,$pass);
            $conn->setAttribute(PDO::ATTR_ERRMODE,
                               PDO::ERRMODE_EXCEPTION );
            $query         =  $conn->prepare("SELECT * FROM user WHERE userID = '" . $userID . "' AND
userpass = '" . $password . "'");
            $query->execute( );
            $resultSet     =  $query->fetchall();
            if( is_array( $resultSet ) and count( $resultSet ) > 0 ) {
               $returnValue        = $resultSet;
            } else {
               $erMessage          = "ID or password not matched";
               $returnValue        = FALSE;
            }
         } catch( PDOexception $e ) {
            $erMessage             = "Error :" . $e->getMessage();
            $returnValue           =  FALSE;
         }
         unset( $query );
         unset( $conn );
         return $returnValue;
      }

      function checkUserID( &$userID, &$password, &$userIDerror, &$passwordError, &$resultSet) {
         $result                   =  true;
         // check for userID
         if( empty( $_POST[ "userID" ] ) ) {
            $userIDerror          = "userID is required";
            $result               = false;
         }
         if( ! preg_match( "/^[a-zA-Z0-9 ]*$/", $_POST[ "userID" ] ) ) {
            $userIDerror         =  "illegal letters included";
            $result               = false;
         }
```

(continued)

```
        if( empty( $_POST[ "password" ] ) ) {
            $passwordError     = "password is required";
            $result                   = false;
        }
        // escaping input data for XSS attack
        $userID                   = escapeForInput( $_POST[ "userID" ] );
        $password               = escapeForInput( $_POST[ "password" ] );
        $password               = $_POST[ "password" ];
        if( $result ) {
            // get user record from database
            $resultSet           =  getUserRec( $userID, $password, $userIDerror, $passwordError );
            if( $resultSet === FALSE ) {
                $result              =  false;
            }
        }
        return $result;
    }
    // process status   true:successfull,  false:have some errors
    $status                       =  true;
    // set initial message / clear error message field
    $userIDerror              =   "Alphabet,Numerics,underscore";
    $passwordError        =   "";
    $errorMessage          =   "";
    // clear echo back field
    $userID                       =   "";
    $password                  =   "";
    $resultSet                  =   FALSE;
    //  check initial call or not
    if( array_key_exists( 'submit', $_POST  ) ) {
        $userIDerror         = "";
        $passwordError    = "";
        // check ID and password by DB. if error, return false
        if( checkUserID( $userID, $password, $userIDerror, $passwordError, $resultSet ) ) {
            $errorMessage  =   $resultSet[0][ "username" ] . "  Successfully logged in";
            $status        = TRUE;
        }
    }
  ?>
  <div>
    <p>*** User Log in Form ***</p>
    <form action="ex52.php" method="POST">
        User ID  :  <input type="text" name="userID" value="<?php echo $userID; ?>" />
                  <span class="error"><?php echo $userIDerror; ?></span>
        <br /><br />
        password :  <input type="password" name="password" value="<?php echo $password; ?>" />
                  <span class="error"><?php echo $passwordError; ?></span>
        <br /><br />
        <span class="error"><?php echo $errorMessage; ?></span>
        <br /><br />
        <input type="submit" name="submit" value="submit" />
    </form>
  </div>
</body>
</html>
```

‘ex52,php’ with guard against SQL injection attacks

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>SQL injection test form</title>
   <style>
     .error {color: #FF0000;}
   </style>
</head>
<body>
   <?php
      //  function for anti XSS attacks
      function escapeForInput( $data ) {
         $data       =  trim($data);
         $data       =  stripslashes($data);
         $data       =  htmlspecialchars($data);
         return $data;
      }
      function getUserRec( $userID, $password, &$erMessage, &$passwordError ) {
         $host       =   'localhost';
         $user       =   'DBuser';
         $pass       =   'password';
         $db         =   'myDB';

         try {
            $conn        =  new PDO( "mysql:host=$host;dbname=$db",$user,$pass);
            $conn->setAttribute(PDO::ATTR_ERRMODE,
                                 PDO::ERRMODE_EXCEPTION );
            // definition of parameter query with placeholders
            $query       =  $conn->prepare("SELECT * FROM user WHERE userID = ?
                                                             AND userpass = md5( ? )");
            // parameter query execution
            $query->execute( array( $userID, $password ) );
            $resultSet  =  $query->fetchall();
            if( is_array( $resultSet ) and count( $resultSet ) > 0 ) {
               $returnValue       = $resultSet;
            } else {
               $erMessage         = "ID or password not matched";
               $returnValue       = FALSE;
            }
         } catch( PDOexception $e ) {
            $erMessage            = "Error :" . $e->getMessage();
            $returnValue          =  FALSE;
         }
         // release objects
         unset( $query );
         unset( $conn );
         return $returnValue;
      }

      function checkUserID( &$userID, &$password, &$userIDerror, &$passwordError, &$resultSet) {
         $result              =  true;
         // check for userID
```

(continued)

```
        if( empty( $_POST[ "userID" ] ) ) {
            $userIDerror      = "userID is required";
            $result           = false;
        }
        if( ! preg_match( "/^[a-zA-Z0-9 ]*$/", $_POST[ "userID" ] ) ) {
            $userIDerror      = "illegal letters included";
            $result           = false;
        }
        if( empty( $_POST[ "password" ] ) ) {
            $passwordError = "password is required";
            $result           = false;
        }
        // avoid single, double quote in password field
        if( ! preg_match( "/^[a-zA-Z0-9 #$%&+*?'=]*$/", $_POST[ "password" ] ) ) {
            $passwordError = "illegal letters included";
            $result           = false;
        }
        // escaping input data for XSS attack
        $userID               = escapeForInput( $_POST[ "userID" ] );
        $password             = escapeForInput( $_POST[ "password" ] );
        $password             = $_POST[ "password" ];
        if( $result ) {
            // get user record from database
            $resultSet        = getUserRec( $userID, $password, $userIDerror, $passwordError );
            if( $resultSet === FALSE ) {
                $result       = false;
            }
        }
        return $result;
    }
    // process status   true:successfull,  false:have some errors
    $status                   = true;
    // set initial message / clear error message field
    $userIDerror              = "Alphabet,Numerics,space,underscore";
    $passwordError            = "Alphabet,Numerics,space,_#$%&+*?=";
    //$passwordError          = "";
    $errorMessage             = "";
    // clear echo back field
    $userID                   = "";
    $password                 = "";
    $resultSet                = FALSE;
    //  check initial call or not
    if( array_key_exists( 'submit', $_POST  ) ) {
        $userIDerror          = "";
        $passwordError        = "";
        // check ID and password by DB. if error, return false
        if( checkUserID( $userID, $password, $userIDerror, $passwordError, $resultSet ) ) {
            $errorMessage  = $resultSet[0][ "username" ] . "  Successfully logged in";
            $status           = TRUE;
        }
    }
?>
```

(continued)

```
    <div>
      <p>*** User Log in Form ***</p>
      <form action="ex52.php" method="POST">
        User ID  :  <input type="text" name="userID" value="<?php echo $userID; ?>" />
              <span class="error"><?php echo $userIDerror; ?></span>
        <br /><br />
        password :  <input type="password" name="password" value="<?php echo $password; ?>" />
              <span class="error"><?php echo $passwordError; ?></span>
        <br /><br />
        <span class="error"><?php echo $errorMessage; ?></span>
        <br /><br />
        <input type="submit" name="submit" value="submit" />
      </form>
    </div>
  </body>
  </html>
```

OOP in PHP

OOP stands for Object Oriented Programming.

1. What's OOP?

   "OOP is a programming paradigm based on the concept of 'objects'".
   (Wikipedia)

   What's "the concept of 'object'"?
   'object' has its states and behaviors;
   state ------- we can call it 'data' or 'fields' or 'attributes' or 'properties'
   behavior ---- we can call it 'methods' or 'procedures'

   Then we can say
   'object' has data/properties which show the state of the 'object'.
   'object' has methods which shows behaviors of the 'object'.

   for example;
   'Car' is one object:
   'Car' has its color, steering, max speed, seating capacity and so on
   Car's maker may also its attributes.
   And 'Car' runs, stops, go straight, turn (to left / right), go back and so on.
   Then we can define 'Car' as an object;

'Car' object	
Data / properties	Color steering transmission max speed seating capacity maker
Methods	Run stop go straight turn to left turn to right go back

   Suppose here are two cars, one is mine and the other is yours;

'Car' object	My car	Your car
Properties	color: blue steering: right transmission: automatic max speed: 180km/h seating capacity: 6 maker: Toyota	color: red steering: left transmission: manual max speed: 240km/h seating capacity: 2 maker: Ford
Methods	run stop go straight turn to left turn to right go back	run stop go straight turn to left turn to right go back

In OOP, 'Car' object is called 'class', 'MyCar' and 'YourCar' is called 'instance'.

We can say a class is a blueprint, an instance is real thing generated by using a class(blueprint)

You remember one PHP code;

```
$conn      = new PDO( "mysql:host=$host;dbname=$db",$user,$pass);
```

This PHP code shows;

$conn is an instance of PDO class generated by 'new' keyword.

The counter term to OOP is 'Procedural Programming'.
'Procedural Programing' doesn't have such concept as 'Object'.

Programing language		
Procedural Programming	COBOL Fortran C Pascal PHP(up to PHP4)	data structure and its process are separated
Object Oriented Programming	Java C++ C# PHP5	data and its behavior(process) are enclosed in a object

[Practice 53] Object, Properties, Methods

Show some examples of Object, Properties and Methods.

It is often said that when we talk about or explain something, we find in our talk many nouns and verbs, and nouns may be properties and verbs may be methods of the object which is the subject we are talking about.

Suppose we are talking about 'animals'.
We may talk; animals has a head, two eyes, a mouth, two hands(some have more than two), sleep, eat, run. Some animals walk on foot. Some can smile while others can't.

2. Class and instance(or object)

In the previous page, we find 'Class is a kind of blueprint' and 'instance is generated by a class(blueprint)'.

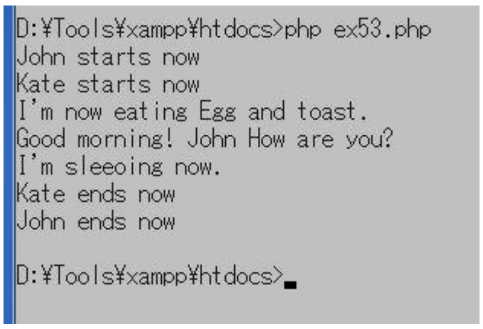
We can show one example with PHP code;

[Practice 54] example of class and instance ex54.php

```
<?php
  class human
  {                                         class variable
    $name;

    function  __construct( $name = "" ) {        constructor
      $this->name  =  $name;
      echo $this->name . " starts now\n";
    }

    function  __destruct( ) {                    destructor
      echo $this->name . " ends now\n";
    }

    function sleep() {
      echo "I'm sleeoing now.\n";
    }

    function eat( $eatThis ) {
      echo "I'm now eating " . $eatThis . ".\n";
    }

    function wakeup( $toWhom ) {
      echo "Good morning! " . $toWhom . " How are you?\n";
    }
  }

  $John   = new human( "John" );
  $Kate   = new human( "Kate" );

  $John->eat( "Egg and toast" );
  $Kate->wakeup( "John" );
  $John->sleep();
  exit();
?>
```

This program may be executed on a prompt window.

The program 'ex54.php' will produce output on prompt screen as follows;



```
D:¥Tools¥xampp¥htdocs>php ex53.php
John starts now
Kate starts now
I'm now eating Egg and toast.
Good morning! John How are you?
I'm sleeoing now.
Kate ends now
John ends now

D:¥Tools¥xampp¥htdocs>
```

__constructor, __destructor

'__constructor' is a method which is called automatically when an instance is generated by 'new' keyword.
You can see two '… starts now' message which are called by two 'new' keywords.

Usually the constructor is used for necessary initialization of an instance.

We can neglect to define constructor, then PHP will execute one constructor defined by PHP itself.

In the program above, one start message will be shown.

'__destructor' is called when all objects in one program don't have any reference from other or when a PHP program execute 'exit()'.
You can see two '… ends now' messages which executed destructor in two instances when 'exit()' is executed.

3. 3(Three) important mechanisms in OOP

There are 3 main mechanisms in OOP.

1) Encapsulation
2) Inheritance
3) Polymorphism

3-1 Encapsulation

Encapsulation is to avoid from accessing to data(properties) in an object directly. Data(properties) in an object are always accessed and modified by a certain method defined in the object, which are called 'setter' for modifying and 'getter' for referring. Some programing language call them 'accessor' and 'mutator'.

For example, in the Practice54, we defined human class and in the class we have one property named 'name'.
in OOP manner, we should define property's scope as 'private'. And we should define 'getter' and 'setter' to refer to/modify it;

```
<?php
  class human
  {
    private $name;

    public function getName() {           //  getter for $name
      return $this->name;
    }
    Public function setName( $name )  {  //  setter for $name
      $this->$name = $name;
    }
```

In a program using class 'human';

```
$human   =   new human();

$human->setName( "Johnson" );

eho  "My name is " .  $human->getName();
```

By setting scope of properties as 'private' and defining 'setter','getter' method for properties, properties can be guarded from being directly accessed and modified.

3-2 Inheritance

Suppose common characteristics of animals.
Animals walk, run, sleep, eat, ….
Animals has a head, hands, legs, eyes, colors, height, weight, ….

Can we define 'Animal' as a class?   Let's try...

```
<?php
  class animal
  {
    private $head;
    private $hand;
    private $leg;
    private $eye;
    private $color;
    private $height;
    private $weight;

  public function setHead( $head ) {
    $this->head    =   $head;
  }
```

(continued)

```
    public function getHead() {
      return $this->head;
    }
    ……..
    Public function walk( $toward ) {
      Echo "I'm walking toward " . $toward;
    }
    Public function run( $speed ) {
      Echo "I'm running " . $speed;
    }
    Public function eat( $eatThis ) {
      Echo "I'm eating " . $eatThis;
    }
  }
```

This is a simple example of 'animal' class.
Human is a kind of 'animal', and as you know, human can smile while other animals can't.

How can we define 'human' class? Should we define 'human' class by repeating 'walk', 'run', 'eat' methods?

In such cases, 'Inheritance' mechanism can help us.

In the case above, we can define 'animal' class as 'parent' class, and 'human' class is 'child' class, which can inherit properties and methods from the parent class.

```
<?php
  class human  extends animal
  {
    //  add unique method in child class
    public function smile() {
      echo "I'm smiling now.";
    }
  }
```

In the code above;

'extends' keyword indicates that Class 'human' inherit Class 'animal'.
All properties and methods in 'animal' class are inherited in 'human' class.
In addition, 'human' class defines its own method 'smile()'.

Abstract class

Sometimes we can't define detail behavior in a methods, that means if we can define different behaviors with different instances, then we can't define only one behavior in a parent class.

Suppose in 'animal' class, we want to define one behavior named 'greet'.
'human' will say 'Hello!', but 'dog' may greet with "bow-wow".

In such cases, we define ‘abstract method’, and such classes which have ‘abstract

method’ is defined as ‘abstract class’.
See codes below;

```
<?php
  abstract class animal
  {
    private $head;
    private $hand;
    private $leg;
    private $eye;
    private $color;
    private $height;
    private $weight;

    public function setHead( $head ) {
      $this->head    =   $head;
    }
    public function getHead() {
      return $this->head;
    }
    ……..
    Public function walk( $toward ) {
      echo “I’m walking toward “ . $toward;
    }
    Public function run( $speed ) {
      echo “I’m running “ . $speed;
    }
    Public function eat( $eatThis ) {
      echo “I’m eating “ . $eatThis;
    }
    abstract public function greet();
  }
```

Abstract class can’t be instanciated.
Abstract class should be inherited and have abstract methods in it.

And abstract method must not have any code in its content, that means, it has only definition of methods(function).
In the inherited class, abstract methods must be implemented with codes.

```
<?php
  class human  extends  animal
  {
    public function greet() {
      echo “Hello!”;
    }
  }

  $personA    =    new  human();
  $personA->grret();

  class dog extends  animal
  {
    public function greet() {
      echo “Bow wow!”;
    }
  }
  $dogB    =   new  dog();
  $dogB->grret();
```

In PHP, multi inheritance is not allowed.

```
<?php
  class engine {l
    public function start() {
      echo “engine started”;
    }
  }
  class body {
    public function setColor( $color ) {
      echo “body color is changed to “ . $color;
    }
  }

  class car extends engine, body {  ←  Multi inheritance is not allowed
    ……


  }
```

Multi inheritance is not allowed to avoid complexity in codes.

3-3 Polymorphism

Interface

Interface has a set of method without contents(codes), which means Interface will provide us what this interface can do and not how this interface implements its methods.

While Abstract class can have concrete methods and abstract methods as well, Interface can only have abstract methods.

Differences between Abstract method and Interface are;

	Interface	Abstract class
Multi inheritance	A class can inherit several interfaces	A class can inherit(extend) only one abstract class
Default implementations	A interface can’t have any codes.	An abstract class can have complete codes and/or details that have to be overridden,
Access modifiers	Interfaces can’t have access modifier, always are assumed as public.	An abstract class can have access modifier.
Fields, constants	No properties(variables) can be defines in an interface.	An abstract class can have properties and constants,

Interface is used to define an abstract type that contains no data or code, but defines behaviors as method signatures.

Polymorphism

Polymorphism is a mechanism where classes have different functionality while they share common interface.

Example;

[Practice 55] example of polymorphysm ex55.php

```
<?php
  // Shape interface, it provides only 'getArea' function signature
  interface Shape {
    public function getArea();
  }

  // Square class implements Shape interface
  class Square implements Shape {
    private $width;
    private $height;

    public function __construct($width, $height) {
      $this->width  = $width;
      $this->height = $height;
    }

    public function getArea(){ // implementation of 'getArea()' for Square class
      return $this->width * $this->height;
    }
  }

  class Circle implements Shape {
    private $radius;

    public function __construct($radius) {
      $this->radius = $radius;
    }

    public function getArea(){ // implementation of 'getArea()' for Circle class
      return 3.14 * $this->radius * $this->radius;
    }
  }

  function calculateArea(Shape $shape) {
     return $shape->getArea();
  }

  $square  = new Square(8, 6);
  $circle  = new Circle(9);

  echo "Area od square is : " . calculateArea($square) , "\n";
  echo "Area od circle is : " . calculateArea($circle);

?>
```

3-4 “Program to an Interface, not to an Implementation”

‘Interface’ represents only “what”, which means “what the class can do”, and doesn’t represent “how it will do it” , which is represented in actual implementation.

In Practice55, of course we can code the last several lines as follows;

```
 ........

  $square   =  new Square(8, 6);
  $circle    =  new Circle(9);

  echo "Area of square is : " . $square -> getArea() , "\n";
  echo "Area of circle is : " . $circle -> gerArea();

?>
```

The above code will work, but from OOP programing manner, isn’t recommended.

It is recommended to program to an interface, not to an implementation.

MVC pattern

MVC stands for 'Model – View – Controller'.

1. What's 'MVC pattern'?

   MVC pattern is a software methodology or architectural pattern to separate application's concern. It proposes three main components or objects to be used in software development.

   1-1 Model

   A Model represents logical aspects of an application. It has a logic to update the current state of the application.

   1-2 View

   A View represents user interface aspects of an application. It will visualize data which a Model contains.

   1-3 Controller
   A Controller controls data flow into Model objects and updates View objects when data changes.

2 What's an advantage of developing an application with 'MVC pattern'?

In Web application development, we (software engineers) need to cooperate with web designers who develop web page design in HTML and CSS.

When our PHP codes and designers' HTML codes are mixed up disorderly, it may cause confusion; designers may make changes PHP codes by mistake, or programmers may make changes HTML codes by mistake.

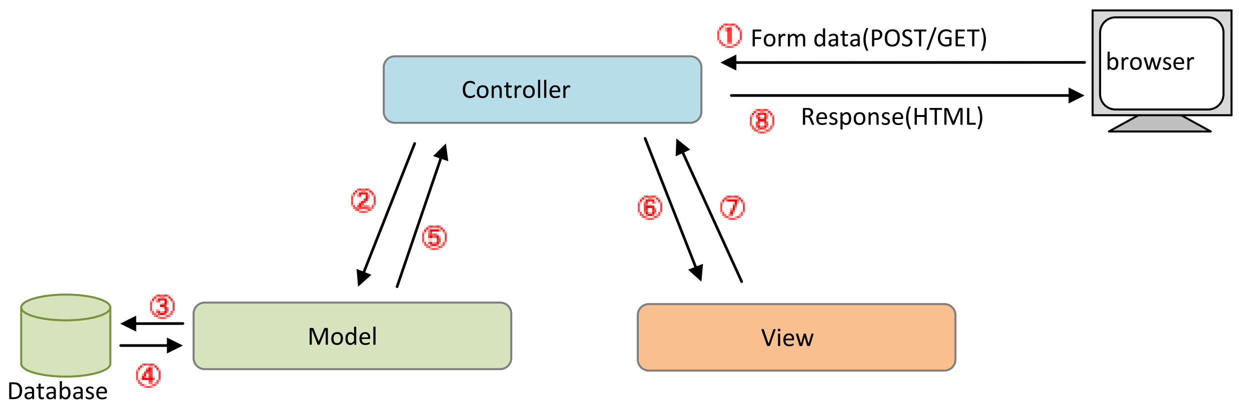
To avoid such confusion, we'd better separate our tasks from designers' tasks.
MVC pattern is one of such methodology.

3 MVC pattern process structure

In MVC pattern process, 'Controller' will play the main role.

'Controller' will get data from application user, transfer them to 'Model',
'Model' will do checking, manipulate database, generate data for 'view' and
return them to 'Controller'.
'Controller' will transfer generated data to 'View', which will generate HTML for
response. 'Controller' will display HTML generated by 'View'.
You can see the flow and structure of process in the next page.

Flow and structure of MVC pattern



MVC pattern process scenario

①Application user sends data from a ‘form’ on a browser to the application.
In the application, ‘Controller’ receives data from browser.

②‘Controller’ find which ‘Model’ is to be called and transfer data to ‘Model’.
‘Model’ do input check,

③‘Model’ accesses database if necessary.

④‘Model’ gets data set from database.

⑤‘Model’ processes data, generate data for ‘View’ and transfer them to ‘Controller’.

⑥‘Controller’ lets ‘View’ generate HTML to be output.

⑦‘View’ generates HTML.

⑧‘Controller’ responds HTML to user.

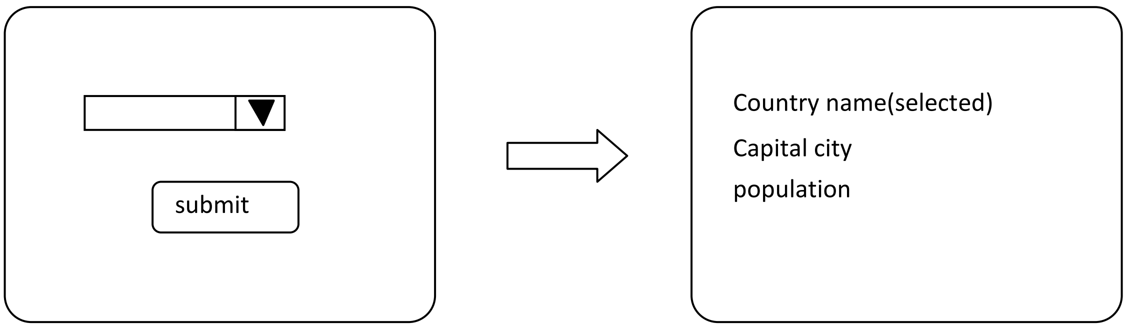
Here we’ll make one very simple example.

[Practice 56] simple web application ex56.php

This application will have

1) A form with one dropdown list with country name.
2) We can select one country from the list and submit it.
3) Our application will search the country in the ‘nations.txt’ and
4) Show the capital city and population on the next screen.

Screens for this Practice are;



(ex56.php) (ex56b.php)

First, we'll make this application without using MVC pattern in Practice 56,
and arrange it with MVC pattern in Practice 56. (Before starting Practice 56, we need to get knowledge on SMARTY, a PHP web template tool.

For country list, you can use 'nations.txt'.

In this practice, we use text file, read it into drop down list. ex56.php

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http: //www.w3.org/1999/xhtml">
<head>
   <title>Non-MVC pattern example</title>
</head>
<body>
   <form action="ex56b.php" method="POST" >
      Select country name and press 'submit':<br />
         <select  name="nations" >
            <option value="">Select ….</option>
         <?php
            $fh   =  fopen( "nations.txt",  "r" );
            while( $line  =  fgets( $fh ) ) {
               $line   =  explode( ",", $line );
               echo "<option value='" . $line[0]  . "'>" . $line[0] . "</option>";
            }
            fclose( $fh );
            unset( $fh );
         ?>
        </select>
        <br />
        <input type="submit" value="submit" name="submit"  />
   </form>
</body>
</html>
```

Function;

explode() : split a string by given strings

syntax : *$result* = explode(*delimiter*, *string*);

delimiter :	the boundary string
string :	the target string
$result :	return value. An array of strings

ex)
```
$target  =  "123,456,789";
$array   =  explode( ",", $target );
```

→ $array is ("123", "456", "789")

ex56b.php
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http: //www.w3.org/1999/xhtml">
<head>
   <title>Non-MVC pattern example</title>
</head>
<body>
   <?php
      $arrayKey  = $_POST[ "nations" ];    // get form data
      $nations   = file( "nations.txt" );    //  read nations.txt into array
      $index     =  array_search( $arrayKey, $nations ); // search country name in array(file)
      $nation    = explode( ",", $nations[ $index ] );  // split target nation in array by comma
   ?>
   You select <?php echo $nation[0]; ?><br />
   Capital city is <?php echo $nation[1]; ?><br />
   Population is <?php echo $nation[2]; ?><br /></body>
</html>
```

Function;

array_search() : search a specified string in an array. If found, return the corresponding key. If not found, return FALSE.

syntax : *$result* = array_search(*target*, *array*, *[strict]*);

target : the searched value. If the target is string, the comparison will be done in case-sensitive manner.

array : The array.

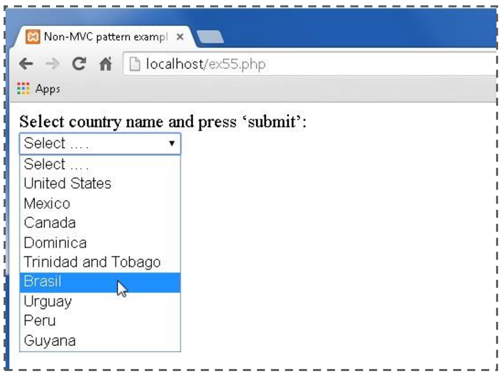
strict : Boolean value. If True, then the comparison will be done with type and value. Default value is FALSE.

example :

$array = (“water”, “air”, “sea”, “mountain”, “lake”);
$index = array_search(“sea”, $array);
→ $index = 2

$array = (“water”, “air”, “sea”, “mountain”, “lake”);
$index = array_search(“SEA”, $array);
→ $index = FALSE (the function will search in case-sensitive mode)

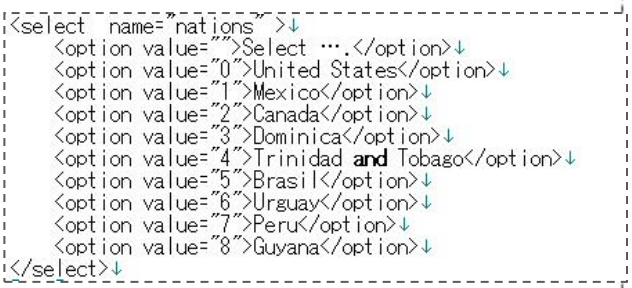
Dropdown list in HTML;



Non-MVC pattern exampl
localhost/ex55.php
Apps

Select country name and press ‘submit’:

Select ....
Select ....
United States
Mexico
Canada
Dominica
Trinidad and Tobago
Brasil
Urguay
Peru
Guyana



```
<select  name="nations" >
    <option value="">Select ….</option>
    <option value="0">United States</option>
    <option value="1">Mexico</option>
    <option value="2">Canada</option>
    <option value="3">Dominica</option>
    <option value="4">Trinidad and Tobago</option>
    <option value="5">Brasil</option>
    <option value="6">Urguay</option>
    <option value="7">Peru</option>
    <option value="8">Guyana</option>
</select>
```

Using ‘select’ tag, we can define a dropdown list shown above.

If we select ‘Brasil’ on the list above, then $_POST or $_GET array will have an element shown below;

‘nations’ => ‘5’

‘Brasil’ has ‘5’ as ‘value’ of ‘option’

name of ‘select’ tag

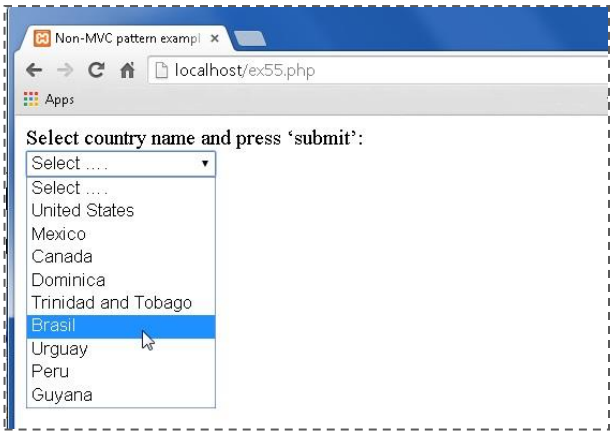
You can define an array of nations as follows;

$nations = (“United States”, “Mexico”, “Canada”,“Dominica”,
“Trinidad and Tobago”, “Brasil”,“Urguay”, “Peru”,
“Guyana”);

Then you can get the selected data by using $_POST(or $_GET);

```
$index = $_POST( "nations" );
$nation_selected = $nations[ $index ];
```

If you make two PHP program, ex56.php and ex56b.php, then you can complete Practice56.



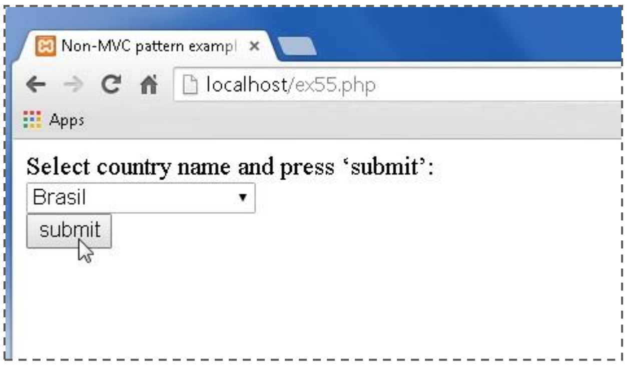
Non-MVC pattern exampl

localhost/ex55.php

Select country name and press 'submit':

```
Select ....
Select ....
United States
Mexico
Canada
Dominica
Trinidad and Tobago
Brasil
Urguay
Peru
Guyana
```

ex56.php



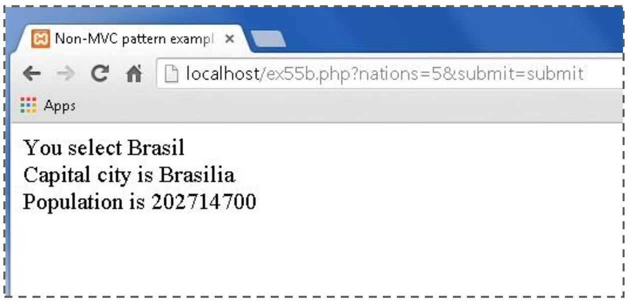
Non-MVC pattern exampl

localhost/ex55.php

Select country name and press 'submit':

Brasil

ex56b.php



Non-MVC pattern exampl

localhost/ex55b.php?nations=5&submit=submit

```
You select Brasil
Capital city is Brasilia
Population is 202714700
```

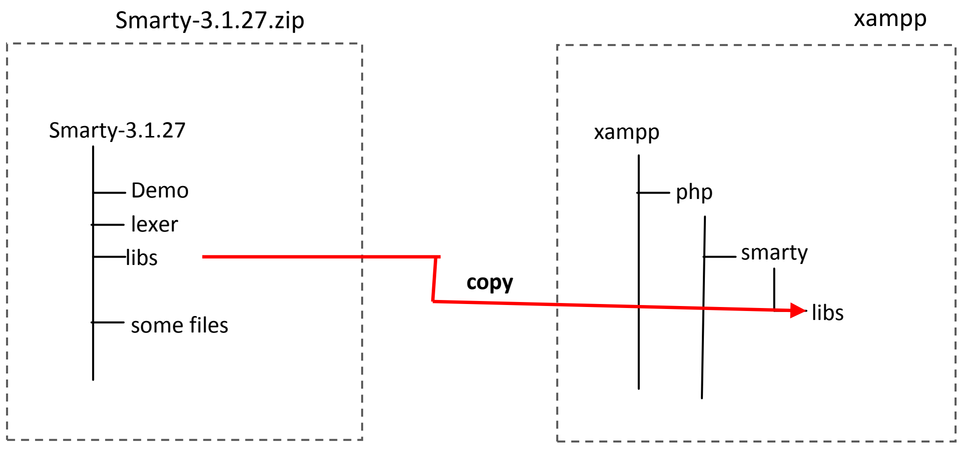
In these two programs, PHP codes and HTML are mixed in sources.

In the next step, we'll separate PHP codes and HTML by MVC pattern.
Before starting, we need to install 'SMARTY', which is a template engine written in PHP and works as a tool for 'View' in MVC pattern.

SMARTY installation

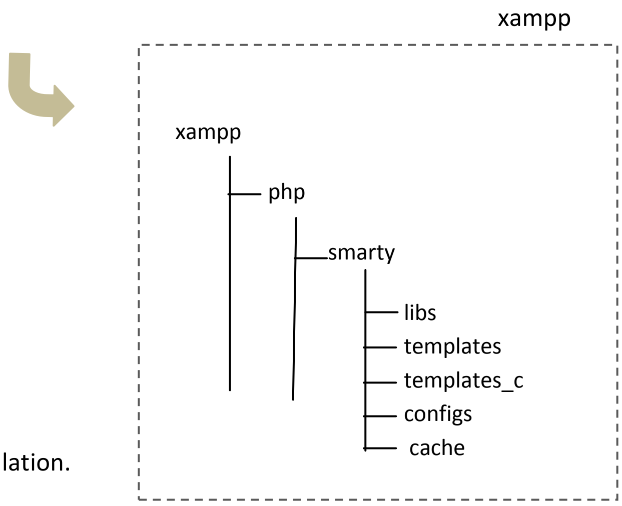
Copy 'smarty-3.1.27.zip' to your PC.
Decompress it, then you'll see one folder 'smarty-3.1.27', under which you'll see 3 folders 'demo','lexer' and 'libs' and some files.

Make one folder 'smarty' under 'xampp/php' folder, and copy 'libs' folder to 'xampp/php/smarty' folder.



Set up 4 folders for Smarty under 'xampp/php/smarty' folder;

templates	.... all template files come here
templates_c	.... all compiled template objects come here
configs	.... for configuration
cache	.... for cache



That's all for SMARTY installation.

When you use SMARTY in your PHP programs, you need to include(require) Smarty file and generate Smarty object as shown below;

```
$pathToSmarty  =  "<path to smarty>";
require_once( 'libs/Smarty.class.php' );
$smarty   =   new Smarty();
$smarty->setTemplateDir( $ pathToSmarty . 'templates' );
$smarty->setCompileDir($pathToSmarty . 'templates_c' );
$smarty->setConfigDir($pathToSmarty . 'configs' );
$smarty->setCacheDir($pathToSmarty . 'cache' );
```

```
$pathToSmarty  =  "<path to smarty>";
```

<path to smarty> : we installed Smarty under a folder "<xamp folder>/php". Don't forget to end with '/'(slash).

ex) $pathToSmarty = "c:/xampp/php/";

SMARTY templates

Smarty templates contains;

HTML tags

Smarty tags and logics inside which application contents(PHP variables' value) are assigned.

Templates file has file extension '.tpl'.

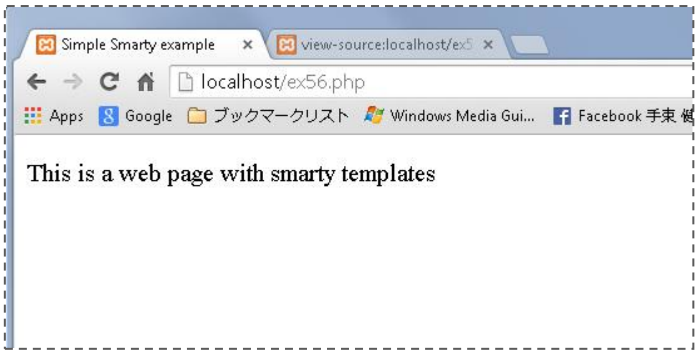
Let's try very simple Smarty example;

[Practice 57] simple Smarty example ex57.php

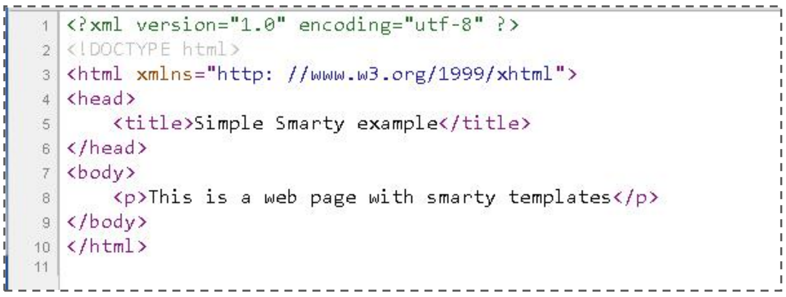
```
<?php
   $pathToSmarty  =  "<path to smarty>";
   require_once( 'libs/Smarty.class.php' );
   $smarty          =  new Smarty();
   $smarty->setTemplateDir( $pathToSmarty . 'templates'     );
   $smarty->setCompileDir(  $pathToSmarty .  'templates_c' );
   $smarty->setConfigDir(    $pathToSmarty .   'configs'       );
   $smarty->setCacheDir(    $pathToSmarty .   'cache'          );

   $smarty->assign( "title", "Simple Smarty example" );
   $smarty->assign( "message", "This is a web page with smarty templates" );
   $smarty->display( 'ex57.tpl' );
   unset( $smarty );
?>
```

ex57.tpl

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http: //www.w3.org/1999/xhtml">
<head>
   <title>{$title}</title>
</head>
<body>
   <p>{$message}</p>
</body>
</html>
```



This is a web page with smarty templates



```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html>
<html xmlns="http: //www.w3.org/1999/xhtml">
<head>
    <title>Simple Smarty example</title>
</head>
<body>
    <p>This is a web page with smarty templates</p>
</body>
</html>
```

Then we can try to modify ex55.php with Smarty.
ex55.php with Smarty

There are two factors which are new for us in Smarty,
one is “How we can express loop to process an array in Smarty”.
The other is “How we can express dropdown list in Smarty”.

Loop process in Smarty

There are 2 ways to process arrays in Smarty;

section	:	*section* is used to process sequentially indexed arrays of data.
foreach	:	*foreach* is used to process associated arrays of data

{section}

Syntax : {*section* name=*name*, loop=*loop*, start=*start*
[,step=*step*] [,max=*max*] }
{/*section*}

name	:	name of the section
loop	:	array
start	:	index position where loop will begin to loop
step	:	step value that will be used to traverse the loop
max	:	maximum times section will loop

example :

PHP program

```
$array   = array( “red”, “blue”, “green”, “white”, yellow”, “black” );
$smarty->assign( “colors”, $array );
```

Smarty template

```
{section  name=color  loop=$colors}
   ({$smarty.section.color.iteration})  {$colors[color]}<br />
{/section}
```

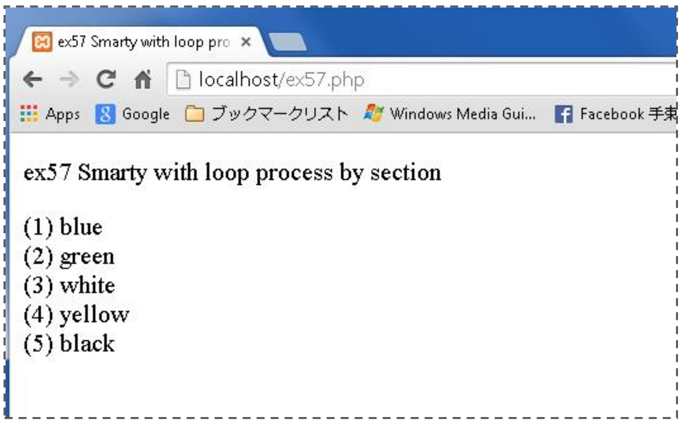
```
(1)   red
(2)   blue
(3)   green
(4)   white
(5)   yellow
(6)   black
```

(*) iteration : a property of ‘section’ which shows index number which start from 1.

index : also an index property but start from 0

[Practice 58] loop process by {section} in Smarty ex58.php

Make a PHP program ‘ex57.php’ and Smarty template ‘ex58.tpl’ to process above example.



ex57 Smarty with loop process by section

```
(1) blue
(2) green
(3) white
(4) yellow
(5) black
```

{foreach}

Syntax : {*foreach* from=*from*, item=*item*, key=*key*, name=*name* }
{*/foreach*}

from : array which is currently processed

item : current element

key : variable name of the current key

name : name of foreach loop for accessing foreach properties

example :

PHP program

```
$array   = array( array( “id”=>“1002”, “name”=>“Mike”, “country”=>”USA”),
                        “id”=>“1013”, “name”=>“Kate”, “country”=>”Canada”),
                        “id”=>“1163”, “name”=>“James”, “country”=>”UK”)     );

$smarty->assign( “students”, $array );
```

Smarty template

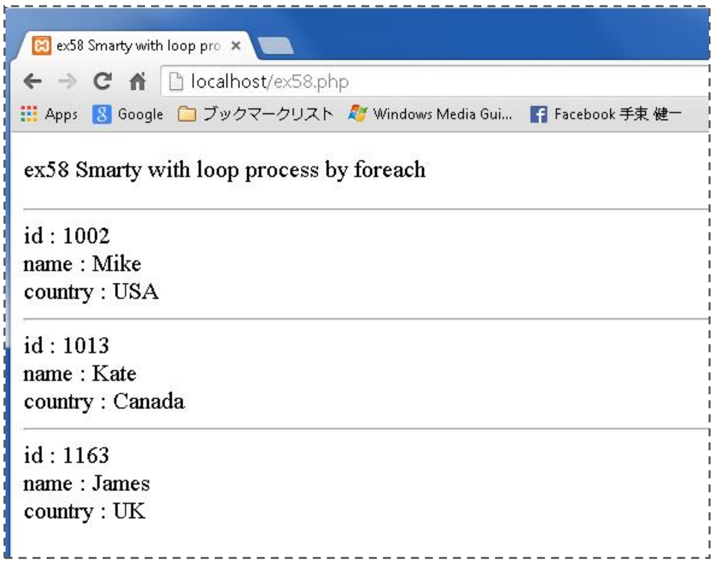
```
{foreach from=$students  item=student  key=key}
     <hr />
     {foreach  from=$student  key=key  item=item}
         {$key}    :    {$item}<br />
     {/foreach}
{/foreach}
```

⇩

```
-------------------
id        1002
name      Mike
country   USA
--------------------
id        1013
name      Kate
country   Canada
--------------------
id        1163
name      James
country   UK
```

[Practice 59] loop process by {foreach} in Smarty ex59.php

Make a PHP program ‘ex58.php’ and Smarty template ‘ex59.tpl’ to process above example.



```
ex58 Smarty with loop process by foreach

id : 1002
name : Mike
country : USA

id : 1013
name : Kate
country : Canada

id : 1163
name : James
country : UK
```

Through Practice 57 and 58, you can see that PHP codes and HTML codes are separated by using Smarty. Let’s go back to Practice 55 and make ex55.php with Smarty.

In Practice 55, we need a dropdown list. Smarty has its own style for dropdown list, which is *html_options* tag in Smarty.

Dropdown list in Smarty ;

nations.txt

```
United States,Washington.D.C,311630000
Mexico,Mexico City,112322767
Canada,Ottawa,33573000
Dominica,Roseau,67000
Trinidad and Tobago,Port of Spain,1339000
Brasil,Brasilia,202714700
Urguay,Montevideo,3477780
Peru,Lima,29132000
Guyana,Georgetown,780000
```

```
$array = ( ‘0’=>’United States’, ‘1’=>’Mexico’, ‘2’=>’Canada’, …. );
$smarty->assign( “nations”, $array );
```

{html_options}

Syntax(1) : {html_options name=*name* options=*options* selected=*selected*} *name* : name of <select> tag in generated HTML

options : associated array for dropdown list

selected : selected option element

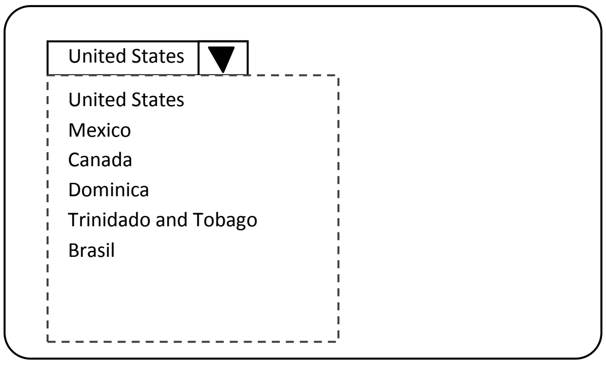
<u>example :</u>

PHP program

```
$array   = array( ‘0’=>’United States’, ‘1’=>’Mexico’, ‘2’=>’Canada’,
                  ‘3’=>’Dominica’, ‘4’=>’Trinidado and Tobago’, ‘5’=>’Brasil’  );

$smarty->assign( “nations, $array );
```

Smarty template

```
{html_options  name=nations  options=$nations  selected=’0’}
```


United States

United States
Mexico
Canada
Dominica
Trinidado and Tobago
Brasil

Syntax(2) : {html_options name=*name* values=*values* output=*output* selected=*selected*}

name, selected : same as Syntax(1)

values : array of values of dropdown list

output : array of output on dropdown list

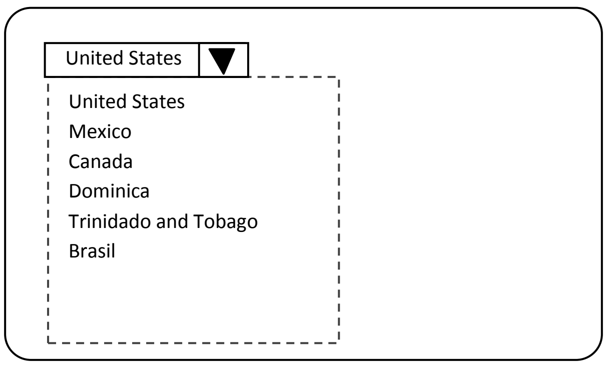
example :

PHP program

```
$array   = array( ‘United States’,  ‘Mexico’, ’Canada’,’Dominica’,
                  ’Trinidado and Tobago’, ’Brasil’  );
$smarty->assign( $values, array( 0,1,2,3,4 ) );
$smarty->assign( “nations, $array );
```

Smarty template

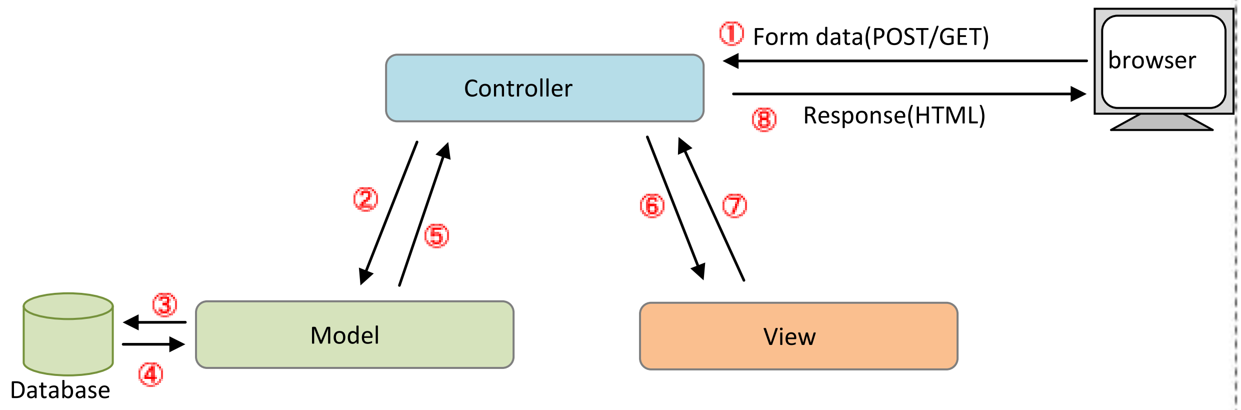
```
{html_options  name=nations  output=$nations  values=$values   selected='0'}
```



Class structures for MVC pattern

We'll place again a picture for MVC pattern flow and structure here.

Flow and structure of MVC pattern



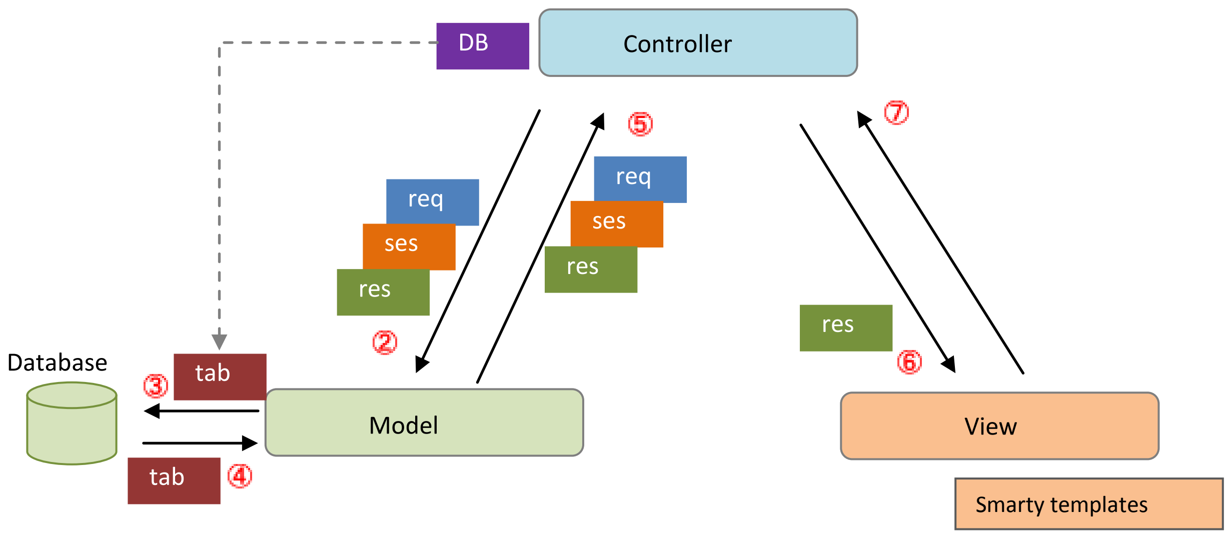
The Controller can be one class.
Each screen for a web application can be one class as a Model module.
For View, Smarty template has a role as View module.
Classes for database table access will work as a part of Model module.

The Controller will make instances to keep data for Model and View.
The instances (class objects) are;

Request object	:	it keeps all POST/GET data from user's browser
Session object	:	it keeps all Session data
Result object	:	it keeps all data for View, which can be Smarty variables

Each Model module (in other words, modules for each screen) will access tables in database through Table object. Each table has one class for columns and behavior.

In OOP, we can say 'Each object dispatches messages to other object', where 'messages' means 'objects'.



DB	Database connection object (<u>static object by Singleton Pattern</u>)
tab	table object
req	Request data object
ses	Session data object
res	Result data (View) object

DB connection object must be common and unique among Model modules. Singleton pattern is applied for such cases, where the class has private constructor, which means we can't 'new' this object and the class provides a public method to return common and unique instance/object to client.

Practice 55 with MVC pattern : application structure is shown below

Controller

index.php	Entry point of application. All input data comes to this module. It configures app's environment and generates objects call 'controller.php'
controller.php	It calls screen process programs by '_REQUEST' parameter, hands objects to the screen process and prepares output data, calls Smarty template and lets it display next screen

Model

ex55.php	It reads 'nations.txt' and make an array with its data.
ex55b.php	It gets country's name, capital city and population by the result of the dropdown list, and set them in the result object.
Request.php	A class to keep POST/GET variables.
Session.php	A class to keep Session variables.
Result.php	A class to keep View variables for Smarty.
comfunc.php	functionss commonly called by Model modules.
DefineConst.php	Module to keep user defined constants .
RandomString.php	Module to generate randomized strings In this practice, it's not used.

View

ex55.tpl	View for 'ex55' screen
ex55b.tpl	View for 'ex55b' screen

You can see some sample source files for this application.

And you'll find PHP codes and HTML codes are separated, Smarty variables and some Smarty tags are used to combine PHP and HTML.

This practice is very simple one, so there's no case for branches in one screen.
If one screen can have more than one state (for example, normal case and error process, etc.), then we need to keep 'state' variable in Session or in some Smarty variable on screen.('action' in ex55)

As you see, there's only one entry point on this application, that is, 'index.php'.
All forms in this application has 'action' attribute with the value 'index.php'.
We don't need to search other module for entry point from HTML form.
In addition, all output (which is 'next web page') are output from only one point, which is 'dispPage' function in 'Controller.php'. Any other module for screen process should not have output point for next page.
"One entry, one exit" will save your time to search modules and make application easy to read.

One more point;

You can try another example for 'Practice 55',

    localhost/ex55s.php

In 'ex55s.php', you'll see request and response are shown in the same window('ex55s.tpl').
In this program, pay attention to the following line;

```
<div style="display: {$display}; ">
```

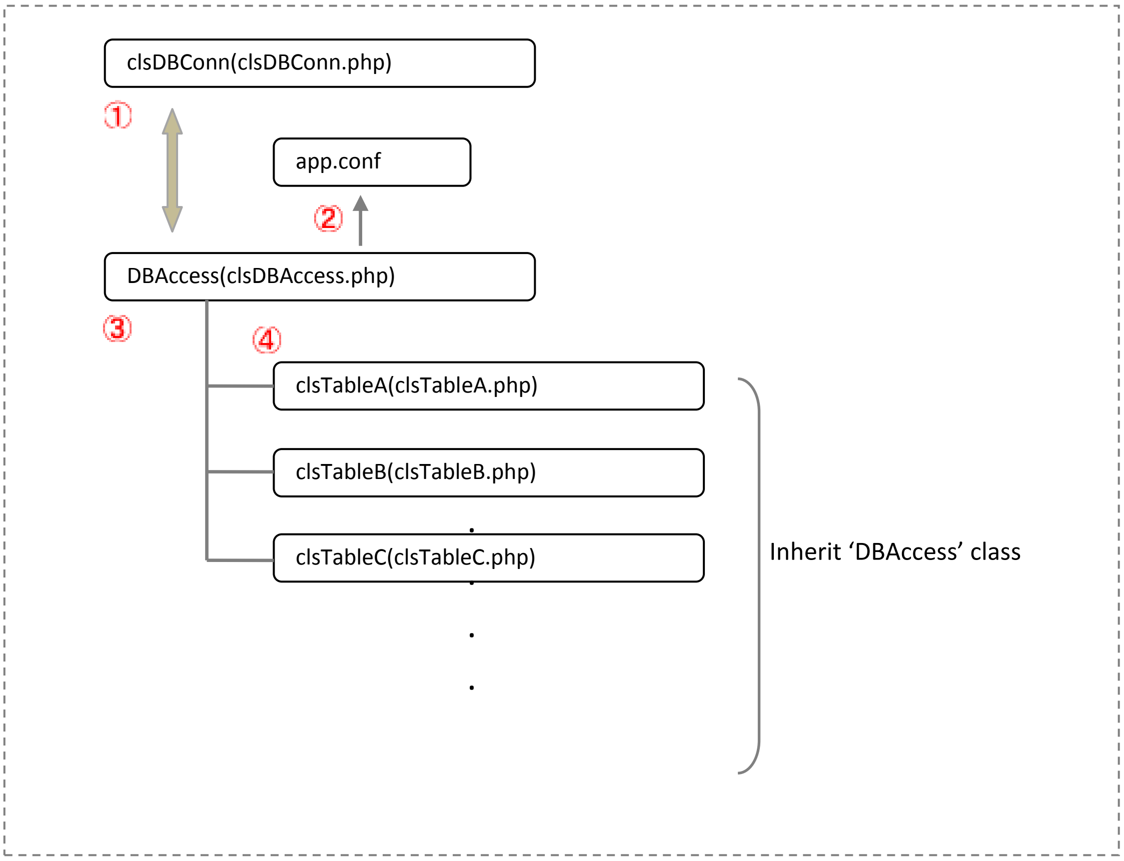
Using CSS 'display' property, we can arrange the contents of the window, hiding result texts or showing them. For details of CSS 'display' property, visit http://www.w3schools.com/css/css_display_visibility.asp

Development of Database access class

Here I will introduce one example of how to develop database access classes.

This is an example which I've developed in my career, so there must be much better way for it. You can copy them, you can develop your own access classes based on my classes.

1. Access class structure



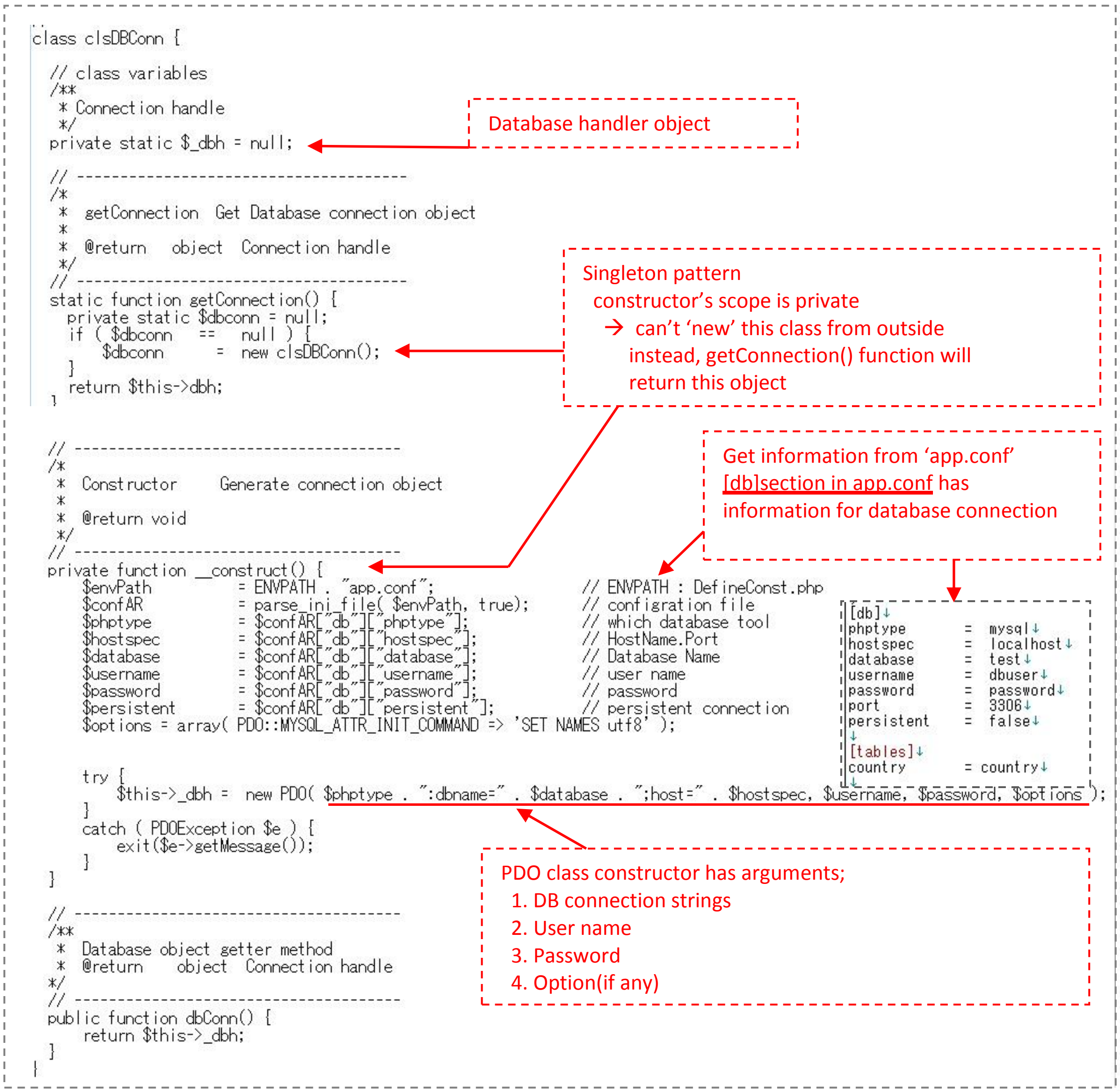
① clsDBConn : Instance of this class will keep 'Database connection object'.
All actions on database will be done on this object.
To avoid operation mistakes, we usually make only one object for same database. It is realized by using 'Singleton Pattern' mechanism.

② app.conf : A text file where application's configuration is defined in 'ini' format file.
We can make use of the contents of this file not only to define configuration of the application but also to avoid hard coding of constants in PHP sources.

③ clsDBAccess : This is a class for accessing to a table, it contains 'database connection object', 'query(executing SQL)' method, 'fetch rows' method and so on.
This class is a parent class of accessor classes for each table.

④ clsXxxx : One table in a database has one accessor class and each accessor class inherits DBAccess class.

2. Database connection class(clsDBConn.php)



```
class clsDBConn {

  // class variables
  /**
   * Connection handle
   */
  private static $_dbh = null;

  // ----------------------------------------
  /*
   *  getConnection  Get Database connection object
   *
   *  @return   object  Connection handle
   */
  // ----------------------------------------
  static function getConnection() {
    private static $dbconn = null;
    if ( $dbconn     ==    null ) {
       $dbconn       =    new clsDBConn();
    }
    return $this->dbh;
  }


  // ----------------------------------------
  /*
   *  Constructor     Generate connection object
   *
   *  @return void
   */
  // ----------------------------------------
  private function __construct() {
      $envPath           = ENVPATH . "app.conf";                  // ENVPATH : DefineConst.php
      $confAR            = parse_ini_file( $envPath, true);       // configration file
      $phptype           = $confAR["db"]["phptype"];              // which database tool
      $hostspec          = $confAR["db"]["hostspec"];             // HostName.Port
      $database          = $confAR["db"]["database"];             // Database Name
      $username          = $confAR["db"]["username"];             // user name
      $password          = $confAR["db"]["password"];             // password
      $persistent        = $confAR["db"]["persistent"];           // persistent connection
      $options = array( PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8' );

      try {
          $this->_dbh =  new PDO( $phptype . ":dbname=" . $database . ";host=" . $hostspec, $username, $password, $options );
      }
      catch ( PDOException $e ) {
          exit($e->getMessage());
      }
  }

  // ----------------------------------------
  /**
   *  Database object getter method
   *  @return    object  Connection handle
  */
  // ----------------------------------------
  public function dbConn() {
      return $this->_dbh;
  }
}
```

```
[db]↓
phptype        =  mysql↓
hostspec       =  localhost↓
database       =  test↓
username       =  dbuser↓
password       =  password↓
port           =  3306↓
persistent     =  false↓
↓
[tables]↓
country        = country↓
↓
```

Database handler object

Singleton pattern
constructor's scope is private
→ can't 'new' this class from outside
instead, getConnection() function will
return this object

Get information from 'app.conf'
[db]section in app.conf has
information for database connection

PDO class constructor has arguments;
1. DB connection strings
2. User name
3. Password
4. Option(if any)

By singleton pattern, there's only one database object for the target database in the application space. If you write only Standard SQL, then you can prepare DB connection class for several database tools like MySQL, ORACLE, MSSQLserver and so on. You don't need add any other module than clsDBconn with different 'DB connection strings'.

3. Table accessor class (clsDBAccess.php)

3-1 constructor

```
$this->mDBconn  =  clsDBConn::getConnection()
```

this executes ‘getConnection’ method of ‘clsDBConn’ class, which is static method
we don’t need to instanciate ‘cksDBConn’ class.
‘getConnection’ method will return database objet, which will be saved in a class variable ‘mDBConn’.

3-2 DBAqueryRS($sql)

This method receives SQL statement as an argument .

```
$this->mStmt  =  $this->mDBConn->query( $sql )
```

‘query’ method of the PDO class will execute given SQL statement and return record set(if any) as ‘Statement’ object. We can get records from ‘Statement’ object by ‘fetch’ or ‘fetchall’ method.

3-3 DBAqueryRSP($sql, $ar)

This method receives SQL statement as an argument and an array of parameters,
execute a ‘prepared query’(=’parameter query’) and return ‘statement’ object with record set.

```
$this->mStmt  =  $this->mDBConn->prepare( $sql );
$this->mStmt->execute( $ar ) );
```

‘prepare’ method of PDO object will return ‘statement’ object. The SQL statement must have at least one placeholder(‘?’) in it, and actual value for placeholder must be given in an array which will be given to ‘execute’ method of ‘statement’ object.
‘execute’ method will return true(if success) or false(if failure).

We can get records from ‘Statement’ object by ‘fetch’ or ‘fetchall’ method.

3-4 DBAquery($sql)

This method will execute SQL statement but will not return any record set.
Such SQL statements as ‘add’, ‘update’, ‘create table’ and so on may be executed by this method.

If success, return true, otherwise return false.

3-5 DBAqueryP($sql, $ar)

This method is a ‘parameter query version of DBAquery()’.
This doesn’t return record set in ‘Statement’ object.
If success, return true, otherwise return false.

If SQL statement contains user input data such as ‘add’, ‘update’ sql, you would better use this ‘DBAqueryP’ parameter query than ‘DBAquery’.

3-6 DBAgetNext()

This method will get next record set.
If valid record exists, then return true and the record set will be set into 'mRS' as associated array(<field name> => <field value>).

```
$this->mRS  =  $this->mStmt->fetch( PDO::FETCH_ASSOC );
```

If valid record exists, mRS will get associated array of the record and return true, otherwise return false.

```
array( [field_name_1] => [field_value_1],
       [field_name_2] => [field_value_2],
            ....                         );
```

3-7 abstract methods

There are 3 abstract methods in DBAccess class;

1) 'add' method   method to add a record(row) into a table
2) 'update' method method to update a record(row)
3) 'delete' method method to delete record(row)

these methods should be overridden by sub-class(child class).

4. Table accessor class for each table(clsXXXX.php)

Table accessor class for each table inherits clsDBAccess class.

4-1 table 'country' accessor class (clsCountry.php)

Table 'country' is shown below;

Column name	Data type	size	attribute	remarks
id	Int		PRIMARY KEY AUTO INCREMENT	record ID
name	Varchar	128		country name
capitalcity	Varchar	128		capital city
population	Int		UNSIGNED	population
iso2	Char	2		2 digits country code
iso3	char	3		3 digits country code

1) definition of column variables is shown on the next page.
every variable for column(field) should have 'private' scope and have 'setter','getter' methods.

Variables for table columns

```
/**↓
 * record ID  (primary key)↓
 */↓
private $mID;↓
/**↓
 * country name↓
 */↓
private $mName;↓
/**↓
 * capital city name↓
 */↓
private $mCapital;↓
/**↓
 * population↓
 */↓
private $mPopulation;↓
/**↓
 * country code ( ISO 2 )↓
 */↓
private $mISO2;↓

/**↓
 * country code ( ISO 3 )↓
 */↓
private $mISO3;↓
```

'setter' and 'getter' methods for each column

```
// getter method for properties↓
function getID()                    { return $this->mID;      }↓
function getName()                  { return $this->mName;    }↓
function getCapital()               { return $this->mCapital; }↓
function getISO2()                  { return $this->mISO21;   }↓
function getISO3()                  { return $this->mISO31;   }↓
↓
// setter method for properties↓
function setID( $dt )               { $this->mID         = $dt; }↓
function setName( $dt )             { $this->mName       = $dt; }↓
function setCapital( $dt )          { $this->mCapital    = $dt; }↓
function setPopulation( $dt )       { $this->mPopulation = $dt; }↓
function setISO2( $dt )             { $this->mISO2       = $dt; }↓
function setISO3( $dt )             { $this->mISO3       = $dt; }↓
↓
```

2) getAll() method

A method to get all records from table 'country'.
Return value : true(if successful) or false(if failed)
Records can be got by 'DBAfetch' method or 'DBAfetchAll' method.

3) getByID() method

A method to get a record which matches with specified record ID.
argument : record ID
return value : true(success) or false(failure)
If successful, each field can be got by 'getter' method for fields.

```
$mCountry  =  new clsCountry();
          .
          .
if( $mCountry->getByID( $id ) ) {
   echo "country name is " . $mCountry->getName();
} else {
   echo "Failed to get country record ";
}
```

4) getByName() method

A method to get a record which matches with specified country name.
argument : country name
return value : true(success) or false(failure)
If successful, each field can be got by 'getter' method for fields.

```
$mCountry  =  new clsCountry();
          .
          .
if( $mCountry->getByName( 'Guyana' ) ) {
   echo "Capital city is " . $mCountry->getCapital();
} else {
   echo "Failed to get country record ";
}
```

5) add() method (implementation of 'add' abstract method in DBAccess class)

Before calling 'add' method, we need to set values to column variables.
'add' method will issue 'parameter query' as shown below;

```
function add() {
   $wSql  = "INSERT INTO " . $this->mTableName;
   $wSql .= " ( name, capital, population, iso2, iso3 ) ";
   $wSql .= " VALUES ( ?, ?, ?, ?, ? )";                      Place holder
   $wAr   = array( $this->getName(),
                   $this->getCapital(),
                   $this->getPopulation(),                    parameters
                   $this->getISO2(),
                   $this->getISO3()
                 );
   $wRes  =  $this->DBAqueryP( $wSql, $wAr );
   if( $wRes ) {
      return true;
   } else {
      // failed  :  set error message
      $this->mMessage  = "clsCountry:add SQL error:" .
                              parent::DBAgetMessage() ." SQL:".$wSql . ":";
      return false;
   }
}
```

6) update() method (implementation of 'update' abstract method in DBAccess class)

Before calling 'update' method, we need to set values to column variables and do necessary modification.
'update' method will issue 'parameter query' as shown below;

```
function update() {
   $wSql  =  "UPDATE ".$this->mTableName." SET ".
           " name        =  ?,".
           " capital     =  ?,".
           " population  =  ?,".                              Place holder
           " iso2        =  ?,".
           " iso3        =  ? "
           " WHERE id   =  ?";
   $wAr   = array( $this->getName() ),
                 $this->getCapital(),
                 $this->getPopulation(),
                 $this->getISO2(),                            parameters
                 $this->getISO3(),
                 $this->getID()
               );
   $wRes  =  $this->DBAqueryP( $wSql, $wAr );
   if( $wRes ) {
      $this->message   =   "update successful:".$this->getName().":";
      return true;
   } else {
      $this->message   =   "update failed:".$this->getName().": Message:" .
                                              parent::DBAgetMessage();
      return false;
```

7) delete() method (implementation of 'delete' abstract method in DBAccess class)

Before calling 'delete' method, we need to get the target record.
This method will not show any confirmation alert to user, so if you need to show confirmation alert message before deleting, you must do it yourself
in this method.

```
function delete() {
    $wSql   = "DELETE FROM  ".$this->mTableName."  WHERE id = ? ";        Place holder
    $wAr    = array( $this->getID() );        parameters
    $wRes   = $this->DBAqueryP( $wSql, $wAr );
    if( $wRes ) {
        $this->message   =   "Delete successful:".$this->getName().":";
        return true;
    } else {
        $this->message   =   "Delete failed:".$this->getName().": Message:" .
                                             parent::DBAgetMessage();
        return false;
    }
}
```

These 3 implemented methods may have same PHP codes for other tables.
After developing table accessor classes, we can access to database tables from our Model module (= module for each screen).

5. Access to database table from modules for each screen
   We define an instance variable for a table.
   In the constructor of a module for a screen, make table instance by 'new' keyword.

```
private mCountry;        //  instance variable for table 'country'

function __constructor() {
        …….

    $mCountry   =   new clsCountry();     //  make 'country' instance
        ……..
}
```

Using table instace, we can access to the table in the module for a screen.

```
public function xxxxxxx() {
    …..
    $this ->mCountry->getByName( $wName );
    ……
```

Final exercise

Make a web application with specification shown below;
Make a new folder 'phptraining' under 'xampp/htdocs' and programs are saved here. Specification of the web application

1. Using 'myDB' database, make a table 'person' as follows;

No.	column	type	length	remarks
1	userID	varchar	Max 20	Primary key
2	password	varchar	Max 128	encrypted by md5()
3	name	varchar	Max 128	
4	gender	int		1 : male 2 : female
5	email	varchar	Max 128	
6	nationality	int		'id' in the 'nations' table

On 'myDB' database, make a table 'country' as follows;

No.	column	type	length	remarks
1	id	int		Primary key, auto increment
2	name	varchar	Max 128	Country name
3	capital	varchar	Max 128	Capital city name
4	population	int		
5	iso2	char	2	ISO country code(2 digits)
6	iso3	char	3	ISO country code(3 digits)

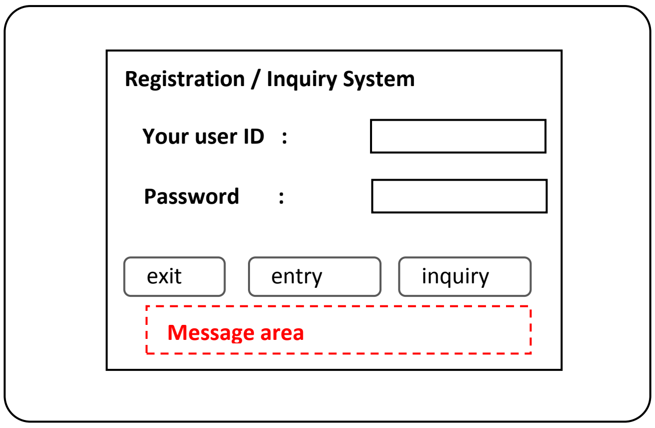
sample

name	Capital city	population	iso2	iso3
United States of America	Washington	321793000	US	USA
Mexico	Mexico City	121740000	MX	MEX
Canada	Ottawa	35749600	CA	CAN
Guyana	Georgetown	746900	GY	GUY
Brazil	Brasilia	20487800	BR	BRA
France	Paris	66212000	FR	FRA
Nigeria	Abuja	182202000	NG	NGA

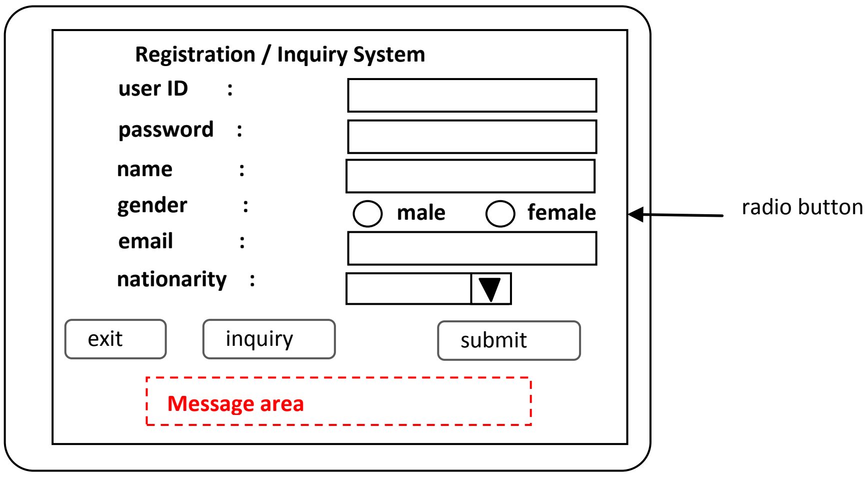
2.This is an application for ‘registration and inquiry of person’.

It has 3 screens shown below;

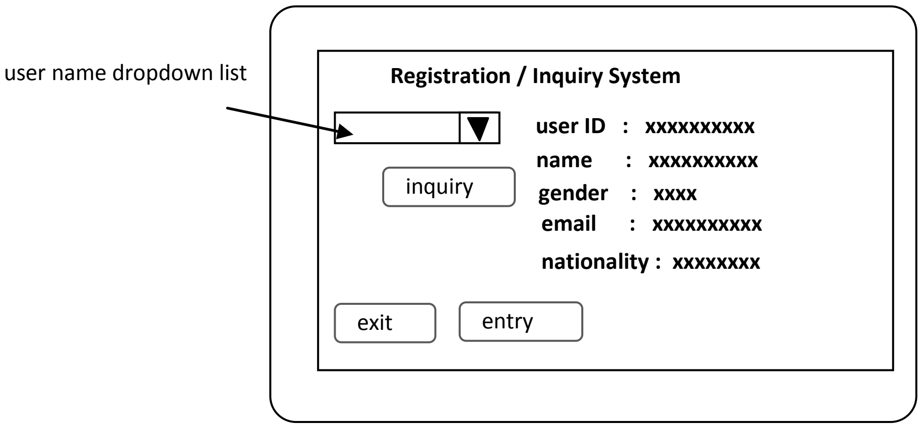
2-1 Log-in screen program : login.php



2-2 Registration screen program : entry.php



2-3 Inquiry screen program : inquiry.php



2-4 administrator

1) We need one user for administration as follows;

userID	:	'admin'
password	:	'password'
name	:	'administrator'
gender	:	1 or 2 (whichever you like)
email	:	your email address
nationality	:	0

after setting up 'person' table, you need to make this user manually.

2-5 screens

1) 'log-in' screen
'userID' and 'password' should be matched with a record on 'person' table.

Take measure to guard from 'SQL injection attack'.

After successful 'log-in',
if 'entry' button is pressed, the next screen is 'registration screen',
if 'inquiry' button is pressed, the next screen is 'inquiry screen'.
When we have more than one destination page(.php) in one form,
we need to change the value of 'action' attribute in 'form' tag, it
will be done by using javascript, see '2-6 javascript'.

If 'exit' is pressed, the application will quit. Don't forget to release objects.

2) 'registration' screen
'userID' must be unique
'userID' can have only alphabets, numbers and underscore.
maximum length is 20 letters.

'password' can have any letters.
'password' should be saved with md5() encrypted.
md5() function will generate 32 digits of random letters.

'email' must be checked with legal combination of email address characters.

All fields are mandatory.

When ‘submit’ button is pressed, input data are checked and if no
error is detected, user record will be registered and successful message will be shown on ‘message’ area. If any error is detected, error message will be shown on ‘message’ area. In both case, input data should be echoed back.

When ‘inquiry’ button is pressed, it will make screen transition to ‘inquiry’ screen. In case of pressing this button, input data will be neglected.

If ‘exit’ is pressed, the application will quit. Don’t forget to release objects.

3) ‘inquiry’ screen
After selecting one name from ‘person’ dropdown list and pressing ‘inquiry’ button, ‘person’ table will be searched by the selected ‘user’.

Getting record from ‘person’ table, columns’ value are shown on the screen.

If ‘entry’ button is pressed, the next screen is ‘registration screen’,

If ‘exit’ is pressed, the application will quit. Don’t forget to release objects.

2-6 javascript

In the application, you’ll see multiple transition destinations from one form. In such cases, we need to change ‘action’ attribute value (next PHP program) by data sent from a form.

In the form, ‘inquiry’ button will request ‘inquiry’ screen which will be processed by ‘inquiry.php’ and ‘entry’ button will request ‘entry’ screen which will be processed by ‘entry.php’.

Then, we define as follows;

```
<input type=”submit” name=”submit” value=”inquiry”
        onclick=”form.action=’inquiry.php’;return true;” />
<input type=”submit” name=”submit” value=”entry”
        onclick=”form.action=’entry.php’;return true;” />
```

In ‘onclick’ event, you’ll see a small javascript.
‘form’ means current form, ‘action’ is ‘action’ attribute of ‘form’ tag in HTML.
So, when ‘inquiry’ button is pressed, the ‘action’ attribute of ‘form’ tag will be changed into ‘inquiry.php’ by this javascript code.

In tis way, we can change the destination program dynamically by ‘onclick’ event and javascript code in the definition of ‘submit’ button.

2-7 radio button on Smarty

Radio button on Smarty has two syntax just like dropdown list on Smarty.

{html_radios}

Syntax(1) : {html_radios name=*name* options=*options*
selected=*selected* separator=*separator*}

name : name of <select> tag in generated HTML

options : associated array for radio buttons

selected : selected option element

separator : string or text to separate each radio button

example :

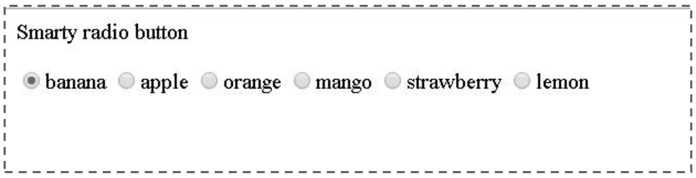
PHP program

```
$array   = array( ‘0’=>’banana’, ‘1’=>’apple’, ‘2’=>’orange’,
                  ‘3’=>’mango’, ‘4’=>’strawberry’, ‘5’=>’lemon’ );

$smarty->assign( “fruits”, $array );
```

Smarty template

```
{html_radios name=fruits options=$fruits selected=’0’ separator=” “ }
```

⇩


Smarty radio button

● banana ○ apple ○ orange ○ mango ○ strawberry ○ lemon

Syntax(2) : {html_radios name=*name* value=*values* output=*output*
selected=*selected* separator=*separator*}

name : name of <select> tag in generated HTML *values* : array of values for radio buttons

output : array of output for radio buttons

selected : selected option element

separator : string or text to separate each radio button

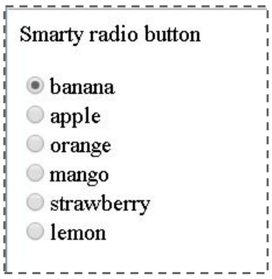
example :

PHP program

```
$values   = array( 0, 1, 2, 3, 4, 5 );
$output   = array( “banana”, “apple”, “orange”, “mango”, “strawberry”, “lemon” );

$smarty->assign( “values”, $values );
$smarty->assign( “output”, $output );
```

Smarty template

```
{html_radios  name=fruits output=$output value=$values
                                        selected=’0’ separator=”<br />“ }
```


Smarty radio button

● banana
○ apple
○ orange
○ mango
○ strawberry
○ lemon

To arrange radio buttons vertically, set ‘separator’ attribute value to ‘<br />’.

3.Multiple states in one web page

In this application, there are multiple states in one page.

For example, in ‘Log-in’ page,

1) to show page initially (all fields and message area are cleared)
2) to echo back input data and show message

How can we identify each state?

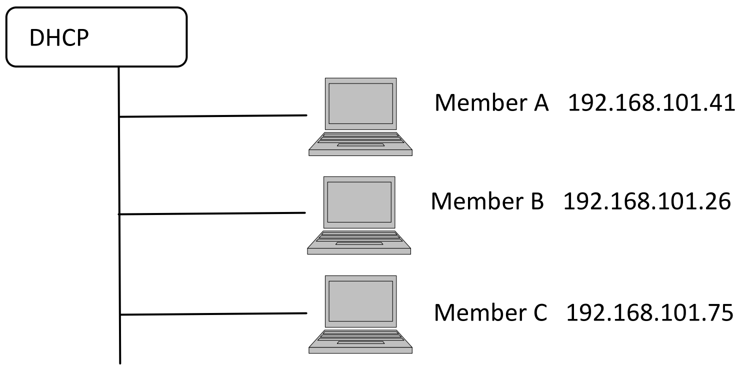
One point is;

if ‘subnit’ button(‘inquiry’ or ‘entry’) is posted, it is ‘echo back’ state, otherwise it's 'initial' state. We can check the '$_POST' array and whether it has an element of one of each button or not.

```
<input type="submit" name="submit"  value="inquiry" />
  → if( array_key_exists( $_POST["submit"] ) ) {   //  if 'submit' button is pressed,
        ( process for 'echo back' state )
```

After testing this application on your own PC, then we can test this application in a local network.

In GWI, we(training members) have a local IP address 192.168.101.xxx



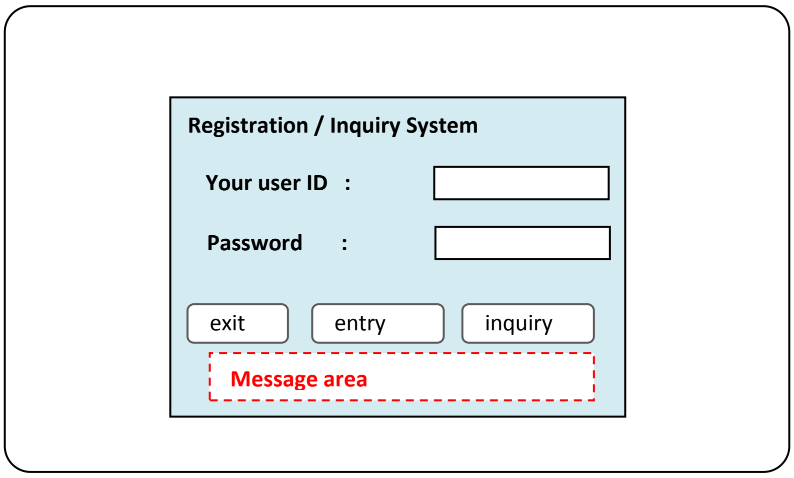
URL strings shown below can access to 'member A's XAMPP web pages;
http://192.168.101.41/phptraining/

each member can register his/her own user profile to other's site and test others' application on his/her own PC.

In the above case, member A is the web server, and web server can be shifted day by day.
You can say "Today's web server is Member C. Access to IP 192.168.101.75. Let's start!".

4.Some tips

4-1 To show a web page block in the middle of the browser

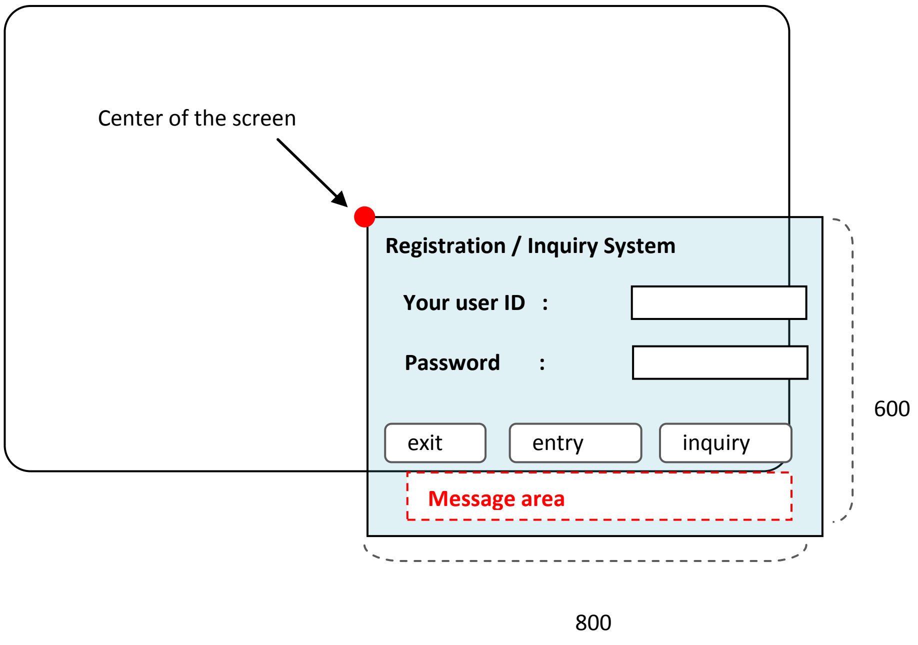


Using CSS and box layout, we can put our web page (box) in the middle of the browser screen.

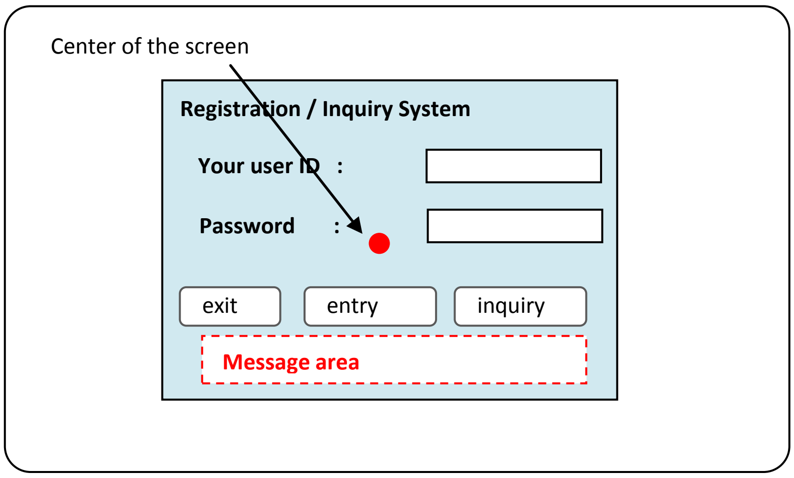
In the CSS;

```
/*
    container :  box for whole area
*/
#container {
          position:            absolute;
          top:                 50%;          /* move down box top to center   */
          left:                50%;          /* move left side to center     */
          width:               800px;        /* width   */
          height:              600px;        /* height  */
          margin-left:         -400px;       /* move to left by half of width */
          margin-top:          -300px;       /* move up by half of height    */
          background-color:  #efefea;     /* back ground color : light blue */
}
```

‘top: 50%;’ and ‘left: 50%;’ will place the upper left of ‘container’ at the center;



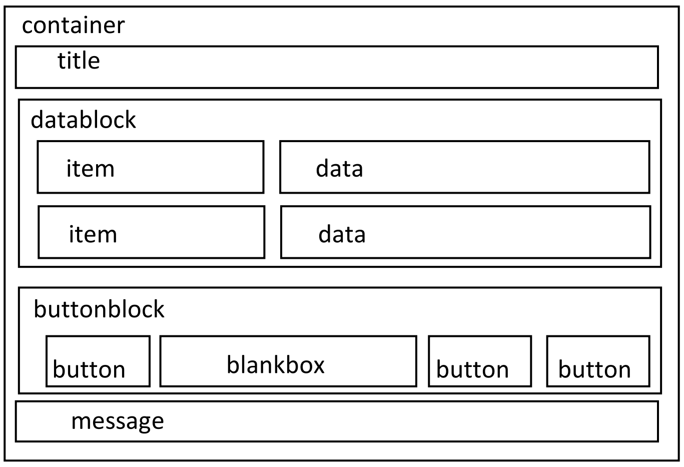
Next step is to move ‘container’ box to the proper position.
‘‘margin-left: -400px:’ will move the box 400px to the left.
“margin-top: -300px:’ will move the box 300px to the top.
400px and 300px means half of the vertical and horizontal size of the box. Then ‘container’ box will come to the center of the screen.



Even when you resize browser window, the ‘container’ will keep the center position.

4-2 Layout of the boxes for this application

This ‘login’ window is made up with nested boxes shown below;



Application structure for final exercise is shown below;

Controller

index.php	Entry point of application. All input data comes to this module. It configures app's environment and generates objects call 'controller.php'
controller.php	It calls screen process programs by '_REQUEST' parameter, hands objects to the screen process and prepares output data, calls Smarty template and lets it display next screen

Model

login.php	Log-in window.
entry.php	User profile registration window
inquiry.php	User profile inquiry window
Request.php	A class to keep POST/GET variables.
Session.php	A class to keep Session variables.
Result.php	A class to keep View variables for Smarty.
comfunc.php	functions commonly used/called by Model modules.
DefineConst.php	Module to keep user defined constants .
RandomString.php	Module to generate randomized strings In this practice, it's not used.
DBconn.php	A class to keep Database connection object using Singleton pattern
DBAccess.php	A parent class to access to a table in a database. An access class for each table will be inherited from this parent class.
clsPerson.php	A class to access 'person' table. It inherits DBAccess.php'
clsCountry.php	A class to access 'country' table. It inherits DBAccess.php'

View

login.tpl	View for 'login' window
entry.tpl	View for 'entry' window
inquiry.tpl	View for 'inquiry' window

Acknowledgements :

This text book was made for “PHP & Web programing training” at ICT department in GWI(Guyana Water Incorporate) conducted by JICA(Japan International Cooperation Agency) Senior Volunteer(ICT) in August and September, 2015.

Special thanks to web sites shown below for valuable information and hints;

http://www.w3schools.com/php/

http://www.w3schools.com/html/

http://www.w3schools.com/php/php_mysql_intro.asp

http://dev.mysql.com/doc/refman/5.7/en/index.html

https://www.apachefriends.org/index.html

http://php.net/manual/en/index.php

http://www.regular-expressions.info/php.html

http://www.smarty.net/

https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Computer_science

In this text, I dared to avoid to refer to PHP framework tools.

It is because PHP beginners would better write PHP source codes, SQL statements, HTML&CSS definitions for themselves to get familiar with these language syntax. Framework tools might generate PHP codes automatically and divide SQL statements into pieces with parameters, which could give us high productivity but might hide basic knowledge on these languages aside.

After having completed basic training, framework tools like cakePHP and others would be better to use for higher productivity.

Kenichi Tezuka
September 14, 2015