

Pemrograman Web – PHP 1

Antonius RC

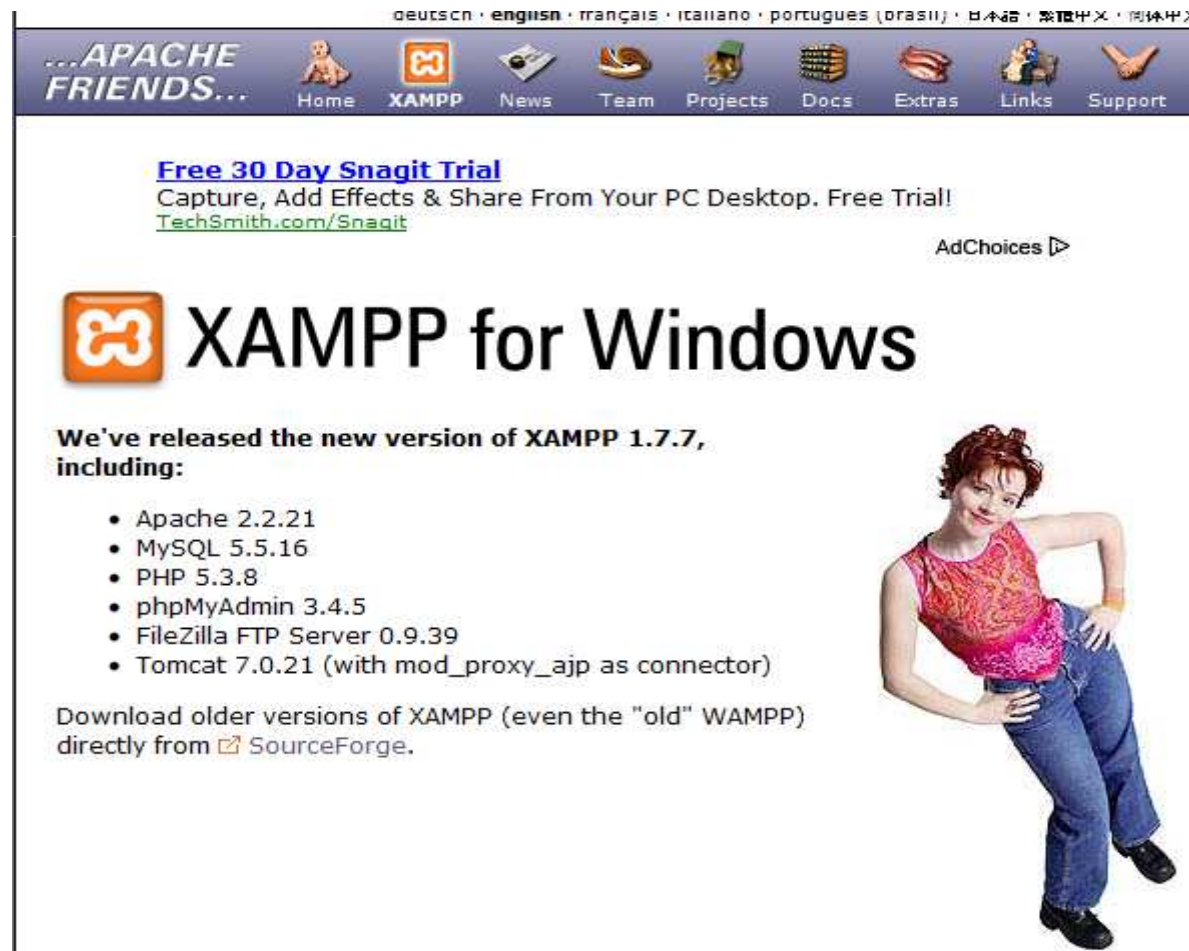
© 2012

Server side

- PHP merupakan salah bahasa pemrograman yang berjalan pada sisi **server**
 - Dapat menggenerate HTML!
- PHP merupakan bahasa pemrograman yang menggunakan sintaks keturunan **C/C++**
- PHP files have a file extension of ".php", ".php3", or ".phtml"
- PHP singkatan dari **PHP Hypertext Preprocessors**
 - www.php.net (latest stable: 5.3.8)
 - Singkatan pertama: Personal Home Page, Professional Home Page

Web Server

- ApacheFriends – XAMPP:
 - Bundle: PHP, MySQL, Perl, Mail Server, FTP Server




The screenshot shows the ApacheFriends website. At the top, there is a navigation bar with the text "...APACHE FRIENDS..." and several icons representing different services: Home (a dog), XAMPP (the ApacheFriends logo), News (a newspaper), Team (a group of people), Projects (a green folder), Docs (a book), Extras (a stack of papers), Links (a hand pointing), and Support (a hand holding a heart). Above the navigation bar, there are language links: deutsch, english, français, italiano, portugues (brasil), 日本語, 繁體中文, 简体中文.

Below the navigation bar, there is a promotional banner for "Free 30 Day Snagit Trial" with the text "Capture, Add Effects & Share From Your PC Desktop. Free Trial!" and the URL "TechSmith.com/Snagit". To the right of this banner is an "AdChoices" link.

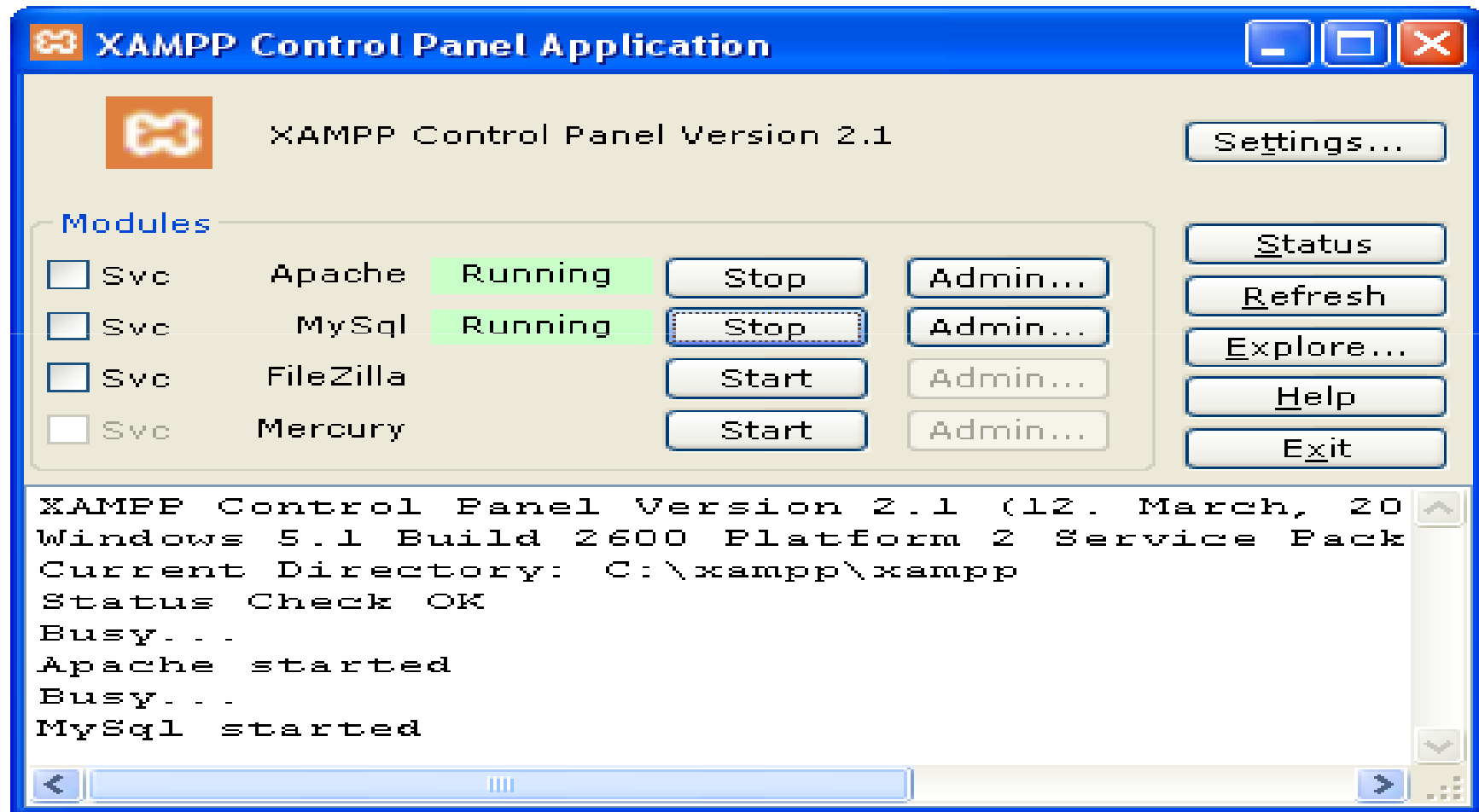
The main heading is "XAMPP for Windows" with the ApacheFriends logo to the left. Below this, the text reads "We've released the new version of XAMPP 1.7.7, including:" followed by a bulleted list of components:

- Apache 2.2.21
- MySQL 5.5.16
- PHP 5.3.8
- phpMyAdmin 3.4.5
- FileZilla FTP Server 0.9.39
- Tomcat 7.0.21 (with mod_proxy_ajp as connector)

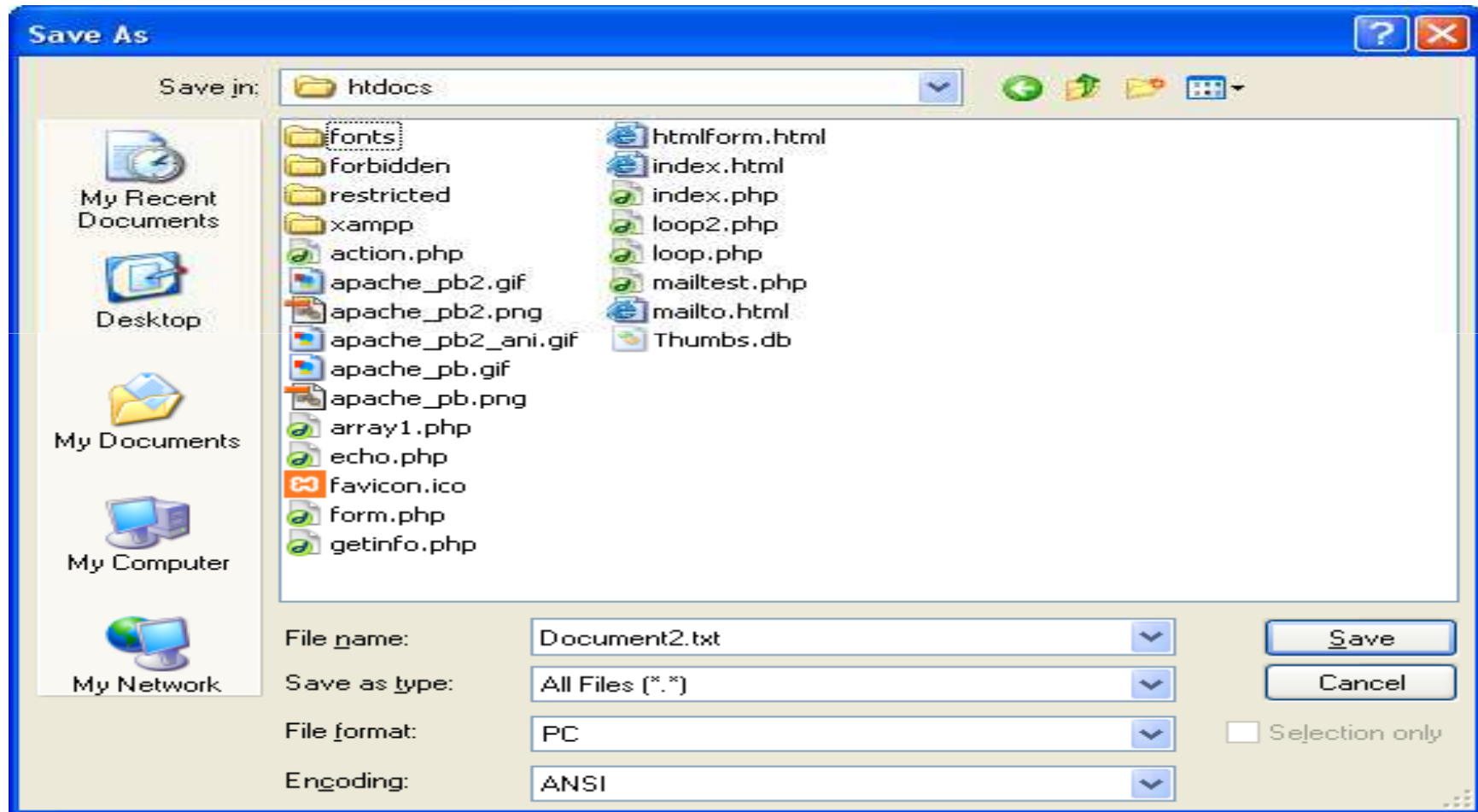
Below the list, the text says "Download older versions of XAMPP (even the 'old' WAMPP) directly from  SourceForge."

On the right side of the page, there is a photograph of a woman with red hair, wearing a pink and orange patterned sleeveless top and blue jeans, leaning against a white surface.

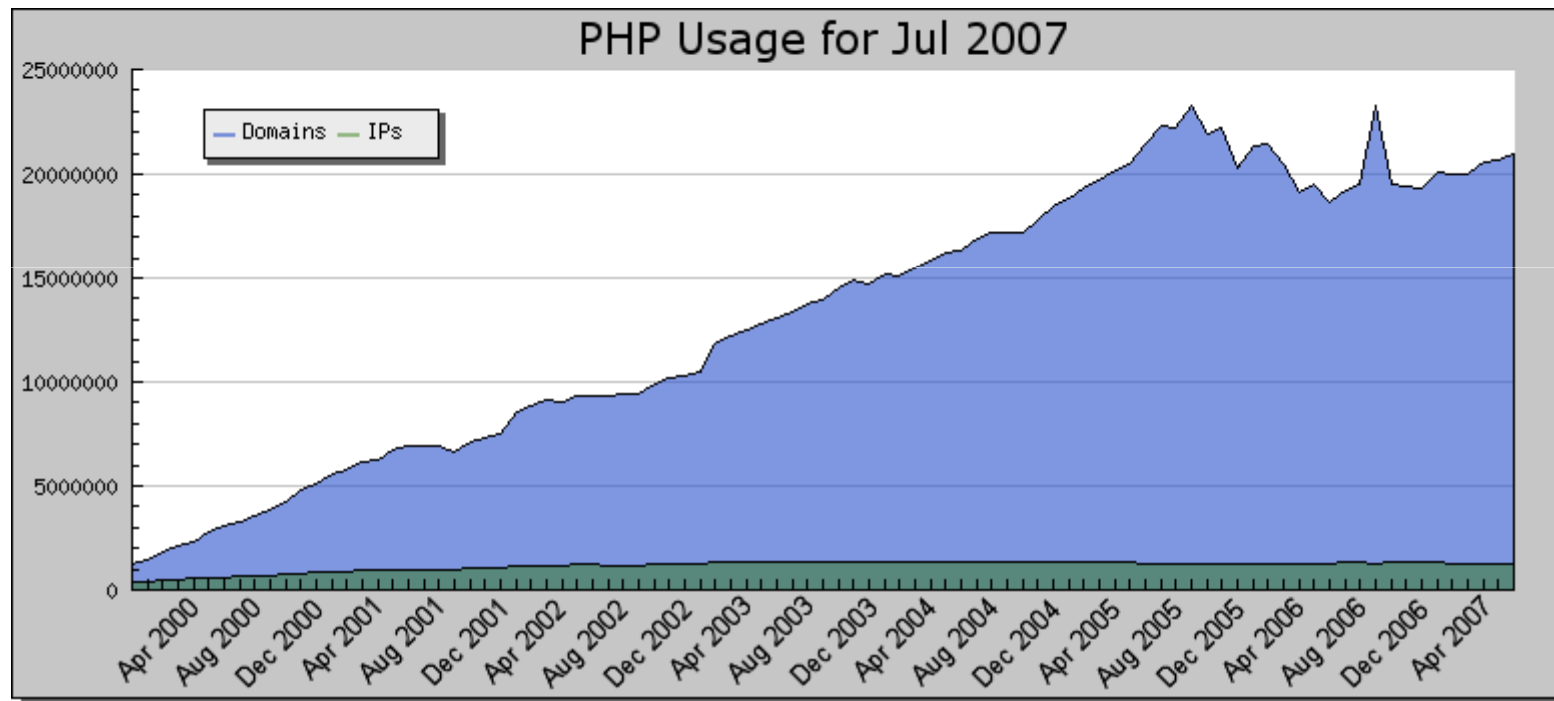
Xampp Control panel running



For xampp, put **php** here:
xampp\apache\htdocs



PHP Usage

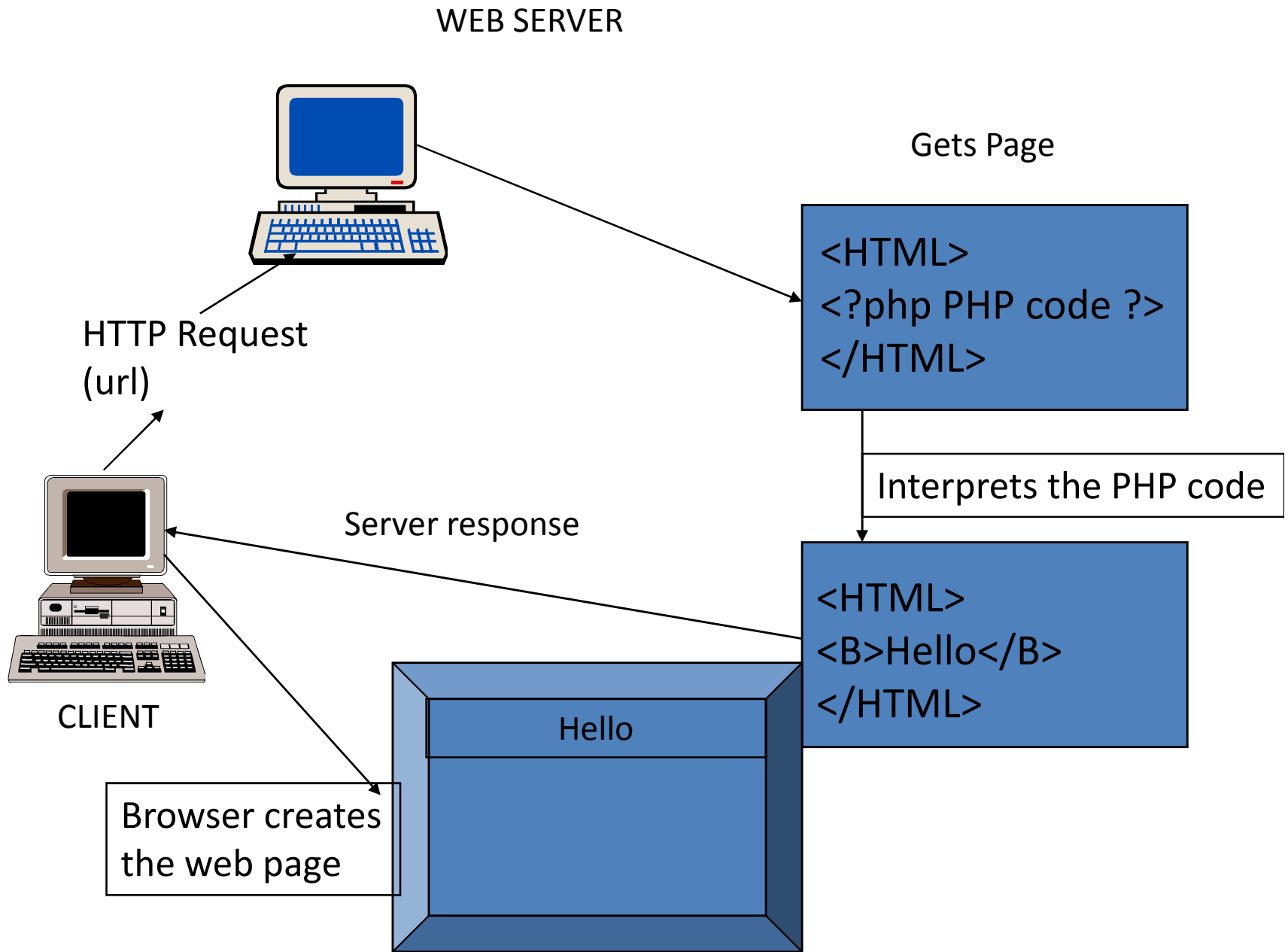


PHP in TIOBE Index

Position Sep 2011	Position Sep 2010	Delta in Position	Programming Language	Ratings Sep 2011	Delta Sep 2010	Status
1	1	=	Java	18.761%	+0.85%	A
2	2	=	C	18.002%	+0.86%	A
3	3	=	C++	8.849%	-0.96%	A
4	6	↑↑	C#	6.819%	+1.80%	A
5	4	↓	PHP	6.596%	-1.77%	A
6	8	↑↑	Objective-C	6.158%	+2.79%	A
7	5	↓↓	(Visual) Basic	4.420%	-1.38%	A
8	7	↓	Python	4.000%	-0.58%	A
9	9	=	Perl	2.472%	+0.03%	A
10	11	↑	JavaScript	1.469%	-0.20%	A
11	10	↓	Ruby	1.434%	-0.47%	A
12	12	=	Delphi/Object Pascal	1.313%	-0.27%	A
13	24	↑↑↑↑↑↑↑↑↑↑	Lua	1.154%	+0.60%	A
14	13	↓	Lisp	1.043%	-0.04%	A
15	15	=	Transact-SQL	0.860%	+0.09%	A
16	14	↓↓	Pascal	0.845%	+0.06%	A-
17	20	↑↑↑	PL/SQL	0.720%	+0.08%	A--
18	19	↑	Ada	0.682%	+0.01%	B
19	17	↓↓	RPG (OS/400)	0.666%	-0.05%	B
20	30	↑↑↑↑↑↑↑↑↑↑	D	0.609%	+0.20%	B

Sejarah PHP

- Dibuat oleh Rasmus Lerdorf (1994)
 - PHP/FI2.0 (Personal Homepage)
- Versi PHP3 (1997)
 - Zeev Suraski and Andi Gutmans (**Zend**)
- Versi PHP4 (2000) menjadi sangat populer di web
 - Berkembang ke OOP (Java, C#, VB.NET)
- Kemudian munculah PHP5 yang support OOP

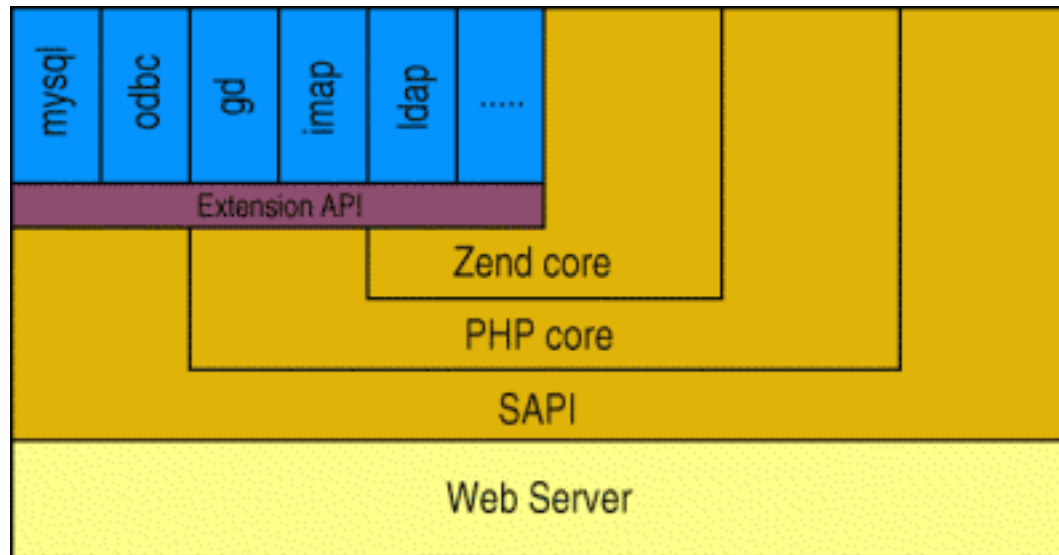


Yang baru di PHP 5

- New and improved **MySQL extension** (mysqli.dll)
- PHP 5 bundles **SQLite** (sqlite.dll sqlite3.dll)
- Supports **SimpleXML** extension (XML)
- New Iterators and Structure Programming Language
- More Error Handling and Debugging
- Supports Streams, Filters, and Wrappers

PHP 5 Architecture

- **Zend** engine as parser (Andi Gutmans and Zeev Suraski)
- **SAPI** (Server Application Programming Interface) is a web server abstraction layer - php5apache2.dll
- PHP **components** now self contained (ODBC, Java, LDAP, etc.)
- This structure is a good general design for software (compare to OSI model, and middleware applications)



Why PHP?

- Cross Platform
 - Dapat jalan hampir di semua web server pada beberapa sistem operasi
 - Web server: Apache, Microsoft IIS, Caudium, Netscape Enterprise Server
 - Sistem operasi: NIX (HP-UX, OpenBSD, Solaris, Linux), Mac OSX, Windows NT/98/2000/XP/2003
 - Database: Adabas D, dBase, Empress, FilePro (read-only), Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL, SQLite, Solid, Sybase, Velocis, Unix dbm
- Keuntungan dalam biaya
 - PHP is free.

PHP Block

- PHP code block is embedded within the **<?php** and **?>** tags.
- When the server encounters the **PHP tags** it **switches** from the HTML to PHP mode.
- There are four different ways to embed the PHP code
 - `<?php echo("Some PHP code"); ?>`
 - `<? echo("Some PHP code"); ?>`
 - `<SCRIPT Language='php'> echo("Some PHP code"); </SCRIPT>`
 - `<% echo("Some PHP code"); %>`

PHP - Variables

- Prefixed with a **\$**
- Assign values with **= operator**
- No need to define type
- Variable names are case sensitive
 - \$author and \$Author are different
- A variable name must start with a letter or an underscore "_"
- A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z, 0-9, and _)
- A variable name should not contain spaces
- If a variable name is more than one word, it should be separated with an underscore (\$my_string), or with capitalization (\$myString)

PHP Data Type

- Three **basic** data types
 - Integer
 - Double
 - String
- More data types
 - Array
 - Object
- PHP is an untyped language
 - variables type can change on the fly.

Float

- Ukuran float tergantung pada platform, walaupun maksimum $\sim 1.8e208$ (64 bit format IEEE)
 - \$a = 1.234;
 - \$b = 1.2e4;
 - \$c = 7E-10;

String

- Literal string dideklarasikan dengan:
 - Petik ganda (“ ”). contoh: `$a = “Nama: $nama\n”;`
 - Petik tunggal (' '). contoh: `$b = 'c:*. *';`
- Pengaksesan karakter string
 - `$a = “ini test”;`
 - `$pertama = $a[0];`
 - `$ketiga = $a[2];`
 - `$akhir = $a[strlen($a)-1];`

First PHP

- `<html>`
`<body>`

```
<?php  
echo "<strong>Hello PHP</strong>";  
echo "<br/>I love web programming";  
?>
```

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

Konversi String ke Angka

- Contoh:

- \$a = 1 + "10.5"; 11.5
- \$a = 1 + "-1.3e3"; -1299
- \$a = 1 + "bob-1.3e3"; 1
- \$a = 1 + "bob3"; 1
- \$a = 1 + "10 ayam kate"; 11
- \$a = 1 + "10.2 ayam kate"; 11.2
- \$a = "10.0 ayam" + 1; 11

Contoh Tipe Data

```
$testing = 5; //integer  
echo gettype($testing)."<br>";  
$testing = "five"; //string  
echo gettype($testing)."<br>";  
$testing = 5.0; //double  
echo gettype($testing)."<br>";  
$testing = true; //boolean  
echo gettype($testing)."<br>";
```

integer
string
double
boolean

```
$varumum = 3.14;  
$vardouble = (double)$varumum;  
echo gettype($vardouble)."<br>"; //double  
$varstring = (string)$varumum;  
echo gettype($varstring)."<br>"; //string
```

double
string

Useful string functions

- `str_replace()`
- `trim()`, `ltrim()`, `rtrim()`
- `implode()`, `explode()`
- `addslashes()`, `stripslashes()`
- `htmlentities()`, `html_entity_decode()`,
`htmlspecialchars()`
- `striptags()`

Contoh

```
$kalimat = "<strong>Percobaan pakai bold</strong><hr>";  
echo $kalimat;  
$kalimat = strip_tags($kalimat);  
echo $kalimat."<br>";  
  
$kalimat = str_replace("bold", "<strong>tebal</strong>", $kalimat);  
echo $kalimat."<br>";  
  
$coba = "Ibu berkata, 'Nak saya harus pergi sekarang!'";  
$coba = addslashes($coba);  
echo $coba;
```

Percobaan pakai bold

Percobaan pakai bold

Percobaan pakai tebal

Ibu berkata, \'Nak saya harus pergi sekarang!\'

Operator

- Untuk String : titik (.)
- Untuk Numerik : +, -, *, /, %, ++, --
- Untuk Boolean: &&, ||, !

Operator Perbandingan

Example	Name	Result
<code>\$a == \$b</code>	Equal	TRUE if \$a is equal to \$b.
<code>\$a === \$b</code>	Identical	TRUE if \$a is equal to \$b, and they are of the same type. (PHP 4 only)
<code>\$a != \$b</code>	Not equal	TRUE if \$a is not equal to \$b.
<code>\$a <> \$b</code>	Not equal	TRUE if \$a is not equal to \$b.
<code>\$a !== \$b</code>	Not identical	TRUE if \$a is not equal to \$b, or they are not of the same type. (PHP 4 only)
<code>\$a < \$b</code>	Less than	TRUE if \$a is strictly less than \$b.
<code>\$a > \$b</code>	Greater than	TRUE if \$a is strictly greater than \$b.
<code>\$a <= \$b</code>	Less than or equal to	TRUE if \$a is less than or equal to \$b.
<code>\$a >= \$b</code>	Greater than or equal to	TRUE if \$a is greater than or equal to \$b.

Ternary operator

```
<?php
// Contoh Ternary Operator
$action = (empty($_POST['action'])) ? 'default' : $_POST['action'];

// sama dengan if/else berikut:
if (empty($_POST['action'])) {
    $action = 'default';
} else {
    $action = $_POST['action'];
}
?>
```

Perbandingan

```
<?php
    if ($a > $b) {
        echo "a is bigger than b";
    } elseif ($a == $b) {
        echo "a is equal to b";
    } else {
        echo "a is smaller than b";
    }
?>
```

Switch

```
<?php
if ($i == 0) {
    echo "i equals 0";
} elseif ($i == 1) {
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
}
```

```
switch ($i) {
case 0:
    echo "i equals 0";
    break;
case 1:
    echo "i equals 1";
    break;
case 2:
    echo "i equals 2";
    break;
}
?>
```

Switch (2)

```
<?php
switch ($i) {
case 0:
case 1:
case 2:
    echo "0 <= i < 3 ";
    break;
case 3:
    echo "i is 3";
}
?>
```

```
<?php
$i = 0;
switch ($i) {
default:
    echo "default";
case 0:
case 1:
case 2:
    echo "test";
    break;
case 3:
    echo "i is 3";
}
?>
```

Struktur While

```
<?php
    $i = 1;
    while ($i <= 10) {
        echo $i++;
    }
    $i = 1;
    while ($i <= 10):
        echo $i;
        $i++;
    endwhile;
?>
```

```
<?php
    $i = 10;
    do{
        echo $i;
    }while($i>=1);
?>
```

Struktur for

```
<?php
/* example 1 */

for ($i = 1; $i <= 10; $i++) {
    echo $i;
}
```

```
/* example 2 */

for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    echo $i;
}
```

```
/* example 3 */
$i = 1;
for (; ; ) {
    if ($i > 10) {
        break;
    }
    echo $i;
    $i++;
}
```

```
/* example 4 */
for ($i = 1; $i <= 10; print $i,
    $i++);
?>
```

Cara Looping

- Dapat digunakan pada array!

```
$arr = array("one", "two", "three");  
//cara pertama  
while (list($key, $value) = each ($arr)) {  
    echo "Key: $key; Value: $value<br />";  
}  
//cara kedua  
foreach ($arr as $key => $value) {  
    echo "Key: $key; Value: $value<br />";  
}
```

```
Key: 0; Value: one  
Key: 1; Value: two  
Key: 2; Value: three  
Key: 0; Value: one  
Key: 1; Value: two  
Key: 2; Value: three
```

- Tanpa sebut index

```
foreach ($arr as $value) {  
    echo "Value: $value<br />";  
}
```

```
Value: one  
Value: two  
Value: three
```

Break

```
<?
$counter=1;
while($counter<10){
    if($counter==5){
        echo "berhenti jika counter bernilai 5 <br>";
        break;
    }
    echo $counter."<br>";
    $counter++;
}
?>
```


Continue

```
<?
```

```
for($counter=1;$counter<=10;$counter++){
```

```
    if($counter==5) continue;
```

```
        echo "counter bernilai: $counter<br>";
```

```
}
```

```
?>
```

Array

- In PHP, there are three kind of arrays:
 - **Numeric array** - An array with a numeric index
`$cars=array("Saab","Volvo","BMW","Toyota");`
 - **Associative array** - An array where each ID key is associated with a value
`$ages = array("Peter"=>32, "arc"=>30, "Joe"=>34);`
 - **Multidimensional array** - An array containing one or more arrays

Array

- Mendefinisikan array dengan array() function
`$users = array("tom", "sharon", "jhon", "hary");`

- Dengan array identifier

```
$user[] = "tom"; $user[] = "sharon";
```

```
$user[] = "jhon"; $user[] = "hary";
```

- Contoh penggunaan array

```
<? //foreach.php
```

```
$user= array("tom", "bert", "sharon", "jhon");
```

```
foreach ($user as $key=>$value) {
```

```
    echo "Array Ke-" . $key . " isinya:
```

```
    " . $value . "<br>"; }
```

```
?>
```

Associative Array

```
$multi = array("Peter"=>"Northon", "Quagmire"=>30, "Joe"=>"true");  
foreach($multi as $key=>$val){  
    echo $key." = ".$val."<br>";  
}
```

Peter = Northon

Quagmire = 30

Joe = true

Custom Array

```
$a = array(5 => 43, 32, 56, "b" => 12);  
$b = array(5 => 43, 6 => 32, 7 => 56, "b" => 12);  
  
foreach($a as $key=>$val){  
    echo $key." = ".$val."<br>";  
}
```

```
5 = 43  
6 = 32  
7 = 56  
b = 12
```

- Size of array is **not defined**
- If you add a new element the maximum of the integer indices is taken, and the new key will be that maximum value + **1**

Fungsi-fungsi ttg Array

- implode() -> menggabungkan array menjadi string
- explode() -> memecah string menjadi array
- sort() -> mengurutkan array scr asc
- rsort() -> mengurutkan array scr desc
- ksort() -> mengurutkan index string array scr asc
- array_pop() -> menghapus array dr urutan terakhir
- array_push() -> menambah array
- array_flip() swaps keys for values
- array_count_values() returns an associative array of all values in an array, and their frequency
- array_rand() pulls a random element
- array_unique() removes duppies
- count() returns the number of elements in an array
- array_search() returns the key for the first match in an array

Contoh penggunaan explode/implode

```
$city[0]="Jogja";  
$city[1]="Bandung";  
$city[3]="Surabaya";  
$strgabung = implode("-", $city);  
echo "Stlh digabung: ".$strgabung."<br>";  
$cityarray = explode("-", $strgabung);  
foreach ($cityarray as $key=>$value){  
    echo "Array ke-".$key." = ".$value."<br>";  
}
```

Stlh digabung: Jogja-Bandung-Surabaya

Array ke-0 = Jogja

Array ke-1 = Bandung

Array ke-2 = Surabaya

Function (by value)

- **Function definition**

```
function my_function()  
{  
    echo 'My function was called';  
}
```

```
function fungsi_return($a,$b){  
    return $a + $b;  
}
```

- **Calling function**

```
my_function();  
$c = fungsi_return(1,2);
```


Function (by reference)

<?

```
function tax(&$salary){
```

```
    $salary = $salary-(( $\$salary/100$ )*20);
```

```
    return $salary;
```

```
}
```

```
$salary = 2000;
```

```
echo tax($salary); //hasil 1600
```

```
echo $salary; //hasil 1600
```

?>

Parameter default function

<?

```
function tax($salary=2000){  
    $salary = $salary-(($salary/100)*20);  
    return $salary;  
}
```

```
echo tax();
```

?>

Include vs Require

- The **include()** function takes **all the content** in a specified file and includes it in the current file.
- If an error occurs, the **include()** function generates a **warning**, but the script will **continue** execution.
- The **require()** function is **identical** to `include()`, **except** that it handles **errors** differently.
- If an error occurs, **require()** generates a **fatal error**, and the script will **stop**.

Contoh include

```
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a> |
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Tutorial</title>
</head>

<body>

<div class="leftmenu">
<?php include("leftmenu.php"); ?>
</div>

<h1>Welcome to my home page.</h1>
<p>Ini teks biasa.</p>

</body>
</html>
```

[Home](#) [Tutorials](#) [References](#) [Examples](#) [About Us](#) [Contact Us](#)

Welcome to my home page.

Ini teks biasa.

Include vs Require

```
<html>
<body>

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

</body>
</html>
```

Warning: require(wrongFile.php) [[function.require](#)]: failed to open stream: No such file or directory in C:\xampplite\htdocs\tutorial\index.php on line 6

Fatal error: require() [[function.require](#)]: Failed opening required 'wrongFile.php' (include_path='.:C:\xampplite\php\PEAR') in C:\xampplite\htdocs\tutorial\index.php on line 6

Diganti require

Warning: include(wrongFile.php) [[function.include](#)]: failed to open stream: No such file or directory in C:\xampplite\htdocs\tutorial\index.php on line 6

Warning: include() [[function.include](#)]: Failed opening 'wrongFile.php' for inclusion (include_path='.:C:\xampplite\php\PEAR') in C:\xampplite\htdocs\tutorial\index.php on line 6

Hello World!

NULL

- NULL menyatakan variabel yang tidak ada nilainya
- Sebuah variabel NULL, jika
 - Dinyatakan sebagai NULL dengan operator “=”
 - Belum pernah diberikan suatu nilai apapun
 - Telah di **unset()**
- Untuk mengecek apakah variabel NULL atau tidak, dapat digunakan fungsi **is_null()**

date() dan mktime()

- Format: `date(format,timestamp)`
 - d - Represents the day of the month (01 to 31)
 - m - Represents a month (01 to 12)
 - Y - Represents a year (in four digits)
- Format:
`mktime(hour,minute,second,month,day,year,is_dst)`

phpinfo()

- The phpinfo() function shows the php environment
- Use this to read **system** and **server variables**, setting stored in **php.ini**, versions, and modules
- Notice that many of these data are in **arrays**

phpinfo()



System	Windows NT ANTONIE-PERPUS 6.1 build 7600 (Unknow Windows version Business Edition) i586
Build Date	Jan 6 2011 17:50:45
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--disable-isapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet" "--with-mcrypt=static"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626

SERVER variable

- `$_SERVER` is an array containing information such as
 - Headers
 - Paths
 - Script locations
- The entries in this array are created by the webserver.
- There is no guarantee that every webserver will provide any of these; servers may omit some, or provide others

SERVER variable

- 'REMOTE_ADDR'
 - The IP address from which the user is viewing the current page.
- 'REMOTE_HOST'
 - The Host name from which the user is viewing the current page. The reverse dns lookup is based off the REMOTE_ADDR of the user.
- 'REMOTE_PORT'
 - The port being used on the user's machine to communicate with the web server.
- \$_COOKIE
 - An associative array of variables passed to the current script via HTTP cookies. Automatically global in any scope.
- \$_POST
 - An associative array of variables passed to the current script via the HTTP POST method.

contoh

```
<html>
<body>
<p>
<?php
    echo $_SERVER['HTTP_USER_AGENT']."<br />";
    echo $_SERVER['SCRIPT_NAME']."<br />";
    echo $_SERVER['PHP_AUTH_USER'];
?>
</p>
</body>
</html>
```

Output:

```
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 1.1.4322; InfoPath.1; .NET CLR
2.0.50727)
/~mdd/test.php
mdd
```

Percobaan Login

- Buat 2 textbox untuk login:
 - Username dan password
- Buat 1 button untuk Submit
- Form action="proseslogin.php"
- Method="POST"

The image shows a login form with a red dashed border. It contains two textboxes: one for 'Username:' and one for 'Password:'. Below the textboxes is a 'Login' button. In the bottom-left corner of the form, there is a small icon of a yellow box with 'PHP' written on it.

Array global pada PHP

- `$_GET["variabel"]` – untuk menerima variabel pada URL secara GET
- `$_POST["variabel"]` – untuk menerima variabel dari POST form
- `$_REQUEST["variabel"]` – untuk menerima variabel GET dan POST
- `$_FILES["variabel"]` – untuk menerima upload file
- `$_SESSION["varname"]` – untuk mengakses session
- `$_COOKIE["varname"]` – untuk mengakses cookies

Session

- `session_start();`
- `$_SESSION["<nama>"] = "<value>";`
- `session_unregister(<nama,nama,nama>`
- `if (isset($_SESSION["<nama>"])){ }`
- `session_regenerate_id();`
- `unset("<nama>")`
- `session_destroy()`

Kode Login

```
<?php
session_start();
$con = mysql_connect("localhost","root","");
mysql_select_db("test",$con);
$user = $_POST["username"];
$pass = $_POST["password"];
$hasil = mysql_query("select * from users3 where username='$user' and
password='$pass'", $con);
if($hasil){
    if(mysql_num_rows($hasil)==1){
        $_SESSION["UNAME"] = $user;
        $_SESSION["PASS"] = $pass;
        $_SESSION["LOG-IN"] = "true";
        session_regenerate_id(true);
        if (!headers_sent()) header("Location: home.php");
    }
    else {
        header("Location: masuk.html");
    }
}
?>
```


Home

```
<?php
session_start();
if(isset($_SESSION["UNAME"]) && $_SESSION["UNAME"] != ""){
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
Hello, <?php echo "<strong><em>".$_SESSION["UNAME"]."</em></strong>"; ?><br />
Anda sedang mengakses halaman <strong>home</strong>.<br />
Silahkan <a href="logout.php">logout</a>
</body>
</html>
<?php
} else echo "Anda belum login!";
?>
```

Hello, *anton*

Anda sedang mengakses halaman **home**.

Silahkan [logout](#)

Logout

```
<?php
session_start();
unset($_SESSION["UNAME"]);
unset($_SESSION["PASS"]);
$_SESSION["LOG-IN"] = "false";
header("Location: masuk.html");
?>
```

NEXT

- PHP Database