

1. PHP – Introduction

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common Uses of PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. The other uses of PHP are:

- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.



Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

"Hello World" Script in PHP

To get a feel of PHP, first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this:

```
<html>
<head>
<title>Hello World</title>
<body>
  <?php echo "Hello, World!";?>
</body>
</html>
```

It will produce the following result:

```
Hello, World!
```

If you examine the HTML output of the above example, you'll notice that the PHP code is not present in the file sent from the server to your Web browser. All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.

All PHP code must be included inside one of the three special markup tags that are recognized by the PHP Parser.

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

<? PHP code goes here ?>

<script language="php"> PHP code goes here </script>
```

Most common tag is the `<?php...?>` and we will also use the same tag in our tutorial.

From the next chapter, we will start with PHP Environment Setup on your machine and then we will dig out almost all concepts related to PHP to make you comfortable with the PHP language.



2. PHP – Environment Setup

In order to develop and run PHP Web pages, three vital components need to be installed on your computer system.

Web Server - PHP will work with virtually all **Web Server** software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here: <http://httpd.apache.org/download.cgi>

Database - PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here: <http://www.mysql.com/downloads/index.html>

PHP Parser - In order to process PHP script instructions, a parser must be installed to generate HTML output that can be sent to the **Web Browser**. This tutorial will guide you how to install PHP parser on your computer.

PHP Parser Installation

Before you proceed, it is important to make sure that you have a proper environment setup on your machine to develop your web programs using PHP.

Type the following address into your browser's address box.

`http://127.0.0.1/info.php`

If this displays a page showing your PHP installation related information, then it means you have PHP and Webserver installed properly. Otherwise you have to follow given procedure to install PHP on your computer.

This section will guide you to install and configure PHP over the following four platforms:

- [PHP Installation on Linux or Unix with Apache](#)
- [PHP Installation on Mac OS X with Apache](#)
- [PHP Installation on Windows NT/2000/XP with IIS](#)
- [PHP Installation on Windows NT/2000/XP with Apache](#)

PHP Installation on Linux or Unix with Apache

If you plan to install PHP on Linux or any other variant of Unix, then here is the list of prerequisites:

- The PHP source distribution <http://www.php.net/downloads.php>
The latest Apache source distribution
<http://httpd.apache.org/download.cgi>



- A working PHP-supported database, if you plan to use one (For example MySQL, Oracle etc.)
- Any other supported software to which PHP must connect (mail server, BCMath package, JDK, and so forth)
- An ANSI C compiler
- Gnu make utility - you can freely download it at <http://www.gnu.org/software/make>

Now here are the steps to install Apache and PHP5 on your Linux or Unix machine. If your PHP or Apache versions are different, then please take care accordingly.

- If you haven't already done so, unzip and untar your Apache source distribution. Unless you have a reason to do otherwise, /usr/local is the standard place.

```
gunzip -c apache_1.3.x.tar.gz
tar -xvf apache_1.3.x.tar
```

- Build the apache Server as follows

```
cd apache_1.3.x
./configure --prefix=/usr/local/apache --enable-so
make
make install
```

- Unzip and untar your PHP source distribution. Unless you have a reason to do otherwise, /usr/local is the standard place.

```
gunzip -c php-5.x.tar.gz
tar -xvf php-5.x.tar
cd php-5.x
```

- Configure and Build your PHP, assuming you are using MySQL database.

```
./configure --with-apxs=/usr/sbin/apxs \
            --with-mysql=/usr/bin/mysql
make
make install
```

- Install the php.ini file. Edit this file to get configuration directives:

```
cd ../../php-5.x
cp php.ini-dist /usr/local/lib/php.ini
```

- Tell your Apache server where you want to serve files from, and what extension(s) you want to identify PHP files. **.php** is the standard, but you can use .html, .phtml, or whatever you want.
 - Go to your HTTP configuration files (/usr/local/apache/conf or whatever your path is)
 - Open httpd.conf with a text editor.



- Search for the word DocumentRoot (which should appear twice), and change both paths to the directory you want to serve files out of (in our case, /home/httpd). We recommend a home directory rather than the default /usr/local/apache/htdocs because it is more secure, but it doesn't have to be in a home directory. You will keep all your PHP files in this directory.
- Add at least one PHP extension directive, as shown in the first line of code that follows. In the second line, we've also added a second handler to have all HTML files parsed as PHP

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php .html
```

- Restart your server. Every time you change your HTTP configuration or php.ini files, you must stop and start your server again.

```
cd ../bin
./apachectl start
```

- Set the document root directory permissions to world-executable. The actual PHP files in the directory need only be world-readable (644). If necessary, replace /home/httpd with your document root below:

```
chmod 755 /home/httpd/html/php
```

- Open a text editor. Type: `<?php phpinfo(); ?>`. Save this file in your Web server's document root as info.php.
- Start any Web browser and browse the file. You must always use an HTTP request (`http://www.testdomain.com/info.php` or `http://localhost/info.php` or `http://127.0.0.1/info.php`) rather than a filename (`/home/httpd/info.php`) for the file to be parsed correctly

You should see a long table of information about your new PHP installation message Congratulations!

PHP Installation on Mac OS X with Apache

Mac users have the choice of either a binary or a source installation. In fact, your OS X probably came with Apache and PHP preinstalled. This is likely to be quite an old build, and it probably lacks many of the less common extensions.

However, if all you want is a quick Apache + PHP + MySQL/PostgreSQL setup on your laptop, this is certainly the easiest way to fly. All you need to do is edit your Apache configuration file and turn on the Web server.

So just follow the steps given below:

- Open the Apache config file in a text editor as root.

```
sudo open -a TextEdit /etc/httpd/httpd.conf
```



- Edit the file. Uncomment the following lines:

```
Load Module php5_module
AddModule mod_php5.c
AddType application/x-httpd-php .php
```

- You may also want to uncomment the `<Directory /home/*/Sites>` block or otherwise tell Apache which directory to serve out of.
- Restart the Web server

```
sudo apachectl graceful
```

- Open a text editor. Type: `<?php phpinfo(); ?>`. Save this file in your Web server's document root as `info.php`.
- Start any Web browser and browse the file. you must always use an HTTP request (`http://www.testdomain.com/info.php` or `http://localhost/info.php` or `http://127.0.0.1/info.php`) rather than a filename (`/home/httpd/info.php`) for the file to be parsed correctly

You should see a long table of information about your new PHP installation message Congratulations!

PHP Installation on Windows NT/2000/XP with IIS

The Windows server installation of PHP running IIS is much simpler than on Unix, since it involves a precompiled binary rather than a source build.

If you plan to install PHP over Windows, then here is the list of prerequisites:

- A working PHP-supported Web server. Under previous versions of PHP, IIS/PWS was the easiest choice because a module version of PHP was available for it; but PHP now has added a much wider selection of modules for Windows.
- A correctly installed PHP-supported database like MySQL or Oracle etc. (if you plan to use one)
- The PHP Windows binary distribution (download it at www.php.net/downloads.php)
- A utility to unzip files (search <http://download.cnet.com> for PC file compression utilities)

Now here are the steps to install Apache and PHP5 on your Windows machine. If your PHP version is different, then please take care accordingly.

- Extract the binary archive using your unzip utility; `C:\PHP` is a common location.
- Copy some .dll files from your PHP directory to your systems directory (usually `C:\Winnt\System32`). You need `php5ts.dll` for every case. You will also probably need to copy the file corresponding to your Web server module - `C:\PHP\Sapi\php5isapi.dll`. It's possible you will also need others from the dlls subfolder - but start with the two mentioned above and add more if you need them.



- Copy either php.ini-dist or php.ini-recommended (preferably the latter) to your Windows directory (C:\Winnt or C:\Winnt40), and rename it php.ini. Open this file in a text editor (for example, Notepad). Edit this file to get configuration directives; We highly recommend new users set error reporting to E_ALL on their development machines at this point. For now, the most important thing is the doc_root directive under the Paths and Directories section. make sure this matches your IIS Inetpub folder (or wherever you plan to serve out of).
- Stop and restart the WWW service. Go to the Start menu -> Settings -> Control Panel -> Services. Scroll down the list to IIS Admin Service. Select it and click Stop. After it stops, select World Wide Web Publishing Service and click Start. Stopping and restarting the service from within Internet Service Manager will not suffice. Since this is Windows, you may also wish to reboot.
- Open a text editor. Type: `<?php phpinfo(); ?>`. Save this file in your Web server's document root as info.php.
- Start any Web browser and browse the file. you must always use an HTTP request (`http://www.testdomain.com/info.php` or `http://localhost/info.php` or `http://127.0.0.1/info.php`) rather than a filename (`/home/httpd/info.php`) for the file to be parsed correctly

You should see a long table of information about your new PHP installation message Congratulations!

PHP Installation on Windows NT/2000/XP with Apache

To install Apache with PHP 5 on Windows follow the following steps. If your PHP and Apache versions are different, then please take care accordingly.

- Download Apache server from www.apache.org/dist/httpd/binaries/win32. You want the current stable release version with the no_src.msi extension. Double-click the installer file to install; C:\Program Files is a common location. The installer will also ask you whether you want to run Apache as a service or from the command line or DOS prompt. We recommend you do not install as a service, as this may cause problems with startup.
- Extract the PHP binary archive using your unzip utility; C:\PHP is a common location.
- Copy some .dll files from your PHP directory to your system directory (usually C:\Windows). You need php5ts.dll for every case. You will also probably need to copy the file corresponding to your Web server module - C:\PHP\Sapi\php5apache.dll. to your Apache modules directory. It's possible that you will also need others from the dlls subfolder, but start with the two mentioned previously and add more if you need them.
- Copy either php.ini-dist or php.ini-recommended (preferably the latter) to your Windows directory, and rename it php.ini. Open this file in a text editor (for example, Notepad). Edit this file to get configuration directives; At this point, we highly recommend that new users set error reporting to E_ALL on their development machines.



- Tell your Apache server where you want to serve files from and what extension(s) you want to identify PHP files (.php is the standard, but you can use .html, .phtml, or whatever you want). Go to your HTTP configuration files (C:\Program Files\Apache Group\Apache\conf or whatever your path is), and open httpd.conf with a text editor. Search for the word DocumentRoot (which should appear twice) and change both paths to the directory you want to serve files out of. (The default is C:\Program Files\Apache Group\Apache\htdocs.). Add at least one PHP extension directive as shown in the first line of the following code:

```
LoadModule php5_module modules/php5apache.dll
AddType application/x-httpd-php .php .phtml
```

- You may also need to add the following line:

```
AddModule mod_php5.c
```

- Stop and restart the WWW service. Go to the Start menu -> Settings -> Control Panel -> Services. Scroll down the list to IIS Admin Service. Select it and click Stop. After it stops, select World Wide Web Publishing Service and click Start. Stopping and restarting the service from within Internet Service Manager will not suffice. Since this is Windows, you may also wish to reboot.
- Open a text editor. Type: <?php phpinfo(); ?>. Save this file in your Web server's document root as info.php.
- Start any Web browser and browse the file. you must always use an HTTP request (http://www.testdomain.com/info.php or http://localhost/info.php or http://127.0.0.1/info.php) rather than a filename (/home/httpd/info.php) for the file to be parsed correctly

You should see a long table of information about your new PHP installation message Congratulations!

Apache Configuration

If you are using Apache as a Web Server, then this section will guide you to edit Apache Configuration Files.

PHP.INI File Configuration

The PHP configuration file, php.ini, is the final and most immediate way to affect PHP's functionality.

Just Check it here: [PHP.INI File Configuration](#)

Windows IIS Configuration

To configure IIS on your Windows machine you can refer your IIS Reference Manual shipped along with IIS.



Apache Configuration for PHP

Apache uses `httpd.conf` file for global settings, and the `.htaccess` file for per-directory access settings. Older versions of Apache split up `httpd.conf` into three files (`access.conf`, `httpd.conf`, and `srml.conf`), and some users still prefer this arrangement.

Apache server has a very powerful, but slightly complex, configuration system of its own. Learn more about it at the Apache Web site: www.apache.org

The following section describes settings in `httpd.conf` that affect PHP directly and cannot be set elsewhere. If you have standard installation, then `httpd.conf` will be found at `/etc/httpd/conf`:

Timeout

This value sets the default number of seconds before any HTTP request will time out. If you set PHP's `max_execution_time` to longer than this value, PHP will keep grinding away but the user may see a 404 error. In safe mode, this value will be ignored; you must use the timeout value in `php.ini` instead

DocumentRoot

`DocumentRoot` designates the root directory for all HTTP processes on that server. It looks something like this on Unix:

```
DocumentRoot ./usr/local/apache_1.3.6/htdocs.
```

You can choose any directory as document root.

AddType

The PHP MIME type needs to be set here for PHP files to be parsed. Remember that you can associate any file extension with PHP like `.php3`, `.php5` or `.htm`.

```
AddType application/x-httpd-php .php
AddType application/x-httpd-phps .phps
AddType application/x-httpd-php3 .php3 .phtml
AddType application/x-httpd-php .html
```

Action

You must uncomment this line for the Windows `apxs` module version of Apache with shared object support:

```
LoadModule php4_module modules/php4apache.dll
```

or on Unix flavors:

```
LoadModule php4_module modules/mod_php.so
```



AddModule

You must uncomment this line for the static module version of Apache.

```
AddModule mod_php4.c
```

PHP.INI file Configuration

The PHP configuration file, `php.ini`, is the final and most immediate way to affect PHP's functionality. The `php.ini` file is read each time PHP is initialized. In other words, whenever `httpd` is restarted for the module version or with each script execution for the CGI version. If your change isn't showing up, remember to stop and restart `httpd`. If it still isn't showing up, use `phpinfo()` to check the path to `php.ini`.

The configuration file is well commented and thorough. Keys are case sensitive, keyword values are not; whitespace, and lines beginning with semicolons are ignored. Booleans can be represented by 1/0, Yes/No, On/Off, or True/False. The default values in `php.ini-dist` will result in a reasonable PHP installation that can be tweaked later.

Here we are explaining the important settings in `php.ini` which you may need for your PHP Parser.

short_open_tag = Off

Short open tags look like this: `<? ?>`. This option must be set to Off if you want to use XML functions.

safe_mode = Off

If this is set to On, you probably compiled PHP with the `--enable-safe-mode` flag. Safe mode is most relevant to CGI use. See the explanation in the section "CGI compile-time options". earlier in this chapter.

safe_mode_exec_dir = [DIR]

This option is relevant only if safe mode is on; it can also be set with the `--with-exec-dir` flag during the Unix build process. PHP in safe mode only executes external binaries out of this directory. The default is `/usr/local/bin`. This has nothing to do with serving up a normal PHP/HTML Web page.

safe_mode_allowed_env_vars = [PHP_]

This option sets which environment variables users can change in safe mode. The default is only those variables prepended with "PHP_". If this directive is empty, most variables are alterable.

safe_mode_protected_env_vars = [LD_LIBRARY_PATH]

This option sets which environment variables users can't change in safe mode, even if `safe_mode_allowed_env_vars` is set permissively.



disable_functions = [function1, function2...]

A welcome addition to PHP4 configuration and one perpetuated in PHP5 is the ability to disable selected functions for security reasons. Previously, this necessitated hand-editing the C code from which PHP was made. Filesystem, system, and network functions should probably be the first to go because allowing the capability to write files and alter the system over HTTP is never such a safe idea.

max_execution_time = 30

The function `set_time_limit()` won't work in safe mode, so this is the main way to make a script time out in safe mode. In Windows, you have to abort based on maximum memory consumed rather than time. You can also use the Apache timeout setting to timeout if you use Apache, but that will apply to non-PHP files on the site too.

error_reporting = E_ALL & ~E_NOTICE

The default value is `E_ALL & ~E_NOTICE`, all errors except notices. Development servers should be set to at least the default; only production servers should even consider a lesser value

error_prepend_string = [""]

With its bookend, `error_append_string`, this setting allows you to make error messages a different color than other text, or what you have.

warn_plus_overloading = Off

This setting issues a warning if the `+` operator is used with strings, as in a form value.

variables_order = EGPCS

This configuration setting supersedes `gpc_order`. Both are now deprecated along with `register_globals`. It sets the order of the different variables: Environment, GET, POST, COOKIE, and SERVER (aka Built-in).

You can change this order around. Variables will be overwritten successively in left-to-right order, with the rightmost one winning the hand every time. This means if you left the default setting and happened to use the same name for an environment variable, a POST variable, and a COOKIE variable, the COOKIE variable would own that name at the end of the process. In real life, this doesn't happen much.

register_globals = Off

This setting allows you to decide whether you wish to register EGPCS variables as global. This is now deprecated, and as of PHP4.2, this flag is set to Off by default. Use superglobal arrays instead. All the major code listings in this book use superglobal arrays.

gpc_order = GPC

This setting has been GPC Deprecated.



magic_quotes_gpc = On

This setting escapes quotes in incoming GET/POST/COOKIE data. If you use a lot of forms which possibly submit to themselves or other forms and display form values, you may need to set this directive to On or prepare to use addslashes() on string-type data.

magic_quotes_runtime = Off

This setting escapes quotes in incoming database and text strings. Remember that SQL adds slashes to single quotes and apostrophes when storing strings and does not strip them off when returning them. If this setting is Off, you will need to use stripslashes() when outputting any type of string data from a SQL database. If magic_quotes_sybase is set to On, this must be Off.

magic_quotes_sybase = Off

This setting escapes single quotes in incoming database and text strings with Sybase-style single quotes rather than backslashes. If magic_quotes_runtime is set to On, this must be Off.

auto-prepend-file = [path/to/file]

If a path is specified here, PHP must automatically include() it at the beginning of every PHP file. Include path restrictions do apply.

auto-append-file = [path/to/file]

If a path is specified here, PHP must automatically include() it at the end of every PHP file, unless you escape by using the exit() function. Include path restrictions do apply.

include_path = [DIR]

If you set this value, you will only be allowed to include or require files from these directories. The include directory is generally under your document root; this is mandatory if you're running in safe mode. Set this to . in order to include files from the same directory your script is in. Multiple directories are separated by colons: ./usr/local/apache/htdocs:/usr/local/lib.

doc_root = [DIR]

If you're using Apache, you've already set a document root for this server or virtual host in httpd.conf. Set this value here if you're using safe mode or if you want to enable PHP only on a portion of your site (for example, only in one subdirectory of your Web root).

file_uploads = [on/off]

Turn on this flag if you will upload files using PHP script.

upload_tmp_dir = [DIR]

Do not uncomment this line unless you understand the implications of HTTP uploads!

session.save-handler = files

Except in rare circumstances, you will not want to change this setting. So don't touch it.



ignore_user_abort = [On/Off]

This setting controls what happens if a site visitor clicks the browser's Stop button. The default is On, which means that the script continues to run to completion or timeout. If the setting is changed to Off, the script will abort. This setting only works in module mode, not CGI.

mysql.default_host = hostname

The default server host to use when connecting to the database server if no other host is specified.

mysql.default_user = username

The default user name to use when connecting to the database server if no other name is specified.

mysql.default_password = password

The default password to use when connecting to the database server if no other password is specified.

