PHP 7

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

PHP 7 is the most awaited and is a major feature release of PHP programming language. PHP 7 was released on 3rd Dec 2015.

This tutorial will teach you the new features of PHP 7 and their usage in a simple and intuitive way.

Audience

This tutorial has been prepared for PHP developers from a beginner’s point of view. After completing this tutorial, you will find yourself at a moderate level of expertise in the knowledge of PHP from where you can take yourself to next levels.

Prerequisites

We assume that you already know about the older versions of PHP and now you can start learning the new features of PHP 7.

Execute PHP-7 Online

For most of the examples given in this tutorial, you will find an option **Try it.** Just use this option to execute your PHP-7 programs at the spot and enjoy your learning.

Try the following example using **Try it** option available at the top right corner of the below sample code box –

```
<html>
   <head>
      <title>Online PHP-7 Script Execution</title>
   </head>
      <body>
      <?php
         echo "<h1>Hello, PHP-7!</h1>";
      ?>
   </body>
</html>
```

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial i
Audience i
Prerequisites i
Execute PHP-7 Online i
Copyright & Disclaimer ii
Table of Contents iii

1. PHP 7 – Introduction 1
What is PHP 7? 1
New Features 1

2. PHP 7 – Performance 3
Magento 1.9 3
Drupal 7 4
Wordpress 3.6 4
Comparison of Dynamic Languages 5

3. PHP 7 – Environment Setup 6
Try it Option Online 6
PHP Parser Installation 7
Installation on Linux/Unix 7
Installation on Mac OS X 10
Installation on Windows with IIS 11
Installation on Windows with Apache 12
Apache Configuration 13
PHP Configuration in Apache 13
PHP.INI File Configuration 14
PHP.INI Configuration 14
Windows IIS Configuration 18

4. PHP 7 – Scalar Type Declarations 19
Example – Coercive Mode 19

5. PHP 7 – Return Type Declarations 21
Example - Valid Return Type 21
Example - Invalid Return Type 21

6. PHP 7 – Null Coalescing Operator 23

7. PHP 7 – Spaceship Operator 24

8. PHP 7 – Constant Arrays 26

9. PHP 7 – Anonymous Classes 27

10. PHP 7 – Closure::call() 28
Example – Pre PHP 7 28
Example – PHP 7+ 28

11. PHP 7 – Filtered unserialize() 30

12. PHP 7 – IntlChar 32

13. PHP 7 – CSPRNG 33
random_bytes() 33
random_int() 34

14. PHP 7 – Expectations 35
Configuration directives for assert() 35

15. PHP 7 – use Statement 37

16. PHP 7 – Error Handling 38

17. PHP 7 – Integer Division 40

18. PHP 7 – Session Options 41

19. PHP 7 – Deprecated Features 42
PHP 4 Style Constructors 42
Static Calls to Non-Static Methods 42
password_hash() salt option 43
capture_session_meta SSL context option 43

20. PHP 7 – Removed Extensions & SAPIs 44

1. PHP 7 – INTRODUCTION

What is PHP 7?

PHP 7 is a major release of PHP programming language and is touted to be a revolution in the way web applications can be developed and delivered for mobile to enterprises and the cloud. This release is considered to be the most important change for PHP after the release of PHP 5 in 2004.

New Features

There are dozens of features added to PHP 7, the most significant ones are mentioned below -

- **Improved performance** - Having PHPNG code merged in PHP7, it is twice as fast as PHP 5.

- **Lower Memory Consumption** - Optimized PHP 7 utilizes lesser resource.

- **Scalar type declarations** - Now parameter and return types can be enforced.

- **Consistent 64-bit support** - Consistent support for 64-bit architecture machines.

- **Improved Exception hierarchy** - Exception hierarchy is improved.

- **Many fatal errors converted to Exceptions** - Range of exceptions is increased covering many fatal error converted as exceptions.

- **Secure random number generator** - Addition of new secure random number generator API.

- **Deprecated SAPIs and extensions removed** - Various old and unsupported SAPIs and extensions are removed from the latest version.

- **The null coalescing operator (??) -** New null coalescing operator added.

- **Return and Scalar Type Declarations -** Support for return type and parameter type added.

- **Anonymous Classes -** Support for anonymous added.

- **Zero cost asserts -** Support for zero cost assert added.

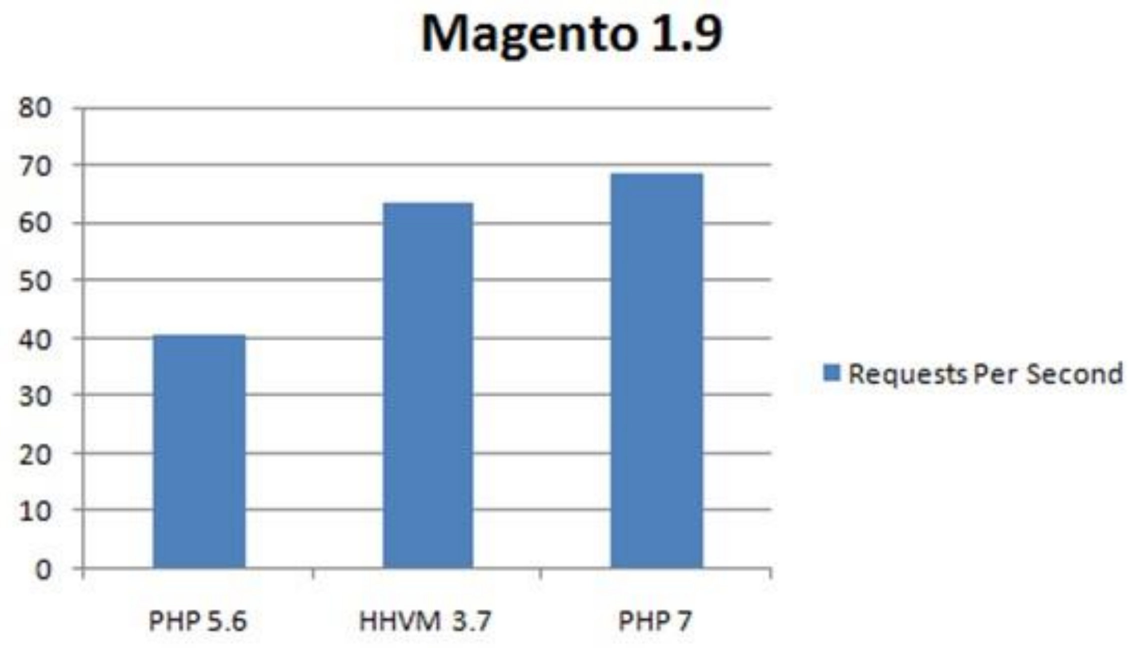
PHP 7 uses new Zend Engine 3.0 to improve application performance almost twice and 50% better memory consumption than PHP 5.6. It allows to serve more concurrent users without requiring any additional hardware. PHP 7 is designed and refactored considering today's workloads.

2. PHP 7 – PERFORMANCE

As per the Zend team, following illustrations show the performance comparison of PHP 7 vs PHP 5.6 and HHVM 3.7 on popular PHP based applications.

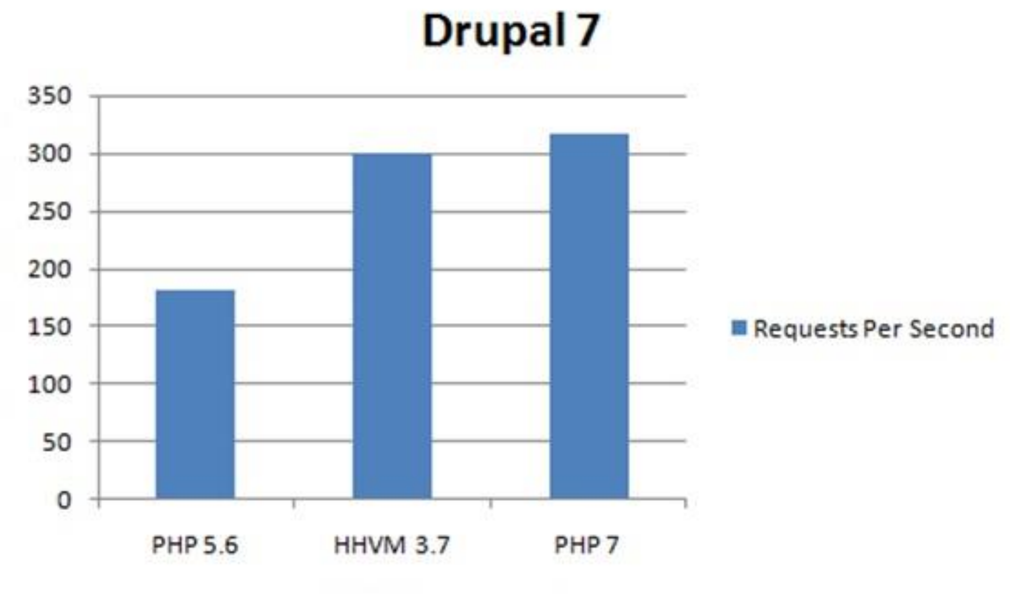
Magento 1.9

PHP 7 proves itself more than twice as faster, as compared to PHP 5.6 while executing Magento transactions.



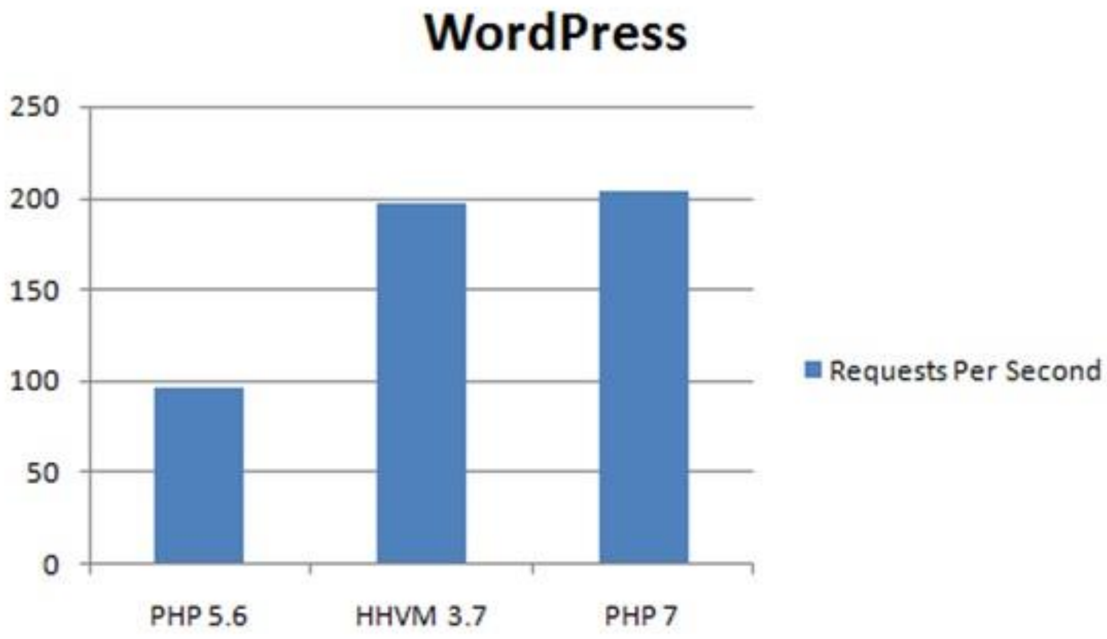
Drupal 7

PHP 7 proves itself more than twice as faster, as compared to PHP 5.6 while executing Drupal transactions.

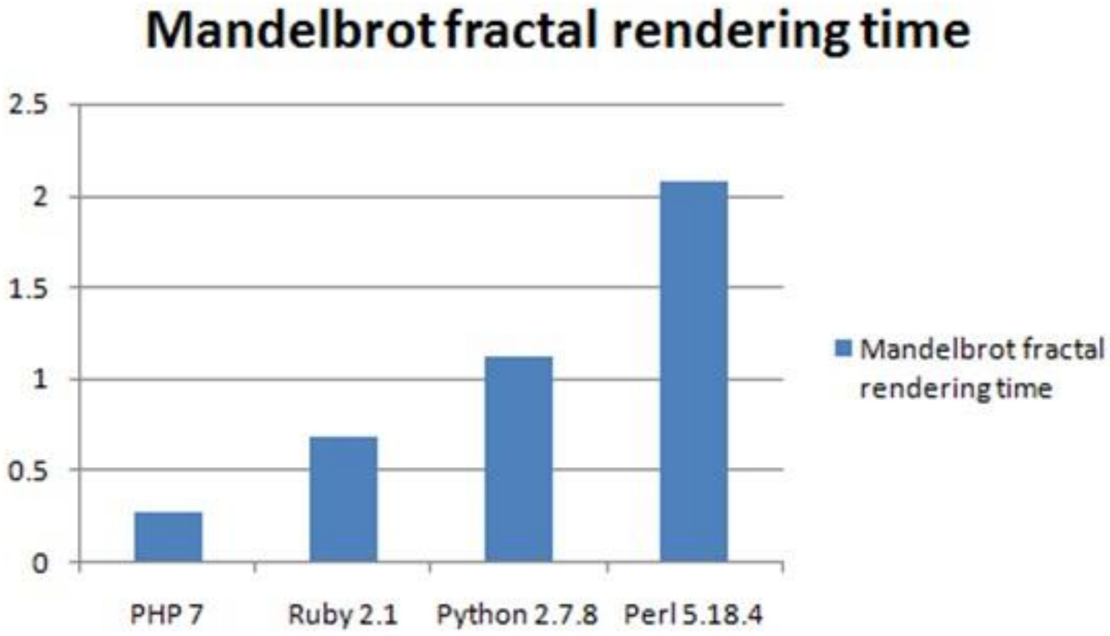


Wordpress 3.6

PHP 7 proves itself more than twice as faster as compared to PHP 5.6 while executing Wordpress transactions.



Comparison of Dynamic Languages



3. PHP 7 – ENVIRONMENT SETUP

Try it Option Online

We have set up the PHP Programming environment on-line, so that you can compile and execute all the available examples online. It gives you confidence in what you are reading and enables you to verify the programs with different options. Feel free to modify any example and execute it online.

Try the following example using our online compiler available at CodingGround.

```
<html>
   <head>
      <title>Online PHP Script Execution</title>
   </head>
   <body>
      <?php
         echo "<h1>Hello, PHP!</h1>";
      ?>
   </body>
</html>
```

For most of the examples given in this tutorial, you will find a **Try it** option in our website code sections at the top right corner that will take you to the online compiler. So just use of and enjoy your learning.

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

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

- **Database** – PHP works with virtually all database software, including Oracle and Sybase but most commonly used is MySQL database. Download MySQL for free here –http://www.mysql.com/downloads/

- **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 proper environment setup on your machine to develop your web programs using PHP. Store the following php file in Apache's htdocs folder.

phpinfo.php

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

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

```
http://127.0.0.1/phpinfo.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 the 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

Installation on Linux / Unix

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.

Step 1

If you have not 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_2.4.x.tar.gz

tar -xvf apache_2.4.x.tar
```

Step 2

Build the apache Server as follows -

```
cd apache_2.4.x

./configure --prefix=/usr/local/apache --enable-so

make

make install
```

Step 3

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

```
gunzip -c php-7.x.tar.gz

tar -xvf php-7.x.tar

cd php-7.x
```

Step 4

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
```

Step 5

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

```
cd ../../php-7.x

cp php.ini-development /usr/local/lib/php.ini
```

Step 6

- Tell your Apache server where you want to serve files from, and what extension(s) you want to identify PHP files. A **.php** extension is 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 the 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 does not 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 the code that follows. In the second line, we have 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
```

Step 7

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
```

Step 8

Set the documentroot-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
```

Step 9

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 will see a long table of information about your new PHP installation message **Congratulations!**

Installation on Mac OS X

Mac users have a 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.

Just follow the steps given below–

Step 1

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

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

Step 2

Edit the file. Uncomment the following lines –

```
Load Module php7_module

AddModule mod_php7.c

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

Step 3

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

```
sudo apachectl graceful
```

Step 4

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 will see a long table of information about your new PHP installation message **Congratulations**!

Installation on Windows 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.

1. Extract the binary archive using your unzip utility; C:\PHP is a common location.

2. 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\php7isapi.dll. It is possible you will also need other files from the dlls subfolder, but start with the two files mentioned above and add more if you need them.

3. Copy either the php.ini-development 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 the configuration directives. We highly recommend the new users to 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 the IIS Inetpub folder (or wherever you plan to serve out of).

4. 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 the World Wide Web Publishing Service and click Start. Stopping and restarting the service from within the Internet Service Manager will not suffice. Since this is Windows, you may also wish to reboot.

5. Open a text editor. Type: <?php phpinfo(); ?>. Save this file in your Web server's document root as info.php.

6. 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 will see a long table of information about your new PHP installation message **Congratulations**!

Installation on Windows with Apache

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

Step 1

- 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:\php7 is a common location.

- Rename php.ini-development to php.ini. Open this file in a text editor (for example, Notepad). Edit this file to get the configuration directives. At this point, we highly recommend that the 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 the 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 the 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 –

```
AddHandler application/x-httpd-php .php

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

LoadModule php7_module "C:/php7/php7apache2_4.dll"

PHPiniDir "c:/php7"
```

Step 2

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 will 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.

Check here – PHP Configuration in Apache Server

PHP Configuration in Apache

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 srm.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 the 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; instead, you must use the timeout value in php.ini.

DocumentRoot

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

```
DocumentRoot ./usr/local/apache_2.4.0/htdocs.
```

You can choose any directory as the 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 php7_module modules/php7apache2_4.dll
```

on Unix flavors –

```
LoadModule php7_module modules/mod_php.so
```

AddModule

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

```
AddModule mod_php7.c
```

PHP.INI File Configuration

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

Check here – PHP.INI File Configuration

PHP.INI Configuration

The PHP configuration file, php.ini, is the final and immediate way to affect PHP's functionality. The php.ini file is read each time PHP is initialized. In other words, httpd is restarted for the module version or with each script execution for the CGI version. If your change is not showing up, remember to stop and restart httpd. If it is still not 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 the XML functions.

safe_mode = Off

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

safe_mode_exec_dir = [DIR]

This option is relevant only if the safe mode is ON; it can also be set with the --with-exec-dir flag during the Unix build process.