= PHP =

PHP is a server @-@ side scripting language designed for web development but also used as a general @-@ purpose programming language . Originally created by Rasmus Lerdorf in 1994 , the PHP reference implementation is now produced by The PHP Group . PHP originally stood for Personal Home Page , but it now stands for the recursive backronym PHP : Hypertext Preprocessor .

PHP code may be embedded into HTML code , or it can be used in combination with various web template systems , web content management systems and web frameworks . PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable . The web server combines the results of the interpreted and executed PHP code , which may be any type of data , including images , with the generated web page . PHP code may also be executed with a command @-@ line interface (CLI) and can be used to implement standalone graphical applications .

The standard PHP interpreter , powered by the Zend Engine , is free software released under the PHP License . PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform , free of charge .

The PHP language evolved without a written formal specification or standard until 2014 , leaving the canonical PHP interpreter as a de facto standard . Since 2014 work has gone on to create a formal PHP specification .

During the 2010s there have been increased efforts towards standardisation and code sharing in PHP applications by projects such as PHP @-@ FIG in the form of PSR @-@ initiatives as well as Composer dependency manager and the Packagist repository .

= = History = =

= = = Early history = = =

PHP development began in 1995 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C , which he used to maintain his personal homepage . He extended them to work with web forms and to communicate with databases , and called this implementation " Personal Home Page / Forms Interpreter " or PHP / FI .

PHP / FI could be used to build simple , dynamic web applications . To accelerate bug reporting and improve the code , Lerdorf initially announced the release of PHP / FI as " Personal Home Page Tools (PHP Tools) version 1 @.@ 0 " on the Usenet discussion group comp.infosystems.www.authoring.cgi on June 8 , 1995 . This release already had the basic functionality that PHP has as of 2013 . This included Perl @-@ like variables , form handling , and the ability to embed HTML . The syntax resembled that of Perl but was simpler , more limited and less consistent .

Early PHP was not intended to be a new programming language , and grew organically , with Lerdorf noting in retrospect : " I don ? t know how to stop it , there was never any intent to write a programming language [?] I have absolutely no idea how to write a programming language , I just kept adding the next logical step on the way . " A development team began to form and , after months of work and beta testing , officially released PHP / FI 2 in November 1997 .

The fact that PHP was not originally designed but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters . In some cases , the function names were chosen to match the lower @-@ level libraries which PHP was " wrapping " , while in some very early versions of PHP the length of the function names was used internally as a hash function , so names were chosen to improve the distribution of hash values .

= = = PHP 3 and 4 = = =

Zeev Suraski and Andi Gutmans rewrote the parser in 1997 and formed the base of PHP 3 , changing the language 's name to the recursive acronym PHP : Hypertext Preprocessor . Afterwards , public testing of PHP 3 began , and the official launch came in June 1998 . Suraski and Gutmans then started a new rewrite of PHP 's core , producing the Zend Engine in 1999 . They also founded Zend Technologies in Ramat Gan , Israel .
On May 22 , 2000 , PHP 4 , powered by the Zend Engine 1 @.@ 0 , was released . As of August 2008 this branch reached version 4 @.@ 4 @.@ 9 . PHP 4 is no longer under development nor will any security updates be released .

= = = PHP 5 = = =

On July 14 , 2004 , PHP 5 was released , powered by the new Zend Engine II . PHP 5 included new features such as improved support for object @-@ oriented programming , the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases) , and numerous performance enhancements . In 2008 PHP 5 became the only stable version under development . Late static binding had been missing from PHP and was added in version 5 @.@ 3 .
Many high @-@ profile open @-@ source projects ceased to support PHP 4 in new code as of February 5 , 2008 , because of the GoPHP5 initiative , provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5 .
Over time , PHP interpreters became available on most existing 32 @-@ bit and 64 @-@ bit operating systems , either by building them from the PHP source code , or by using pre @-@ built binaries . For the PHP versions 5 @.@ 3 and 5 @.@ 4 , the only available Microsoft Windows binary distributions were 32 @-@ bit x86 builds , requiring Windows 32 @-@ bit compatibility mode while using Internet Information Services (IIS) on a 64 @-@ bit Windows platform . PHP version 5 @.@ 5 made the 64 @-@ bit x86 @-@ 64 builds available for Microsoft Windows .

= = = PHP 6 and Unicode = = =

PHP received mixed reviews due to lacking native Unicode support at the core language level . In 2005 , a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP , by embedding the International Components for Unicode (ICU) library , and representing text strings as UTF @-@ 16 internally . Since this would cause major changes both to the internals of the language and to user code , it was planned to release this as version 6 @.@ 0 of the language , along with other major features then in development .
However , a shortage of developers who understood the necessary changes , and performance problems arising from conversion to and from UTF @-@ 16 , which is rarely used in a web context , led to delays in the project . As a result , a PHP 5 @.@ 3 release was created in 2009 , with many non @-@ Unicode features back @-@ ported from PHP 6 , notably namespaces . In March 2010 , the project in its current form was officially abandoned , and a PHP 5 @.@ 4 release was prepared containing most remaining non @-@ Unicode features from PHP 6 , such as traits and closure re @-@ binding . Initial hopes were that a new plan would be formed for Unicode integration , but as of 2014 none have been adopted .

= = = PHP 7 = = =

During 2014 and 2015 , a new major PHP version was developed , which was numbered PHP 7 . The numbering of this version involved some debate . While the PHP 6 Unicode experiment had never been released , several articles and book titles referenced the PHP 6 name , which might have caused confusion if a new release were to reuse the name . After a vote , the name PHP 7 was chosen .
The foundation of PHP 7 is a PHP branch that was originally dubbed PHP next generation (phpng) . It was authored by Dmitry Stogov , Xinchen Hui and Nikita Popov , and aimed to optimize PHP

performance by refactoring the Zend Engine while retaining near @-@ complete language compatibility . As of 14 July 2014 , WordPress @-@ based benchmarks , which served as the main benchmark suite for the phpng project , showed an almost 100 % increase in performance . Changes from phpng are also expected to make it easier to improve performance in the future , as more compact data structures and other changes are seen as better suited for a successful migration to a just @-@ in @-@ time (JIT) compiler . Because of the significant changes , the reworked Zend Engine is called Zend Engine 3 , succeeding Zend Engine 2 used in PHP 5 .

Because of major internal changes in phpng , it must receive a new major version number of PHP , rather than a minor PHP 5 release , according to PHP 's release process . Major versions of PHP are allowed to break backward @-@ compatibility of code and therefore PHP 7 presented an opportunity for other improvements beyond phpng that require backward @-@ compatibility breaks . In particular , it involved the following changes :

Many fatal- or recoverable @-@ level legacy PHP error mechanisms were replaced with modern object @-@ oriented exceptions

The syntax for variable dereferencing was reworked to be internally more consistent and complete , allowing the use of the operators - > , [ ] , ( ) , { } , and : : with arbitrary meaningful left @-@ hand @-@ side expressions

Support for legacy PHP 4 @-@ style constructor methods was deprecated

The behavior of the foreach statement was changed to be more predictable

Constructors for the few classes built @-@ in to PHP which returned null upon failure were changed to throw an exception instead , for consistency

Several unmaintained or deprecated server application programming interfaces (SAPIs) and extensions were removed from the PHP core , most notably the legacy mysql extension

The behavior of the list ( ) operator was changed to remove support for strings

Support for legacy ASP @-@ style PHP code delimiters (< % and % > , < script language = php > and < / script >) was removed

An oversight allowing a switch statement to have multiple default clauses was fixed

Support for hexadecimal number support in some implicit conversions from strings to number types was removed

The left @-@ shift and right @-@ shift operators were changed to behave more consistently across platforms

Conversions between integers and floating point numbers were tightened and implemented more consistently across platforms

PHP 7 also included new language features . Most notably , it introduces return type declarations for functions , which complement the existing parameter type declarations , and support for the scalar types (integer , float , string , and boolean) in parameter and return type declarations .

= = = Release history = = =

Beginning on June 28 , 2011 , the PHP Group implemented a timeline for the release of new versions of PHP . Under this system , at least one release should occur every month . Once per year , a minor release should occur which may include new features . Every minor release should at least be supported for two years with security and bug fixes , followed by at least one year of only security fixes , for a total of a three @-@ year release process for every minor release . No new features , unless small and self @-@ contained , are to be introduced into a minor release during the three @-@ year release process .

= = Mascot = =

The mascot of the PHP project is the elePHPant , a blue elephant with the PHP logo on its side , designed by Vincent Pontier in 1998 . The elePHPant is sometimes differently colored when in plush toy form .

= = Syntax = =

The following " Hello , World ! " program is written in PHP code embedded in an HTML document :
However , as no requirement exists for PHP code to be embedded in HTML , the simplest version of Hello , World ! may be written like this , with the closing tag omitted as preferred in files containing pure PHP code
The PHP interpreter only executes PHP code within its delimiters . Anything outside its delimiters is not processed by PHP , although non @-@ PHP text is still subject to control structures described in PHP code . The most common delimiters are < ? php to open and ? > to close PHP sections . The shortened form < ? also exists . This short delimiter makes script files less portable , since support for them can be disabled in the local PHP configuration and it is therefore discouraged . However , there is no recommendation against the use of the echo short tag < ? = . Prior to PHP 5 @.@ 4 @.@ 0 , this short syntax for echo ( ) only works with the short _ open _ tag configuration setting enabled , while for PHP 5 @.@ 4 @.@ 0 and later it is always available . The purpose of all these delimiters is to separate PHP code from non @-@ PHP content , such as JavaScript code or HTML markup .
The first form of delimiters , < ? php and ? > , in XHTML and other XML documents , creates correctly formed XML processing instructions . This means that the resulting mixture of PHP code and other markup in the server @-@ side file is itself well @-@ formed XML .
Variables are prefixed with a dollar symbol , and a type does not need to be specified in advance . PHP 5 introduced type hinting that allows functions to force their parameters to be objects of a specific class , arrays , interfaces or callback functions . However , before PHP 7 @.@ 0 , type hints could not be used with scalar types such as integer or string .
Unlike function and class names , variable names are case sensitive . Both double @-@ quoted (" ") and heredoc strings provide the ability to interpolate a variable 's value into the string . PHP treats newlines as whitespace in the manner of a free @-@ form language , and statements are terminated by a semicolon . PHP has three types of comment syntax : / * * / marks block and inline comments ; / / as well as # are used for one @-@ line comments . The echo statement is one of several facilities PHP provides to output text , e.g. , to a web browser .
In terms of keywords and language syntax , PHP is similar to the C style syntax. if conditions , for and while loops , and function returns are similar in syntax to languages such as C , C + + , C # , Java and Perl .

= = = Data types = = =

PHP stores integers in a platform @-@ dependent range , either a 64 @-@ bit or 32 @-@ bit signed integer equivalent to the C @-@ language long type . Unsigned integers are converted to signed values in certain situations ; this behavior is different from that of other programming languages . Integer variables can be assigned using decimal (positive and negative) , octal , hexadecimal , and binary notations .
Floating point numbers are also stored in a platform @-@ specific range . They can be specified using floating point notation , or two forms of scientific notation . PHP has a native Boolean type that is similar to the native Boolean types in Java and C + + . Using the Boolean type conversion rules , non @-@ zero values are interpreted as true and zero as false , as in Perl and C + + .
The null data type represents a variable that has no value ; NULL is the only allowed value for this data type .
Variables of the " resource " type represent references to resources from external sources . These are typically created by functions from a particular extension , and can only be processed by functions from the same extension ; examples include file , image , and database resources .
Arrays can contain elements of any type that PHP can handle , including resources , objects , and even other arrays . Order is preserved in lists of values and in hashes with both keys and values , and the two can be intermingled . PHP also supports strings , which can be used with single quotes , double quotes , nowdoc or heredoc syntax .

The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes .

= = = Functions = = =

PHP defines a large array of functions in the core language and many are also available in various extensions ; these functions are well documented in the online PHP documentation . However , the built @-@ in library has a wide variety of naming conventions and associated inconsistencies , as described under history above .
Custom functions may be defined by the developer , e.g. :
In 2016 , the output of the above sample program is ' I am currently 35 years old.'
In lieu of function pointers , functions in PHP can be referenced by a string containing their name . In this manner , normal PHP functions can be used , for example , as callbacks or within function tables . User @-@ defined functions may be created at any time without being prototyped . Functions may be defined inside code blocks , permitting a run @-@ time decision as to whether or not a function should be defined . There is a function _ exists function that determines whether a function with a given name has already been defined . Function calls must use parentheses , with the exception of zero @-@ argument class constructor functions called with the PHP operator new , in which case parentheses are optional .
Until PHP 5 @.@ 3 , support for anonymous functions and closures did not exist in PHP . While create _ function ( ) exists since PHP 4 @.@ 0 @.@ 1 , it is merely a thin wrapper around eval ( ) that allows normal PHP functions to be created during program execution . PHP 5 @.@ 3 added syntax to define an anonymous function or " closure " which can capture variables from the surrounding scope :
In the example above , getAdder ( ) function creates a closure using passed argument $ x (the keyword use imports a variable from the lexical context) , which takes an additional argument $ y , and returns the created closure to the caller . Such a function is a first @-@ class object , meaning that it can be stored in a variable , passed as a parameter to other functions , etc .
Unusually for a dynamically typed language , PHP supports type declarations on function parameters , which are enforced at runtime . This has been supported for classes and interfaces since PHP 5 @.@ 0 , for arrays since PHP 5 @.@ 1 , for " callables " since PHP 5 @.@ 4 , and scalar (integer , float , string and boolean) types since PHP 7 @.@ 0 . PHP 7 @.@ 0 also has type declarations for function return types , expressed by placing the type name after the list of parameters , preceded by a colon . For example , the getAdder function from the earlier example could be annotated with types like so in PHP 7 :
By default , scalar type declarations follow weak typing principles . So , for example , if a parameter 's type is int , PHP would allow not only integers , but also convertible numeric strings , floats or booleans to be passed to that function , and would convert them . However , PHP 7 has a " strict typing " mode which , when used , disallows such conversions for function calls and returns within a file .

= = = Object @-@ oriented programming = = =

Basic object @-@ oriented programming functionality was added in PHP 3 and improved in PHP 4 . This allowed for PHP to gain further abstraction , making creative tasks easier for programmers using the language . Object handling was completely rewritten for PHP 5 , expanding the feature set and enhancing performance . In previous versions of PHP , objects were handled like value types . The drawback of this method was that code had to make heavy use of PHP 's " reference " variables if it wanted to modify an object it was passed rather than creating a copy of it . In the new approach , objects are referenced by handle , and not by value .
PHP 5 introduced private and protected member variables and methods , along with abstract classes , final classes , abstract methods , and final methods . It also introduced a standard way of declaring constructors and destructors , similar to that of other object @-@ oriented languages such

as C + + , and a standard exception handling model . Furthermore , PHP 5 added interfaces and allowed for multiple interfaces to be implemented . There are special interfaces that allow objects to interact with the runtime system . Objects implementing ArrayAccess can be used with array syntax and objects implementing Iterator or IteratorAggregate can be used with the foreach language construct . There is no virtual table feature in the engine , so static variables are bound with a name instead of a reference at compile time .

If the developer creates a copy of an object using the reserved word clone , the Zend engine will check whether a _ _ clone ( ) method has been defined . If not , it will call a default _ _ clone ( ) which will copy the object 's properties . If a _ _ clone ( ) method is defined , then it will be responsible for setting the necessary properties in the created object . For convenience , the engine will supply a function that imports the properties of the source object , so the programmer can start with a by @-@ value replica of the source object and only override properties that need to be changed .

The following is a basic example of object @-@ oriented programming in PHP :

The visibility of PHP properties and methods is defined using the keywords public , private , and protected . The default is public , if only var is used ; var is a synonym for public . Items declared public can be accessed everywhere. protected limits access to inherited classes (and to the class that defines the item) . private limits visibility only to the class that defines the item . Objects of the same type have access to each other 's private and protected members even though they are not the same instance . PHP 's member visibility features have sometimes been described as " highly useful . " However , they have also sometimes been described as " at best irrelevant and at worst positively harmful . "

= = Implementations = =

The original , only complete and most widely used PHP implementation is powered by the Zend Engine and known simply as PHP . To disambiguate it from other implementations , it is sometimes unofficially referred to as " Zend PHP " . The Zend Engine compiles PHP source code on @-@ the @-@ fly into an internal format that it can execute , thus it works as an interpreter . It is also the " reference implementation " of PHP , as PHP has no formal specification , and so the semantics of Zend PHP define the semantics of PHP itself . Due to the complex and nuanced semantics of PHP , defined by how Zend works , it is difficult for competing implementations to offer complete compatibility .

PHP 's single @-@ request @-@ per @-@ script @-@ execution model , and the fact the Zend Engine is an interpreter , leads to inefficiency ; as a result , various products have been developed to help improve PHP performance . In order to speed up execution time and not have to compile the PHP source code every time the web page is accessed , PHP scripts can also be deployed in the PHP engine 's internal format by using an opcode cache , which works by caching the compiled form of a PHP script (opcodes) in shared memory to avoid the overhead of parsing and compiling the code every time the script runs . An opcode cache , Zend Opcache , is built into PHP since version 5 @.@ 5 . Another example of a widely used opcode cache is the Alternative PHP Cache (APC) , which is available as a PECL extension .

While Zend PHP is still the most popular implementation , several other implementations have been developed . Some of these are compilers or support JIT compilation , and hence offer performance benefits over Zend PHP at the expense of lacking full PHP compatibility . Alternative implementations include the following :

HipHop Virtual Machine (HHVM) ? developed at Facebook and available as open source , it converts PHP code into a high @-@ level bytecode (commonly known as an intermediate language) , which is then translated into x86 @-@ 64 machine code dynamically at runtime by a just @-@ in @-@ time (JIT) compiler , resulting in up to 6 × performance improvements .

Parrot ? a virtual machine designed to run dynamic languages efficiently ; Pipp transforms the PHP source code into the Parrot intermediate representation , which is then translated into the Parrot 's bytecode and executed by the virtual machine .

Phalanger ? compiles PHP into Common Intermediate Language (CIL) bytecode
HipHop ? developed at Facebook and available as open source , it transforms the PHP scripts into C + + code and then compiles the resulting code , reducing the server load up to 50 % . In early 2013 , Facebook deprecated it in favor of HHVM due to multiple reasons , including deployment difficulties and lack of support for the whole PHP language , including the create _ function ( ) and eval ( ) constructs .

= = Licensing = =

PHP is free software released under the PHP License , which stipulates that :
Products derived from this software may not be called " PHP " , nor may " PHP " appear in their name , without prior written permission from group @ php.net. You may indicate that your software works in conjunction with PHP by saying " Foo for PHP " instead of calling it " PHP Foo " or " phpfoo " .
This restriction on use of " PHP " makes the PHP License incompatible with the General Public License (GPL) , while the Zend License is incompatible due to an advertising clause similar to that of the original BSD license .

= = Development and community = =

PHP includes various free and open @-@ source libraries in its source distribution , or uses them in resulting PHP binary builds . PHP is fundamentally an Internet @-@ aware system with built @-@ in modules for accessing File Transfer Protocol (FTP) servers and many database servers , including PostgreSQL , MySQL , Microsoft SQL Server and SQLite (which is an embedded database) , LDAP servers , and others . Numerous functions familiar to C programmers , such as those in the stdio family , are available in standard PHP builds .
PHP allows developers to write extensions in C to add functionality to the PHP language . PHP extensions can be compiled statically into PHP or loaded dynamically at runtime . Numerous extensions have been written to add support for the Windows API , process management on Unix @-@ like operating systems , multibyte strings (Unicode) , cURL , and several popular compression formats . Other PHP features made available through extensions include integration with IRC , dynamic generation of images and Adobe Flash content , PHP Data Objects (PDO) as an abstraction layer used for accessing databases , and even speech synthesis . Some of the language 's core functions , such as those dealing with strings and arrays , are also implemented as extensions . The PHP Extension Community Library (PECL) project is a repository for extensions to the PHP language .
Some other projects , such as Zephir , provide the ability for PHP extensions to be created in a high @-@ level language and compiled into native PHP extensions . Such an approach , instead of writing PHP extensions directly in C , simplifies the development of extensions and reduces the time required for programming and testing .
The PHP Group consists of ten people (as of 2015) : Thies C. Arntzen , Stig Bakken , Shane Caraveo , Andi Gutmans , Rasmus Lerdorf , Sam Ruby , Sascha Schumann , Zeev Suraski , Jim Winstead , Andrei Zmievski .
Zend Technologies provides a PHP Certification based on PHP 5 @.@ 5 exam for programmers to become certified PHP developers .

= = Installation and configuration = =

There are two primary ways for adding support for PHP to a web server ? as a native web server module , or as a CGI executable . PHP has a direct module interface called Server Application Programming Interface (SAPI) , which is supported by many web servers including Apache HTTP Server , Microsoft IIS , Netscape (now defunct) and iPlanet . Some other web servers , such as OmniHTTPd , support the Internet Server Application Programming Interface (ISAPI) , which is a

Microsoft 's web server module interface . If PHP has no module support for a web server , it can always be used as a Common Gateway Interface (CGI) or FastCGI processor ; in that case , the web server is configured to use PHP 's CGI executable to process all requests to PHP files .
PHP @-@ FPM (FastCGI Process Manager) is an alternative FastCGI implementation for PHP , bundled with the official PHP distribution since version 5 @.@ 3 @.@ 3 . When compared to the older FastCGI implementation , it contains some additional features , mostly useful for heavily loaded web servers .
When using PHP for command @-@ line scripting , a PHP command @-@ line interface (CLI) executable is needed . PHP supports a CLI SAPI as of PHP 4 @.@ 3 @.@ 0 . The main focus of this SAPI is developing shell applications using PHP . There are quite a few differences between the CLI SAPI and other SAPIs , although they do share many of the same behaviors .
PHP has a direct module interface called SAPI for different web servers ; in case of PHP 5 and Apache 2 @.@ 0 on Windows , it is provided in form of a DLL file called php5apache2.dll , which is a module that , among other functions , provides an interface between PHP and the web server , implemented in a form that the server understands . This form is what is known as a SAPI .
There are different kinds of SAPIs for various web server extensions . For example , in addition to those listed above , other SAPIs for the PHP language include the Common Gateway Interface (CGI) and command @-@ line interface (CLI) .
PHP can also be used for writing desktop graphical user interface (GUI) applications , by using the PHP @-@ GTK extension . PHP @-@ GTK is not included in the official PHP distribution , and as an extension it can be used only with PHP versions 5 @.@ 1 @.@ 0 and newer . The most common way of installing PHP @-@ GTK is compiling it from the source code .
When PHP is installed and used in cloud environments , software development kits (SDKs) are provided for using cloud @-@ specific features . For example :
Amazon Web Services provides the AWS SDK for PHP
Windows Azure can be used with the Windows Azure SDK for PHP .
Numerous configuration options are supported , affecting both core PHP features and extensions . Configuration file php.ini is searched for in different locations , depending on the way PHP is used . The configuration file is split into various sections , while some of the configuration options can be also set within the web server configuration .

= = Use = =

PHP is a general @-@ purpose scripting language that is especially suited to server @-@ side web development , in which case PHP generally runs on a web server . Any PHP code in a requested file is executed by the PHP runtime , usually to create dynamic web page content or dynamic images used on websites or elsewhere . It can also be used for command @-@ line scripting and client @-@ side graphical user interface (GUI) applications . PHP can be deployed on most web servers , many operating systems and platforms , and can be used with many relational database management systems (RDBMS) . Most web hosting providers support PHP for use by their clients . It is available free of charge , and the PHP Group provides the complete source code for users to build , customize and extend for their own use .
PHP acts primarily as a filter , taking input from a file or stream containing text and / or PHP instructions and outputting another stream of data . Most commonly the output will be HTML , although it could be JSON , XML or binary data such as image or audio formats . Since PHP 4 , the PHP parser compiles input to produce bytecode for processing by the Zend Engine , giving improved performance over its interpreter predecessor .
Originally designed to create dynamic web pages , PHP now focuses mainly on server @-@ side scripting , and it is similar to other server @-@ side scripting languages that provide dynamic content from a web server to a client , such as Microsoft 's ASP.NET , Sun Microsystems ' JavaServer Pages , and mod _ perl . PHP has also attracted the development of many software frameworks that provide building blocks and a design structure to promote rapid application development (RAD) . Some of these include PRADO , CakePHP , Symfony , CodeIgniter , Laravel

, Yii Framework , Phalcon and Zend Framework , offering features similar to other web frameworks .

The LAMP architecture has become popular in the web industry as a way of deploying web applications . PHP is commonly used as the P in this bundle alongside Linux , Apache and MySQL , although the P may also refer to Python , Perl , or some mix of the three . Similar packages , WAMP and MAMP , are also available for Windows and OS X , with the first letter standing for the respective operating system . Although both PHP and Apache are provided as part of the Mac OS X base install , users of these packages seek a simpler installation mechanism that can be more easily kept up to date .
As of April 2007 , over 20 million Internet domains had web services hosted on servers with PHP installed and mod _ php was recorded as the most popular Apache HTTP Server module . As of October 2010 , PHP was used as the server @-@ side programming language on 75 % of all websites whose server @-@ side programming language was known (as of February 2014 , the percentage had reached 82 %) , and PHP was the most @-@ used open source software within enterprises . Web content management systems written in PHP include MediaWiki , Joomla , eZ Publish , eZ Platform , SilverStripe , WordPress , Drupal , Moodle , the user @-@ facing portion of Facebook , and Digg .
For specific and more advanced usage scenarios , PHP offers a well defined and documented way for writing custom extensions in C or C + + . Besides extending the language itself in form of additional libraries , extensions are providing a way for improving execution speed where it is critical and there is room for improvements by using a true compiled language . PHP also offers well defined ways for embedding itself into other software projects . That way PHP can be easily used as an internal scripting language for another project , also providing tight interfacing with the project 's specific internal data structures .
PHP received mixed reviews due to lacking support for multithreading at the core language level , though using threads is made possible by the " pthreads " PECL extension .
As of January 2013 , PHP was used in more than 240 million websites (39 % of those sampled) and was installed on 2 @.@ 1 million web servers .

= = Security = =

In 2013 , 9 % of all vulnerabilities listed by the National Vulnerability Database were linked to PHP ; historically , about 30 % of all vulnerabilities listed since 1996 in this database are linked to PHP . Technical security flaws of the language itself or of its core libraries are not frequent (22 in 2009 , about 1 % of the total although PHP applies to about 20 % of programs listed) . Recognizing that programmers make mistakes , some languages include taint checking to automatically detect the lack of input validation which induces many issues . Such a feature is being developed for PHP , but its inclusion into a release has been rejected several times in the past .
There are advanced protection patches such as Suhosin and Hardening @-@ Patch , especially designed for web hosting environments .
There are certain language features and configuration parameters (primarily the default values for such runtime settings) that make PHP applications prone to security issues . Among these , magic _ quotes _ gpc and register _ globals configuration directives are the best known ; the latter made any URL parameters become PHP variables , opening a path for serious security vulnerabilities by allowing an attacker to set the value of any uninitialized global variable and interfere with the execution of a PHP script . Support for " magic quotes " and " register globals " has been deprecated as of PHP 5 @.@ 3 @.@ 0 , and removed as of PHP 5 @.@ 4 @.@ 0 .
Another example for the runtime settings vulnerability comes from failing to disable PHP execution (via engine configuration directive) for the directory where uploaded images are stored ; leaving the default settings can result in execution of malicious PHP code embedded within the uploaded images . Also , leaving enabled the dynamic loading of PHP extensions (via enable _ dl configuration directive) in a shared web hosting environment can lead to security issues .
Also , implied type conversions that result in incompatible values being treated as identical against

the programmer 's intent can lead to security issues . For example , the result of the comparison 0e1234 = = 0 comparison is true because the first compared value is treated as scientific notation having the value (0 × 101234) , i.e. zero . This feature resulted in authentication vulnerabilities in Simple Machines Forum , Typo3 and phpBB when MD5 password hashes were compared . Instead , either the function strcmp or the identity operator (= = =) should be used ; 0e1234 = = = 0 results in false .

In a 2013 analysis of over 170 @,@ 000 website defacements , published by Zone @-@ H , the most frequently (53 %) used technique was exploitation of file inclusion vulnerability , mostly related to insecure usage of the PHP functions include , require , and allow _ url _ fopen .