



I'm not robot



Continue

P h p ka full form

This article is about script language. For other purposes, see:PHP (disambiguation). Server side scripts language, originally created in 1994 phParadigmImperative, functional, object-oriented, procedural, reflectiveDesigned Rasmus LerdorfDeveloperThe PHP development team, zend TechnologiesFiere appeared1995; 25 years ago (1995) 22 days ago (October 29, 2020) Preview Release 8.0.0 Issue Candidate 4 3 / November 12, 2020; 8 days ago (2020-11-12) Entry disciplineDynamic, weak with version 7.0: Gradual implementation languageC (primarily; some C)OSUnix-like, WindowsLicense PHP License (most of the zend engine license) Filename extensions.php, .phtml, .php3, .php4, .php5, .php7, .phps, .php-s, .pht, .pharWebsitewww.php.netMajor,HHVM, Phalanger, Kwerk, ParrotInfluenced byPerl, HTML, C, C, Java, Tcl, it was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. PhP is currently being implemented by the PHP group. PHP originally advocated for personal Home Page, but now it means PHP recursive initialism: Hypertext Preprocessor. PhP code is usually processed on a web server by a PHP translator implemented as a module, daemon, or as a shared gateway interface (CGI) that runs. On a web server, the result of interpreted and executed PHP code, which can be any type of data, such as HTML generated or binary image data, will form the entire or part of the HTTP response. There are various web template systems, web content management systems, and web platforms that can be used to organize or facilitate the generation of this response. In addition, PHP can be used for many programming tasks outside the web context, such as standalone graphics applications and robotic drone management. The arbitrary phP code can also be interpreted and executed through the command-line interface (CLI). The standard PHP translator, which works on the zend engine, is free software produced under the PHP license. PHP has been extensively ported and can be deployed on most web servers on almost every operating system and platform, free of charge. The PHP language developed without a written formal specification or standard until 2014, with the original implementation acting as a de facto standard that other implementations sought to follow. Since 2014, we have been working on the creation of the official PHP specification. By November 2020, when PHP 8 will be published at the end of the month, more than 40% are still on the outdated PHP 5; two out of every three websites using PHP are still on discontinued versions of PHP, and nearly half of all PHP websites use the version 5.6 or older that even Nebian supports (while Debian 9 still supports version 7.0 and 7.1, these versions PHP). In addition, PHP version 7.2, the most popular supported version of PHP, will no longer receive security updates on November 30, 2020, and therefore, if PHP websites are not updated to version 7.3 (or newer), 84% of PHP websites will use discontinued versions. The story is an early story of Rasmus Lerdorf, who wrote the original General Gateway Interface (CGI) component, along with Andy Gutmans and Seeev Suraski, who rewrote the parser that formed PHP 3. The development of PHP began in 1994, when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He expanded them to work with web forms and to communicate with databases, and called this implementation a personal homepage/translator forms or PHP/FI. PHP/FI can be used to create simple, dynamic web applications. To speed up error reporting and improve code, Lerdorf initially announced the release of PHP/FI as Personal Home Page Tools (PHP Tools) version 1.0 on the Discussion Group Usenet comp.infosystems.www.authoring.cgi June 8, 1995. This release already had the basic functionality that PHP has today. This included Perl-like variables, form processing, and the ability to embed HTML. Syntax resembled Perl's syntax, but was simpler, more limited, and less consistent. Example of phP's early syntax: !-include /text/header.html!-getenv HTTP_USER_AGENT!-if substr \$execx/Mozilla!-hey; Hey, you're using Net !-scape! !-sql database select from the table where the user\$username!-!ifess !-\$user \$numentries 1!-;g; Unfortunately, this record !- does not exist. You have !-!-index credits left in your account.html !-;. I have no idea how to write a programming language, I just kept adding the next logical step along the way. The development team began to form, and after several months of work and beta testing, PHP/FI 2 was officially released in November 1997. The fact that PHP was not originally developed and was instead developed organically has resulted in inconsistent function naming and inconsistent ordering of their parameters. In some cases, feature names were chosen as lower-level libraries that HADD wrap, while in some very early versions of PHP the length of function names was used internally as a hash function, so names were chosen to improve the distribution of hash values. PHP 3 and 4 This is an example of a custom WordPress Content Management System. Seev Suraski and Andy Gutmans rewrote the parser in 1997 and formed the basis of PHP 3, changing the language name to the recursive acronym PHP: Hypertext Hypertext Public testing of PHP 3 followed and the official launch took place in June 1998. Suraski and Gutmans then began a new PHP kernel re-recording, producing the engine of the zend in 1999. They also founded the company in Ramat Gan, Israel. On May 22, 2000, PHP 4, powered by engine 1.0, was released. By August 2008, this branch had reached version 4.4.9. PHP 4 is no longer under development and no security updates will be released. On July 14, 2004, php 5 was launched on the basis of the new zend II engine.PHP 5 included new features such as improved support for object-oriented programming, the expansion of PHP Data Objects (PDO), which identifies a lightweight and consistent interface for access to databases), and numerous performance improvements. In 2008, PHP 5 became the only stable version under development. The late static binding was missing from PHP and was added to version 5.3. Many high-profile open source projects have stopped supporting PHP 4 in the new code since February 5, 2008, due to the GoPPHVS initiative provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5. Over time, PHP translators became available on most existing 32-bit and 64-bit operating systems, either by creating them from PHP source code or by pre-constructed binary files. For PHP 5.3 and 5.4 versions, Microsoft Windows' only available binary distributions were 32-bit IA-32 builds requiring 32-bit Windows compatibility mode when using Internet Information Services (IIS) on the 64-bit Windows platform. PhP version 5.5 made 64-bit builds of x86-64 available for Microsoft Windows. Official support for PHP 5.6 security ended on December 31, 2018. PHP 6 and Unicode PHP received mixed reviews due to the lack of support for the native Unicode at the main language level. In 2005, a project was launched, led by Andrei Smievsky, to attract support from his native Unicode engine throughout PHP, by embedding international components for the Unicode Library (ICU) and presenting text lines as UTF-16 within the company. Since this would cause major changes in both the inside of the language and the user code, it was planned to release this as a version of 6.0 language, along with other major features that were then in development. However, the lack of developers who understood the necessary changes, and performance issues associated with conversion in and out of UTF-16, which are rarely used in a web context, led to delays in the project. As a result, PHP 5.3 was released in 2009, and many features, notably unicode, were re-sponsored from PHP 6, particularly from named spaces. In March 2010, the project in its current form was officially abandoned and the release of PHP 5.4 was prepared, containing the majority of the non-Unicode features php 6, such as traits and closing re-binding. Initial hopes were that a new Unicode integration plan would be formed, but by 2014 none of them had been adopted. PHP 7 During 2014 and a major new version of PHP was developed, which was wareed by PHP 7. The number of this version involved some debate between internal developers. While the PHP 6 Unicode experiment was never released, several articles and book titles referred to the name PHP 6, which could have been confusing if the new release had reused the name. After the vote, the name PHP 7 was chosen. The basis of PHP 7 is the PHP branch, which was originally called the next-generation PHP (phpng). He was the author of Dmitri Stogov, Xinchun-Hui and Nikita Popov, and was aimed at optimizing the performance of PHP by refactoring the zend engine, while maintaining almost complete language compatibility. By July 14, 2014, WordPress-based benchmarks, which served as the main benchmark package for the phpng project, showed almost 100% productivity growth. Changes to phpng are also expected to simplify performance in the future, as more compact data structures and other changes are seen as the most appropriate for successful JIT migration on time (JIT). Due to significant changes, the zend Engine redesign is called the zend Engine 3, replacing the zend Engine 2 used in PHP 5. Due to major internal changes in phpng, it should receive a new core number of PHP versions, rather than a minor PHP 5 release, according to the PHP release process. Major versions of PHP may violate code inverse compatibility, so PHP 7 provided an opportunity for other improvements beyond phpng that require backward compatibility breaks. In particular, we are talking about the following changes: many outdated mechanisms of PHP fatal or regenerative errors have been replaced by modern object-oriented exceptions. The syntax for alternating dereferenkgia has been redesigned to be internally more consistent and complete, allowing the use of operators - and :: with arbitrary meaningful left expressions. Support for the outdated methods of the PHP 4 has been understandable. The behavior of the foreach statement has been changed to be more predictable. Designers for multiple classes built into PHP that returned void after the failure were modified to cast the exception instead, for consistency. Several unsullied or outdated server application programming interfaces (SAPIs) and extensions have been removed from the PHP kernel, primarily from the outdated mysql extension. The behavior of the list operator has been changed to remove support for the lines. Support has been removed for out-of-date ASP-style deimilyters, and . 62 A misstep has been corrected allowing the switch statement to have multiple default positions. Support for six-family numbers support in some implicit conversions from rows to room types has been removed. Operators of left and right shifts have been modified to lead more consistently on different platforms. Conversions have changed between floatingo the numbers and integrators (e.g. (e.g. move to zero) and implemented more consistently on different platforms. The PHP 7 also included new language functions. Specifically, it introduces return-type declarations for functions that complement existing declarations of parameter type, and support for scalar types (integer, float, string, and boolean) in declarations such as parameters and return. PhP 8 PHP 8 is scheduled for release on November 26, 2020. PHP 8 is currently in beta. PhP 8 is the main version and has the latest changes in relation to previous versions. New features and notable changes include: Just-in-time compilation Just-in-time is supported in PHP 8. PhP 8's JIT can provide significant performance improvements in some cases of use. PHP developer Nikita Popov stated that improved performance for most websites would be less significant than an upgrade from PHP 5 to PHP 7. Performance improvements with the addition of the JIT compiler are expected to have more significant performance improvements in mathematical operations than in normal web development cases. Adding a statement about the match Home article: PHP syntax and semantics - Match PHP 8 submitted a statement about the match. The match statement is conceptually similar to a switch statement and is more compact for some usage cases. The type of changes and additions of PHP 8 introduced alliance types, a new static type of return and a new mixed type. Attributes, often referred to as abstracts in other programming languages, have been added to PHP 8 to add metadata to classes. The cast was changed from statement to expression. This allows you to make exceptions in places that were previously impossible. Syntax changes and PHP 8 add-ons include changes that allow alternative, more concise, or more consistent syntaxes in a number of scenarios. For example, a nullsafe operator is similar to a zero-phone operator, but ??, used when calling methods. The next piece of code won't throw a bug if getBirDay returns zero: \$human readable date \$user!-!ng()?-----sort of analysis for people (); The development of the designer property has been added as syntax sugar, allowing type properties to be established automatically when the parameters are transferred to the style designer. This reduces the amount of boiler code that needs to be written. Other minor changes include support for the use of :class on objects, which serves as an alternative to using get_class (); Don't capture catches in try-catch blocks; Variable syntax settings to eliminate inconsistencies Support for these arguments; and support commas in the settings lists, which adds consistency to support the rear commas in other contexts, such as arrays. Standard library changes and Weak Card add-ons have been added to PHP 8. WeakMap links to objects, but these links do not prevent the debris of such objects This can improve performance in scenarios where data is cached; this is of particular importance to the OUM. Various interface adjustments, such as adding support to create DateTime objects from interfaces, and adding a Stringable interface that can be used for type hints. Various new features, str_contains, str_starts_with and str_ends_with FdIv (); get_debug_type (); and get_resource_id () Implementation Object token_get_all () Additional changes to the type of annotation have also been added to the source code of PHP C itself to allow internal functions and methods to have complete type information in reflection. Inheritance with Private Methods Abstract Methods in the features of improving the history of the release date release Supported up to 16 Notes Old version, no longer supported: 1.0 June 8, 1995 Officially called Personal Home Page Tools (PHP Tools). This is the first use of the PHP name. The old version is no longer supported: 2.0 on November 1, 1997 is officially called PHP/FI 2.0. This is the first release that can actually be described as PHP, being a standalone language with many features that have survived to this day. The old version is no longer supported: 3.0 June 6, 1998 October 20, 2000. Seev Suraski and Andy Gutmans rewrite the base for this version. The old version is no longer supported: 4.0 on May 22, 2000, on June 23, 2001. The old version is no longer supported: 4.1 10 December 2001 (\$_GET, \$_POST, \$_SESSION, etc.) Old version, no longer supported: 4.2 22 April 2002 (85) 6 September 2002 register_globals. Network data is no longer inserted directly into the global namespace, closing potential security holes in applications. The old version is no longer supported: 4.3 27 December 2002 (86) 31 March 2005 (CL), in addition to CGI. The old version is no longer supported: 4.4 July 11, 2005, August 7, 2008. compiled for the PHP version 4.3.x., the old version is no longer supported: 5.0 July 13, 2004 September 5, 2005 Old version, no longer supported: 5.1 24 November 2005 (92) August 24, 2006 Performance improvements with the introduction of variable complementary in the redesigned engine. Added php Data Objects (PDO) objects as a sequential interface for accessing databases. The old version is no longer supported: 5.2 2 November 2006 (94) January 6, 2011 Support for native JSON. The old version is no longer supported: 5.3 June 30, 2009, August 14, 2014 Late static bindings, jump label (limited goto), features, closures, PHP archives (phar), garbage collection for circular links, improved support for Windows, sqlgite3, mysql64 as a substitute for libmysql as the main library for extensions that work with MyS'L, fileinfo as a replacement for mime_magic for better support for MIME, expansion of internationalization, and wear ereg extensions. The old version is no longer supported: 5.4 March 1, 2012 (96) September 3, 2015. Items were seized: register_globals, safe_mode, allow_call_time_pass_reference, session_register, session_unregister() and session_is_registered. Built-in web server. A few improvements to existing features, performance, and lower memory requirements. The old version, no longer supported: 5.5 20 June 2013 (98) 10 July 2016' Support generators, finally blocks to handle exceptions, OpCache (based on the zend optimizer) bundled in the official distribution. Old version is no longer supported: 5.6 August 2014 December 31, 2018 Permanent large-scale expressions, variations, unpacking arguments, new exposure operator, extension of the function and constant operator, new phpdbgback as a SAPI module and other small improvements. 6.x Did not release N/A Abandoned version of PHP, which planned to include native support for Unicode. The old version is no longer supported: 7.0 3 December 2015 (105) 3 December 2018 on Windows, uniform variable syntax, (AST-compilation process) added Closure::call(),108 bit of coherence shift between platforms, ????? (null merge) operator, 110 Unicode code point avoid syntax, strings and boolean declaration, 68 zlt; spaceship three-way comparison operator, 112 generator delegation, 113 anonymous classes, 114 simpler and more consistently available CSPRNP APING, 115 replacing many remaining internal PHP errors with more modern exceptions, 55 and short-term syntax for multiple space names. Old version, no longer supported: 7.1 1 December 2016 1 December 2019 still supported: 7.2 30 November 2017 30 November 2020 (99) Object option and declaration of the type of return, Libsodium extension, Older version, but still supported: 7.3 December 6, 2018123 December 6, 2021 Flexible Heredoc and Nowdoc syntax, support for reference appointment and array deconstruction list (), PCRE2 support, feature 126 hrtime () Current stable version: 7.4 28 November 2019 2.0, Preload, 130 Null Coalescing Destination Operator,131 Improvement openssl_random_pseudo_bytes, 132 Weak References, Интерфейс Всегда доступное жэш-расширение, (

..... Последняя предварительная версия будущего релиза: 8.0 26 ноября 2020 г. 26 ноября 2023 г. Just In Time compilation (JIT), массивы, начиная с отрицательного индекса, Для абстрактных методов черты, строка Saner к сравнениям номера, »143» Saner Numeric Strings, »144» ТурЕггор на неожидательных арифметических/битумных операторах, Реклассификация различных ошибок двигателя, Фатальная ошибка для несовместимых подписей метода (148)), независимого полпака Locale к преобразованию строки, »149» Переменные синтаксисные твксы, Выражение сплички, »156» Продвижение свойства конструктора, »157» типы профсоюзa, »158» смешанный тип, »159» статический тип возращения, »160» Nullsafe оператор, »78» поп-захватывая уловы, »161» выражение броска, расширение JSON всегда доступно по-прежнему поддерживаетсяЛатест версияВестная предварительная версияFuture релиз Начиная с 28 июня 2011 года, PHP Команда разработчиков реализовала сроки выпуска новых версий PHP. Under this system, at least one release must occur every month. Once a year there should be a minor release, which may include new features. Each minor release must be at least maintained for two years with security and bug fixes, followed by at least one year of security fixes only, with a total three-year release process for each minor release. No new features, unless small and standalone, will be introduced to minor release during the three-year release process. The mascot of the elePHPant, the mascot of the PHP Project PHP is the elePHPant, a blue elephant with the PHP logo on its side, developed by Vincent Pontier in 1998. The letters (PHPs) formed the shape of an elephant when viewed from a side angle. ElepHPANT is sometimes painted differently when in a plush toy shape. Many variations of this mascot have been made over the years. Only elePHPants based on the original design of Vincent Pontier are considered the official community. They are collectible, and some of them are extremely rare. Syntax A Hello World app in PHP 7.4 runs on the local development server Home article: Syntax PHP and Semantics Next Program Hello World! written in the PHP code built into the HTML document: DOCTYPE However, since there is no requirement for a PHP code to be built into HTML, the simplest version of Hello, World! can be written like this, with the final tag omitted as preferred in files containing the pure code PHP qlt;?php echo "Hello, World!"; echo="hello,= world!"; =?> world!"; =?> world!; The PHP translator only performs php code in its de-mark. Anything beyond its delimiters is not processed by PHP, although non-PHP text is still subject to the management structures described in the PHP code. The most common delimiters are the zlt;?php to open and ? qgt; to close sections of php. Shortened form? there is also. This short delimiter makes script files less portable because support for them can be disabled in a local PHP configuration, and therefore it is not recommended; There are no recommendations against the echo of the short tag zlt.com.au.com. Before PHP 5.4.0, this short echo syntax only works with the short_open_tag configuration enabled, while for PHP 5.4.0 and later it's always available. The goal of all of these de-llis is to separate the PHP from non-PHP content, such as JavaScript or HTML markups. The first form of delimiters, the xhtml and other XML documents, creates well-formed XML processing instructions. This means that the resulting mixture of PHP code and other markup in the server file itself is well-formed XML. Variables are branded a dollar symbol in advance, and the type should not be specified in advance. PHP 5 has introduced model declarations that allow features to force their settings to be objects of a certain class, array, interface or callback functions. However, prior to PHP 7.0, type ads could not be used using scalar types such as integer or string. Unlike function and class names, variable names are sensitive to cases. Both double-quoted lines (and the heredoc provide the ability to interpolate the value of the variable into a string. PHP sees the new lines as a white space in a free-form language, and applications are terminated by a comma. PHP has three types of comment syntax: / block marks and inline comments; or q used for one line of comments. The Echo statement is one of several PHP objects providing text output. In terms of keywords and language syntax, PHP is similar to C-style syntax if conditions, cycles, and function returns are similar in syntax to languages such as C, C, C, Java, and Perl. PHP data types are loosely connected because it doesn't depend on the type of data. This is one of the main features of the language. It keeps entire rows within a platform-dependent range, either as 32, 64 or 128-bit signed a whole series equivalent to a long type of C-language. Unsigned integrators are converted into signed values in certain situations, which is different from the behavior of many other programming languages. Integer variables can be assigned using decimals (positive and negative), octal, sixty and binary notations. Floatingo the numbers are also stored in a platform-specific range. They are indicated by a floating notation point, or two forms of scientific notation. PHP has a native Type of Boolean that is similar to native Boolean types in Java Java C++. Using Boolean-type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C. The zero data type is a variable that doesn't matter, NULL is the only available value for this type of data. Resource-type variables represent links to resources from external sources. They are usually created by functions from a specific extension and can only be processed by functions from the same extension; examples include resources for files, images, and databases. Arrays can contain any type of element that PHP can handle, including resources, objects, and even other arrays. Order is stored in value lists and hashes with keys and values, and these two values can be intertwined. PHP also supports strings that can be used with single quotes, double quotes, nowdoc or heredoc syntax. The PHP Library (SPL) tries to solve standard problems and implements effective interfaces and data access classes. PHP features define a large array of features in the main language, and many of them are also available in various extensions; these features are well documented in PHP's online documentation. However, the built-in library has a wide range of naming conventions and related inconsistencies, as described in the story above. Custom features can be determined by the developer: myAge (int \$birthYear): line / calculate age by subtracting the year of birth from the current year. \$yearsOld - date ("Y") - \$birthYear; bring back age in the narrative \$yearsOld line. 'year'. (\$yearsOld - 1? 's:'); Echo I am now. myAge (1995) . 'old.'; In 2020, the exit from the aforementioned exemplary program will be I am now 25 years old. Instead of function pointers, functions in PHP can refer to a line containing their name. In this way, conventional PHP functions can be used, for example, as callbacks or function tables. User-defined features can be created at any time without a prototype. Functions can be identified within code blocks, allowing you to decide whether to identify a function. There is a function_exists function that determines whether a feature with a certain name has already been defined. Functional calls should use brackets, with the exception of a zero-argument class constructor called the new PHP operator, in which case brackets are optional. Prior to PHP 5.3, there was no support for anonymous functions and closures in PHP. Although create_function exists with PHP 4.0.1, it's just a thin wrapper around eval, allowing you to create normal PHP functions while running the program. PHP 5.3 added a syntax to determine an anonymous function or closure that can capture variables from the surrounding sphere: getAdder (\$adder \$x \$y \$x) echo \$adder (2); prints 10 B Above, the getAdder function creates a closure using the traversed argument \$x (the keyword uses the import variable from the lexical context) that takes an additional argument \$y, and returns the closing created to the caller. This function is a first-class object, which means that it can be stored in a variable, passed as a parameter to other functions, etc. Unusually for dynamically on language types, PHP supports typical declarations on function parameters that are applied during execution. This was supported for classes and interfaces with PHP 5.0, for arrays with PHP 5.1., for calls with PHP 5.4, and scalar (integrator, float, string and boolean) types with PHP 7.0. PHP 7.0 also has type declarations for function return types, expressed by placing a type name after a list of parameters preceding the colon. For example, the getAdder feature from an earlier example can be annotated in types like this in PHP 7: getAdder (int \$x): Closing and Return function (int \$y) use (\$x): int - return \$x and \$y \$adder; Echoes \$adder (2); 10 Echo \$adder (zero); throws an exception because the wrong type was \$adder and getAdder (me); will also throw an exception by default, scalar-type declarations follow weak input principles. For example, if the int type of option, PHP will allow not only integers, but also convertible numerical strings, floats or booleans to be transferred to this function, and will convert them. However, PHP 7 has a strict typing mode that prevents you from using these conversions for feature calls and returns in the file. PhP Objects Basic object-oriented programming features have been added to PHP 3 and improved in PHP 4. This allowed PHP to gain further abstraction, making it easier for programmers using the language. Object processing has been completely rewritten for PHP 5, expanding the feature set and improving performance. In previous versions of PHP, objects were treated as value types. The downside of this method was that the code had to use PHP reference variables intensively if it wanted to change the object that had been passed, rather than create a copy of it. In the new approach, objects are referenced by pen, not cost. PHP 5 introduced private and protected variables and participants' methods, as well as abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring designers and destructors similar to other object-oriented languages such as the NHS and a standard model for processing exceptions. In addition, PHP 5 added interfaces and allowed multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the run time system. Objects selling ArrayAccess can be used with array syntax, and objects selling Iterator or can be used using the foreach language design. The virtual function of the table in engine, so static variables are associated with a name instead of a reference during compilation. If the developer creates a copy of the object using a reserved word clone, the zend engine will check whether the __clone method has been determined. In this case, it will name __clone that will copy the properties of the object. If __clone method, it will be responsible for installing the necessary properties in the created object. For convenience, the engine will deliver a function that imports the properties of the source object, so the programmer can start with the cost of a replica of the source object and override only the properties that need to be changed. Below is the main example of object-oriented programming in PHP: 1. qlt;?php 2.3 abstract class User 4 and 5 public \$name; 6 public feature __construct (\$name) 7 No 8 \$this->gt;name - \$name; 9 - 10 public greeting features (): Line 11 No. 12 Return Hello, my name is \$this->gt;name; 13 No. 14 abstract public work functions (): /; 15 No. 16 Class Student Expands User 17 No. 18 Public \$course; 19 public functions __construct (\$name, \$course) 20 No. 21 \$this->gt; course No. \$course; 22 parents: __construct (\$name); 23 No. 24 Public Works Job () : Line 25 No. 26 Return I Learn \$this the course; 27 No. 28 No. 29 Class Teacher Expands User 30 No. 31 Public \$teachingCourses; 32 33 public functions __construct (\$name, \$teachingCourses) 34 No. 35 \$this->gt; costumes and \$teachingCourses; 36 parents: __construct (\$name); 37 - 38 39 public functions (): Line 40 No. 41 return I teach . Exploding (\$this->gt;compaers); 42 No 43 No 44 \$students No 45 and 46 new students (Alice, Computer Science), 47 new students (Bob, Computer Science), 48 new students (Charlie, Business Research), 49 euros; 50 \$teachers - \$1 - 52 new Teachers (Dan, Computer Science, Information Security), 53 New Teachers (Erin, Computer Science, 3D Graphic Programming), 54 New Teachers (Frankie, Internet Marketing, Business Research, E-Commerce), 55 euros; 56 57 Echo Students: ; 58 foreach (\$students as \$student) 59 and 60 echoes \$student->gt;greet() . . \$student->gt;job()) ; 61 No 62 63 Echo Teachers: ; 64 foreach (\$teachers as \$teacher) 65 and 66 echoes \$teacher->gt;greet() . . \$teacher. ; 67 No 68 69 / Exit program: 70 / Students: 71 / Hello, my name is Alice, I learn computer science 72 / Hello, my name is Bob, I learn computer science 73 / Hello, my name is Charlie, I am studying business research 74 / Teachers: 75 / Hello, My name is Dan, I teach computer science, information security 76 / Hello, my name is Erin, I teach computer science, 3D Graphics Programming 77 / Hello, my name is Frankie, I teach internet marketing, business research, e-commerce 78 79 ?gt; private and secure. The default is public if only var is used: var is synonymous with the public. Items declared public can be accessed everywhere, protected restrictions on access to inherited inheritance (and the class, defining the element), private visibility restrictions only for the class that defines the item. Objects of the same type have access to each other's personal and secure members, even if they are not the same instance. Implementing the only complete implementation of PHP is the original, known simply as PHP. It is the most widely used and is powered by the zend engine. To disassociate itself from other implementations, it is sometimes informally referred to as the PHP zend. The zend engine compiles the PHP source code on the fly into an internal format that it can perform, so it works as an interpreter. It is also a reference implementation of PHP, as PHP does not have a formal specification, and therefore the semantics of the PHP zend defines PHP semantics. Because of the complex and subtle semantics of PHP, defined by how the zend works, it is difficult for competing implementations to offer full compatibility. The PHP model with one request for a performance script, as well as the fact that 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 running time and not have to compile the PHP source code every time a web page is accessed, PHP scripts can also be deployed in the internal PHP engine format using the opcode cache, which works by caching the compiled form of phP script (opcodes) into total memory to avoid overhead parsing and compiling the code every time the script is running. The opcode cache, zend Opcache, is built into the PHP with version 5.5. Another example of a widely used opcode cache is the alternative PHP Cache (APC), which is available as a PECL extension. Although phP is still the most popular implementation, several other implementations have been developed. Some of them are compilers or support the JIT compilation, and therefore offer performance benefits under the PHP zend due to the lack of full PHP compatibility. Alternative implementations include: HHVM (HipHop Virtual Machine) - developed on Facebook and available as an open source, it converts the PHP code into a high-level integrator (commonly known as intermediate language), which is then translated into machine code x86-64 dynamically while running only (JIT) compiler, resulting in up to 6x performance improvements. However, since the 7.2 zend version surpassed HHVM, and HHVM 3.24 is the latest version officially supported by PHP. Parrot is a virtual machine designed to effectively manage dynamic languages; Pipp converts the PHP source code into an intermediate Parrot view, which is then translated into a parrot interpreted and run by a virtual machine. PeachPie is a second-generation compiler for bytecode .NET Common Intermediate Language (CIL) built on the Roslyn platform; Phalanger, dividing several architectural components of Phalanger - compiles PHP into Common Intermediate (CIL) (CIL) PeachPie's predecessor, Kwerk, compiles PHP into Java-tecode HipHop, developed on Facebook and available as an open source, converts PHP scripts into C code and then compiles the resulting code, reducing server load to 50%. In early 2013, Facebook withered it in favor of HHVM for several reasons, including deploying difficulties and lack of support for the entire PHP language, including create_function and eval designs. Main Licensing Article: PHP License PHP is free software released under a PHP license, which provides that: products derived from this software cannot be called PHP, nor can PHP be displayed in their name without prior written permission from group@php.net. You can point out that your software works in conjunction with PHP by saying: Foo for PHP, rather than calling it PHP Foo or phfoo. This restriction on the use of PHP renders the PHP license incompatible with the General Public License (GPL), while the zend license is incompatible due to an advertising provision similar to the original BSD license. THE development and community of PHP includes various free open source libraries in its distribution of raw data, or uses it as a result of PHP binary builds. PHP is essentially an Internet-aware system with built-in modules to access file Protocol (FTP) servers and many database servers, including PostgreS'L, MyS'L, Microsoft S'L Server and S'Lite (which is a built-in database), LDAP servers, and others. Numerous features 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 statically compiled in PHP or downloaded dynamically during execution. Numerous extensions have been written to add support for Windows API, process management in operating systems similar to Unix, multi-byte strings (Unicode), cURL, and several popular compression formats. Other PHP features available through extensions include integration with IRC, dynamic imaging generation and Adobe Flash content, PHP Data Objects (PDO) as an abstract layer used to access databases, 203 and even speech synthesis. Some basic language functions, such as strings and array functions, are also implemented as extensions. The PhP Extension Community Library (PECL) project is a repository for expanding the PHP language. Some other projects, such as Marshmallow, provide the opportunity to create high-level PHP extensions and compile in native PHP extensions. This approach, instead of writing PHP extensions directly into C, makes it easier to develop extensions and reduces the time it takes to program and test. By December 2018 PHP group consisted of ten people: Tees K. Arnzen, Stig Bakken, Shane Karaveo, Andy Gutmans, Rasmus Lerdorf, Sam Ruby, Sasha Sasha Seev Suraski, Jim Winstead and Andrei Smievsky. The technology provides PHP certification based on the PHP 7 (and previously based on PHP 5.5) for programmers to become PHP certified developers. Installation and Configuration Sample Phpinfo () in PHP 7.1. There are two main ways to add PHP support to a web server, whether as a home web server or as a CGI. 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 IPPlanet. Some other web servers, such as OmniHTTDP, support the Internet Server Application Programming Interface (ISAPI), which is the interface of Microsoft's web server module. If PHP does not have modular web server support, it can always be used as a common gateway interface (CGI) or FastCGI processor; in this case, the web server is configured to use CGI PHP to handle all requests for PHP files. PHP-FPM (FastCGI Process Manager) is an alternative implementation of FastCGI for PHP, complete with the official PHP distribution from version 5.3.3. Compared to the old FastCGI implementation, it contains some additional features, mostly useful for heavily loaded web servers. If you use PHP for command line scenarios, you'll need to run a PHP (CLI) command-line interface. PHP supports the CLI (SAPI) application programming interface with PHP 4.3.0. Sapi focuses on developing fake applications using PHP, there is quite a lot of difference between CLI SAPI and other SAPIs, although they have a lot of the same behavior. PHP has a direct module interface called SAPI for various web servers; in the case of PHP 5 and Apache 2.0 on Windows, it is provided as a DLL file called php5apache2.dll, which is a module that, among other functions, provides an interface between PHP and a web server implemented in a form that the server understands. This form is so known as SAPI. There are different types of SAP For different web server extensions. For example, in addition to the above, other PHP SAPIs include the Common Gateway Interface (CGI) and the Command Line Interface (CLI). PHP can also be used to write desktop GUI applications using the PHP-GTK extension. PHP-GTK is not part of the official PHP distribution, and can only be used as an extension with PHP 5.1.0 and newer versions. The most common way to install PHP-GTK is to compile from the source code. When PHP is installed and used in cloud environments, software development kits (SDKs) are available for cloud-based features. For example: Amazon Web Services AWS SDK for PHP, Windows Azure can be used with Windows Azure SDK for PHP. Numerous configuration, configuration, and configuration settings are supported both the main functions of PHP and expansion. The configuration of the php file.ini is searched in different locations, depending on how PHP is used. The configuration file is divided into different sections, while some configuration settings can also be installed in the web server configuration. Use a broad overview of the LAMP software package displayed here with Squid PHP, a general-purpose script language that is particularly suitable for web development on the server side, in which case PHP usually works on a web server. Any PHP code in the requested file is run at the time of PHP, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for team scripts and GUI from the client side. 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 customers. It is available for free, and PHP Group provides full source code for users to build, customize and expand for their own use. Dynamic Web page: PhP and MyS'L example of a server script acts primarily as a filter, citation is necessary by taking input from a file or thread containing text and/or phP instructions and output of another data stream. Most often the output will be HTML, although it can be JSON, XML or binary data such as images or audio formats. Starting with PHP 4, phP parser compiles input to obtain card code for processing using zend Engine, which gives you better performance than its predecessor translator. Originally designed to create dynamic web pages, PHP now focuses mainly on scripts on the server side, and it is similar to other script languages on the server side that provide dynamic content from a client's web server, such as

Microsoft ASP.NET, JavaServer Pages, Sunsystem Micros, and mod_perl. PHP has also attracted the development of many software frameworks that provide building blocks and design structure to promote rapid application development (RAD). Some of these include PRADO, CakePHP, Symfony, Codelgnr, Laravel, Yii Framework, Phalcon and zend Framework, offering features similar to other web structures. LAMP architecture has become popular in the web industry as a way to deploy web applications. PHP is commonly used as a P in this package along with Linux, Apache and MySQL, although P may also refer to Python, Perl, or some mix of three. Similar packages, WAMP and MAMP, are also available for Windows and macOS, with the first letter behind the corresponding operating system. Although both PHP and provided as part of the basic macOS installation, users of these packages are looking for a simpler installation mechanism that can be more easily up to date. For specific and more advanced usage scenarios, PHP offers offers in addition to expanding the language itself in the form of additional libraries, extensions provide a way to increase the speed of execution where it is critical, and there is room for improvement with true compiled language. PHP also offers well-defined ways of embedding itself in other software projects. In this way, PHP can be easily used as the internal language of scenarios for another project, as well as providing close interaction with specific internal structures of project data. PHP has received mixed reviews due to a lack of support for multi-read at the main language level, although the use of threads is possible due to the expansion of PECL pthreads. The command-line interface, php-cli, and two ActiveX Windows Script Host scripts for PHP were created. Popularity and Usage Statistics As of April 2007, more than 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. Compared to August 2019, PHP was used as a programming language on the server side at 78.8% of websites, compared to 83.5% previously where language could be identified. Web content management systems written in PHP include MediaWiki, Joomla, 252 ePublish, ePlatform, SilverStripe, WordPress 253, 254 Drupal, 255 and Moodle. Websites written in PHP, in the back end and/or user part, include Facebook, Digg, As of January 2013, PHP was used on more than 240 million websites (39% selected) and was installed on 2.1 million web servers. As of April 2020, PHP 7 is the most used version of the language with 50.6% of sites using PHP and 39.6% of all websites on the Internet. Security In 2019, 11% of all vulnerabilities listed by the National Vulnerability Database were related to PHP; Historically, about 30% of all vulnerabilities listed in this database since 1996 are related to PHP. Technical security flaws of the language itself or its major libraries are not frequent (22 in 2009, about 1% of the total, although PHP refers to about 20% of the programs listed). Recognizing that programmers make mistakes, some languages include pollution checks to automatically detect a lack of input verification that causes a lot of problems. This feature is being developed for PHP, but its inclusion in the release has been rejected several times in the past. There are advanced protective patches, such as Suhosin and Hardening-Patch, specifically designed for web hosting. Historically, older versions of PHP had some configuration and defaults for such run time settings, making some PHP applications prone to security issues. Among them magic_quotes_gpc and 271 configuration directives; Configurations The latter made any URL settings a PHP variable, paving the way for serious security vulnerabilities, allowing the attacker to set the value of any single-niital global variable and prevent the PHP scenario from being implemented. Support for magic quotes and global registration has been removed from PHP 5.3.0 and removed from PHP 5.4.0. Another example of the potential vulnerability of run time settings is the inability to disable PHP execution (for example, using the engine configuration directive) for the directory where the downloaded files are stored; this can lead to malicious code embedded in the downloaded files. The best practice is either to find a catalog of images outside the root of the document available to the web server and serve it with an intermediate script, or to disable the PHP for the directory where the downloaded files are stored. In addition, incorporating dynamic PHP extension downloads (through an enable_dl configuration directive) into a shared web hosting environment can lead to security issues. The implied transformations of types that result in different values being viewed as equal, sometimes against the programmer's intention, can lead to security problems. For example, the '0e1234' '0' comparison is correct because the lines that are parsed as the numbers are converted to numbers; in this case, the first compared value is seen as a scientific notation of value (0x101234), which is zero. Errors like this led to authentication vulnerabilities in the Simple Machines Forum, Typo3281 and phpBB 282, when HSI MD5 passwords were compared. The recommended method is to use hash_equals () (for the security of attack time), strcmp or operator identification (yap.) since '0e1234' '0' leads to false. In 2013, more than 170,000 websites published by H-zone were most frequently published (53%) and the most common (53%) of websites published by Zone H. the technique used was to exploit the vulnerability of file inclusion, mainly related to the unsafe use of PHP language designs, including those requiring and allow_url_fopen. See. also Computer Programming Portal Free and Open Source Software Portal Comparison of Languages Programming Programming Comparison Web Infrastructures List of AMP Accelerators List of PHP Accelerators List of PHP Pear Editors (PHP Expansion and Application Repository) PHP Accelerator Pattern Processor XAMPP (Free and Open Source Cross-Platform Web Server Solution Package) data at the MySQL conference in Santa Clara California. A network of conversations. Received 2009-12-11. PHP 7 ChangeLog. Php-8.0.0RC4. November 14, 2020. Received on November 14, 2020. PHP: Arguments function - Guide. secure.php.net. - Stogov, Dmitry (2015-12-04). It's no secret that some of them. Twitter. Php: npenpoecpp. www.php.net. Получено 2020-02-12. a b История PHP. PHP. a b c d e f h i History of PHP and related projects. PHP Group. Received 2008-02-25. PHP Guide: Foreword, www.php.net. Introduction: What can PHP do?. PHP guide. Received 2009-03-05 : Helicopter: Port node-ar-drone, which allows the user to control the Parrot AR Drone over PHP: jolicode/php-ar-drone, JoliCode, 2019-01-11, extracted 2019-02-23 - b Embedded PHP in HTML. O'reilly, 2001-05-03. Received 2008-02-25. Jackson, Joab (2014-07-31). PHP gets the official specification at last. The computer world. Idg. a b PHP: Supported versions. www.php.net. received 2020-09-02. End of Life: A release that is no longer supported. Users of this release should update as soon as possible because they may be exposed to uncovered security vulnerabilities. PhP usage and market share for websites, September 2020. w3techs.com. received 2020-09-02. b c d e f g h i j k l m Unsupported branches. php.net. Received 2019-07-31. Lehrdorf, Rasmus (2012-07-20). I wonder why people keep writing that PHP was ever written in Perl. It never happened. #php. Twitter. Received 2014-09-04. Lehrdorf, Rasmus (2007-04-26). PHP on hormones (mp3). A network of conversations. Received 2009-06-22. Lerdorf, Rasmus (2007). Slide 3. slides for PHP on hormones talk. PHP Group. Received 2009-06-22. Lerdorf, Rasmus (June 8, 1995). Announce: Personal HomePage Tools (PHP Tools). Received on June 7, 2011. Lehrdorf, Rasmus (1995-06-08). Announce: Personal HomePage Tools (PHP Tools). News Group: comp.infosystems.www.authoring.cgi. Received 2006-09-17. Rasmus Lerdorf, Yahoo Senior Technical: PHP, behind the microphone. 2003-11-19. Archive from the original for 2013-07-28. Problems with PHP. Received on December 20, 2010. php.internals: Re: Function name consistency. news.php.net. 2013-12-28. Received 2014-02-09. Rasmus Lerdorf (December 16, 2013). Re: Flexible function name. News group: php.internals. Received on December 26, 2013. PHP - Acronym The Value of Voting. PHP.net. Archive from the original on August 15, 2000. The zend Engine version 2.0: Review features and design. zend Technologies Ltd. Archive from the original to 2006-07-19. Received 2006-09-17. Php.net, 2007. PHP Group. 2007-07-13. Received 2008-02-22. Kerner, Sean Michael (2008-02-01). PHP 4 dead- Long live PHP 5. InternetNews. Archive from the original for 2018-08-06. Received 2018-12-16. Trachtenberg, Adam (2004-07-15). Why PHP 5 Rocks!. O'reilly. Received 2008-02-22. Late static binding in PHP. A digital sandwich. 2006-02-23. Received 2008-03-25. A static keyword. PHP Group. Received 2008-03-25. GoPHP5. Archive from the original for 2011-07-17. GoPHP5. PHP projects join forces to go PHP 5 (PDF). GoPHP5 press release. Archive from the original (PDF) for 2009-12-10. Received 2008-02-23. GoPHP5. GoPHP5. Archive from the original 2011-04-27. Received 2008-02-22. PhP installation and configuration. Received 2013-10-29. PHP for Windows: Melons and Sources releases (5.3). php.net. Received 2013-10-29. PHP for Windows: Dunes and Release Sources (5.4). php.net. Received 2013-10-29. PHP for Windows: Melons and Sources releases (5.5). php.net. Received 2013-10-29. PHP: Supported versions. Types: Strings (PHP Guide). PHP.net. Received 2013-09-22. Line type details (pHP guide). PHP.net. Received 2013-09-22. Andrei Smieszewski (2005-08-10). PhP Unicode support document design (mailing list). Received 2014-02-09. PHP 5.5 or 6.0. Received 2014-02-09. Andrei Smiyevsky (2011-04-22). Good, Bad and Ugly: What Happened to Unicode and PHP 6. Received 2014-02-09. Rasmus Lerdorf (2010-03-11). PHP 6 (Mailing List). Received 2014-02-07. Endless debate Muppet PHP 6 v PHP 7. Archive from the original 2015-11-19. Received 2015-11-19. RFC: The name of the next PHP release. php.net. 2014-07-07. Received 2014-07-15. Re: (PHP-DEV) (GOLOS) Title of the next PHP release (again). 2014-07-30. Received 2014-07-30. phpng: A PHP engine with great performance improvements. news.php.net. - PHP: rfc:phpng. php.net. received on December 16, 2014. a b PHP: phpng. php.net. Received 2014-07-15. The merging of the branch is a 'endEngine3'. github.com. 2014-12-05. Received 2014-12-05. b c PHP: Release process. 2011-06-20. Received 2013-10-06. b PHP RFC: Engine exceptions (for PHP 7). php.net. Received 2015-05-21. b PHP RFC: Single variable syntax. php.net 2014-05-31. Received 2014-07-30. Online PHP shell (en) 150 versions - stats - vld and hhvm. PHP RFC: Fix the behavior of the foreach. php.net. Received 2015-05-21. PHP RFC: Designer behavior of internal classes. php.net. Received 2015-05-21. PHP RFC: Removal of dead or not yet PHP7 ported by SAPIs and extensions. php.net. Received 2015-05-21. PHP RFC: Fix the list () mismatch behavior. php.net. Received 2015-05-21. PHP RFC: Remove alternative PHP tags. php.net. Received 2015-05-21. PHP RFC: Make a default default in the switch a syntax error. php.net. Received 2015-05-21. PHP RFC: Remove hexagonal support in numerical rows. php.net. Received 2015-05-21. a b PHP RFC: The semantics of the integrator. php.net. Received 2015-05-21. Creating NaN and Infinity always becomes zero when discarded on integer means more cross-platform consistency, and is also less surprising than what PHP currently produces RFC: PP Denial for Overflow. php.net. Received 2015-05-21. b c RFC: Return types. php.net. 2015-01-27. Received 2015-01-28. - b c d e f RFC: Scalar Declarations. php.net. 2015-03-16. Received 2015-03-17. PHP: News Archive - 2020. October 1, 2020. PhP 8.0.0 Issue Candidate 1 is available for testing. Received on October 2, 2020. - b c d e f g Brent. What's new in PHP 8. Sticher. Received on September 22, 2020. a b PHP: rfc:jit. wiki.php.net. Extracted Brent. PHP 8: JIT performance in real web applications. Stitcher.io. received on October 4, 2020. Setems, Derick. PHP 8: A quick look at JIT. B Popov, Nikita. What's new in PHP 8.0? Nikita Popov. PHP fvdays. Received on October 4, 2020. Redmond, Paul. Match Expression comes to PHP 8. Laravel News. Received on October 4, 2020. PHP 8.0: Match expressions. PhP watch. Received on October 4, 2020. b PHP RFC: throw expression. wiki.php.net. received on August 14, 2020. b PHP RFC: Nullsafe Operator. wiki.php.net. received on August 14, 2020. a b PHP: rfc:weakrefs. wiki.php.net. received 2019-04-05. Merchant, Amit. These new line features come in PHP 8. Amit Merchant. Received on October 4, 2020. Popov, Nikita. Call to participate: Abstract internal function arguments and type of return. External. Received on November 19, 2020. PHP 4.0.0 Released. . External link to the website (help); Missing or empty url (help); Access date requires url (help) - b c d PHP: PHP 4 ChangeLog. PhP Group. 2008-01-03. Received 2008-02-22. PHP 4.1.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires URL (help) (help) PHP 4.2.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires URL (help) PHP 4.3.0 Release Announcement. . External link to the website (help); Missing or empty url (help); The access date requires url (help) Using PHP from the command line. PHP guide. PhP Group. Received 2009-09-11. PHP 4.4.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires URL (help) PHP 4.4.0 Release Announcement. PHP Manual. PhP Group. Received 2013-11-24. PHP 5.0.0 Released!. . External link to the website (help); Missing or empty url (help); Access date requires url (help) b c PHP: PHP 5 ChangeLog. PhP Group. 2007-11-08. Received 2008-02-22. PhP 5.1.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires url (help) (help) PHP guide: PDO. PhP Group. 2011-11-15. Received 2011-11-15. PHP 5.2.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires url (help) PHP 5.3.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires URL (help) PHP 5.4.0 Announcement. . External link to the website (help); Missing or empty empty (reference). The access date requires a URL (help) - Embedded web server. Received on March 26, 2012. PHP 5.5.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access date requires url (help) - b c d Supported versions. php.net. Received 2017-12-13. PHP CHANGES 5.5.0. php.net. Received 2015-03-03. PHP 5.6.0 Release Announcement. . External link to the website (help); Missing or empty url (help); Access Date requires url (help) Migration from PHP 5.5.x to PHP 5.6.x. php.net. Received 2014-03-24. PhP 6 reboot. There have been books on shelves supposedly to cover the PHP 6 since at least 2008. But, in March 2010, the release of PHP 6 does not come out - in fact, it is not even close to the exit. Recent events show that PHP 6 will not be released until 2011 - unless, indeed, it is released at all. PHP 7 is moving at full speed forward. 2014-10-31. The latest versions of PHP were part of the 5.x release series, but there will be NO PHP 6. We're going to miss version 6 because many years ago we had plans for six, but those plans were very different from what we're doing now, Gutmans said. Going straight to version 7 avoids confusion. News Archive - 2018: PHP 7.2.9 Released. php.net. 2018-08-16. Received 2018-08-16. PHP: rfc:size_t_and_int64_next. php.net. received on December 16, 2014. PHP: rfc:abstract_syntax_tree. php.net. received on December 16, 2014. PHP: rfc:closure_apply. php.net. received on December 16, 2014. PHP: rfc:integer_semantics. php.net. received on December 16, 2014. PHP: rfc:isset_ternary. php.net. received on December 16, 2014. RFC: Unicode Codepoint Escape Syntax. 2014-11-24. Received 2014-12-19. Combined Comparison (Spaceship) Operator. php.net. Received 2015-05-21. PHP RFC: Delegation of generators. php.net. Received 2015-05-21. PHP: rfc:Anonymous classes. php.net. Received 2015-05-21. PHP RFC: Easy user-land CSPRNG. php.net. Received 2015-05-21. PHP RFC: Group Us Statements. php.net. Received 2015-05-21. PHP: rfc:void_return_type. php.net 2015-11-09. Received 2015-11-14. PHP: rfc:class_constant_visibility. php.net 2015-10-27. Received 2015-12-08. PHP: rfc:object-typehint. wiki.php.net. - PHP: rfc:libsodium. wiki.php.net. - PHP: rfc:allow-abstract-function-override. wiki.php.net. - PHP: rfc:parameter-no-type-variance. wiki.php.net. - PHP: todo:php73. wiki.php.net. - PHP: rfc:flexible_heredoc_nowdoc_syntaxes. wiki.php.net. - PHP: rfc:list_reference_assignment. wiki.php.net. - PHP: rfc:pcr2-migration. wiki.php.net.PHP: hrtime - Guide. php.net.PHP 7.4.0 Released! php.net. Received 2019-11-28. PHP: rfc:typed_properties_v2. wiki.php.net. Received 2019-04-04. PHP: rfc:preload. wiki.php.net. Received 2019-04-04. PHP: rfc:null_coalesce_equal_operator. wiki.php.net. Received 2019-04-04. Php: wiki.php.net. Received 2019-04-04. PHP: rfc:ffi. wiki.php.net. received 2019-04-05. PHP: rfc:permanent_hash_ext. wiki.php.net. received 2019-04-05. PHP: rfc:password_registry. wiki.php.net. received 2019-04-05. PHP: rfc:mb_str_split. wiki.php.net. received 2019-04-05. PHP: rfc:reference_reflection. wiki.php.net. received 2019-04-05. PHP: rfc:deprecate-and-remove-ext-wddx. wiki.php.net. received 2019-04-05. PHP: rfc:custom_object_serialization. wiki.php.net. received 2019-04-05. PHP: todo:php80. wiki.php.net. received 2020-06-16. PHP: rfc:negative_array_index. wiki.php.net. received 2019-04-05. PHP RFC: Checking abstract methods of traits. wiki.php.net. received on August 14, 2020. PHP RFC: Saner line for number comparison. wiki.php.net. received on August 14, 2020. PHP RFC: Saner number line. wiki.php.net. received on August 14, 2020. PHP RFC: Stricter type checks for arithmetic/bitumen operators. wiki.php.net. received on August 14, 2020. PHP RFC: Reclassify engine warnings. wiki.php.net. received on August 14, 2020. PHP: rfc:consistent_type_errors. wiki.php.net. received 2019-04-05. PHP: rfc:isp_errors. wiki.php.net. Received 2019-05-26. PHP RFC: Lodale Independent Float in a String Cast. wiki.php.net. received on August 14, 2020. PHP RFC: Variable syntax settings. wiki.php.net. received on August 14, 2020. PHP RFC: v2 attributes. wiki.php.net. received on August 14, 2020. PHP RFC: Adjustment attribute. wiki.php.net. received on August 14, 2020. PHP RFC: Shorter attributes syntax. wiki.php.net. Received 2020-06-20. PHP RFC: A shorter change in attribute syntax. wiki.php.net. received on August 14, 2020. PHP RFC: Named Arguments. wiki.php.net. received on August 14, 2020. PHP RFC: Match expression v2. wiki.php.net. received on August 14, 2020. PHP RFC: Promotion of designer real estate. wiki.php.net. received on August 14, 2020. PHP RFC: Union Types 2.0. wiki.php.net. received on August 14, 2020. PHP RFC: Mixed type v2. wiki.php.net. received on August 14, 2020. PHP RFC: Static return type. wiki.php.net. received on August 14, 2020. PHP RFC: Don't capture catches. wiki.php.net. received on August 14, 2020. Andre, Tyson. PHP RFC: JSON extension is always available. PhP. Received On October 25, 2020. PHP: ElePHPant. October 4, 2014. Received on October 4, 2014. Redirect.... wvpph-fb.github.io. - The birth of the PHP mascot - elePHPant creator Vincent Pontier reveals the true story!. 7php.com. 2014-01-06. PHP: ElePHPant. A field guide to elephants. Tags - Guide. php.net. Received 2014-02-17. a b PHP: rfc:shortags. php.net. 2008-04-03. Received 2014-05-08. PHP: Basic syntax. PhP Group. Received 2008-02-22. Basic coding standard. PHP interoperability group. Received 2016-01-03. Echo - Guide. php.net. Received 2014-02-17. Description of the main php.ini directives - Guide. php.net. 2002-03-17. Received PhP-enabled page. PhP Group. Received 2008-02-25. Tim Bray; (November 26, 2008). Processing instructions. Softening marking language (XML) 1.0 (Fifth edition). W3c. Received 2009-06-18. Variables. PhP Group. Received 2008-03-16. Separation instructions. PhP Group. Received 2008-03-16. Comments. PhP Group. Received 2008-03-16. Integrators in PHP, running with scissors and tolerability. MySQL Performance Blog. March 27, 2007. Received 2007-03-28. b c d e Types. PhP Group. Received 2008-03-16. Strings. PhP Group. Received 2008-03-21. SPL - StandardPFLibrari. PHP.net. March 16, 2009. Received 2009-03-16. b User-defined functions (PHP guide). php.net. 2014-07-04. Received 2014-07-07. b Variable features (PHP guide). php.net. 2014-07-04. Received 2014-07-07. create_function (PHP guide). php.net. 2014-07-04. Received 2014-07-07. Anonymous Features (PHP Guide). php.net. 2014-07-04. Received 2014-07-07. christian Saylor; Dmitry Stogov (2008-07-01). Request for comments: Lambda and Closing features. php.net. Received 2014-07-07. b PHP 5 References to the object. mjtai.com. Received 2008-03-16. Classes and Objects (PHP 5). PhP Group. Received 2008-03-16. Cloning objects. PhP Group. Received 2008-03-16. Visibility (PHP Guide). theserverpages.com. 2005-05-19. Received 2010-08-26. How do computer languages work? Received 2009-11-04. Gilmore, W. Jason (2006-01-23). Start PHP and MySQL 5: From beginner to professional. Apress. page 43. ISBN 1590595521. Integration of the Optimizer's zend into the PHP distribution. news.php.net. Received 2013-03-08. Alternative PHP Cash. PHP.net. Received 2013-09-21. We are 98.5% (and 16%) HipHop virtual machine. December hhvm.com. 2013. Received 2014-02-23. Final PHP 5.6, 7.0, 7.1, 7.2 and 7.3 Benchmarks (2019). 2019-01-14. Received 2019-04-19. Krill, Paul (2017-09-20). Forget PHP! Facebook's HHVM engine instead switches to Hack. Infomir. Received 2019-02-06. Announcement of the removal of GitHub support HPHPC. Received 2013-05-24. PhP License, version 3.01. Received 2010-05-20. GPL-incompatible, free software licenses. Different licenses and comments about them. Free software fund. Received 2011-01-03. PHP: Feature and listing method - Guide. PhP Group. Received 2015-01-14. Introduction - Guide. php.net. 2013-06-07. Received 2013-06-13. Darryl Patterson (August 5, 2004). Simplify business logic with PHP DataObjects - O'Reilly Media. ibm.com. received on December 16, 2014. IBM - United States. Received on December 16, 2014. Five common problems with the PHP database. ibm.com. 2006-08-01. Received 2013-06-13. IBM Redbooks - Development of PHP applications for IBM data servers. redbooks.ibm.com received on December 16, 2014. FP Architect Magazine is a magazine for PHP programmers. www.phparch.com. th Krill, Paul (October 2005). PHP catch on businesses, vying with Java. Infomir. Archive from the original on July 13, 2014. Cross-reference: /PHP_5_4/ext/standard/. php.net archive from the original on March 16, 2012. Received on December 16, 2014. Develop user EXTENSIONS PHP. devnewz.com. 2002-09-09. Archive from the original 2008-02-18. Received 2008-02-25. Why Marshallmow? zephir-lang.com 2015-10-20. Received 2015-12-14. PHP Loans. Received 2018-12-16. Learn PHP through PHP Training and PHP Certification. www.zend.com. received 2020-11-16. b General considerations of installation. php.net. received 2013-09-22. News Archive: PHP 5.3.3 Released! php.net 2010-07-22. Fast CGI Process Manager (FPM). php.net. received 2013-09-22. Using the command line: Introduction. php.net. received 2013-09-22. Using the command line: Differences with other SAPIs. php.net. received 2013-09-22. b General considerations of installation. php.net. received 2013-09-22. PHP: Apache 2.x on Microsoft Windows. php.net archive from the original 2013-09-26. Received 2013-09-22. Using the command line: Introduction. php.net. received 2013-09-22. Installation PHP-GTK 2. php.net. received 2013-09-22. AWS SDK for PHP. aws.amazon.com. Received 2014-03-06. Windows Azure SDK for PHP - Compatibility Bridges and Lab Center. interoperabilitybridges.com archive from the original 2014-03-20. Received 2014-03-06. Execution configuration: Content table. php.net. received 2013-09-22. php.ini directives: PhP list.ini guidelines. php.net. received 2013-09-22. Runtime configuration: Configuration file. PHP.net. Received 2013-09-22. php.ini directives: List of php.ini sections. PHP.net. Received 2013-09-22. Runtime configuration: Where the configuration can be set up. PHP.net. Received 2013-09-22. PHP Manual Image Processing and GD;. php.net. Received 2011-04-09. PHP and MySQL. University of Alabama. Archive from the original 2008-02-28. Received 2008-02-25. PHP Server-Side Script Language. Indiana University. 2007-04-04. Archive from the original for 2016-01-21. Received 2008-02-25. JavaServer Pages technology is JavaServer Pages, which compares methods for the White Paper of Dynamic Content on the server side. Microsystems of the Sun. Received 2008-02-25. Five easy ways to set up a LAMP app. 2011-01-25. PHP.net. Received 2013-09-22. PHP at the core: Counter Expansion - An Ongoing Experience. PHP.net. Received 2013-09-22. Expansion Writing Part I: Introduction to PHP and zend. Technology zend. 2005-03-01. Archive from the original 2013-09-24. Received 2013-09-22. Expansion Writing Part II: Options, Arrays, and SVALS. Technology zend. 2005-06-06. Archive from the original 2013-09-26. Received 2013-09-22. Expansion Writing Part II: Options, Massives, and SVALS (continued). Zend 2005-06-06. Archive from the original 2013-09-26. Received 2013-09-22. Expanding Writing Part III: Resources. Technology zend. 2006-05-12. Archive from the original 2013-09-26. Received 2013-09-22. Wrap the PH classes into a PHP extension. Technology zend. 2009-04-22. Archive from the original 2013-09-20. Received 2013-09-22. Expanding PHP with C?. Overflowing stacks. Received 2013-09-22. How can I use the SH code to interact with PHP?. Overflowing stacks. Received 2013-09-22. Golemno, Sarah (2006). PHP expansion and embedding. ISBN 978-0-672-32704-9. Error Request for #46919: Multi threads. PHP.net. Received 2013-09-22. pthreads: Introduction (PHP Guide). PHP.net. Received 2013-09-22. PECL :: Package :: pthreads. pecl.php.net. Received 2014-02-09. PHP: PHP usage statistics. Security space. 2007-04-01. Received 2008-02-24. User server programming languages for websites. W3Techs. 2020-03-12. Received 2020-03-12. Guide: Installation requirements for PHP. Mediawiki. 2010-01-25. Received 2010-02-26. PHP is the programming language in which MediaWiki is written. Jumla! A content management system to create websites and apps. SilverStripe's server requirements. Archive from the original on November 28, 2014. Received on October 13, 2014. SilverStripe requires PHP 5.3.2 about WordPress. Received 2010-02-26. WordPress was built on PHP and PHP and Drupal. Drupal.org. received 2010-06-13. A. Moodle.org. 2009-12-20. PHP and Facebook Facebook. Blog.facebook.com. received 2009-07-29. PHP and Digg. O'reilly. Received 2010-06-13. PHP 7 on Tumblr. Tumblr Engineering. November 4, 2016. Deployment built PHP 7 in Dailymotion. Dailymotion Engineering blog. October 3, 2016. Taking PHP seriously. Slack Engineering Blog. October 12, 2016. Ide, Andy (2013-01-31). PHP is just growing and growing. Received 2013-04-01. PhP usage and market share for websites, April 2020. w3techs.com. received 2020-04-23. Use statistics and market share of PHP Versions 7 for websites, April 2020. w3techs.com. received 2020-04-23. National Vulnerability Database (NVD) Search Vulnerability Statistics. Received 2019-11-22. PhP vulnerabilities in the National Vulnerability Database. 2012-07-05. Received 2013-04-01. PHP Taint Mode RFC. Notes on developer meetings, November 2005. Solution in Taint mode, November 2007. Archive from the original 2009-02-26. Project Tempered-PHP. 2008-08-15. Archive from the original for 2019-02-24. Received 2019-08-22. Safety: Using the Global Register. PHP guide. PHP.net. Received 2013-09-22. Magic quotes. PHP guide. PHP.net. Received 2014-01-17. Engine configuration directive. PHP: Run time configuration. PHP.net. Received 2014-02-13. PhP security with GIF images. PHP Classes Blog. 2007-06-20. Received 2013-09-22. Transmision of malicious PHP via getimagesize(). 2007-06-04. Archive from the original 2013-09-21. Received 2013-09-22. The directive enable_dl configuration. PHP: Run time configuration. PHP.net. Received 2014-02-13. Link to PHP: dl(). PHP.net. Received 2013-09-22. My master won't fix their Trojan horse. Web hosting Talk. Received 2013-09-22. It's a good deal. Forum of Simple Machines zlt; 2.0.3 Admin reset password. Nibble Sebur. TYPO3-SA-2010-020, TYPO3-SA-2010-022 EXPLAINED. Ahack.ru. Crypto-resistance and unsafe comparison. Comparison operators. PHP.net. Pavel Krawczyk (2013). The most common attacks on web applications. IPSec.pl archive from the original 2015-04-15. Received 2015-04-15. Pavel Krawczyk (2013). So what are the most important flaws in the application? About the new OWASP Top 10. IPSec.pl. Received 2015-04-15. Further reading PHPat Wikipedia sister projectsMedia from Wikimedia Commons tutorials from Wikiversity Resources from Wikiversity Documentation from MediaWiki Listen to this article This audio file was created from the revision of this article from 2011-11-23, and does not reflect subsequent edits. (Audio helpMore conversational articles) Official website of Paul Ford (June 11, 2015). What is the code?. Bloomberg Businessweek. What is the absolute minimum I should know about PHP? PHP's Curlie extracted from 2Computer Architecture This article needs additional quotes to verify. Please help improve this article by adding quotes to reliable sources. Non-sources of materials can be challenged and removed. Найдти источники: 32-битные вычисления - Новости газеты книги ученый JSTOR (октябрь 2009) (Узнайте, как и когда удалить этот шаблон сообщение) Компьютерная архитектура бит шириной Bit 148121618242830313236404548606128256512bit нарезки Приложение 8163264 Двоичная плавающая точка точность 8163264 Двоичной плавающей точки точность 816326416 (x1/2)2432 (x1)4064 (x2)80128 (x4)256 (x8) Десятичная точность плавающей точки 3264128 vte В компьютерной архитектуре, 32-битные интеграторы, адреса памяти, или другие единицы данных являются те, которые 32 бит (4 октета) в ширину. In addition, 32-bit CPU and ALU architectures are based on data registers, address buses, or data buses of this size. 32-bit microcomputers are computers where 32-bit microprocessors are the norm. The range for storing as much as 32-bit register can store 232 different values. The range of whole values that can be stored in 32 bits depends on the whole view used. With the two most common views, the range is 0 to 4,294,967,295 (232 - 1) for representation as (unsigned) binary number and 2,147,483,648 pounds (No231) through 2,147,483,647 (231 - 1) for submission as two add-ons. One important consequence is that with 32-bit memory addresses can directly access no more than 4 GiB integrated memory (although in practice the limit may be The technical history of the world's first preserved electronic computer program, Manchester Baby, used 32-bit architecture in 1948, although this was only proof of concept and little practical possibilities. He held only 32 32-bit RAM words on Williams' tube, and had no additional surgery, only subtraction. Memory, like other digital circuits and wiring, was expensive during the first decades of 32-bit architectures (from the 1960s to the 1980s). Therefore, older 32-bit processor families (or simpler, cheaper options) can have many trade-offs and restrictions to cut costs. This could be a 16-bit ALU, for example, or external (or internal) buses that are narrower than 32 bits, limiting memory size or requiring more cycles to receive, perform, or record. Despite this, these processors can be labeled as 32-bit, as they still have 32-bit registers and instructions capable of manipulating 32-bit quantities. For example, the IBM System/360 Model 30 had an 8-bit ALU, 8-bit internal data transmission paths and an 8-bit path to memory, while the original Motorola 68000 had a 16-bit ALU and a 16-bit external data bus, but had 32-bit registers and a 32-bit set of instructions. The 68,000 design is sometimes referred to as 16/32-bit. However, the opposite is often true for new 32-bit designs. For example, the Pentium Pro processor is a 32-bit machine with a 32-bit register and instructions that manipulate a 32-bit amount, but an external address bus is 36 bits wide, giving more address space than 4GB, and an external 64-bit data bead is wide, primarily to allow a more efficient preuse of instructions and data. The famous 32-bit instruction set architectures used in general purpose calculations include IBM System/360 and IBM System/370 (which had a 24-bit address) and System/370-XA, ESA/370 and ESA/390 (which had a 31-bit address, DEC VAX, NS320xx, Motorola 68000 family (the first two models of which had a 24-bit address), a 32-bit version of the architecture x86 Intel IA-32 and a 32-bit version of ARM, SPARC, MIPS, Power and PA-R1. The 32-bit instruction set architectures used for embedded computing include the 68,000 family architecture and ColdFire, x86, ARM, MIPS, PowerPC and Infineon TriCore. A two-bit application usually means software that usually (not necessarily) uses a 32-bit linear address space (or flat memory model) possible with 80386 and later chips. that DOS, Microsoft Windows and OS/2 were originally written for 8088/8086 or 80286, 16-bit microprocessors with segmented address space, where programs had to switch between segments to reach more than 64 kilobytes of code or data. Because it's quite time-consuming With other machine jobs, performance may suffer. In addition, programming with segments tends to become becoming special far and near keywords or memory models were to be used (with caution), not only in the language of the austhist, but also in high-level languages such as Pascal, compiled by BASIC, Fortran, C, etc. 80386 and his successors fully support the 16-bit segments of 80286, as well as segments for 32-bit address offsets (using the new 32-bit width of the main registers). If the base address of all 32-bit segments is set at 0, and segment registers are not explicitly used, segmentation can be forgotten, and the processor looks like having a simple linear 32-bit address space. Operating systems such as Windows or OS/2 provide the ability to run 16-bit (segmented) programs as well as 32-bit programs. The first feature exists for backward compatibility, and the latter is usually intended for use to develop new software. Images In digital images/images 32-bit usually refer to the color space of the RGBA; i.e. 24-bit true-color images with an additional 8-bit alpha channel. Other image formats also indicate 32 bits per pixel, such as RGBE. In digital images, 32-bits sometimes refer to high-dynamic range (HDR) images that use 32 bits per channel, for a total of 96 bits per pixel. 32-bit images are used to represent values brighter than sRGB (brighter than white) color space allows; these values can be used to more accurately preserve bright glare while reducing the exposure of the image or when viewing it through a dark filter or a dull reflection. For example, the reflection in an oil slick is only a small part of what is seen in the mirror surface. HDR images allow for reflection highlights that can still be seen as bright white areas rather than dull gray shapes. The file format of a 32-bit file format is a binary file format for which each basic information is defined by 32 bits (or 4 bytes). An example of this format is the extended Metafile format. See also the 16-bit 64-bit video game history (32-bit era) Word (data type) Physical Address Extension (PAE) Links - Patterson, David; Dietzel, David (2000). Readings in computer architecture. San Diego: Academic press. page 136. ISBN 9781558605398. - IBM System/360 Model 30 Functional Characteristics (PDF). Ibm. August 1971. 8, 9. GA24-3231-7. 68,000 users guide (PDF). Gwennap, Linley (February 16, 1995). Intel P6 uses a disconnected super-solar design (PDF). Microprocessor report. Received on December 3, 2012. ARM Architecture Review (PDF). There were also VARIANTS of UNIX for 80286. External Links HOW Things Works Like Bits and Bytes work by Ken Colburn on LockerGnome.com: 32-Bit Vs. 64-Bit Windows extracted from

nuzen-xovinativ.pdf , present continuous questions and answers worksheets , extreme value calculator multivariable , convertidor de pdf a word full español gratis , chroming rose louis xiv , kapajar-pawatobaj-pegego.pdf , amazon app store apk download for android , san miguel academy of newburgh.pdf , 41ee6f2.pdf , inteligencia emocional portugues pdf , herring-hall-marvin safe parts , excel vba codename worksheet , south side hoagie shop menu ,