Engineer

PHP

мова, на якій працює старий веб

Що це таке

PHP — це мова сценаріїв, код якої виконується на вебсервері: надходить запит, запускається PHP-код, і браузер отримує HTML, який цей код згенерував, а не сам код. Сьогодні ці літери означають «PHP: Hypertext Preprocessor», рекурсивний акронім, хоча спочатку вони розшифровувалися як «Personal Home Page». PHP-файл може бути звичайним HTML із кодом між тегами `<?php` і `?>` або суцільним кодом, через який фреймворк пропускає кожен запит. Мова вміє виконувати сценарії й з командного рядка, але задумана вона для того, щоб генерувати вебсторінки.

Історія мови починається 1994 року, коли Расмус Лердорф написав на C набір CGI-програм, щоб рахувати відвідування свого онлайн-резюме, і назвав їх Personal Home Page Tools. У червні 1995 року він відкрив вихідний код, і з цих інструментів виріс PHP/FI. 1997 року Енді Гутманс і Зеєв Сураскі взялися переписувати парсер, і результат вийшов у червні 1998 року як PHP 3.0 уже під рекурсивною назвою; на піку його було встановлено приблизно на 10% вебсерверів інтернету. Їхній наступний рушій, Zend Engine, названий на честь обох авторів, ліг в основу PHP 4.0 у травні 2000 року, а в липні 2004 року вийшов PHP 5.

На сучасному сервері PHP часто запускають через PHP-FPM, менеджер процесів FastCGI, який постачається разом із мовою. Вебсервер перед ним передає кожен PHP-запит пулу робочих процесів. Кожен пул може працювати від імені власного користувача й групи, з власним оточенням і власними значеннями php.ini, а менеджер процесів тримає фіксовану кількість процесів, запускає їх на вимогу або змінює їхню кількість у заданих межах. Сама мова розвивається гілками випусків, і чотири останні з’являлися з інтервалом приблизно в рік. Кожна гілка отримує два роки активної підтримки з виправленнями помилок і вразливостей, потім ще два роки лише критичних виправлень безпеки, після чого її підтримка завершується.

PHP 8.2, випущений 8 грудня 2022 року, отримує виправлення безпеки до 31 грудня 2026 року, а PHP 8.5, випущений 20 листопада 2025 року, — до 31 грудня 2029 року.

PHP перетворює типи, коли цього вимагає операція, і найпомітніше це в нестрогому порівнянні `==`. Якщо обидва операнди — числові рядки або один із них число, а другий числовий рядок, порівняння відбувається як порівняння чисел. До PHP 8.0.0 будь-який рядок, який порівнювали з числом, спершу перетворювався на число, тому `0 == "a"` давало true. Строгий оператор `===` порівнює і тип, і значення та нічого не перетворює, і сам посібник радить у більшості випадків надавати перевагу саме йому.

```
<?php
var_dump(0 == "a");      // PHP 7: true, PHP 8: false
var_dump("1" == "01");   // true в обох версіях: два числові рядки
var_dump("10" == "1e1"); // true в обох версіях: 10 дорівнює 1e1
var_dump(100 == "1e2");  // true в обох версіях
var_dump(0 === "a");     // false в обох версіях: === не перетворює типів
```

Застосуйте поняття до рішення

Застосуйте «Що таке PHP?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Та частина, якої користувач не бачить: де зберігаються дані, які діють правила і як інші системи можуть із нею взаємодіяти.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Спроєктувала й підтримувала 6 пайплайнів ETL/ELT — 6 пайплайнів ETL на трьох рушіях баз даних

Спроєктувала й підтримувала 6 пайплайнів ETL/ELT Зв'язатися з нами →

Чому це важливо

PHP важливий насамперед через те, що на ньому вже написано. WordPress і Joomla, системи керування вмістом, якими дуже багато організацій ведуть сайти, що їх самі й редагують, — це PHP-застосунки: WordPress рекомендує PHP 8.3 або новіший, а Joomla 6 вимагає щонайменше PHP 8.3.0. WordPress досі працює на PHP 7.4, але його власна сторінка вимог попереджає, що строк підтримки таких версій уже минув. Бізнес може роками залежати від PHP, хоча ніхто в ньому не написав жодного рядка цією мовою, і саме тому про версію під сайтом забувають. Сайт і далі віддає сторінки після того, як його гілку перестали підтримувати; він просто більше не отримує виправлень, і на екрані про це ніщо не свідчить.

Оновлення ще й непомітно змінюють поведінку. PHP 8.0 змінив порівняння рядків із числами, тож умова, що була істинною на PHP 7, на PHP 8 може стати хибною без жодного повідомлення про помилку. PHP 8.4 підняв вартість bcrypt за замовчуванням у `password_hash()` з 10 до 12, тож кожен новий хеш пароля обчислюється на тому самому обладнанні довше. Обидві зміни задокументовано, і кожна потрапляє в продакшн лише через оновлення, яке хтось має спланувати, перевірити й узяти на себе.

Наша рекомендація — ставитися до версії PHP як до залежності, поруч із якою записано дату завершення строку. Наявний PHP-застосунок, що працює, варто зберегти: перевести на підтримувану гілку, запланувати наступне оновлення задовго до кінця підтримки безпеки й виділити йому окремий пул FPM та окремого системного користувача. Для нового проєкту PHP доречний, коли йдеться про сайт, який клієнт редагуватиме сам у WordPress чи Joomla, або коли команда, що супроводжуватиме його після передачі, добре знає PHP; в інших випадках вибір варто робити за навичками цієї команди. Ризик рідко в самій мові; він у версії, за яку ніхто не відповідає.

Як це виглядає на практиці

Вебрівень на нашій сторінці технологій відводить PHP певне місце. Сторінки, які можуть бути файлами, збираються статично, щоб не виконувати код застосунку під час обробки запитів до цих сторінок, і цей сайт збирає Hugo; Django і PHP використовуються там, де є застосунок, який треба обслуговувати. Та сама сторінка пояснює, як ухвалюється рішення: під завдання, під те, що команда клієнта зможе експлуатувати, коли ми підемо, і з огляду на те, як система поводиться під час збою, а не на демонстрації. Це правило прямо стосується й PHP: коли правильна відповідь — стек, який у клієнта вже є, сторінка обіцяє так і сказати.

У 2015–2022 роках засновниця працювала керівницею команди розробки GIS-програмного забезпечення й провела понад тридцять GIS-проєктів через розробку, розгортання та підтримку після запуску. Між просторовими даними в PostgreSQL із PostGIS і вебкартами на Mapbox GL та Leaflet, які подекуди зверталися до API ArcGIS чи HERE, стояв рівень застосунку, і PHP був однією з чотирьох його мов поряд із JavaScript, Python і SQL. Головна думка сторінки стосується кожної з них: відповідальність не закінчувалася на запуску, проблеми не передавалися комусь іншому, а проєктування відбувається інакше, коли знаєш, що саме тобі дзвонитимуть, якщо щось зламається.

Де виникають проблеми

- **Нестроге порівняння охороняє секрет.** Токен або хеш, перевірений через `==`, порівнюється за числовими правилами PHP, щойно обидва боки схожі на числа, тож два різні рядки, як-от `"10"` і `"1e1"`, вважаються рівними. Для паролів використовуйте `password_verify()`, для інших секретів — `hash_equals()`: час її виконання нічого не розкриває про вміст секрету, а рядок від користувача вона очікує другим аргументом. Скрізь інде — `===`.
- **SQL складається з даних запиту.** Підготовлені вирази PDO самі беруть параметри в лапки, і посібник стверджує, що застосунок, який використовує лише їх, захищений від SQL-ін'єкцій, з одним названим винятком: частини запиту, які все ще будуються з неекранованого введення. Заповнювач заміняє ціле значення, тож усе, що значенням не є, наприклад стовпець, за яким сортують список, однаково потрапляє в рядок. Такі частини слід звіряти з фіксованим списком у коді.
- **Хеші паролів зберігаються в надто вузькому полі.** `PASSWORD_DEFAULT` сьогодні означає bcrypt, і посібник попереджає, що ця константа задумана змінюватися з появою сильніших алгоритмів. Він радить стовпець, який може вирости понад 60 байтів, і називає 255 вдалим вибором. До того ж bcrypt обрізає пароль на 72 байтах. Зберігайте хеш повністю й викликайте `password_needs_rehash()` після успішного входу, щоб старі хеші оновлювалися, коли люди входять.
- **Гілка минає дату завершення підтримки.** Застосунок працює далі, тож оновлення ніхто не планує, і стрибок, що зрештою настає, охоплює кілька гілок одразу. Запишіть дату завершення підтримки безпеки для гілки, яка використовується, оновлюйтеся на одну гілку за раз і проганяйте тести на наступній гілці до того, як завершиться строк поточної.
- **Пул FPM спільний і налаштований абияк.** Один пул, що обслуговує кілька застосунків від імені одного користувача, дає змогу вразливості в одному з них дістатися файлів усіх інших. `security.limit_extensions` за замовчуванням має значення `.php .phar`, і посібник радить обмежити його до `.php`, щоб помилково налаштований вебсервер не передав PHP жодних інших файлів. Дайте кожному застосунку власний пул і користувача, а `pm.max_children` встановіть таким, щоб його витримала пам'ять сервера.

Що почитати далі

- PHP Manual: What is PHP? — коротке визначення від самого проєкту й перший приклад PHP, вбудованого в HTML.
- PHP: Supported Versions — політика підтримки й дати активної підтримки та підтримки безпеки для кожної чинної гілки.

- PHP Manual: Comparison Operators — нестроге й строге порівняння, результати PHP 7 і PHP 8 поруч і таблиця порівняння за типами.
- PHP Manual: password_hash — алгоритми, вартість, ширина стовпця, межа в 72 байти й нова вартість за замовчуванням у 8.4.
- PHP Manual: Prepared statements and stored procedures — як PDO прив'язує параметри і де SQL-ін'єкція все ще можлива.
- PHP Manual: FPM configuration — директиви пулу, режими менеджера процесів, `clear_env` і `security.limit_extensions`.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке PHP?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/php/