

# PHP tötet Python

Beau Beauchamp

## Warum PHP beliebter und verhasster ist als je zuvor

Wieder einmal muss ich gegen den Strom der Popkultur schwimmen, um einen Realitätscheck zu liefern, den die meisten von Ihnen, die dies lesen, nicht hören wollen: PHP, das Bastard-Kind unter den Programmiersprachen, ist beliebter denn je!

Warum eigentlich? Warum ist PHP nicht schon tot? Es gibt doch so viele "bessere" Programmiersprachen da draußen, oder?

Nun, nein, die gibt es nicht. Sonst hätten sie die Führung übernommen und zerstört, was PHP schon vor Äonen getan hat. Wir müssen unsere Definition dessen, was eine "bessere" Sprache ist, neu überdenken.

Wenn Sie sich am Kopf kratzen und sich fragen, warum PHP nicht tot ist oder sogar stirbt, dann liegt das wahrscheinlich daran, dass Sie einige Dinge über Programmierung, Programmierer, Märkte und Demokratie nicht verstehen. Dinge, die eigentlich nichts mit Code, Funktionen oder Syntax zu tun haben.

## Es ist Vogue, PHP zu bashen

Um sich im heutigen Programmierzirkus der Ingenieurscliquen einen Namen zu machen, muss man jede Sprache schlecht machen, die mit der konkurriert, die man zu mögen gelernt hat. Hier geht es nicht um

Funktionalität, sondern um Kultur. Eigentlich ist es eher so etwas wie eine Religion.

Programmierer sind ein eigensinniger Haufen. Wir sind arrogant. Finde dich damit ab. Das sind wir. Und wir lieben es, über Sprachen herzuziehen, die wir nicht mögen und nicht benutzen. Es gibt Ingenieure, die Dinge sagen wie: "Java ist scheiße. C# ist scheiße. Go ist scheiße." Es spielt wirklich keine Rolle, um welche Sprache es sich handelt. Sie werden alle von allen kritisiert.

Aber es scheint, als gäbe es eine noch tiefere Verachtung, ja sogar eine unausgesprochene Rivalität zwischen PHP und Python; fast so, als gäbe es eine Verschwörung innerhalb der akademischen Welt, die PHP am liebsten auslöschen und durch Python ersetzen würde.

Sogar jetzt kann man Wikipedia-Artikel finden, in denen der LAMP-Stack nun auf magische Weise Python enthält. HUH? Vor 20 Jahren, als ich LAMP benutzte, war das noch nicht so. PHP war das "P" im LAMP-Akronym. Es war nicht Python oder gar Perl. Ich sage nicht, dass sich Dinge und Definitionen nicht ändern können. Vielleicht bin ich ein Verschwörungstheoretiker? Nicht-inklusiv? Oder haben sich die Python-Anhänger unser Akronym "kulturell angeeignet"?

Die meisten von uns lernen Python an der Uni zusammen mit ein paar anderen Nischensprachen, die wir in der realen Welt wahrscheinlich nie benutzen werden. Entweder unsere Professoren oder andere Anfänger sagen uns: "PHP ist scheiße!", obwohl die meisten von ihnen noch nie eine einzige Zeile PHP geschrieben haben.

Allein der Gedanke daran, dass jemand professionell PHP schreibt, lässt einen völlig aus dem Konzept geraten, so als würden die Menschen zu Füßen des Antichristen ausschwärmen.

Ihre Verachtung für PHP ist nicht so sehr logisch, sondern eher intuitiv oder gemeinschaftlich. Es ist "hip", mit Freunden einen Milchkaffee bei Starbucks zu trinken und dabei über Trump, Nickelback und PHP zu lästern.

Dann kommen die Programmierpuristen, die Stunden ihrer wertvollen Existenz damit vergeudet haben, sich darüber aufzuregen, warum PHP so schlecht ist. Die meisten dieser Argumente drehen sich um Lieblingsvorlieben oder Nuancen, die sie nicht mögen, damit sie ihre eigene Lieblings-Programmierreligion, äh, -sprache, als den einen Ring, der sie alle regiert, anpreisen können!

Die Beschwerden nehmen kein Ende: PHP ist lose typisiert und fördert schlechten Code. (Das gilt auch für Javascript und Node.js, die Sie immer noch verwenden.) PHP hat zu viele Funktionen. (Ja, das ist bei manchen Leuten so eine Sache, es ist zu bequem und macht zu viel.) Die Funktionsnamen sind inkonsistent. (Ja, das passiert, wenn eine Sprache wirklich demokratisiert ist und nicht nur von einem Typen geschrieben wird, der versucht, seinen Dokortitel in Informatik zu verdienen.)

Der Punkt hier ist, dass die Kritik an PHP typischerweise in das Lager der persönlichen Vorlieben oder pingeligen Kleinigkeiten fällt. Wenn wir das wirklich wollten, könnten wir die gleichen Vorlieben auch bei Python, C#, Java, Go, Javascript oder was auch immer finden. Als Programmierer beschwerten wir uns gerne und viel. Das ist unser Job.

Aber der eigentliche Grund, warum wir als Programmierer gerne über PHP herziehen, ist eine Mischung aus der uns innewohnenden Selbstüberschätzung und einer nicht geringen Menge an dünn verschleiertem Neid.

## PHP ist einfach einfacher zu benutzen

PHP gibt es seit Mitte der 90er Jahre, noch bevor einige von uns geboren wurden. Ich habe Anfang der 2000er Jahre angefangen, PHP zu benutzen, als es von Version 4 auf 5 überging, und habe dann angefangen, Anwendungen in CodeIgniter, ZendFramework und meinem eigenen Framework zu schreiben.

Der Grund, warum ich die Langlebigkeit von PHP erwähne, ist, dass dies der Hauptgrund dafür ist, dass die Sprache nicht verschwindet - sie ist nicht nur in neue und ältere Anwendungen auf der ganzen Welt eingebettet, sondern auch in die Menschen, sowohl junge als auch ältere, die sie nicht nur seit ein paar Jahren, sondern seit Jahrzehnten verwenden. Und da *wir* nicht verschwinden *werden*, wird auch der von uns geschriebene kommerzielle Code nicht verschwinden.

Interessanterweise gibt es Python bereits seit 1991, bevor PHP das Licht der Welt erblickte. Doch obwohl es eine noch längere Anlaufzeit hatte, konnte es nicht die gleiche Verbreitung erreichen wie PHP heute.

Und warum ist das so?

Für mich persönlich ist Python im Vergleich zu PHP sehr mühsam zu bedienen. Dieses Problem hat sich mit Python 3 etwas geändert, aber damals, vor 15 Jahren, hatte Python noch nicht all den "Schnickschnack"

(Funktionen), den PHP von Haus aus mitbrachte. Man musste einen Großteil dieser Funktionen manuell installieren.

Das andere Problem, das ich mit Python hatte, damals wie heute, war die Syntax mit den Abständen. Ich hasse es, Code zu schreiben, bei dem die Abstände die Begrenzungssyntax bestimmen. Ich möchte meinen Code so einteilen, wie es mir gefällt, und ich möchte nicht, dass meine Programmiersprache mir dabei in die Quere kommt. Nennen Sie mich, wie Sie wollen, aber es ist mein gottverdammter Code, und ich werde ihn so schreiben, wie ich will, und der Rest von euch PSR-Puristen kann zur Hölle fahren. (Sehen Sie, was ich mit rechthaberisch meine?)

Python wirbt mit einer "leicht lesbaren" Syntax, aber das ist nur Marketing-Schwachsinn. Nennen Sie es einfach so, wie es ist: Python verwendet die schlimmste rechthaberische Syntax, die es gibt.

Es handelt sich um eine hochgradig eigenwillige Syntax, die anfällig für die Einführung von Fehlern war; und ich vermute, dass die Entwickler sie früher, bevor die IDEs all diese Dinge für sie korrigierten, einfach nicht gerne verwendet haben. Ich weiß, dass ich es nicht tat. Und das ist, oder besser gesagt, war wahrscheinlich ein Teil des Grundes, warum die Sprache vor einem Jahrzehnt oder mehr unter einer geringen Akzeptanz litt. Sie war einfach nicht sehr benutzerfreundlich.

Ich glaube auch, dass die Leute einfach nicht wussten, dass sie existiert. Es war in Linux-Distributionen als Skriptsprache eingebettet und wurde hauptsächlich dafür verwendet. Niemand rannte schreiend hinaus: "Oh, Python, jetzt kann ich eine Website erstellen!"

Tatsächlich konnte Python zwar Websites erstellen, aber wirklich gute Ingenieure sahen die Leistungsfähigkeit von Python nicht als die beste Verwendung für das Web an. Abgesehen von meinen persönlichen Vorlieben in Bezug auf die Syntax, ist Python eine SEHR mächtige Sprache funktional, vor allem mit seinen Bibliotheken für Data Science oder andere schwere mathematische Zwecke, wo Präzision ist ein absolutes.

## PHP ist eine Sprache, die jeder benutzen kann

Ich glaube, Python hätte PHP mit den vielen Aktualisierungen, die es in den letzten 15 Jahren erfahren hat, schon längst überholen können - wenn da nicht zwei Dinge wären: Erstens kam es ein Jahrzehnt zu spät mit diesen Aktualisierungen.

Python 3 aus dem Jahr 2008 hat die Abwärtskompatibilität aufgehoben. Python-2-Code läuft nicht mit Python 3, es sei denn, Sie ändern ihn. Und wenn Sie eine Million Codezeilen haben, die aktualisiert werden müssen, dann wird dieser Hund nicht jagen, vor allem nicht in einer Unternehmensumgebung.

Fairerweise muss man sagen, dass PHP 7 und 8 auch einigen PHP 5-Code (es gibt kein PHP 6) zerstört haben. Ich war Teil der [ZendFramework 1-Community](#), die dabei geholfen hat, das Framework zu aktualisieren, damit es auf PHP 7 und 8 läuft. Glücklicherweise ist fast keiner meiner Legacy-Codes für Unternehmen beim Upgrade von ZF1 auf PHP 7 und 8 kaputt gegangen. Ein Großteil des PHP 5.6-Codes ist also immer noch sehr rückwärtskompatibel.

Aber der vielleicht größte Wendepunkt, der PHP zum Durchbruch verhalf, war ein kleines Startup-Unternehmen namens Automattic, das 2005 gegründet wurde.

WordPress hatte bescheidene Open-Source-Anfänge, aber es veränderte die Art und Weise, wie nicht nur Entwickler, sondern auch normale Menschen im Internet veröffentlichen konnten.

Bis zu diesem Zeitpunkt waren Drupal, Joomla oder eine andere BBS-Software die beste Wahl für die Veröffentlichung. (Ich bin mir sicher, dass es noch ein paar andere gab, die ich übersehen habe.) Aber sie waren alle schwer zu bedienen, weil die Leute, die sie geschrieben haben, wie Programmierer dachten und nicht wie echte Endbenutzer.

Wäre WordPress in Python geschrieben worden, ja, dann wäre PHP möglicherweise auf dem absteigenden Ast. Aber ich wage zu behaupten, dass Python den kometenhaften Aufstieg von WordPress gebremst (wenn nicht sogar ganz verhindert) hätte.

Die Akzeptanz und damit die Unterstützung durch die Mitwirkenden ist für das Überleben einer Sprache oder Anwendung auf dem Open-Source-Markt von entscheidender Bedeutung. Und nichts fördert diese Akzeptanz schneller als etwas, das einfach verdammt einfach zu benutzen ist.

Sagen Sie als studierter Software-Ingenieur, was Sie wollen, aber PHP wird nicht nur von gestandenen Software-Ingenieuren verwendet, sondern auch von Nebenjobbern, Hausfrauen, Grafikdesignern, Unternehmensmanagern, Unternehmern und Grundschulkindern, die

mit ihren ersten Computern Webseiten schreiben. Hunderte von Millionen von ihnen.

Man könnte sagen, dass PHP das BASIC (Sprache) des Informationszeitalters ist.

Als Software-Ingenieur *denken* Sie wie ein Software-Ingenieur und nicht wie jemand, der einfach nur eine Website für sein kleines Unternehmen einrichten möchte und dafür nur begrenzte Zeit und Ressourcen zur Verfügung hat.

WordPress und PHP haben das alles geändert.

Ja, WordPress an sich war einfach zu benutzen, aber es hatte nicht alles. Es war die Plugin-Architektur von WordPress (ich verwende den Begriff "Architektur" hier sehr locker), die es ganz normalen Menschen ermöglichte, winzig kleine Codestücke in PHP zu schreiben, um Änderungen vorzunehmen, ohne ganze Klassen umschreiben oder Software-Ingenieure werden zu müssen, um etwas zu erreichen.

Diese Millionen von Amateurprogrammierern teilten dann ihre Erfolge mit anderen im ganzen Web und ließen die Plattform wachsen.

Die leicht verständliche PHP-Sprache ermöglicht es normalen Menschen, funktionierenden Code zu schreiben. Es ist einfach. Sie ist nicht rechthaberisch. Sie ist locker typisiert. Kein Kompilieren. Es funktioniert einfach.

## PHP betreibt das World Wide Web



Spulen wir 15 Jahre zurück, und heute werden 78 % des Internets mit [PHP](#) betrieben. [WordPress](#) allein ist für fast 40 % aller Websites verantwortlich und hat fast 2/3 des CMS-Marktanteils! Der Punkt ist, dass [Python](#) sich wünscht, 10% dieser Märkte zu haben! Trotz des Hasses, den die rivalisierenden Python-Puristen gegen PHP hegen, hat Python einen Anteil von weniger als 1,5 % am Web.

Und so sollte es auch sein.

Warten Sie. WAS?!

Ich wurde von Personalvermittlern mit Firmenkunden kontaktiert, die ihren gesamten Python-Webcode herausreißen und durch PHP ersetzen wollten. Das mag Sie vielleicht schockieren, aber es kommt häufiger vor, als es sollte. Und warum? Weil, ehrlich gesagt, das Web nicht die beste Anwendung für Python-Code ist. Python-Programmierer sind in der Regel Datenwissenschaftler und andere Spezialisten. Und erfahrene Spezialisten sind Mangelware.

Unternehmen müssen oft sehr flexibel sein. Sie haben keine Zeit, wochen- oder monatelang nach einem neuen Python-Experten zu suchen, wenn sie ihr Team erweitern müssen oder wenn jemand ausscheidet. Python-Ingenieure sind auch teuer.

PHP-Programmierer für kleine Unternehmen bis hin zu Unternehmen sind viel, viel einfacher zu finden und sogar zu schulen, wenn Sie es brauchen. Der Punkt ist, dass Sie einen erfahrenen, erfahrenen PHP-Architekten mit jahrzehntelanger Erfahrung viel leichter finden können als die Python-Version.

Der Punkt ist, wenn Sie für ein Unternehmen arbeiten und deren Web- oder Middleware in Python schreiben, ohne einen **wirklich guten Grund** dafür zu haben, sollten Sie eine Ohrfeige bekommen. Sie belasten das Unternehmen mit Code, der sie später viel Geld kosten wird, um ihn herauszureißen und durch das richtige Tool zu ersetzen.

Die erste Aufgabe eines Anwendungsarchitekten besteht darin, dem Unternehmen die effizienteste Technologie für den jeweiligen Anwendungsfall zur Verfügung zu stellen, und nicht nur die eigenen Vorlieben aufzusatteln, weil es die neueste "Modeerscheinung" von Google ist oder das, was man Ihnen in der Schule beigebracht hat.

Es gibt nur sehr wenige Anwendungsfälle, in denen ich empfehlen würde, eine SaaS-Anwendung komplett in Python zu schreiben. Und wenn man den Marktanteil von Python im Web betrachtet, scheint es, als hätten die meisten Anwendungsarchitekten genau das getan. Sie haben Python dort eingesetzt, wo es am besten ist, und PHP dort, wo es ebenfalls am besten ist.

## Schlussfolgerung

Als Ingenieure verwenden wir alle gerne das, was wir wissen, und tun alles, was wir nicht wissen, ab. So sind wir eben alle gestrickt.

Hochschulen und Universitäten haben versucht, Python als kommerzielle Allzwecksprache zu positionieren. Das ist es wirklich nicht. Python ist viel besser als High-End-Sprache für den Betrieb von Servern, Data Science, das Crunchen großer komplexer Zahlen und sogar das Mining von Kryptowährungen positioniert, was es alles sehr gut kann.

PHP wiederum ist die Sprache der Massen und hat sich *de facto* zur Sprache des World Wide Web entwickelt.

Sie können also PHP als das uneheliche Kind der Sprachen verunglimpfen, so viel Sie wollen. Aber am Ende des Tages ist es immer noch die mit Abstand am weitesten verbreitete Sprache im Web; und mit dem anhaltenden kometenhaften Wachstum von WordPress und der neuen Geschwindigkeit von PHP 7 & 8 wird sich dieser Status in absehbarer Zeit nicht ändern.

---

[Beau Beauchamp](#) ist ein Enterprise UI/UX Full-Stack-Webanwendungsarchitekt und leitender Entwickler mit über 20 Jahren Erfahrung in der Entwicklung skalierbarer Anwendungen in der Cloud für einige der größten amerikanischen Unternehmen, darunter: Disney, AAA, Enterprise, Sprint und Dell.