30 Years of Code, 25 Years of Tests

A Journey Through Code and Community

2025

How did we get here?

1995

Bill Clinton

1995

Helmut Kohl

1995

Motorola PowerPC 603e (100 MHz)

1995

Intel Pentium Overdrive (63 MHz)

1995

"Hackers"

1995

Rasmus Lerdorf publishes PHP Tools

From: rasmus@io.org (Rasmus Lerdorf)
Subject: Announce: Personal Home Page Tools (PHP Tools)
Date: 1995/06/08
Message-ID: <3r7pgp$aa1@ionews.io.org>#1/1
X-Deja-AN: 104053006
organization: none
newsgroups: comp.infosystems.www.authoring.cgi

Announcing the Personal Home Page Tools (PHP Tools) version 1.0.

These tools are a set of small tight cgi binaries written in C.

They perform a number of functions including:

. Logging accesses to your pages in your own private log files
. Real-time viewing of log information
. Providing a nice interface to this log information
. Displaying last access information right on your pages
. Full daily and total access counters
. Banning access to users based on their domain
. Password protecting pages based on users' domains
. Tracking accesses ** based on users' e-mail addresses **
. Tracking referring URL's - HTTP_REFERER support
. Performing server-side includes without needing server support for it
. Ability to not log accesses from certain domains (ie. your own)
. Easily create and display forms
. Ability to use form information in following documents

Here is what you don't need to use these tools:

. You do not need root access - install in your ~/public_html dir
. You do not need server-side includes enabled in your server
. You do not need access to Perl or Tcl or any other script interpreter
. You do not need access to the httpd log files

The only requirement for these tools to work is that you have
the ability to execute your own cgi programs. Ask your system
administrator if you are not sure what this means.

The tools also allow you to implement a guestbook or any other
form that needs to write information and display it to users
later in about 2 minutes.

The tools are in the public domain distributed under the GNU
Public License. Yes, that means they are free!

For a complete demonstration of these tools, point your browser
at: http://www.io.org/~rasmus

--
Rasmus Lerdorf
rasmus@io.org
http://www.io.org/~rasmus


#include <ctype.h>
#include <string.h>
#define ishex(x) (((x) >= '0' && (x) <= '9') || ((x) >= 'a' &&
(x) <= 'f') || ((x) >= 'A' && (x) <= 'F'))
int htoi(char *s) {
int value;
char c;
c = s[0];
if(isupper(c)) c = tolower(c);
value=(c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10) * 16;
c = s[1];
if(isupper(c)) c = tolower(c);
value += c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10;
return(value);
main(int argc, char *argv[]) {

1993

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#define ishex(x) (((x) >= '0' && (x) <= '9') || ((x) >= 'a' && \
                  (x) <= 'f') || ((x) >= 'A' && (x) <= 'F'))

int htoi(char *s) {
        int     value;
        char    c;

        c = s[0];
        if(isupper(c)) c = tolower(c);
        value=(c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10) * 16;

        c = s[1];
        if(isupper(c)) c = tolower(c);
        value += c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10;

        return(value);
}

void main(int argc, char *argv[]) {
        char *params, *data, *dest, *s, *tmp;
        char *name, *age;
```

https://talks.php.net/phpbcn19#/hist_1993

1993

```
use CGI qw(:standard);
print header;
print start_html('Form Example'),
    h1('My Example Form'),
    start_form,
    "Name: ", textfield('name'),
    p,
    "Age: ", textfield('age'),
    p,
    submit,
    end_form;
if(param()) {
    print "Hi ",em(param('name')),
        "You are ",em(param('age')),
        " years old";
}
print end_html;
```

https://talks.php.net/phpbcn19#/hist_1993_2

1994-1995

```
<html><head><title>Form Example</title></head>
<body><h1>My Example Form</h1>
<form action="form.phtml" method="POST">
Name: <input type="text" name="name">
Age: <input type="text" name="age">
<br><input type="submit">
</form>
<?if($name):?>
Hi <?echo $name?>, you are <?echo $age?> years old
<?endif?>
</body></html>
```

https://talks.php.net/phpbcn19#/hist_1994

C API FOR THE WEB

```
void Cos(void) {
    Stack *s;
    char temp[64];

    s = Pop();
    if(!s) {
        Error("Stack error in cos");
        return;
    }
    sprintf(temp,"%f",cos(s->douval));
    Push(temp,DNUMBER);
}
```

AND YOU COULD THEN USE IT LIKE THIS:

```
<html><head><title>Cos Example</title></head>
<body><h1>Cos Example</h1>
<?echo Cos($input)>
</body></html>
```

https://talks.php.net/phpbcn19#/7

thePHP.cc

1998

PHP 3


APACHE
MIRRORED WORLDWIDE
APACHE SOFTWARE FOUNDATION
<?php

From: Kristian Koehntopp <kk () shonline ! de>
Date: Tue, 19 May 1998 12:56:15 +0000
To: php-general
Subject: [PHP3] A PHP standard library (was: Persistent Variables in PHP3)
X-MARC-Message: https://marc.info/?l=php-general&m=90222497032618

PHP should ship with, and install by default, a standard library.
This library should offer certain standard functionality that is
often required in larger web projects.

The language itself does not need to support any of the
functionality that should be in the library (see below). In fact,
quite a lot of PHP special functions are probable candidates for
library functions that could (and perhaps should) be removed from
the language core, if you are into language purity (I am not).
What PHP needs in my personal opinion is a more stable, better
debugged and better equipped modularization mechanism (and better
namespace management).

1998

PHPLIB

https://marc.info/?l=php-general&m=902224970326l8&q=mbox

1998

PHPLIB

https://marc.info/?l=phplib&m=94064176914442&q=mbox

I can help you with

- adopting PHPUnit,
- optimizing its use,
- refining development processes, and
- writing more testable code

thePHP.cc

2000

PHP 4

2000

PEAR

2000

First PHP conference (in Japan)

2000

Second PHP conference (in Cologne)

2000

I start working on PHPUnit

2002

Xdebug

Apparently Zend made Dmitry an offer he couldn't refuse :)

From chat log of July 2nd, 2003:

```
<@Zeev> Turck_ is copying every darn thing that we're doing
<@Zeev> I'm really not sure why this Dmitri guy is doing it
<@Derick> Zeev: ever asked him?
```

On internals@php.net yesterday:

```
From: Dmitry Stogov <dmitry - zend - com>
To: internals@lists.php.net
Subject: [PHP-DEV] CVS Account Request: dmitry

Overall & engine development
```

https://derickrethans.nl/godfather.html

2004

PHP 5

HOW THE OPEN-SOURCE WORLD PLANS TO SMACK DOWN MICROSOFT, AND ORACLE, AND . . .

By David Kirkpatrick

Steve Ballmer made a sudden and unscheduled trip to Munich last winter. The CEO of Microsoft had been vacationing with his family in Europe when he got word that the Bavarian capital was about to scrap the Windows operating system on its 14,000 PCs and switch to free "open source" Linux software to run its machines. Loath to lose a prominent government customer, Ballmer jumped into a business suit and rushed to Munich. But he was too late. The city decided to go open source.

What happened in Germany is a microcosm of a change that is sweeping the $200-billion-a-year software industry. Open-source software is popping up in servers that power the world's websites and in giant corporate and government systems. Today the biggest challenge confronting Microsoft—and Oracle and IBM and virtually every other major software maker—is chillingly simple: How do you compete with programs that can be had free?

In just a few years, a grassroots approach to creating software has shaken the status quo. In 1991, Linus Torvalds, a college kid in Finland, posted his Linux operating system online and invited friends to use and improve it. The availability of this basic, powerful software, which works on Intel's ubiquitous microprocessors, coincided with the explosive growth

GARAGE BAND They may look casual, but the foursome in this San Jose driveway are serious about rocking the software world. Each masterminds a popular open-source product. From left: Marten Mickos is president of MySQL, a Swedish startup taking on Oracle and IBM in databases; Rasmus Lerdorf created PHP, which companies use to write custom applications; Greg Stein chairs the Apache Software Foundation, which gives away the server software that runs most websites; Linus Torvalds is the father of Linux (the house and car are his). Though the four rarely meet, their software is used in combination by thousands of corporations. The four-program bundle is so popular that it has picked up what in the geek world passes for a brand—an acronym. It's LAMP, short for Linux, Apache

2005

Symfony 1.0

2006

PHP_CodeSniffer

2007

Zend Framework 1.0

From: php@stefan-marr.de (Stefan Marr)
Subject: RFC: Traits for PHP
Date: 2008/02/18

Hi,

during last six months I've studied a language construct called Traits.
It is a construct to allow fine-grained code reuse and in my opinon
this would be a nice feature for PHP, which I did like to propose here.
The following RFC deals with the questions what Traits are, how they are
used, why they are usefull and how they do look like in PHP.
A patch implementing this new language construct is available, too.

Thank you for your attention and I'm looking forward to hear your comments
:)

Kind Regards
Stefan

2009

PHP 5.3 = PHP 6 - Unicode

2009

PhpStorm


PRIVATE
PACKAGIST

2011

Composer

2011

Laravel 1.0


FORUMPHP
PARIS2015

From: dmitry@zend.com (Dmitry Stogov)
Subject: phpng: Refactored PHP Engine with Big Performance Improvement
Date: 2014/05/05

For people who know me it's not a secret that PHP performance is my main responsibility and passion at Zend. Actually, starting from PHP 5.0 we already made 6 times speedup on synthetic benchmarks and about 2 times speedup on real-life applications. We endlessly made improvements in PHP engine and OPCache. However, by PHP 5.5 release we weren’t be able to make any serious progress, and among other things started to experiment with memory managers, JIT technologies and other potential ideas.

I spent a significant amount of time experimenting with JIT, and even created a PoC of transparent LLVM based JIT compiler embedded into OPCache. The results on bench.php was just amazing – (0.219 seconds against 2.175 – *10 times speedup of PHP 5.5*), but on real-life apps we got just few percent speedup. This made us look much deeper into some of the runtime characteristics and what was truly the bottleneck to making more substantial progress.

It was clear the VM is already highly optimized, but works with data structures that require endless memory allocation, deallocation and reference counting. Typical real-life PHP application spends about 20% of the CPU time in memory manager, 10% doing hash tables operations, 30% in internal functions and only 30% in VM. Of course, we tried to JIT only VM code and in most cases it had to perform the same memory allocations. So we decided to change focus and work on the big bottlenecks. The idea was to change our data types to minimize heap allocations. This was a very difficult decision because we had to start with a huge refactoring, and we had no idea whether it’s going to have any impact or not.

Now I'm glad to present you a result of our recent four month work. It's a refactoring of the PHP engine that significantly improves performance, memory usage and builds a foundation for a lot more future performance improvements incl. JIT.

2012

PHP-CS-Fixer

2014

PHP 5.6

(75% faster than PHP 5.0)

2015

PHP 7

(150% - 400% faster than PHP 5.6)

2016

Composer 1.0

2016

Phan

2016

Psalm

2016

PHPStan

2017

Infection

2020

PHP 8

(Up to 20% faster than PHP 7.4)

2020

Composer 2.0

2021

The PHP Foundation

2024

Security Audit of PHP

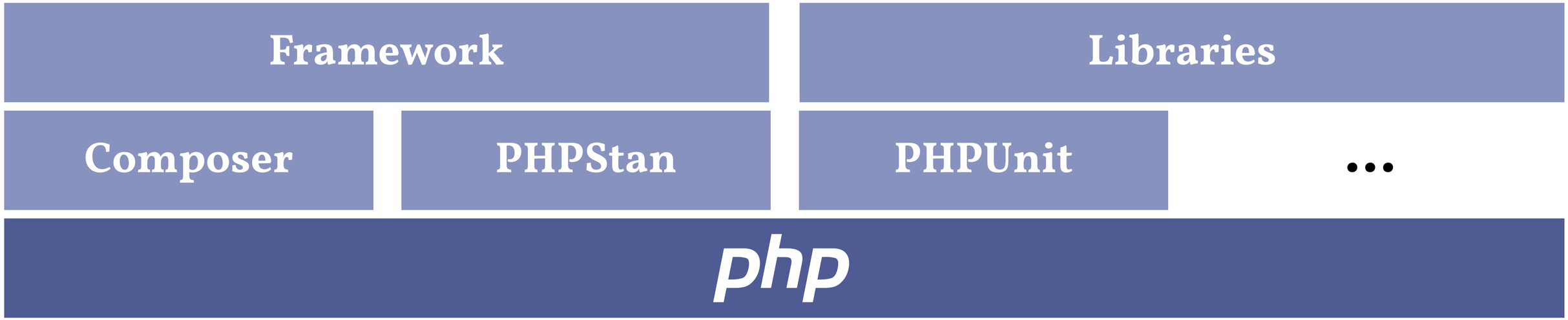
thePHP.cc

2025

PIE

Hardware
Kernel
Web Browser
Libraries
Framework
PHP Application
Framework
Libraries
Composer
PHPStan
PHPUnit
...
php
Webserver
...
libc
libxml2
OpenSSL
PCRE
...
C/C++ Compiler
Kernel
Hardware

This is (part of) your supply chain:



Are you already sponsoring it?

thePHP.cc

php

PHPUnit

thePHP.cc

This presentation has a home on the web:

thePHP.cc

Thank you!

https://phpunit.expert

sebastian@thephp.cc

@sebastian@phpc.social

thePHP.cc

Image Credits

- https://www.pexels.com/de-de/foto/menschen-silhouette-wahrend-des-sonnenuntergangs-853168
- https://www.pexels.com/de-de/foto/zeitrafferfotografie-wahrend-der-nacht-1103969
- "Day of the Tentacle", Copyright 1993 LucasArts
- https://phpics.com/picture.php?/18386
- https://www.slengpung.com/?id=3840
- https://www.flickr.com/photos/akrabat/8134304039
- Fortune, February 2004, 149(3):40-X
- https://www.youtube.com/watch?v=zekEqhaPmag