

PHPUnit: Past, Present, Future

Sebastian Bergmann



1968

1968

Rasmus Lerdorf is born

1969

Alan Kay et. al. create Smalltalk

1978

Hello world! (I am born.)

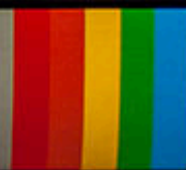
“There came a moment in your life when maybe you were in a store somewhere and you walked up to a Commodore 64 [...] and you typed a little BASIC program that would just print your name infinitely and you would go ‘YES! I am a god!’ And you wanted to be a programmer.”

— Robert C. Martin

GENERATION 64

GENERATION 64

Jimmy Wilhelmsson
Kenneth Grönwall



1986

I realize that I want to be a programmer



MATTHEW BRODERICK
as DAVID LIGHTMAN

ALLY SHEEDY
as JENNIFER KUDERHINE WICK

DANIEL COLEMAN
as MORTIMER

WAR GAMES

IS IT A GAME, OR IS IT REAL?

MGM PICTURES PRESENTS
A FILM BY JOHN DAHL
Signalnoise Skayles

WAR GAMES TM & © 1993 - 1994 MCA-MGM PICTURES INC. ALL RIGHTS RESERVED.
WRITTEN BY JAMES WHITE FOR MUSEUM, CANADA, WWW.MUSEUM.CA



1989

Kent Beck creates SUnit

1990

I get an Amiga 500



1990

I start programming in BASIC

1993

I get an Amiga 1200

1993

I start programming in C and Assembly

```

init:
    move #$ac,d7
    move #1,d6
mainloop:
wframe:
    cmp.b #$ff,$dff006
    bne wframe
    add d6,d7
    cmp #$f0,d7
    blo ok1
    neg d6
ok1:
    cmp.b #$40,d7
    bhi ok2
    neg d6
ok2:
waitras1:
    cmp.b $dff006,d7
    bne waitras1
    move.w #$fff,$dff180
waitras2:
    cmp.b $dff006,d7
    beq waitras2
    move.w #$116,$dff180
    btst #6,$bfe001
    bne mainloop
    rts

```


1994

James Gosling creates Java

1995

Rasmus Lerdorf creates PHP

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>

```
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[]) {  
    char *params, *data, *dest, *s, *tmp;  
    char *name, *age;
```

```
    puts("Content-type: text/html\r\n");  
    printf("Content-type: text/html\r\n");
```

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;

    puts("Content-type: text/html\r\n");
    puts("<HTML><HEAD><TITLE>Form Example</TITLE></HEAD>");
```

https://talks.php.net/etsy_interns19#/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/etsy_interns19#/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/etsy_interns19#/hist_1994

1997

Erich Gamma and Kent Beck create JUnit

1997

PHP/FI



1998

PHP 3

1998

I get my first X86-based PC

1998

I start programming in PHP

“[T]he Amiga's free-software community was certainly the most sophisticated and active in the world of personal computing prior to Linux. It should not be a surprise that countless Amiga users migrated to Linux and other open-source OSs as the Amiga's necessary ultimate fate became clear, for much of the spirit of the Amiga free-software community persisted in these communities without being tied to a single corporation's decisions and fate. Indeed, the Amiga's fate serves as an object lesson for the modern open-source movement, speaking to the way that even excellent hardware and software can wither when said excellence is proprietary and closed source.”

— Jimmy Maher

1999

Stig Sæther Bakken starts working on PEAR

2000

PHP 4

2000

I start contributing to PHP

```

$ git log --author="Sebastian Bergmann" --full-history --no-merges --date=relative --pretty="%cd %s"
1 year, 8 months ago  Fix typo
2 years, 2 months ago  Fix
2 years, 2 months ago  Fix
3 years, 5 months ago  Leftover from 5230541ef59e0637d5522293a7d099bf18ce6af3
3 years, 5 months ago  Tweak
3 years, 6 months ago  Use "Mac" instead of "OSX" to identify macOS in PHP_OS_FAMILY
3 years, 8 months ago  Fix bug #74409
.
.
.
9 years ago           Close #55490.
10 years ago          Add optional argument to debug_backtrace() and debug_print_backtrace() to limit the amount of stack frames returned.
10 years ago          Fix prototype of zend_fetch_debug_backtrace().
.
.
.
11 years ago          Add ReflectionMethod::setAccessible() for invoking non-public methods through the Reflection API.
.
.
.
19 years ago          Rename storage* to container*.
19 years ago          Fix ZTS build.
19 years ago          Even more TSRM cleanup.
19 years ago          More tsrm-related cleanup.
19 years ago          Fixed some TSRMLS_CC instances (at least it looked odd to me the way it was before).
19 years ago          Remove more duplicate TSRMLS_FETCH() calls.
19 years ago          Another one bites the dust.
19 years ago          Remove duplicate TSRMLS_FETCH() call.
20 years ago          Removed files from Win32 workspace that were removed in libmysql 3.23
20 years ago          Removed obsolete entries from php.ini.

$ git shortlog --author="Sebastian Bergmann" --summary --email
 15 Sebastian Bergmann <sb@sebastian-bergmann.de>
 58 Sebastian Bergmann <sbergmann@php.net>
552 Sebastian Bergmann <sebastian@php.net>

```



2000

World's ~~first~~ second PHP conference in Cologne

2000

Michael Feathers creates CppUnit

2000

I start working on PHPUnit

<http://sourceforge.net/projects/phpunit> already existed



PHPUnit is developed on `cvs.php.net`

PHP: Revision 63330

svn.php.net/viewvc?view=revision&revision=63330

php SVN SVNROOT phpdoc pear doc pecl

./[svn]

Jump to revision: 63330 Go ↺ ↻

Author: sebastian

Date: Tue Nov 27 16:37:38 2001 UTC (18 years ago)

Changed paths: 7

Log Message: Initial commit of PHPUnit.

Changed paths

Path	Details
pear/packages/PHPUnit/trunk/PHPUnit.php	added
pear/packages/PHPUnit/trunk/PHPUnit_Assert.php	added
pear/packages/PHPUnit/trunk/PHPUnit_TestCase.php	added
pear/packages/PHPUnit/trunk/PHPUnit_TestFailure.php	added
pear/packages/PHPUnit/trunk/PHPUnit_TestResult.php	added
pear/packages/PHPUnit/trunk/PHPUnit_TestSuite.php	added
pear/packages/PHPUnit/trunk/package.xml	added

Copyright © 2001-2013 The PHP Group Other PHP.net sites Privacy policy

<http://svn.php.net/viewvc?view=revision&revision=63330>



2004

PHP 5

2004

PHPUnit 2

2005

PHP 5.1

2006

PHP 5.2



2006

PHPUnit moves to svn.phpunit.de

2006

PHPUnit 3



2009

Sebastian + Arne + Stefan = thePHP.cc

sharing experience

thePHP.cc



2009

PHP 5.3 = PHP 6 - Unicode



PHP 5.3 Release Party in Munich

2009

PHPUnit moves to GitHub

2011

Nils Adermann and Jordi Boggiano create Composer

2012

PHP 5.4

2012

PHPUnit 3.7

2012

PHPUnit is installable using Composer



PHPUnit is distributed as PHP Archive (PHAR)

2013

PHP 5.5

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.

2014

PHP 5.6

2014

PHPUnit 4

2015

PHP 7

2015

PHPUnit 5

2016

Composer 1.0.0



COMPOSER V1.0.0 2016
ANNIVERSARY GOLD EDITION
COPYRIGHT © JORDI BOGGIANO
NILS ADERMAN

DISK 1: INSTALL



1.44MB
formatted for use

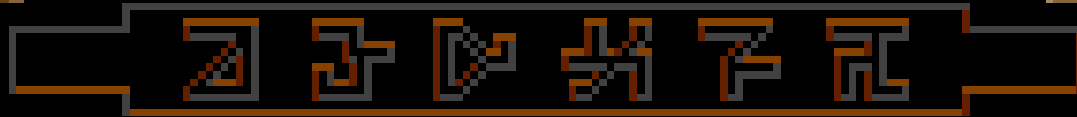
2HD 1371
format

2016

PHP 7.1

2017

PHPUnit 6



Danger Zone Master

"The dark secret of the Dungeon Master is that he knows every trap in the existing legacy software, because he was the one to leave the traps around."

– Alberto Brandolini: "The rise and fall of the Dungeon Master", 2016

I am sorry, but I became the



of PHPUnit

"The problem is in software design often the consequences of your decisions don't become apparent for years. One of the advantages of having to live with JUnit for 8 years, is now we can look back and see which decisions we made worked nicely and which we would have done differently."

– Kent Beck, 2006



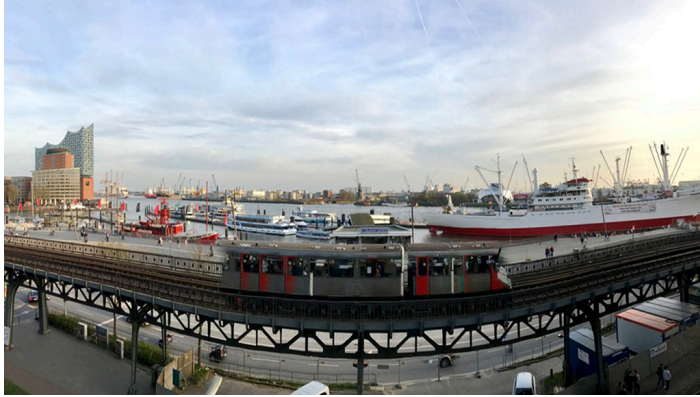
- I am sorry if a design decision I made caused problems for you

- I am sorry if a design decision I made caused problems for you
- I am sorry if correcting such a design decision caused problems for you

- I am sorry if a design decision I made caused problems for you
- I am sorry if correcting such a design decision caused problems for you
- Background on some of these decisions as well how and why they were corrected is available here: <https://thephp.cc/news/2019/02/help-my-tests-stopped-working> and <https://thephp.cc/news/2020/02/migrating-to-phpunit-9>



First PHPUnit Code Sprint



Hamburg, Germany (March 2017)



Düsseldorf, Germany (October 2017)



Hamburg, Germany (April 2018)



Munich, Germany (September 2018)



Würzburg, Germany (April 2019)



Mannheim, Germany (September 2019)

<https://phpunit.de/code-sprints/>

2017

PHP 7.2

2018

PHPUnit 7

2018

PHP 7.3

2019

PHPUnit 8



PHPUnit finally has a logo

PHPUnit

2019

PHP 7.4



2020

PHPUnit 9

2020

20 years of PHPUnit

2020

PHP 8

2020

25 years of PHP

What has changed?

What has changed?

- Performance

What has changed?

- Performance
- Type Safety

What has changed?

- Performance
- Type Safety
- Consistency

What has changed?

- Performance
- Type Safety
- Consistency
- Paradigms

What has changed?

- Performance
- Type Safety
- Consistency
- Paradigms
- Tools

What has changed?

- Performance
- Type Safety
- Consistency
- Paradigms
- Tools
- Processes

Goals for PHPUnit

Goals for PHPUnit

- Allow tests to be written more easily and clearly

Goals for PHPUnit

- Allow tests to be written more easily and clearly
- Run and evaluate tests better

PHPUnit's Release Cycle

PHPUnit's Release Cycle

- New major version (10.0, for example) in February

PHPUnit's Release Cycle

- New major version (10.0, for example) in February
- New minor version (10.1, 10.2, 10.3, 10.4, and 10.5, for example) in April, June, August, October, December

PHPUnit's Release Cycle

- New major version (10.0, for example) in February
- New minor version (10.1, 10.2, 10.3, 10.4, and 10.5, for example) in April, June, August, October, December
- Support for major version X ends two years after its initial release when major version (X+2) is released

PHPUnit's Release Cycle

- New major version (10.0, for example) in February
- New minor version (10.1, 10.2, 10.3, 10.4, and 10.5, for example) in April, June, August, October, December
- Support for major version X ends two years after its initial release when major version (X+2) is released
- Support for minor version X.Y ends when minor version X.(Y+1) is released

PHPUnit's Release Cycle

- New major version (10.0, for example) in February
- New minor version (10.1, 10.2, 10.3, 10.4, and 10.5, for example) in April, June, August, October, December
- Support for major version X ends two years after its initial release when major version (X+2) is released
- Support for minor version X.Y ends when minor version X.(Y+1) is released
- After regular support: *Life Support* to keep an otherwise unsupported version compatible with new versions of PHP

What's next for PHPUnit?

What's next for PHPUnit?

- PHPUnit 10 in February 2021

What's next for PHPUnit?

- PHPUnit 10 in February 2021
- Event System (developed by Arne Blankerts and Andreas Möller)

What's next for PHPUnit?

- PHPUnit 10 in February 2021
- Event System (developed by Arne Blankerts and Andreas Möller)
- Support for PHP 8 Attributes

What's next for PHPUnit?

- PHPUnit 10 in February 2021
- Event System (developed by Arne Blankerts and Andreas Möller)
- Support for PHP 8 Attributes
- Lots of cleanup under the hood

 [**https://thephp.cc**](https://thephp.cc)

 [**https://talks.thephp.cc**](https://talks.thephp.cc)

 [**sebastian@thephp.cc**](mailto:sebastian@thephp.cc)

 [**@s_bergmann**](https://twitter.com/s_bergmann)

sharing experience

thePHP.cc

