

PHP Doesn't Suck



Who are you?



John Hobbs

- Developer at What Cheer
- @jmhobbs
- <https://github.com/jmhobbs>
- <http://velvetcache.org/>

PHP Sucks

T_PAAMAYIM_NEKUDOTAYIM

`get_class() => gettype()`

PHPE9568F34-D428-11d2-A769-00AA001ACF42

`sort(&$array)`

www.phpsadness.com

PHP Sucks



Low Barrier To Entry
Weak Community
Google-Copy-Paste
NIH

“In the end I think you will find that your homegrown small framework has saved you time and aggravation and you end up with a better product.” - Rasmus

<http://toys.lerdorf.com/archives/38-The-no-framework-PHP-MVC-framework.html>

**I can write
terrible code in
any language.**

- Me

Making It Not Suck

Learn the Language



<http://php.net/manual>

Making It Not Suck

Test Your Code



<http://www.simpletest.org/>

<http://www.phpunit.de/>

<http://qa.php.net/write-test.php>

Making It Not Suck

NO LIVE EDITS

(mostly)



CAPISTRANO WORKS FINE WITH PHP

Making It Not Suck

Share & Be shared



Github

PSR-0 (<http://tinyurl.com/yh6jydd>)

Making It Not Suck

Have A Style



http://www.mediawiki.org/wiki/Manual:Coding_conventions

Making It Not Suck

Try The REPL

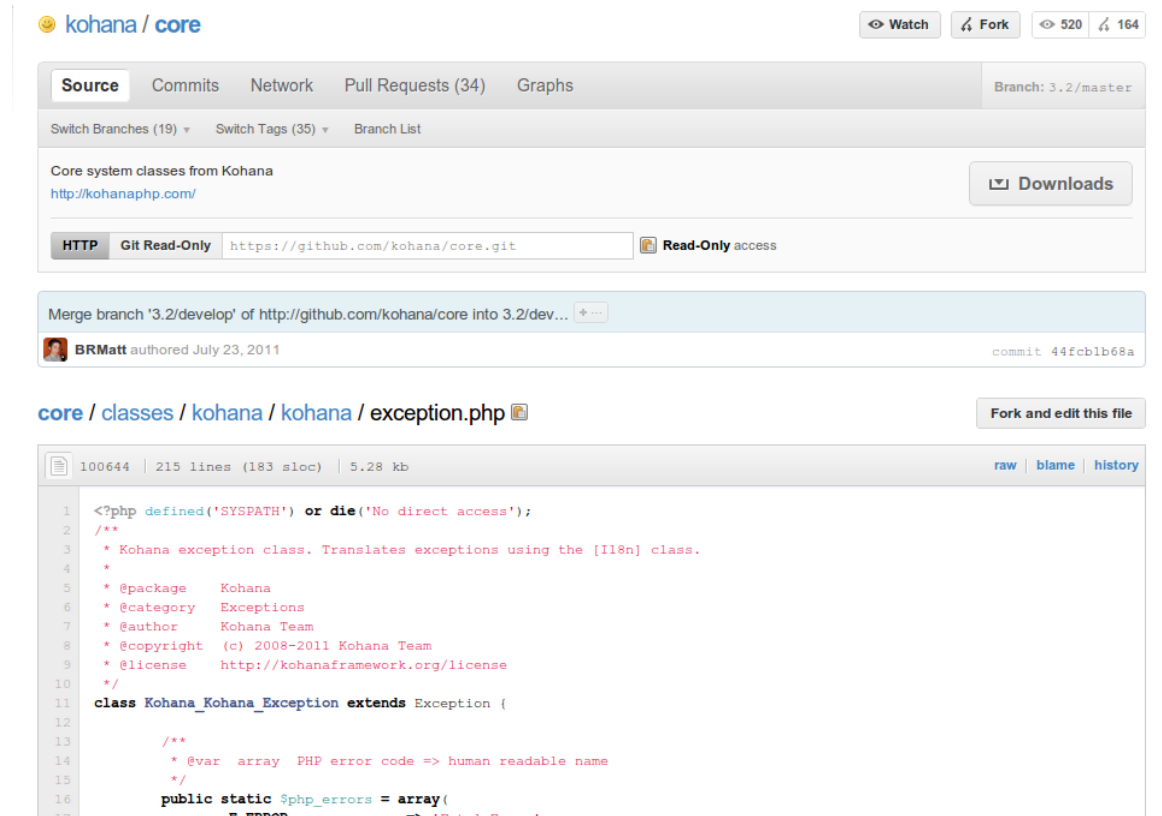
```
jmhobbs@Cordelia:~$ phpsh
Starting php
type 'h' or 'help' to see instructions & features
php> echo "Oh hai!";
Oh hai!
php> include "test.php";

php> whoa();
lol, functions!1!
php>
jmhobbs@Cordelia:~$
```

<http://www.phpsh.org/>

Making It Not Suck

Read Code (Including other languages!)



The screenshot displays the GitHub interface for the 'kohana / core' repository. At the top, it shows the repository name and navigation tabs: Source, Commits, Network, Pull Requests (34), and Graphs. The 'Source' tab is active, showing the file structure and a 'Downloads' button. Below this, there's a section for 'Core system classes from Kohana' with a link to 'http://kohanaphp.com/'. The main content area shows the source code for 'exception.php', which is a PHP file. The code includes a header with a check for 'SYSPATH' and a series of comments describing the Kohana exception class. The class 'Kohana_Kohana_Exception' extends 'Exception' and contains a static property '\$php_errors'.

kohana / core

Watch Fork 520 164

Source Commits Network Pull Requests (34) Graphs Branch: 3.2/master

Switch Branches (19) Switch Tags (35) Branch List

Core system classes from Kohana
<http://kohanaphp.com/> Downloads

HTTP Git Read-Only <https://github.com/kohana/core.git> Read-Only access

Merge branch '3.2/develop' of <http://github.com/kohana/core> into 3.2/dev... commit 44fcb1b68a

BRMatt authored July 23, 2011

core / classes / kohana / kohana / exception.php Fork and edit this file

100644 | 215 lines (183 sloc) | 5.28 kb raw blame history

```
1 <?php defined('SYSPATH') or die('No direct access');
2 /**
3  * Kohana exception class. Translates exceptions using the [1]n class.
4  *
5  * @package    Kohana
6  * @category   Exceptions
7  * @author     Kohana Team
8  * @copyright  (c) 2008-2011 Kohana Team
9  * @license    http://kohanaframework.org/license
10 */
11 class Kohana_Kohana_Exception extends Exception {
12
13     /**
14      * @var array PHP error code => human readable name
15      */
16     public static $php_errors = array(
17         E_ERROR => 'Fatal Error'
```

USE A FRAMEWORK

(Hint: Not Zend)

Use A Framework



- HMVC
- Useful helpers
- Super flexible
- Great modules system
- Great modules

<http://kohanaframework.org/>

Use A Framework



```
namespace app\controllers;

class PostsController extends \lithium\action\Controller {

    public function index() {
        return array('foo' => 'bar', 'title' => 'Posts');
    }
}
```

<http://lithify.me/>

Use A Framework

Skunk

```
// Initialize
$s = new Skunk();

// Add a route
$s->get(
    '/hi(<name>)',
    function ( &$s, $name = "world" ) {
        $s->body = "Hello $name!";
    }
);

// Run it!
$s->run();
```

<https://github.com/jmhobbs/Skunk>

So...

It's Getting Better

5.4

- <http://www.php.net/manual/en/language.oop5.traits.php>
(Traits => Mixins)
- <http://svn.php.net/viewvc?view=revision&revision=313641>
(Short Array Syntax: ["a" => "b"])
- <http://news.php.net/php.internals/55293>
(SVN => git)

Discussion

Questions?

Comments?

PHP Doesn't Suck



This is a presentation about PHP.

If you are looking for a fight, leave now.

Who are you?



John Hobbs

- Developer at What Cheer
- @jmhobbs
- <https://github.com/jmhobbs>
- <http://velvetcache.org/>

PHP Doesn't Suck

I'm John Hobbs.

I make fancy websites at What Cheer.

I have a twitter and a github and I blog but no one reads it.

If I'm known for anything on the Internet it's Beef Taco.

Just google Beef Taco and it's the top item.

I kid you not.

PHP Sucks

T_PAAMAYIM_NEKUDOTAYIM

`get_class() => gettype()`

PHPE9568F34-D428-11d2-A769-00AA001ACF42

`sort(&$array)`

www.phpsadness.com

PHP Doesn't Suck

PHP Sucks.

The language itself has some fundamental flaws that crept in early in it's life and have not been excised yet, mostly for compatibility issues.

Crappy error messages.

Inconsistent naming conventions.

Stupid “features”

In place array sorts.

There is a decent chunk of these, phpsadness.com is a good place to read and track them.

PHP has a weird history of full rewrites, but it's held onto backwards compatibility through a lot of them.

Zend (4/5) is a decent engine, but the old cruft is lame.

PHP Sucks



Low Barrier To Entry
Weak Community
Google-Copy-Paste
NIH

"In the end I think you will find that your homegrown small framework has saved you time and aggravation and you end up with a better product." - Rasmus

<http://toys.lerdorf.com/archives/38-The-no-framework-PHP-MVC-framework.html>

PHP Doesn't Suck

Why else does PHP suck?

Well, it runs everywhere. It's billed as an easy language to learn, fast. And it is, if you want to write crappy code.

The community is, compared to Python and Ruby, fragmented and not sophisticated. Testing is a big problem here. Ever used a phpt?

Because PHP is everywhere, you can Google for anything and get some code. Too often people don't read it, they just copy-paste. So they lose a chance to learn and also integrate crappy code.

There also seems to be a serious not-invented-here syndrome in PHP. This includes Rasmus. Don't listen to Rasmus.

**I can write
terrible code in
any language.**

- Me

PHP Doesn't Suck

I can write terrible code in any language.

The point is that the quality of the code has a lot to do with who is writing it, and how much they understand.

Don't blame everything on the language.

So how can we write better PHP?

Making It Not Suck

Learn the Language



<http://php.net/manual>

PHP Doesn't Suck

Learn the language.

It sounds simple, but I bet there is lots of stuff that even good PHP programmers don't know about PHP.

Has anyone here ever written a `__clone` method?

How much of the manual have you read?

Have you ever compiled an extension? Written one?

Making It Not Suck

Test Your Code



<http://www.simpletest.org/>
<http://www.phpunit.de/>
<http://qa.php.net/write-test.php>

PHP Doesn't Suck

Test your code.

This is a huuuuge deal. There are good testing tools for PHP, use them. I like SimpleTest, but it's a personal choice, and an easy one.

TDD makes you think, something PHP doesn't always make you do.

Something else that unit testing does is force you to separate code. PHP is it's own template language, so it's super easy to mix business logic with presentation. Don't. Testing will help.

Making It Not Suck

NO LIVE EDITS

(mostly)



CAPISTRANO WORKS FINE WITH PHP

PHP Doesn't Suck

No live edits. Stay off of production.

I admit, I don't use version control on some stuff. One off tools and pages, wordpress blogs, stuff that doesn't really matter.

If you are writing an app or a library, you should be using version control and running deployments after all your tests pass.

Capistrano gets along fine with PHP, and if you would rather you can use phing and other tools.

PHP tempts you to edit live in production. Resist.

Making It Not Suck

Share & Be shared



Github

PSR-0 (<http://tinyurl.com/yh6jydd>)

PHP Doesn't Suck

Share your code, and use shared code.

You don't have to come up with your own version of a lot of things.

But don't just blindly copy code or use any old module.

Look at the code and make a judgment call. It's okay to fork or rewrite if it's in bad shape, but then share it back to the community!

If you are writing a library, be compatible. If it's generic, there are naming conventions like PSR-0.

(I admit I am bad at being compatible)

Making It Not Suck

Have A Style



Have a style you stick to.

Being clear and consistent in your code is essential, no sloppy formatting.

Bad style is endemic of google-copy-paste code.

If you are working on someone else's project, use their style before you make a pull request.

It's polite, it's clean it's essential.

I use a variant of MediaWiki's conventions.

Making It Not Suck

Try The REPL

```
jmhobbs@Cordelia:~$ phpsh
Starting php
type 'h' or 'help' to see instructions & features
php> echo "Oh hai!";
Oh hai!
php> include "test.php";

php> whoa();
lol, functions!!
php>
jmhobbs@Cordelia:~$
```

<http://www.phpsh.org/>

PHP Doesn't Suck

Other languages have a great REPL.

PHP, not so much.

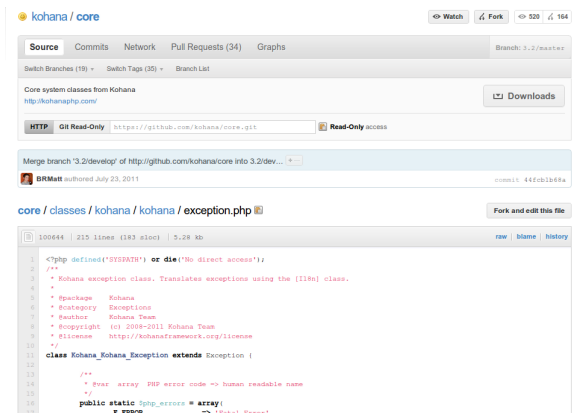
Luckily we have a good one from Facebook, written mostly in Python actually.

Get it and give it a shot.

When you forget argument ordering it's often easier to try it in the shell than to go look it up on php.net.

Making It Not Suck

Read Code (Including other languages!)



PHP Doesn't Suck

Actually **READ** the code you use.

This one is a big one. Read other people's code, the good and the bad.

Identify what you like and what you don't like about other people's code.

Try out new things you find in other peoples code.

And don't be *just* a PHP programmer.

Read and write some other languages, Python, Ruby, whatever.

I am convinced the more well read you are, in any language, the better you will be as a programmer.

USE A FRAMEWORK

(Hint: Not Zend)

PHP Doesn't Suck

Use a framework.

A framework will make you organize your code within a set of conventions. Which conventions don't matter as much as the fact that they exist.

I recommend not Zend because it frightens me.

Learn your framework well.

Read it's core, submit patches, be involved.

Here are few examples of frameworks:

Use A Framework



- HMVC
- Useful helpers
- Super flexible
- Great modules system
- Great modules

<http://kohanaframework.org/>

PHP Doesn't Suck

I'm a Kohana user.

It's not all roses though, there are no migrations either, which is gross.

Development moves fast, and breaks api a lot. It's managed with versioning though, and there is plenty of backporting and bug fixes.

Use A Framework



```
namespace app\controllers;  
  
class PostsController extends \lithium\action\Controller {  
    public function index() {  
        return array('foo' => 'bar', 'title' => 'Posts');  
    }  
}
```

<http://lithify.me/>

PHP Doesn't Suck

Another cool one is Lithium.

It's very “leading edge” and uses namespaces like they are going out of style.

Use A Framework

Skunk

```
// Initialize
$s = new Skunk();

// Add a route
$s->get(
    '/hi(<name>)',
    function ( &$s, $name = "world" ) {
        $s->body = "Hello $name!";
    }
);

// Run it!
$s->run();
```

<https://github.com/jmhobbs/Skunk>

PHP Doesn't Suck

Finally I included a Sinatra and Bottle style framework that I wrote.

I wouldn't necessarily *recommend* using Skunk, but it's a good example a little, single file framework with an interesting API and feel.

So...

PHP Doesn't Suck

So.

Those are my thoughts on how to write better PHP.

As for the language itself...

It's Getting Better

5.4

- <http://www.php.net/manual/en/language.oop5.traits.php>
(Traits => Mixins)
- <http://svn.php.net/viewvc?view=revision&revision=313641>
(Short Array Syntax: ["a" => "b"])
- <http://news.php.net/php.internals/55293>
(SVN => git)

PHP Doesn't Suck

It's getting better.

5.4 is on it's way and it's bringing some fixes for major pain points.

Discussion

Questions?
Comments?

PHP Doesn't Suck

Please be gentle.