

PHP 6

Beliebte Webskriptsprache wird erwachsen



- Linux User Group Bern
- 14.05.2009
- René Moser <mail@renemoser.net>

Inhalt

1. Wie entstand PHP?
2. Was PHP?
3. Warum PHP?
4. Wie installiere ich PHP?
5. Wie programmiere ich in PHP?
6. Was ist gutes Programmierdesign?
7. Weiteres? Fragen?

Was ist PHP?

- Skriptsprache
- Open Source Software
- Lizenz: PHP License
- Einflüsse: Java, C, C++, Perl
- Paradigmen: funktional und objektorientiert

Wer setzt PHP ein?

- Facebook
- Wikipedia
- Yahoo!
- Youtube
- Wordpress
- Digg
- Flickr

Wo wird PHP eingesetzt?

- MediaWiki – Wiki Software
- Typo3, Joomla!, Drupal – CMS
- PhpMyAdmin – MySQL Administration
- PHPbb, vBulletin – Forum
- Wordpress, Serendipity – Blog Software
- Moodle, ILIAS– Lernplattform
- eCommerce, Magento – Onlineshop

Warum PHP?

Pro:

- Einfacher Syntax
- Schnelle Ergebnisse
- Weite Verbreitung von LAMP bei Webhostern
- Viele Applikationen in PHP
- Datenbankbindung einfach zu machen

Warum nicht PHP?

Kontra:

- Unkoordinierte Entwicklung PHP Interpreters
- Einfach, unsicheren Code zu schreiben
- Es gibt Python, Ruby, JSP...

Wofür PHP?

- Verwendet für dynamische Webseiten
- Webanwendungen
- CLI Scripting

Wie entstand PHP?

- 1995: PHP 1 von Rasmus Lerdorf
- 1998: PHP 3 von Andi Gutmanns und Zeev Suraski neu geschrieben
- 2000: PHP 4 von Zend Technologies mit Zend Engine 1
- 2004: PHP 5 mit Zend Engine 2
- Heute: PHP5 5.3rc2

PHP 4 vs PHP 5

- Bessere OOP Unterstützung:
 - Exceptions
 - Private Methoden und Attribute
- DOM
- XML
- u.a.

PHP 4 vs PHP 5

- Flickr: „Wir haben mit der Umstellung von PHP 4 auf PHP 5 die CPU-Last um 20% gesenkt.“
- PHP 4 ist „End Of Life“
- PHP 4 will niemand mehr...

Was ist neu in PHP 5.3

- Namespaces
- Verbesserter XML Support
- Goto
- Closures
- Natives PHP Archiv „Phar“

Die Zukunft: PHP 6

- Durchgehend Unicode!
- `ereg()` entfernen
- Aufräumen: Weg mit `'register_globals'`, `'magic_quotes'` und `'safe_mode'`

Wie installiere ich PHP ein?

- Für Entwicklungsumgebung:
XAMPP von <http://Apachefriends.org>:
 - Alles dabei: Apache 2, PHP 5, MySQL 5, PEAR, etc...
 - Mehrere unterschiedliche XAMPPs auf einem System
 - Starten wenn man es braucht...

Und so gehts...



Wie programmiere ich in PHP?

- Dateiextension: .php oder .phtml
- Beginn mit `<?php` (oder `<?`)
- Hervorragendes Online Manual:
<http://php.net/>

Beispiele in PHP

- Ein einfaches „Hallo Welt“
- Eine einfache Funktion
- Eine einfache Datenbankbindung
- Eine einfache Klasse

Register Globals?

- Der Grund warum PHP so beliebt und berüchtigt wurde → Ein Beispiel
- Ist seit PHP 4.2 Jahr 2002 deaktiviert

Ist PHP sicher?

- National Vulnerability Database: „Overall proportion of PHP-related vulnerabilities:
- 20% in 2004
- 28% in 2005
- 43% in 2006
- 36% in 2007
- 35% in 2008
- 36% in 2009

Ist Java sicherer?

- National Vulnerability Database: „Overall proportion of Java-related vulnerabilities:
- 2% in 2004
- 3% in 2005
- 3% in 2006
- 3% in 2007
- 3% in 2008
- 4% in 2009

Ist PHP sicher?

- Ja, aber PHP macht's einem nicht von Hause aus einfach

Professionell mit PHP entwickeln

- Was ist schlechter Code?
 - Einige Beispiele
- Was ist guter Code?

Guter Code ist...

1. Fehlerfreier Code
2. Wartbarer Code
3. Testbarer Code
4. Erweiterbarer Code
5. Dokumentierter Code

Guter Programmierstil

- DRY – Don't Repeat Yourself
- KISS – Keep It Simple and Stupid
- YAGNI – You ain't gonna need it

Fehlerfreier Code

- Was ist ein Fehler?
- Fehler in Software: Bugs!
- „Fehler: Keine Verbindung mit...“
- Exception! → So sieht das aus

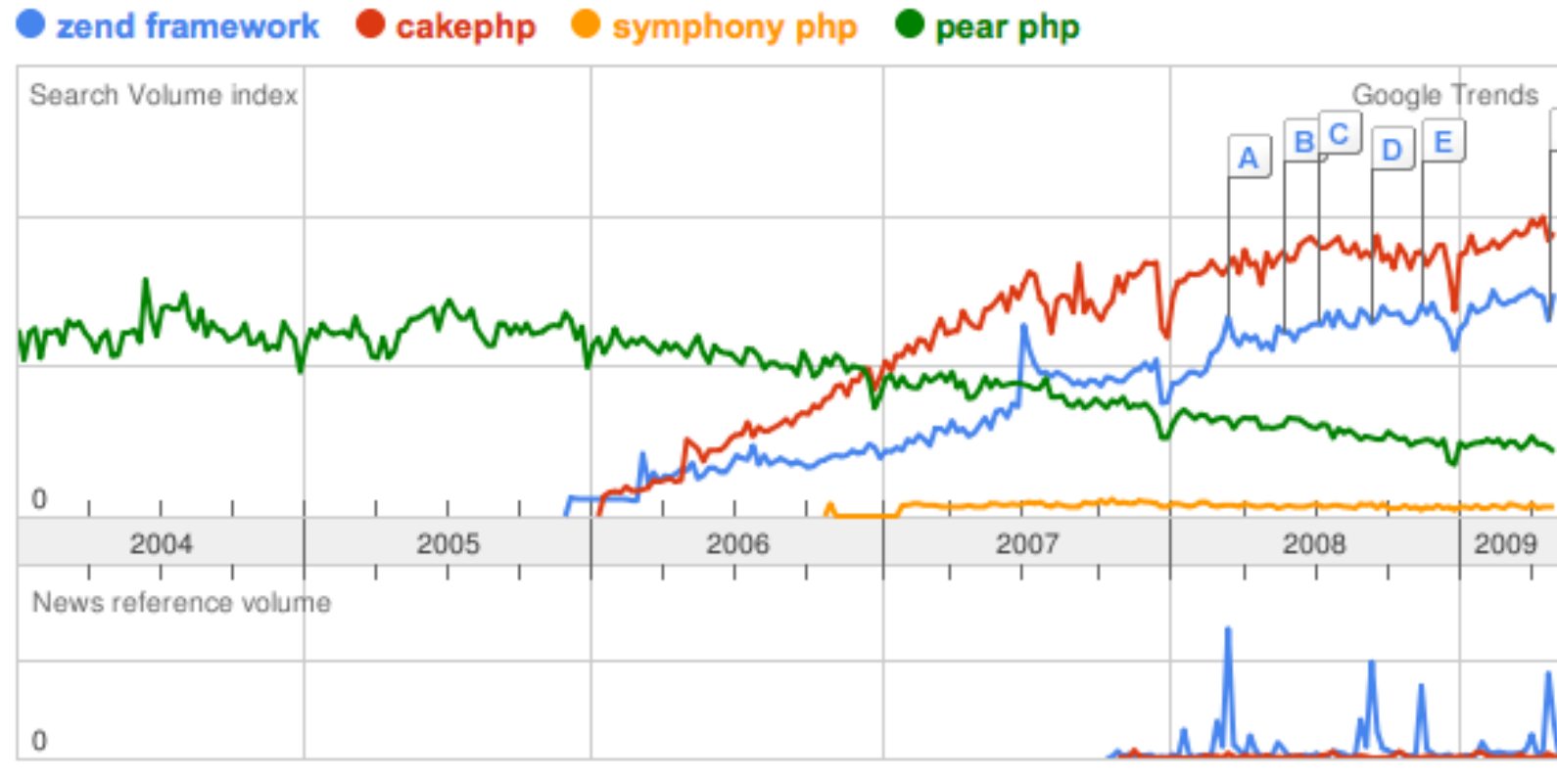
Wartbarer Code

- Code wiederverwenden
- Modularer Code → Objektorientierter Code
- Separation of Concerns →
Model View Controller MVC
- → Frameworks!

PHP Frameworks

- PEAR:
 - PHP 4 / 5, PHP Lizenz
- Zend Framework
 - PHP 5, BSD Lizenz
- CakePHP
 - PHP 4, MIT Lizenz
- Symphony
 - PHP 5, MIT Lizenz

Framework Trends



Testbarer Code

- Tests in der Realität → Auto: Bremse geht nicht..
- Programmlogik definieren
 - Formular: Nicht alle Felder ausgefüllt?
 - Keine MySQL Connection?
 - Formular: Email Syntax nicht korrekt?
- Unit Tests → PHPUnit
 - Installieren via PEAR

PHP Unit

```
<?php
require_once 'PHPUnit/Framework.php';

class ArrayTest extends PHPUnit_Framework_TestCase
{
    public function testNewArrayIsEmpty()
    {
        // Create the Array fixture.
        $fixture = array();

        // Assert that the size of the Array fixture is 0.
        $this->assertEquals(0, sizeof($fixture));
    }

    public function testArrayContainsAnElement()
    {
        // Create the Array fixture.
        $fixture = array();

        // Add an element to the Array fixture.
        $fixture[] = 'Element';

        // Assert that the size of the Array fixture is 1.
        $this->assertEquals(1, sizeof($fixture));
    }
}
?>
```

Running the Test

```
phpunit ExceptionTest
```

```
PHPUnit 3.3.0 by Sebastian Bergmann.
```

```
F
```

```
Time: 0 seconds
```

```
There was 1 failure:
```

```
1) testException(ExceptionTest)
```

```
Expected exception InvalidArgumentException
```

```
FAILURES!
```

```
Tests: 1, Assertions: 1, Failures: 1.
```

Erweiterbarer Code

- OOP
- MVC
- Framework
- Design Patterns
 - Singleton (Einzelstück)
 - Factory Method (Fabrikmethode)
 - Abstract Factory (Abstrakte Fabrik)

Design Patterns?

```
<?php
switch ($type) {
    case 'HTTP': {
        include_once 'partner_interface/http.php';
        $interface = new PartnerInterface_HTTP;
    }
    break;

    case 'SOAP': {
        include_once 'partner_interface/soap.php';
        $interface = new PartnerInterface_SOAP;
    }
    break;

    case 'XML-RPC': {
        include_once 'partner_interface/xml-rpc.php';
        $interface = new PartnerInterface_XMLRPC;
    }
    break;
}
?>
```

Abstrakte Fabrik

```
<?php  
$interface = PartnerInterface::factory($type);  
?>
```

```
<?php
abstract class PartnerInterface {
    protected function __construct() {}

    public static function factory($type) {
        $source = 'PartnerInterface/' . $type . '.php';

        if (@require_once($source)) {
            $class = 'PartnerInterface_' . $type;
            $object = new $class;

            return $object;
        } else {
            throw new Exception(
                sprintf(
                    'Konnte kein Objekt vom Typ %s erzeugen.',
                    'PartnerInterface_' . $type
                )
            );
        }
    }

    public abstract function import($data);
    public abstract function export();
}
?>
```

```
<?php
require_once 'PartnerInterface.php';

class PartnerInterface_HTTP extends PartnerInterface {
    public function import($data) {
        // ...
    }

    public function export() {
        // ...
    }
}
?>
```

Dokumentierter Code

- Anzahl Zeilen Code = Anzahl Zeilen Dokumentation
- Zeit wird an Dokumentation gespartt →
Muss einfach und zeitnah sein!
- phpDocumentor
 - Installation via PEAR

Zend Framework

- Wie benutzt man Zend Framework?

Tools?

- Was es sonst noch für Tools gibt
 - IDE: Eclipse PDT
 - Editor: JEdit, gEdit, !DreamWeaver
 - SCM: Git, SVN
 - → Vortrag über SVN
 - Phing
 - Build Tool

Links

- Projektseite von PHP: <http://php.net>
- Zend Framework: <http://framework.zend.com>
- PEAR: <http://pear.php.net>
- Eclipse PDT: <http://www.eclipse.org/pdt/>
- PHPUnit: <http://www.phpunit.de>

Fragen?

- Besten Dank für die Aufmerksamkeit!