Web development with PHP

Kore Nordmann <kore@php.net>
Tobias Schlitt <toby@php.net>
Jakob Westhoff <jakob@php.net>

08.12.2009

Speaker

- The speakers
  - Kore Nordmann <kore@php.net>
  - Tobias Schlitt <toby@php.net>
  - Jakob Westhoff <jakob@php.net>

Speaker

- The speakers
  - Kore Nordmann <kore@php.net>
  - Tobias Schlitt <toby@php.net>
  - Jakob Westhoff <jakob@php.net>
- Active in various PHP related open source projects
  - PHPUnit [37], eZ Components [6], PEAR [13], PHP [18] itself, . . .

Speaker

- The speakers
  - Kore Nordmann <kore@php.net>
  - Tobias Schlitt <toby@php.net>
  - Jakob Westhoff <jakob@php.net>
- Active in various PHP related open source projects
  - PHPUnit [37], eZ Components [6], PEAR [13], PHP [18] itself, ...
- Speaker on various webtechnology related conferences
  - International PHP Conference, WebtechCon, FrOSCon, ...

Speaker

- The speakers
  - Kore Nordmann <kore@php.net>
  - Tobias Schlitt <toby@php.net>
  - Jakob Westhoff <jakob@php.net>
- Active in various PHP related open source projects
  - PHPUnit [37], eZ Components [6], PEAR [13], PHP [18] itself, ...
- Speaker on various webtechnology related conferences
  - International PHP Conference, WebtechCon, FrOSCon, ...
- Regular authors in various magazines
  - PHP Magazine, Linux Magazine, ...

General

- Please ask!
  - Whenever a question occurs
  - How stupid you think the question might be

General

- Please ask!
  - Whenever a question occurs
  - How stupid you think the question might be
- Slides will be online later today on
  - `http://udo.edu/.../php.zip`

Who uses PHP?

- Do you?
  - What do you do with PHP?
  - Do you like / hate PHP?
  - Did you already use PHP 4 (maybe even 3)?

Who uses PHP?

- Do you?
  - What do you do with PHP?
  - Do you like / hate PHP?
  - Did you already use PHP 4 (maybe even 3)?
- In the “real” world:
  - Yahoo, Facebook, StudiVZ, Wikipedia, . . .
  - Siemens, T-Online, . . .
  - Add $bigCompany here

What is PHP?

- Most important programming language in the web
    - Third popular language (TIOBE) [40]

What is PHP?

- Most important programming language in the web
  - Third popular language (TIOBE) [40]
- Easy to learn

What is PHP?

- Most important programming language in the web
  - Third popular language (TIOBE) [40]
- Easy to learn
- Web-centric, but not web-only
  - CLI [20]
  - GTK [22]
  - Embedded [?]

What is PHP?

- Most important programming language in the web
  - Third popular language (TIOBE) [40]
- Easy to learn
- Web-centric, but not web-only
  - CLI [20]
  - GTK [22]
  - Embedded [?]
- Portable

Use Case: StudiVZ

- 550 High-End-Server
- 2,200 Processor Cores
- 7,000 GB RAM
- Peek 4,300 Mbit / sec.
- Peek 150,000 database queries / sec.
- Statistics estimated in June 2009

Open Source

- Vital Open Source community

Open Source

- Vital Open Source community
  - Professionals
    - Zend (Zend Engine 2)
    - IBM (Project Zero)
    - Yahoo! (APC)
    - eZ Systems (Xdebug)
    - ...

Open Source

- Vital Open Source community
  - Professionals
    - Zend (Zend Engine 2)
    - IBM (Project Zero)
    - Yahoo! (APC)
    - eZ Systems (Xdebug)
    - ...
  - Volunteers
    - Extension developers
    - Testers
    - Documentors (12 languages)
    - User-space libraries
    - (PEAR, Zend Framework, eZ Components,...)

What comes next?

History

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)
  - Collection of Perl scripts from Rasmus Lerdorf
  - Converted to C a bit later

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)
  - Collection of Perl scripts from Rasmus Lerdorf
  - Converted to C a bit later
- PHP 3 (PHP Hypertext Preprocessor)

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)
  - Collection of Perl scripts from Rasmus Lerdorf
  - Converted to C a bit later
- PHP 3 (PHP Hypertext Preprocessor)
  - Rewrite by Andi Gutmans and Zeev Suraski
  - Zend develops the Zend Engine 1

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)
  - Collection of Perl scripts from Rasmus Lerdorf
  - Converted to C a bit later
- PHP 3 (PHP Hypertext Preprocessor)
  - Rewrite by Andi Gutmans and Zeev Suraski
  - Zend develops the Zend Engine 1
- PHP 4

The past

- PHP/FI (Personal Homepage Tools / Form Interpreter)
  - Collection of Perl scripts from Rasmus Lerdorf
  - Converted to C a bit later
- PHP 3 (PHP Hypertext Preprocessor)
  - Rewrite by Andi Gutmans and Zeev Suraski
  - Zend develops the Zend Engine 1
- PHP 4
  - Merges parts of PHPlib (Sessions)
  - Very simple object orientation

The present

- PHP 5 / Zend Engine 2

The present

- PHP 5 / Zend Engine 2
  - “Real” object orientation
  - Important features
    - Better XML handling
    - Database abstraction layer
    - Major speed improvements

The present

- PHP 5 / Zend Engine 2
  - “Real” object orientation
  - Important features
    - Better XML handling
    - Database abstraction layer
    - Major speed improvements

- PHP 5 has actually been released four years ago.
- Support for PHP 4 has officially been dropped end 2007.

The future

- PHP 5.3
  - Namespaces [25]
  - Late static binding [24]
  - Closures / lambda functions [19]
  - Goto (sic!)

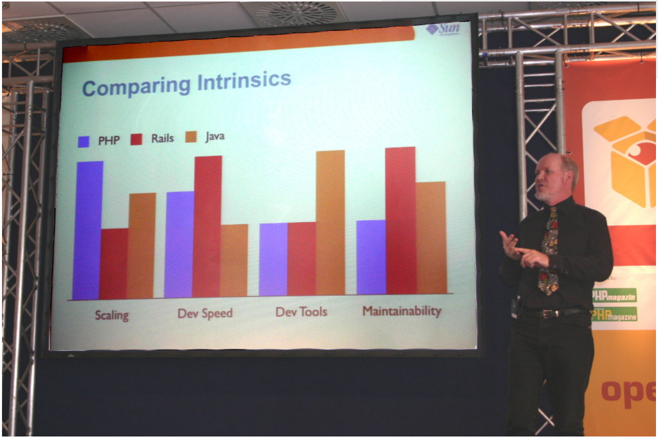
The future

- PHP 5.3
  - Namespaces [25]
  - Late static binding [24]
  - Closures / lambda functions [19]
  - Goto (sic!)
- PHP 6 / PHP 5.4 (no promises!)
  - Native Unicode support
  - Traits
  - Cleanup of deprecated features
    - Register globals
    - Safe mode
    - ...

What comes next?

Architecture

Tim Bray (Sun):


Sun
Comparing Intrinsics
PHP
Rails
Java
Scaling
Dev Speed
Dev Tools
Maintainability
PHPmagazin
PHPmagazine
ope

Language properties (1/2)

- Typing:
  - Dynamic (similar to Python, Perl)
  - Weak (similar to JavaScript, Python, Ruby)

Language properties (1/2)

- Typing:
  - Dynamic (similar to Python, Perl)
  - Weak (similar to JavaScript, Python, Ruby)
- Paradigms
  - Imperative
  - Procedural
  - Object oriented
  - Functional
  - Structured
  - (Aspect oriented)

Language properties (1/2)

- Typing:
  - Dynamic (similar to Python, Perl)
  - Weak (similar to JavaScript, Python, Ruby)
- Paradigms
  - Imperative
  - Procedural
  - Object oriented
  - Functional
  - Structured
  - (Aspect oriented)
- Freedom vs. Clarity

Language properties (2/2)

- Libraries
  - PHP uses C/C++ libraries as “extensions”

Language properties (2/2)

- Libraries
  - PHP uses C/C++ libraries as “extensions”
  - Several core extensions
    - Standard
    - Array
    - . . .

Language properties (2/2)

- Libraries
  - PHP uses C/C++ libraries as “extensions”
  - Several core extensions
    - Standard
    - Array
    - ...
  - Optional extensions shipped with the core

Language properties (2/2)

- Libraries
  - PHP uses C/C++ libraries as “extensions”
  - Several core extensions
    - Standard
    - Array
    - ...
  - Optional extensions shipped with the core
  - Additional extensions installable via PECL [16]

Share nothing

- Designed for HTTP

Share nothing

- Designed for HTTP
  - Stateless
  - Resources are freed at the end of a request
  - Environment reinitilisation

Share nothing

- Designed for HTTP
  - Stateless
  - Resources are freed at the end of a request
  - Environment reinitilisation
- No application server

Share nothing

- Designed for HTTP
  - Stateless
  - Resources are freed at the end of a request
  - Environment reinitilisation
- No application server
- Persistent states in shared memory
  - RAM
    - memcache / APC / ...
  - File systems
  - Databases

Share nothing

- Designed for HTTP
    - Stateless
    - Resources are freed at the end of a request
    - Environment reinitilisation
- No application server
- Persistent states in shared memory
    - RAM
        - memcache / APC / ...
    - File systems
    - Databases
- Threadsafe environment
    - No threading in language

Installation

- Normally used inside a web server
  - Apache [1], Lighttpd [11], nginx, IIS, . . .
  - Standalone applications possible

Installation

- Normally used inside a web server
  - Apache [1], Lighttpd [11], nginx, IIS, . . .
  - Standalone applications possible
- Interfacing through:
  - mod_php
  - FCGI (clustering)
  - CGI

Installation

- Normally used inside a web server
  - Apache [1], Lighttpd [11], nginx, IIS, ...
  - Standalone applications possible
- Interfacing through:
  - mod_php
  - FCGI (clustering)
  - CGI
- `http://www.php.net/manual/en/install.php`

PEAR Installer

- Unified installation infrastructure for PHP [13]

PEAR Installer

- Unified installation infrastructure for PHP [13]
  - Manage
    - PHP code packages
    - C-Extensions to PHP

PEAR Installer

- Unified installation infrastructure for PHP [13]
  - Manage
    - PHP code packages
    - C-Extensions to PHP
  - Channel support
    - Provide your own code via PEAR Installer

PEAR Installer

- Unified installation infrastructure for PHP [13]
  - Manage
    - PHP code packages
    - C-Extensions to PHP
  - Channel support
    - Provide your own code via PEAR Installer
  - Application setup facilities
    - Pre- / post install scripts
    - Shipped with PHP since 4.3.0

Documentation

- http://docs.php.net/(extension—function—class)
  - Good coverage of core functions / classes
  - English version very up to date
  - Translations sometimes outdated

Documentation

- http://docs.php.net/(extension—function—class)
  - Good coverage of core functions / classes
  - English version very up to date
  - Translations sometimes outdated
- Books
  - Many many books...
  - ... including quite some bad ones.

Documentation

- http://docs.php.net/(extension—function—class)
  - Good coverage of core functions / classes
  - English version very up to date
  - Translations sometimes outdated
- Books
  - Many many books…
  - … including quite some bad ones.
- Magazines
  - PHP Magazin (`http://it-republik.de/php/`)
  - php—architect (`http://www.phparch.com/`)
  - …

What comes next?

Request handling

Request handling

- Request data is available in *superglobal* arrays [28]
  - `$_GET`

Request handling

- Request data is available in *superglobal* arrays [28]
  - $_GET
  - $_POST

Request handling

- Request data is available in *superglobal* arrays [28]
  - `$_GET`
  - `$_POST`
  - `$_COOKIE`

Request handling

- Request data is available in *superglobal* arrays [28]
  - $_GET
  - $_POST
  - $_COOKIE
  - $_FILES

Request handling

- Request data is available in *superglobal* arrays [28]
  - $_GET
  - $_POST
  - $_COOKIE
  - $_FILES
  - ($_SERVER / $_ENV)

Simple form

```
<?php
if ( ( isset( $_POST['submit'] ) ) &&
     ( $_POST['Login'] === 'Webtech' ) &&
     ( $_POST['Password'] === '2009' ) ) {
    $status = 'You are logged in now.';
} else {
    $status = 'Login failed.';
}
?>
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Simple form example</title></head>
<body>
  <h1>Simple form example</h1>
  <h2><?php echo htmlspecialchars( $status, ENT_QUOTES, 'UTF-8' ); ?></h2>
  <form method="post">
  <fieldset>
    <legend>Simple login form</legend>
    <label for="login">Username</label>
      <input type="text" id="login" name="Login" />
    <label for="password">Password</label>
      <input type="password" id="password" name="Password" />
    <button type="submit" name="submit">Login</button>
  </fieldset>
  </form>
</body>
</html>
```

Session handling

- Session data in superglobal array
    - `$_SESSION`

Session handling

- Session data in superglobal array
  - $_SESSION
- Data stored serialized in:
  - File system (standard)
  - User extendable
    - Memcache (-server)
    - Database

Session handling

- Session data in superglobal array
  - `$_SESSION`
- Data stored serialized in:
  - File system (standard)
  - User extendable
    - Memcache (-server)
    - Database
- Session-user-association:
  - Trans-ID
  - Cookie

Start a session

```php
<?php

session_start();

// Handle login
if ( ( isset( $_POST['submit'] ) ) &&
     ( $_POST['Login'] === 'Webtech' ) &&
     ( $_POST['Password'] === '2008' ) )
{
    $_SESSION['loggedin'] = true;
}

// ...

// Check if user is logged in
if ( isset( $_SESSION['loggedin'] ) &&
     ( $_SESSION['loggedin'] === true ) )
{
    echo 'You␣are␣logged␣in.';
}
else
{
    echo 'You␣are␣not␣logged␣in.';
}

?>
```

OOP Features

OOP Features

- Classes
  - Methods
  - Attributes
  - Constants
  - Constructor / Destructor
  - Scopes

OOP Features

- Classes
  - Methods
  - Attributes
  - Constants
  - Constructor / Destructor
  - Scopes
- Interfaces

OOP Features

- Classes
  - Methods
  - Attributes
  - Constants
  - Constructor / Destructor
  - Scopes
- Interfaces
- Abstract classes

OOP Features

- Classes
  - Methods
  - Attributes
  - Constants
  - Constructor / Destructor
  - Scopes
- Interfaces
- Abstract classes
- Static

OOP Features

- Classes
  - Methods
  - Attributes
  - Constants
  - Constructor / Destructor
  - Scopes
- Interfaces
- Abstract classes
- Static
- Namespaces

OOP Example

```php
<?php
namespace vorgon\poetry;

class DeepThought
{
    const answer = "42";

    protected $question;

    public function __construct()
    {
        $this->question = "The answer to Life, the universe and everything else?";
    }

    public function getResult( $input )
    {
        if ( $this->question == $input )
        {
            return self::answer;
        }
    }
}

?>
```

Specialities

- Autoloading

Specialities

- Autoloading
- Type hints

Specialities

- Autoloading
- Type hints
- Interceptors

What comes next?

Databases

Supported databases

- Mysql
- Oracle
- PostgreSQL
- Sqlite
- Firebird
- MSSQL
- DB2
- CouchDB
- ... nearly any other DBMS you can find.

PDO

- PDO = PHP Data Objects [26]

PDO

- PDO = PHP Data Objects [26]
- Object oriented interface

PDO

- PDO = PHP Data Objects [26]
- Object oriented interface
- Standardised API (kind of)
  - No SQL abstraction
  - No automatic data persistence

Database abstraction

- Abstraction layers written in PHP instead of C

Database abstraction

- Abstraction layers written in PHP instead of C
- SQL abstraction
  - eZ Database [7]
  - PEAR MDB2 [15]

Database abstraction

- Abstraction layers written in PHP instead of C
- SQL abstraction
  - eZ Database [7]
  - PEAR MDB2 [15]
- "Active record"-like implementations
  - PEAR DB_DataObject [14] (PHP 4)
  - eZ PersistentObject [8] (PHP 5)
  - Zend ActiveRecord [42]
  - Doctrine [4]

Templating

PHP is a templating language

- Original idea for PHP

PHP is a templating language

- Original idea for PHP
- Problems
  - PHP got too powerful
  - User provided templates
  - No automatic output escaping
  - Designers don't want to write real code

Template languages

- Restricted functionality

Template languages

- Restricted functionality
- Automatic output escaping

Template languages

- Restricted functionality
- Automatic output escaping
- Simple syntax

Template languages

- Restricted functionality
- Automatic output escaping
- Simple syntax
- Highly specialized template languages written in PHP
  - Smarty [39]
  - eZ Template [9]
  - PHPTAL [35]
  - . . .

What comes next?

Security

Is PHP an unsafe language?

- Security problems are not language specific

Is PHP an unsafe language?

- Security problems are not language specific
- There are bugs in all language interpreters / virtual machines

Is PHP an unsafe language?

- Security problems are not language specific
- There are bugs in all language interpreters / virtual machines
  - The web environment is an “easy” target

Is PHP an unsafe language?

- Security problems are not language specific
- There are bugs in all language interpreters / virtual machines
  - The web environment is an “easy” target
- ext/filter for general input filtering
  - Global prefiltering

Common infrastructure

Related technology (1/2)

- Bytecode caches
  - APC (open source) [2]
  - Zend Platform (commercial) [43]

Related technology (1/2)

- Bytecode caches
    - APC (open source) [2]
    - Zend Platform (commercial) [43]
- XML [29]
    - Different APIs, with different memory / read characteristics

Related technology (1/2)

- Bytecode caches
  - APC (open source) [2]
  - Zend Platform (commercial) [43]
- XML [29]
  - Different APIs, with different memory / read characteristics
- JSON [23]
  - Integrated

Related technology (1/2)

- Bytecode caches
  - APC (open source) [2]
  - Zend Platform (commercial) [43]
- XML [29]
  - Different APIs, with different memory / read characteristics
- JSON [23]
  - Integrated
- Webservices
  - XML-RPC [30]
  - SOAP [27]
  - REST

Related technology (2/2)

- Development tools
  - IDEs (Netbeans [12], Eclipse PDT [5], . . .)
  - Standalone Debuggers (XDebug [41], pecl/DTrace)
  - Profiler (XDebug [41])

Related technology (2/2)

- Development tools
  - IDEs (Netbeans [12], Eclipse PDT [5], . . .)
  - Standalone Debuggers (XDebug [41], pecl/DTrace)
  - Profiler (XDebug [41])
- Quality Assurance tools
  - Documentation generation (phpDocumentor [33])
  - Build tools (phing [17])
  - Unit testing (PHPUnit [37])
  - Acceptance testing (Selenium [38])
  - Continuous Integration (phpUnderControl [36], Arbit [3], Hudson [10])
  - Static code analysis tools (phpCodeSniffer [31], pDepend [21], phploc [34], phpcpd [32], . . .)

Related technology (2/2)

- Development tools
  - IDEs (Netbeans [12], Eclipse PDT [5], . . .)
  - Standalone Debuggers (XDebug [41], pecl/DTrace)
  - Profiler (XDebug [41])
- Quality Assurance tools
  - Documentation generation (phpDocumentor [33])
  - Build tools (phing [17])
  - Unit testing (PHPUnit [37])
  - Acceptance testing (Selenium [38])
  - Continuous Integration (phpUnderControl [36], Arbit [3], Hudson [10])
  - Static code analysis tools (phpCodeSniffer [31], pDepend [21], phploc [34], phpcpd [32], . . .)
- (Rails like) Frameworks

Conclusion

- PHP is easy and fast to hack
  - Beginners might publish (unsecure) apps
  - C-like syntax

Conclusion

- PHP is easy and fast to hack
  - Beginners might publish (unsecure) apps
  - C-like syntax
- Object orientation
  - Re-usable components
  - Good web architecture still not easy
  - PHP gives you freedom

Conclusion

- PHP is easy and fast to hack
  - Beginners might publish (unsecure) apps
  - C-like syntax
- Object orientation
  - Re-usable components
  - Good web architecture still not easy
  - PHP gives you freedom
- PHP scales
  - Development wise
  - Performance wise
  - Customer wise

Thank you for listening

- Any questions?
  - Kore Nordmann <kore@php.net> / @koredn
  - Tobias Schlitt <toby@php.net> / @tobySen
  - Jakob Westhoff <jakob@php.net> / @jakobwesthoff
- We hope you enjoyed the session

References I

[1] Apache Webserver.
`http://httpd.apache.org/`.

[2] APC.
`http://pecl.php.net/APC`.

[3] Arbit.
`http://arbitracker.org/`.

[4] Doctrine.
`http://www.doctrine-project.org/`.

[5] Eclipse PDT.
`http://www.eclipse.org/pdt/`.

[6] eZ Components.
`http://ezcomponents.org`.

[7] eZ Components Database.
`http://ezcomponents.org/s/Database`.

References II

[8] eZ Components PersistentObject.
http://ezcomponents.org/s/PersistentObject.

[9] eZ Components Template.
http://ezcomponents.org/s/Template.

[10] Hudson.
https://hudson.dev.java.net/.

[11] Lighttpd Webserver.
http://www.lighttpd.net/.

[12] Netbeans IDE.
http://netbeans.org/.

[13] PEAR - PHP Extension and Application Repository.
http://pear.php.net.

[14] PEAR DB_DataObject.
http://pear.php.net/DB_DataObject.

References III

[15] PEAR MDB2.
http://pear.php.net/MDB2.

[16] PECL - PHP Extension Code Library.
http://pecl.php.net.

[17] Phing.
http://phing.info/trac/.

[18] PHP.
http://php.net.

[19] PHP Closures / lambda functions.
http://php.net/manual/en/functions.anonymous.php.

[20] PHP Command Line Interface.
http://php.net/cli.

[21] PHP Depend.
http://pdepend.org/.

References IV

[22] PHP-GTK.
http://gtk.php.net/.

[23] PHP JSON.
http://php.net/json.

[24] PHP Late Static Binding.
http://php.net/manual/en/language.oop5.late-static-bindings.php.

[25] PHP Namespaces.
http://php.net/namespaces.

[26] PHP PDO.
http://php.net/pdo.

[27] PHP SOAP.
http://php.net/soap.

[28] PHP Superglobals.
http://php.net/manual/en/language.variables.superglobals.php.

References V

[29] PHP XML APIs.
http://php.net/manual/en/refs.xml.php.

[30] PHP XML-RPC.
http://php.net/xmlrpc.

[31] phpCodeSniffer.
http://pear.php.net/PHP_CodeSniffer.

[32] phpcpd.
http://github.com/sebastianbergmann/phpcpd/.

[33] phpDocumentor.
http://www.phpdoc.org/.

[34] phploc.
http://github.com/sebastianbergmann/phploc.

[35] PHPTAL.
http://phptal.org/.

References VI

[36] phpUnderControl.
http://phpundercontrol.org/.

[37] PHPUnit.
http://www.phpunit.de.

[38] Selenium.
http://seleniumhq.org/.

[39] Smarty.
http://www.smarty.net/.

[40] TIOBE Programming Community Index.
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

[41] Xdebug.
http://xdebug.org.

[42] Zend Framework Database.
http://framework.zend.com/manual/en/zend.db.html.

References VII

[43] Zend Platform.
http://www.zend.com/products/platform/.