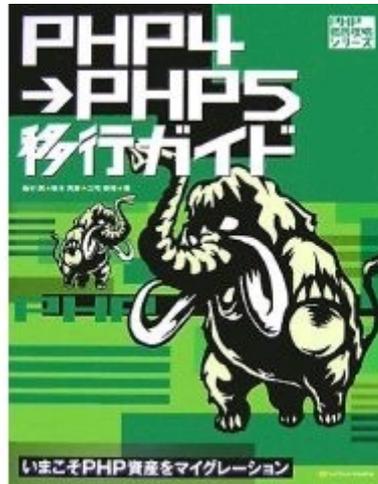


PHP4からPHP5への移行について



2008-03-06 jus勉強会

株式会社 オープンソース総合研究所

<http://www.osri.co.jp/>

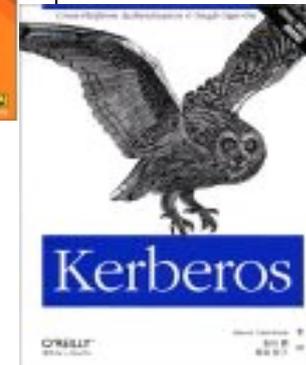
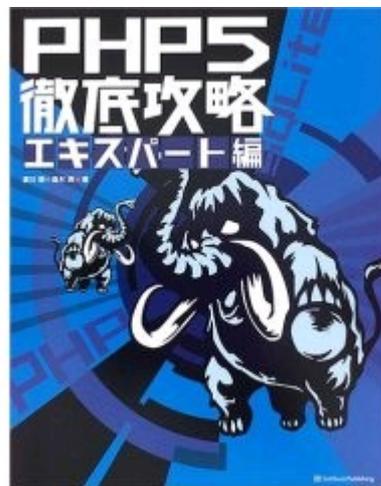


Copyright 2008

Open
Source
Research
Institute, Inc.

講演者

- 桑村 潤 KUWAMURA, Jun
 - 株式会社 オープンソース総合研究所 研究員
コンサルティンググループ



Copyright 2008

Open
Source
Research
Institute, Inc.

目次

PHPをとりまく現状
PHP5のPHP4との違い
 PHP5で改良されたこと
 PHP5の新機能
 PHP5で変更になった機能
移行におけるポイント



Copyright 2008

Open
Source
Research
Institute, Inc.

Part 1.

PHP4⇒PHP5移行

PHPをとりまく現状



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP歴史年表

	PHP/FI	PHP3	PHP4	PHP5	PHP6
1995	1.91 (PHPとFIのマージ、C言語で書き直し)				
1997/11	2.0 (各種データベースサポート)				
1998	2.0.1 (Apache DSO)				
1998/06	. +3.0 (ZendEngine 0.5)				
1999	. (Apache DSO)				
	.				
	. 3.0.7jp				
2000	.				
2000/05	. +4.0 (ZendEngine 1)				
	. 3.0.18 (OOPモデル)				
2001	.				
	. 4.0.6 (multibyte拡張)				
	. 4.1 (スーパーグローバル配列)				
2002	. v3.0.18j3				
	. 4.2 (ソケットAPI変更、Shift-JIS内部コード)				
	. 4.3 (CLI、汎用I/OストリームAPI)				
2003	.				
	v2.0.2				
2004					
2004/07	+5.0 (ZendEngine 2)				
	(OOP大幅強化、SQLite、SOAP)				
2005					
	4.4				
2006	5.1 (パーサ改良、PDO、高速化)				
	5.2 (メモリ管理改良、安定化)				
2007					
	5.2.5				
2008	. 4.4.8				
	(8月迄)				
	(5.3 namespace)				
	(6.0)				
	(ICUでUnicode/i18n強化、				
	OOP強化、namespace、				
	脆弱機能撤去、APC)				
	PHP/FI	PHP3	PHP4	PHP5	PHP6



Copyright 2008

Open
Source
Research
Institute, Inc.

PHPの歩み(1)

- 1995/10 PHP/FI 1.91
 - FI(Form Interface)にPHP(Personal Home Page Tools)をマージ
- 1995/10 PHP/FI 1.95 GDイメージ操作のサポート
- 1996/05 PHP/FI 1.99
 - Postgres95 DBMSのサポート, Apacheモジュール 1.1API
- 1997/11 PHP/FI 2.0 各種データベースサポート
- 1998/06 PHP (PHP: Hypertext Preprocessor) 3.0
- 1999/05 PHP 3.0.7jp PHP3 日本語パッチ初版
- 2000/10 PHP 3.0.18 PHP3 最終版
- 2002/03 PHP 3.0.18-i18n-ja-3 PHP3日本語パッチ最終版



Copyright 2008

Open
Source
Research
Institute, Inc.

PHPの歩み(2)

- 2000/05 PHP 4.0 Zend Engine 1、オブジェクト指向
- 2001/06 PHP 4.0.6 mbstring エクステンションバンドル
- 2001/12 PHP 4.1 スーパーグローバル(\$_GET、\$_POST、...)
- 2002/04 PHP 4.2 ソケットAPI、Shift-JIS内部コード対応
- 2002/12 PHP 4.3 CLI版、汎用I/O ストリームAPI
- 2005/07 PHP 4.4 メモリ衝突問題の修正、API 変更

- 2004/07 PHP 5.0 Zend Engine2, OOP、XMLの強化
- 2005/11 PHP 5.1 パーサエンジンの改良による高速化、PDO
- 2006/11 PHP 5.2 メモリマネージャの改良による安定化



Copyright 2008

Open
Source
Research
Institute, Inc.

時代の要求にマッチ

- ICTブーム
 - 1990年代 インターネット・WWW
 - 2000年代 Webサービス・XML
 - LAMPソリューション(Linux/Apache/MySQL/PHP)
- Webアプリ開発環境
 - スクリプティングツール
 - プログラミング言語
- 大規模サイト
 - Yahoo!(2002～)

<http://public.yahoo.com/bfrance/radwin/talks/yahoo-phpcon2002.htm>

- 楽天(2003～)

<http://itpro.nikkeibp.co.jp/members/SI/oss/20031007/1/>



Copyright 2008

Open
Source
Research
Institute, Inc.

Webアプリとして 効率のよい開発スタイル

- 言語としてとっつきやすい
- HTML中の埋め込みコード記述可能
- Perl4やCコードからの移植性が高い
- C/C++ 拡張モジュールが利用可能
- 国際化(i18n)をサポート可能
- オープンソース故の相互サポート
- スクリプト言語としても確立



Copyright 2008

Open
Source
Research
Institute, Inc.

Webサービス開発言語としてのPHP

- 開発効率
 - PEARクラスレポジトリ、PECLライブラリ
- 開発手法
 - テンプレート、MVC、フレームワーク
- 開発環境
 - ブラウザ、IDE
- 移植性
 - Java, Perl, C, C++, Fortran
- 高可用性
 - アクセラレータ、エンコーダ
- 安価な導入コスト
 - オープンソース、統合パッケージ
- 安価なトレーニングコスト
 - 手続き型言語、急峻な学習曲線



Copyright 2008

Open
Source
Research
Institute, Inc.

PHPの思想

- “物心一如の繁栄”

- 松下幸之助

- http://research.php.co.jp/about/php_right.php

- “面倒な問題を簡単に”

- 「Webアプリケーションに関する面倒な問題を、きれいに解決することは困難だ。PHPの目的は、この問題を“簡単に”解決することにある。」 -- Rasmus氏の言葉

- <http://itpro.nikkeibp.co.jp/members/SI/oss/20030926/2/>



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP3からPHP4に移行した理由

- 日本語文字列取扱の改善
 - jstring => i18n_* => mb_*
- PHP3開発終了後のサポートの問題
 - サポートはPHP4のみ
- 言語仕様改善
 - if ブロック、while ブロックなど
- 高速パーサの出現
 - Zend Engine



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP4に留まっていた理由

- オブジェクトモデルの利用
 - PEARクラス
- PECLによる拡張性
 - 既存Cライブラリの利用
- PHP5リリース後のPHP4開発の長期継続
 - PHP5の初期版が不安定
- PHP5からのバックポート



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5への移行材料

- 5.0での漠然とした不安
 - バグ多、性能劣化
 - オブジェクトモデルの違い
 - SimpleXML
- 5.1 での大幅な機能追加
 - PDO
 - 高速化
- 5.2で安定したPHP5
 - 2006/11リリースから1年以上経過
 - メモリーまわりの改善による安定
 - さらに高速化



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP4サポート終了

- PHP4のサポートは2007年12月末日で終了
 - セキュリティフィックスのリリースは2008年8月8日まで
 - PHP 4.4.8 が2008年1月3日にリリース

PHP 4 end of life announcement

[13-Jul-2007]

Today it is exactly three years ago since PHP 5 has been released. In those three years it has seen many improvements over PHP 4. PHP 5 is fast, stable & production-ready and as PHP 6 is on the way, PHP 4 will be discontinued.

The PHP development team hereby announces that support for PHP 4 will continue until the end of this year only. After 2007-12-31 there will be no more releases of PHP 4.4. We will continue to make critical security fixes available on a case-by-case basis until 2008-08-08. Please use the rest of this year to make your application suitable to run on PHP 5.

For documentation on migration for PHP 4 to PHP 5, we would like to point you to our migration guide. There is additional information available in the PHP 5.0 to PHP 5.1 and PHP 5.1 to PHP 5.2 migration guides as well.



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP4の継続利用にあたって

- 利用者減少にともなう情報の減少
 - メーリングリストでの話題性
- バグ、セキュリティの問題
 - 対応に遅れの可能性
- PHP4セキュリティサポート有り
 - 有償サポート開始 (SRAOSS 2007/12~)



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5そして、PHP6への流れ

- 名前空間(namespace)のサポート
 - PHP 5.3にバックポート予定
- PHP4レガシーコード削除
 - 非推奨のregister_globals,magic_*,safe_mode,ze1_* を削除
- Unicodeに内部対応
 - ICU(International Components for Unicode)を使用
 - 5.3では拡張モジュールの予定
- PEAR2はPHP5.2以降のみサポート
 - (2008/01リリース予定)
- Cycle collector
 - ガーベッジコレクタ自己参照改善



Copyright 2008

Open
Source
Research
Institute, Inc.

Part 2.

PHP4⇒PHP5移行

PHP5のPHP4との違い



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5のPHP4との違い

PHP5で改良されたこと



Copyright 2008

Open
Source
Research
Institute, Inc.

アプリ制作上での差異

- **メンテナンスおよびアップデート**
 - pearコマンドによりPEARのインストールや更新が簡単
 - peclコマンドによりモジュールのインストールや更新が簡単
- **パフォーマンス面**
 - PDOによりDBアクセスが高速に
 - メモリー管理の改善
 - 独自にヒープ管理を行うことで精度と速度が向上
- **セキュリティ面**
 - オブジェクト内のデータの隠蔽
 - ラッパー/フィルタによる入出力の制御



Copyright 2008

Open
Source
Research
Institute, Inc.

オブジェクトモデル実装の比較

項目	PHP4	PHP5
クラス (Class)	○	○
継承 (extends)	○	○
クラスの自動ロード	×	○ (__autoload)
コンストラクタ	○ (クラス名)	○ (__construct)
デストラクタ	×	○
アクセス制限	× (varのみ)	○ (public、private、protected)
スコープ決定演算子 (::)	○	○
staticキーワード	× (::メンバ関数)	○
クラス定数 (const)	×	○
抽象クラス (abstract)	×	○
オブジェクトインタフェース	×	○ (interfaceで定義、implementsで実装)
finalキーワード	×	○
イタレータ	×	○
例外処理	×	○
メンバのオーバーロード	×	○ (__get, __set, __isset, __unset)
メソッドオーバーロード	×	○ (__call)
シリアル化の特殊関数	○	○ (__sleep, __wakeup)
文字列出力	×	○ (__toString)
クラスや配列のエクスポート	×	○ (__set_state)
オブジェクトのコピー	×	○ (__clone)
タイプヒント	×	○
Reflection API	×	○
Standard PHP Class (SPL)	×	○
名前空間 (namespace)	×	○ (PHP5.3で予定)



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5のPHP4との違い

PHP5の新機能



Copyright 2008

Open
Source
Research
Institute, Inc.

新しいキーワード

- 追加されたキーワード

final, php_user_filter, interface, implements,
public, private, protected, abstract, clone,
exception, try, catch, throw, this

- PHP6で予定されるキーワード

namespace, import, goto



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(1)

+ 配列

array_combine() array_diff_uassoc() array_udiff() array_udiff_assoc()
array_udiff_uassoc() array_walk_recursive() array_uintersect_assoc()
array_uintersect_uassoc() array_uintersect()
array_fill_keys()

+ InterBase(FireBird)

ibase_affected_rows() ibase_backup() ibase_commit_ret()
ibase_db_info() ibase_drop_db() ibase_errcode()
ibase_free_event_handler() ibase_gen_id() ibase_maintain_db()
ibase_name_result() ibase_num_params() ibase_param_info()
ibase_restore() ibase_rollback_ret() ibase_server_info()
ibase_service_attach() ibase_service_detach()
ibase_set_event_handler() ibase_wait_event()



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(2)

+ iconv

iconv_mime_decode() iconv_mime_decode_headers() iconv_mime_encode()
iconv_strlen() iconv_strpos() iconv_strrpos() iconv_substr()

+ ストリーム

stream_copy_to_stream() stream_get_line() stream_socket_accept()
stream_socket_client() stream_socket_get_name()
stream_socket_recvfrom() stream_socket_sendto() stream_socket_server()

+ 日付・時刻関連

idate() date_sunset() date_sunrise() time_nanosleep()

+ 文字列・データ

convert_uudecode() convert_uuencode() image_type_to_extension()
str_split() strpbrk() substr_compare()



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(3)

+ pspell

pspell_config_data_dir() pspell_config_dict_dir()

+ image

imagefilter()

imagegrabscreen() imagegrabwindow()

+ mbstring

mb_list_encodings()

mb_list_encodings_alias_names() mb_list_mime_names() mb_stripos()

mb_stristr() mb_strrchr() mb_strrichr() mb_stripos() mb_strstr()

+ ming

ming_setSWFCompression() swfmovie::namedanchor() swfmovie::protect()



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(4)

+ openssl

openssl_csr_get_public_key() openssl_csr_get_subject()

openssl_pkey_get_details()

+ spl

spl_object_hash() iterator_apply()

+ pcre

pcre_last_error()

+ posix

posix_initgroups()

+ gmp

gmp_nextprime()



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(5)

+ xmlwriter

xmlwriter_full_end_element() xmlwriter_write_raw()

xmlwriter_start_dtd_entity() xmlwriter_end_dtd_entity()

xmlwriter_write_dtd_entity()

+ データベース関係

dba_key_split() dbase_get_header_info() dbx_fetch_row()

fbsql_set_password() sqlite_fetch_column_types()

+ pgsql

pg_version() pg_field_table()

+ ファイル・URL関係

file_put_contents() get_headers() headers_list() http_build_query()

setrawcookie() scandir()



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい関数(6)

+ ネットワーク関係

`curl_copy_handle()` `ftp_alloc()` `imap_getacl()` `ldap_sasl_bind()` `snmp_read_mib()`

+ 実行プロセス関係

`pcntl_getpriority()` `pcntl_wait()` `proc_nice()`

+ PHP コア

`get_declared_interfaces()` `php_check_syntax()` `php_strip_whitespace()`

`error_get_last()` `image_type_to_extension()`

`memory_get_peak_usage()` `sys_get_temp_dir()`

`timezone_abbreviations_list()` `timezone_identifiers_list()`

`timezone_name_from_abbr()` `stream_socket_shutdown()`



Copyright 2008

Open
Source
Research
Institute, Inc.

新しいクラス(1)

+ SPL

ArrayObject ArrayIterator AppendIterator
CachingIteratorCachingRecursiveIterator DirectoryIterator
RecursiveDirectoryIteratorRecursiveIteratorIterator FilterIterator
LimitIterator ParentIteratorSplFileObject RegexIterator
RecursiveRegexIterator Exception ErrorException RuntimeException ...

+ DOM

DOMAttr DOMCharacterData DOMComment DOMDocument
DOMDocumentFragmentDOMElement DOMEntityReference
DOMImplementation DOMNamedNodeMapDOMNode DOMNodeList
DOMProcessingInstruction DOMText DOMXPath

+ PDO

PDO PDOStatement

+SimpleXML

SimpleXMLElement SimpleXMLIterator



Copyright 2008

Open
Source
Research
Institute, Inc.

新しいクラス(2)

+ SOAP

SoapServer

+ Zip

ZipArchive

+ Tidy

tidyNode

+ DateTime

DateTime DateTimeZone

+ XMLReader

XMLReader

+ XMLWriter

XMLWriter

+XSL

XSLTProcessor



Copyright 2008

Open
Source
Research
Institute, Inc.

追加されたディレクティブ

- `mail.force_extra_paramaters`
- `register_long_arrays`
- `session.hash_function`
- `session.hash_bits_per_character`
- `zend.ze1_compatibility_mode`

- `allow_url_include`
- `pcre.backtrack_limit`
- `pcre.recursion_limit`
- `session.cookie_httponly`
- `max_input_nesting_level`



Copyright 2008

Open
Source
Research
Institute, Inc.

追加されたエラー報告

- **E_STRICT**

- 廃止予定の関数を使用した場合などに STRICTメッセージを発生
- 移行の際に修正箇所の指針となる

- **E_RECOVERABLE_ERROR**

- ユーザ定義のエラーハンドラで処理が可能



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい日付/時刻のサポート

- 新しい日付/時刻のサポートタイムゾーン設定は次方法
 - php.ini で date.timezone ディレクティブを設定
 - システムの環境変数 TZ を設定
 - スクリプト中で関数 date_default_timezone_set() を使用
- 新しいタイムゾーンの書式例: Asia/Tokyo
- タイムゾーン取得は次の順
 - (1) date_default_timezone_set() 関数を使用して設定したタイムゾーン
 - (2) 環境変数 TZ の値 (もし空でなければ)
 - (3) もし OS がサポートしていれば、その推測
 - (4) デフォルトは、UTC



Copyright 2008

Open
Source
Research
Institute, Inc.

新しいDBのサポート

- SQLiteライブラリのバンドル
 - SQLite2
 - PDO_SQLiteはSQLite3
- MySQLi(Improved MySQL)
 - MySQL 4.1以降に対応高速
- PDO(PHP Data Object)
 - 高速なI/O



Copyright 2008

Open
Source
Research
Institute, Inc.

新しい拡張モジュール

- **新しい拡張モジュール(5.2)**
 - Filter – データの検証とフィルタリング
 - JSON – JavaScript Object Notationデータ交換フォーマットサポート
 - Zip – ZIP 圧縮されたアーカイブ中のファイルの読み書き
- **新しい拡張モジュールのクラス定数**
 - Zend Engine 2.1 の API
 - 拡張モジュールの開発者がクラス定数を定義することが可能
 - SPL や PDO、XMLReader そして date など



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5のPHP4との違い

PHP5で変更になった機能



Copyright 2008

Open
Source
Research
Institute, Inc.

互換性のない変更点(1)

- <http://www.php.net/manual/ja/migration5.php>
- **なくなったキーワード**
 - cfunction, old_function
- **変更になった関数**
 - strrpos()とstripos()
needleとして文字列全体を使用
 - array_merge()
指定できるのは配列のみ
 - get_class(), get_parent_class(), get_class_methods()
ケース依存の名前を返す
 - ip2long()
無効なIPアドレスには FALSE を返す



Copyright 2008

Open
Source
Research
Institute, Inc.

互換性のない変更点(2)

- 挙動に関する変更

- CLI 版

- 常にグローバル変数 \$argc および \$argv が設定される

- プロパティを持たないオブジェクト

- “空”とはみなされない

- interfaces など

- classを使用する前に宣言の必要がある場合もある

- 読み込まれたファイルの中の関数

- メインファイルの中で使用可能(二度読みでエラー)

- Windows上の include_once() と require_once()

- ファイルパスを正規化

- 内部仕様の変更

- CGI版は php-cgi.exe(以前は、php.exe)

- CLI版はメインディレクトリに置かれる(以前は、cli/php.exe)

- CLI版は常にグローバル変数 \$argv と \$argc が設定される



Copyright 2008

Open
Source
Research
Institute, Inc.

互換性のない変更点(3)

•内部仕様の変更

- PHPDoc形式のコメント `/** */`
T_DOC_COMMENT と認識
- 文字列オフセットの不正な使用
E_WARNING ではなく E_ERROR を発生
- サーバー変数 PATH_TRANSLATED
Apache2 SAPI では暗黙のうちに設定されない
- 定数 T_ML_CONSTANT
Tokenizerエクステンションで 定義されなくなった
- error_reporting にE_ALL を指定した場合
PHPは notice を生成



Copyright 2008

Open
Source
Research
Institute, Inc.

拡張モジュール

- PHP コアから削除された拡張モジュールの代替

拡張モジュール	代替モジュール/開発状況
cpdf	pecl/pdflib
dbx	pecl/dbx
dio	pecl/dio
fam	-
ingres_ii	pecl/ingres
ircg	-
mcve	pecl/mcve
mnogosearch	-
oracle	ext/oci8 あるいは ext/pdo_oci
ovrimos	-
pfpro	-
w32api	pecl/ffi
yp	-
activescript	pecl/activescript
Filepro	-
Hyperwave	-



Copyright 2008

Open
Source
Research
Institute, Inc.

データベースのサポートの変更

- PDOのバンドル
 - PDO(PHP Data Object)はPHPの新共通DBドライバ
 - PECLとして開発され高速に動作する
- MySQLサポート
 - MySQLクライアントライブラリのバンドル削除
 - 従来のmysqlではMySQL <= 4.0、新しいmysqliでは MySQL >= 4.1をサポートPDO_MYSQLでMySQLの各バージョン(3,4,5)をサポート
- SQLiteサポート
 - SQLiteライブラリを標準バンドル従来のsqliteではSQLite2をサポート
 - sqlite2でSQLite2を、PDO_SQLITEでSQLite3をサポート



Copyright 2008

Open
Source
Research
Institute, Inc.

関数およびクラス

- [] の読み込
 - 引数の宣言がオブジェクトではない変数に [] を付けた引数は渡せない
- 関数のパラメータとしての整数値
 - 文字列が整数として渡された際に E_NOTICE を発行
- クラスとオブジェクト
 - is_a() は非推奨となり、instanceof 演算子に置換
 - 抽象 private メソッドは禁止
 - インターフェイス内での関数定義では public アクセス修飾子のみ



Copyright 2008

Open
Source
Research
Institute, Inc.

変更のあった関数(1)

- <http://www.php.net/manual/ja/migration5.functions.php>

+ PHP コア

`htmlentities()` - `double_encode` 追加

`htmlspecialchars()` - `double_encode` 追加

`base64_decode()` - `strict` 追加

`setcookie()` - `httponly` 追加

`setrawcookie()` - `httponly` 追加

`session_set_cookie_params()` - `httponly` 追加

`memory_get_usage()` - `real_usage` 追加

`get_loaded_extensions()` - `zend_extensions` 追加

+ curl

`curl_multi_info_read()` - `msgs_in_queue` 追加

+ datetime

`date()` - 書式文字 "u" (ミリ秒) 追加



Copyright 2008

Open
Source
Research
Institute, Inc.

変更のあった関数(2)

+ imap

imap_open() - n_retries 追加

imap_reopen() - n_retries 追加

+ mbstring

mb_strrpos() - offset 追加

+ ming

swfmovie::streamMP3() - skip 追加

+ openssl

openssl_verify() - signature_algo 追加

+ pgsql

pg_escape_bytea() - connection 追加

pg_escape_string() - connection 追加



Copyright 2008

Open
Source
Research
Institute, Inc.

Part 3.

PHP4⇒PHP5移行

PHP5移行例題



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5移行例題

オブジェクトの違い



Copyright 2008

Open
Source
Research
Institute, Inc.

オブジェクトの参照渡し

```
<?php
```

```
class myC {  
    var $c;  
    function myC ($c)  
    { $this->setC($c); }  
    function getC ()  
    { return $this->c; }  
    function setC ($c)  
    { $this->c = $c; }  
}
```

```
$c1 = new myC(1);
```

```
$c2 = $c1;
```

```
$c2->setC(2);
```

```
?>
```

結果

	PHP4	PHP5
\$c1->c	1	2
\$c2->c	2	2

対策

```
$c2 = clone $c1;
```



Copyright 2008

Open
Source
Research
Institute, Inc.

関数にオブジェクト引数渡し

```
<?php
class myC {
    var $c;

    function myC ($c)
    { $this->setC($c); }

    function setC ($c)
    { $this->c = $c; }
}

function myFunc ($f)
{ $f->setC(2); }

$c = new myC(1);

myFunc ($c) ;

echo $c->c

?>
```

結果

	PHP4	PHP5
\$c->c	1	2



Copyright 2008

Open
Source
Research
Institute, Inc.

オブジェクトの比較

- PHP 4

```
$o1=new ClassA();
```

```
$o2=new ClassA()
```

同一クラスの2つのインスタンス

```
$o2=$o1;
```

インスタンスのコピー

```
$o2=new ClassB();
```

2つの異なるクラスのインスタンス

```
$o2=&$o1
```

参照渡し

- PHP 5

```
$o1=new ClassA();
```

```
$o2=new ClassA();
```

同一クラスの2つのインスタンス

```
$o2=$o1;
```

同じインスタンスへの2つのリファレンス

```
$o2=new ClassB();
```

2つの異なるクラスのインスタンス

```
$o2=clone $o1
```

インスタンスのクローン



Copyright 2008

Open
Source
Research
Institute, Inc.

オブジェクトの比較

	<code>o1 == o2</code>	<code>o1 != o2</code>	<code>o1 === o2</code>	<code>o1 !== o2</code>
PHP4 同一クラスの2つのインスタンス	T	F	T	F
PHP4 インスタンスのコピー	T	F	T	F
PHP4 2つの異なるクラスのインスタンス	F	T	F	T
PHP4 参照渡し	T	F	T	F
PHP5 同一クラスの2つのインスタンス	T	F	F	T
PHP5 同じインスタンスへの2つのリファレンス	T	F	T	F
PHP5 2つの異なるクラスのインスタンス	F	T	F	T
PHP5 インスタンスのクローン	T	F	F	T



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5移行例題

データベースアクセス



Copyright 2008

Open
Source
Research
Institute, Inc.

プリペアドステートメント

```
<?php
$dbc = pg_connect("dbname=mydb");

// 実行クエリの準備
$result = pg_prepare($dbc,
                    "que", 'SELECT * FROM shops WHERE name = $1');

// プリペアドステートメントでは文字列 "Joe's Sample"の(')のエスケープは不要
$result = pg_execute($dbc, "que", array("Joe's Sample"));

// 同じプリペアドステートメントを別のパラメータで実行する
$result = pg_execute($dbc, "que", array("Ronny Carter"));

print_r(pg_fetch_all($result));
?>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

PDOでプリペアドステートメント

```
<?php
try {
    $db = new PDO('pgsql:dbname=test host=localhost', $user, $pass);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // プリペアドステートメントを用意
    $stmt = $db->prepare("SELECT * FROM shops WHERE name = ?");

    // プリペアドステートメントを実行
    $stmt->execute(array("Joe's Widgets"));

    while(($rows = $stmt->fetch())) {
        print_r($rows);
    }
} catch (PDOException $e) {
    print $e;
}
?>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

※ PDOコンストラクタの第一引数はDBMSにより異なるので注意が必要。
(たとえば、mysqlでは区切り文字がセミコロンとなる)

PDOでプリペアドステートメント

```
<?php
```

```
$dbc = new PDO('pgsql:dbname=mydb', 'username', 'password');  
$dbc->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
$stmt = $dbh->prepare(  
    "INSERT INTO shops (id, name) VALUES (:id, :name)");  
$stmt->bindParam('id', $id);  
$stmt->bindParam('name', $name);  
  
$id = 5;  
$name = 'Candy';  
$stmt->execute();  
  
// bindParamをしないで連想配列にして渡すことも可能  
$stmt->execute( array(':id' => 6, ':name' => 'Soyee') );
```

```
?>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5移行例題

XMLの取り扱い



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP5のXMLのサポート

- **SAXインターフェース**
 - パース処理のプログラムを記述する前に、ハンドラ関数の定義をする必要がある(PHP4互換)
- **DOMインターフェース**
 - XMLのデータツリー構造を理解できる
 - PHP4のdom_xmlとは別のモジュールで非互換
- **SimpleXML(Simple API for XML)**
 - 新機能、簡単にXMLデータを扱える
 - DOMツリーからオブジェクトを作ることも可能



Copyright 2008

Open
Source
Research
Institute, Inc.

DOMによるRSSデータ読み込み

```
<?php
$dom = new domDocument();
$dom->load('http://www.postgresql.org/news.rss');
$root = $dom->documentElement;
foreach ($root->childNodes as $elems) {
    if ($elems->nodeType == XML_ELEMENT_NODE) {
        foreach ($elems->childNodes as $elem) {
            if ($elem->nodeType == XML_ELEMENT_NODE) {
                $d = array();
                foreach ($elem->childNodes as $item) {
                    if ($schild = $item->firstChild) {
                        if ($schild->nodeType == XML_TEXT_NODE) {
                            $d[$item->nodeName] = $schild->nodeValue;
                        }
                    }
                }
                if (! empty($d)) print_r($d);
            }
        }
    }
}
?>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

RSSデータ例

<http://www.postgresql.org/news.rss>

```
<rss version="0.91" xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <channel>
    <title>PostgreSQL News</title>
    <link>http://www.postgresql.org/</link>
    <description>PostgreSQL News</description>
    <item>
      <title>AnySQL Maestro 8.2 released</title>
      <link>http://www.postgresql.org/about/news.923</link>
      <description>SQL Maestro Group announces the release of
        a freeware for PostgreSQL.</description>
      <author>lists@sqlmaestro.com</author>
      <dc:date>2008-02-13T00:00:00+00:00</dc:date>
    </item>
    <item>
      <title>OilCan 2.1 available</title>
      <link>http://www.postgresql.org/about/news.922</link>
      <description>OilCan 2.1, a free, native Cocoa and
        PostgreSQL query tool is available.</description>
      <author>david@spanware.com</author>
      <dc:date>2008-02-08T00:00:00+00:00</dc:date>
    </item>
    <item>
      <title>EMS SQL Manager for PostgreSQL 4.3</title>
      ...
    </item>
  </channel>
</rss>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

SimpleXMLによるRSS読み込み

```
<?php
define('DC', 'http://purl.org/dc/elements/1.1/');

$rss = simplexml_load_file(
    'http://www.postgresql.org/news.rss');
foreach ($rss->channel->item as $item) {
    $d['title'] = ".$item->title.";
    $d['link'] = ".$item->link.";
    $d['description'] = ".$item->description.";
    $d['dc:date'] = ".$item->children(DC)->date.";
    print_r($d);
}
?>
```

※ このRSSに含まれる日付はDublin Coerモジュールの名前空間で定義された要素で、他の要素と同じようにはアクセスできない



Copyright 2008

Open
Source
Research
Institute, Inc.

DOMからSimpleXMLに

```
<?php
define('DC', 'http://purl.org/dc/elements/1.1/');
$dom = new domDocument();
$dom->load('http://www.postgresql.org/news.rss');
$root = $dom->documentElement;

foreach ($root->childNodes as $selems) {
    if ($selems->nodeType == XML_ELEMENT_NODE) {
        $selem = simplexml_import_dom($selems);
        foreach ($selem->item as $item) {
            $d['title'] = ".$item->title.";
            $d['link'] = ".$item->link.";
            $d['description'] = ".$item->description.";
            $d['dc:date'] = ".$item->children(DC)->date.";
            print_r($d);
        }
    }
}
```



Copyright 2008

Open
Source
Research
Institute, Inc.

SimpleXMLによるパース例

```
<?php
$p = array();

// XMLファイルをパースし読み込む
$xml = simplexml_load_file( $config );

// 最初のノードの子ノードを見付けて順次値を連想配列にセット
foreach ( $xml->children() as $name => $node) {
    $value = ''. $node .'';
    $p[$name] = mb_convert_encoding( $value, 'EUC-JP');
}
?>
```



Copyright 2008

Open
Source
Research
Institute, Inc.

Part 4.

PHP4⇒PHP5移行

移行におけるポイント



Copyright 2008

Open
Source
Research
Institute, Inc.

既存のシステムをマイグレート

- 多くの部分はそのまま稼動
 - テストは重要
 - マイグレート用の環境を用意する
 - PHP4とPHP5共存環境も構築可能
- 計画的に行う
 - 作業ボリュームの洗い出し
 - 作業分担
 - ツール開発
- 移行手順を明確に
 - 段取りが大切
 - リハーサル
 - 通達を忘れない



Copyright 2008

Open
Source
Research
Institute, Inc.

プログラマの立場から

- 代入、渡しに気をつける
 - オブジェクトの場合は要注意
 - 値の場合そのままを忘れずに
- 変更のある関数にも注意
 - 使用箇所をあらかじめ抽出しておく
- 作業量の見積り
 - 修正箇所はあらかじめ洗い出す
 - 同一パターン多数の場合はツールを用意
- 動作検証
 - テストパターンはあらかじめ用意
 - 修正したら必ずテストをする
 - エンバグにも気をつける



Copyright 2008

Open
Source
Research
Institute, Inc.

システム設計者の立場から

- 設計見直しの機会
 - 次のPHP6への移行を考慮
 - 仕様変更となったモジュールに関する再設計
 - 統計情報を収集
- ハードウェア・OSにも気配り
 - リプレースの機会
 - 耐用年数に注意
 - 予測に見合った構成に変更
- 作業量と工期の兼ね合い
 - 目的を忘れず
 - 無理をしない
- 動作検証
 - テスト環境にも配慮
 - プログラマにも配慮



Copyright 2008

Open
Source
Research
Institute, Inc.

まとめ

- 無闇に怖がらない
- 基本を押さえる
- 違いを把握する
- 移行は計画的に



Copyright 2008

Open
Source
Research
Institute, Inc.

ご清聴ありがとうございました



Copyright 2008

Open
Source
Research
Institute, Inc.

PHP4⇒PHP5移行ガイド目次

目次

はじめに: PHP4とPHP5	iii
本書のテーマ	iii
本書の内容	iii
謝辞	iv
著者について	v

第1章 PHP4からPHP5への発展とその影響 1

1.1 PHP 5.0、5.1、5.2の主な特徴と変更点	2
1.1.1 PHP 5.0の特徴と変更点	2
1.1.2 PHP 5.1の特徴と変更点	3
1.1.3 PHP 5.2の特徴と変更点	3
1.1.4 PHP 5.3の予定	3
1.2 PHP 歴史年表	4
1.3 PHP4サポート終了の問題	6
1.3.1 メジャーバージョンアップの持つ意味	6
1.3.2 新機能、使いやすさ、性能の向上	7
1.3.3 バグやセキュリティホールへの迅速な対応	7
1.3.4 維持管理と将来の移行性	8
1.3.5 オブジェクト指向による品質と開発効率の向上	8
1.3.6 PHP6で予定される更新	9

第2章 開発上の相違点 11

2.1 PHP5のオブジェクト指向プログラミング	12
2.1.1 PHP4のオブジェクト指向	12
2.1.2 オブジェクト指向言語使用上の注意点	13
2.1.3 PHP5のオブジェクト指向	14
2.1.4 関数主体からクラス主体へ	16
2.2 PHP4とPHP5の機能の違い	19
2.2.1 予約語の違い	19
2.2.2 参照渡し	20
2.2.3 オブジェクトの比較	23
2.2.4 プロパティのないオブジェクト	27
2.2.5 クラスの宣言位置	27
2.2.6 クラスおよびメソッド名に関する関数	28
2.2.7 \$thisへの代入不可	29
2.2.8 剰余を求めるゼロ除りの警告	30
2.3 PHP4とPHP5の関数や挙動に関する主な違い	30
2.3.1 文字列走査のコード	30
2.3.2 文字オフセット間違いのエラーの取り扱い	31
2.3.3 array_merge()の引数	31
2.3.4 ext/mysqlからext/mysqlへの移植の際の注意	32
2.3.5 ip2long()の戻り値	32
2.3.6 ファイル読み込みのタイミング	33
2.3.7 PATH_TRANSLATED変数	33
2.3.8 コメント定数の変更	33
2.3.9 CLI版のためのvariables_orderの変更	33
2.4 PHP5での日本語環境	34
2.5 データベース	35
2.5.1 コンフィグレーション	38
2.5.2 DBMSのバージョンについての注意	39
2.5.3 PostgreSQLの関数	41
2.5.4 MySQLの関数	47
2.5.5 SQLiteの関数	55
2.6 XML	58
2.7 ストリームラッパーとストリームフィルタ	63
2.7.1 ストリーム	63
2.7.2 ストリームラッパー	65
2.7.3 ストリームラッパーの登録	68
2.7.4 ストリームフィルタ	73
2.7.5 カスタムフィルタの定義	75
2.8 SOAP、Webサービス	77
2.8.1 SOAP 拡張モジュール	78
2.9 JSON	84
2.9.1 XMLとJSON	85
2.9.2 PHPのJSON関数	85
2.10 標準ライブラリ SPL (Standard PHP Library)	87

2.11 SPLのイテレータ	89
2.11.1 PHP4で実現できなかったイテレーション	89
2.11.2 PHP5でのイテレータの実装	90
2.11.3 PHP5におけるforeach文とIteratorインタフェース	92
2.11.4 イテレータを使うメリット	92
2.11.5 SPLを用いたイテレータの実装	93
2.11.6 配列演算子を用いたオブジェクトへのアクセス	97
2.11.7 イテレータに関連するその他のインタフェース	104
2.11.8 SPLであらかじめ実装されているクラスの利用	105
2.11.9 FilterIteratorクラス	106
2.11.10 DirectoryIteratorクラスおよびRecursiveDirectoryIteratorクラス	107
2.11.11 ArrayObjectクラスおよびArrayIteratorクラス	111
2.11.12 SimpleXMLIteratorクラス	115
2.12 SPLの例外処理	117
2.12.1 実装方法の比較	117
2.12.2 SPLにおける例外処理クラス	121
2.12.3 複数の例外クラスを用いる	127
2.12.4 PHP5における例外処理の特徴	132
2.12.5 try文の処理終了時に必ず実行させたい処理	135
2.12.6 (参考) アストラクタと例外処理	142

第3章 PEARとフレームワーク 145

3.1 PEAR	146
3.1.1 PEARとは?	146
3.1.2 PEARの存在意義	146
3.1.3 PEARコマンド	146
3.1.4 PEARパッケージのインストール	150
3.1.5 PEARパッケージ	152
3.2 PEAR::DB	152
3.2.1 PEAR::DBとは?	152
3.2.2 PEAR::DBパッケージのインストール	153
3.2.3 DSN	153
3.2.4 接続の方法	155
3.2.5 クエリの実行	155
3.2.6 エラー処理	156
3.3 PEAR::Auth	160
3.3.1 PEAR::Authとは?	160
3.3.2 PEAR::Authパッケージのインストール	160
3.3.3 スレージョンテンプレート	160
3.3.4 Authパッケージの使用例	161
3.3.5 Smartyを用いたリファクタリング	162
3.3.6 PEAR::Authのその他のメソッド	163
3.4 PEAR::HTML_QuickForm	166
3.4.1 PEAR::HTML_QuickFormとは?	166
3.4.2 PEAR::HTML_QuickFormパッケージのインストール	167
3.4.3 フォームの生成	167
3.4.4 フォーム検証機能	169
3.4.5 フィルタとコールバック関数	172
3.4.6 SmartyとHTML_QuickForm	173
3.4.7 Smartyの使い方	176
3.5 PEAR::HTML_AJAX	178
3.5.1 Ajaxとは?	178
3.5.2 非同期通信の仕組み	178
3.5.3 PEAR::HTML_AJAX	179
3.5.4 PEAR::HTML_AJAXパッケージのインストール	180
3.5.5 開発時の注意点	180
3.5.6 HTML_AJAXの概要	181
3.5.7 HTML_AJAXのメソッド	183
3.6 PEAR::MDB2	199
3.6.1 PEAR::MDB2とは?	199
3.6.2 PEAR::MDB2パッケージのインストール	200
3.6.3 DSN (Data Source Name)	200
3.6.4 データベースへの接続	201
3.6.5 クエリの実行	202
3.6.6 トランザクションの利用	203
3.7 フレームワーク	205
3.7.1 フレームワーク導入のメリットとデメリット	205
3.7.2 Symfonyの特徴	207
3.7.3 SymfonyのMVCモデル	208
3.7.4 Symfonyプロジェクトの立ち上げ	209
3.7.5 ActionとView	212
3.7.6 SymfonyでHello World	215
3.7.7 SymfonyでHello World++	217
3.7.8 フォームの検証とリダイレクト処理	220
3.7.9 YAML	223
3.8 Symfony 版高書管理アプリケーションの作成	227
3.8.1 スタートアップ	227
3.8.2 ログイン画面の作成	234
3.8.3 検索機能の作成	246
3.8.4 足場(管理画面)の作成と不正遷移対策	252
3.8.5 デザインの最適化	256
3.8.6 完成!	257
3.8.7 もう一度フレームワークの意義を考える	258

第4章 システム設定と運用のポイント 259

4.1 PHP4互換モード	260
4.1.1 PHP4互換モードとは?	260
4.1.2 PHP4互換モードの利用方法	260
4.2 互換性チェックとバージョン比較	262
4.2.1 E_STRICTについて	262
4.2.2 PHPバージョンの比較	262
4.3 SuhosinによるPHPのセキュリティ強化	263
4.4 APCによる高速化	268
4.5 PHP 6.0での変更内容	270

第5章 PHP5で変更・追加された関数 273

5.1 PHP5で変更された関数	274
5.1.1 配列関数	274
5.1.2 クラス/オブジェクト関数	275
5.1.3 日付/時刻関数	276
5.1.4 dtxt関数	278
5.1.5 ディレクトリ関数	279
5.1.6 ファイルシステム関数	279
5.1.7 URL関数	283
5.1.8 InterBase関数	284
5.1.9 OpenSSL関数	284
5.1.10 プログラムの実行関数	285
5.1.11 ストリーム関数	285
5.1.12 文字列関数	285
5.1.13 マルチバイト関数	289
5.1.14 変数ための関数	289
5.1.15 その他の関数	290
5.2 PHP5で追加された関数	293
5.2.1 配列関数	293
5.2.2 クラス/オブジェクト関数	295
5.2.3 Standard PHP Library (SPL) 関連関数	295
5.2.4 日付/時刻関数	297
5.2.5 dbase関数	301
5.2.6 dBase関数	301
5.2.7 dtxt関数	301
5.2.8 fbsql関数	301
5.2.9 mysqli関数	302
5.2.10 PostgreSQL関数	302
5.2.11 SQLite関数	305
5.2.12 InterBase関数 (概略のみ)	305
5.2.13 MS SQLサーバ関数	307
5.2.14 Oracle Interface (OCI)関数	308
5.2.15 Pqsql関数	308
5.2.16 SNMP関数	308
5.2.17 ファイルシステム関数	308
5.2.18 URL関数	309
5.2.19 iconv関数	310
5.2.20 イメージ関数	312
5.2.21 IMAP関数	313
5.2.22 POSIX関数	313
5.2.23 プロセス制御関数	314
5.2.24 プログラムの実行関数	315
5.2.25 マルチバイト文字列関数	315
5.2.26 ストリーム関数	316
5.2.27 文字列関数	322
5.2.28 Perl互換正規表現 (PCRE) 関数	324
5.2.29 readline関数	324
5.2.30 inet関数	325
5.2.31 URL関数	325
5.2.32 XML関数	325
5.2.33 XSL関数	326
5.2.34 ハッシュ関数	326
5.2.35 OpenSSL関数	328
5.2.36 JSON関数	328
5.2.37 その他の関数	329
5.3 PHP5で追加された主な関数群	330
5.3.1 MySQL 拡張高書管理サポート (mysqli) 関数群	330
5.3.2 SQLite 関数群	334
5.3.3 SPL クラスメソッド	335
5.3.4 DOM (Document Object Model) XML クラスメソッド	338
5.3.5 SimpleXML クラスメソッド	342
5.3.6 XSL クラスメソッド	342
5.3.7 PDO クラスメソッド	343

付録 補足説明 345

付録 A PHP4までの足取り	346
A.1 Personal Home Page Tools 1.0からPHP/FI 2.0へ	346
A.2 PHP3の登場	346
A.3 PHP4の登場	347
A.4 PHP4の発展	348
A.5 PHP4までの国際化	349

付録 B Amazon Webサービス	355
B.1 SOAP 拡張モジュールを使ったAWSの利用	355
B.2 PEAR SOAPを使ったAWSの利用	356
B.3 RESTを使ったAWSの利用	357
B.4 AWSのWSDL ファイル	359

付録 C 高書検索アプリケーションの開発	363
C.1 ここでの開発方針	363
C.2 本アプリケーションでのMVCの割り当て	363
C.3 機能ごとの簡単な仕様	364
C.4 本アプリケーションでのディレクトリ構造	366
C.5 ユーティリティ関数	367
C.6 エラー処理	367
C.7 ModelとView	369
C.8 ログイン画面の作成	371
C.9 検索画面の生成	376
C.10 検索登録機能	385

付録 D PHP 6.0での変更内容の詳細	391
-----------------------	-----

索引	395
----	-----

コラム

csvからのチェックアウト	10
クッキーの取得	66
オブジェクトを関数するには	141
PEAR::DB	159
リファクタリングとは?	163
代入文における下位互換性	204
ORマッピングとは?	207
URLとコントロール	215
開発モードと本番モード	239
モジュールが先に実装が先か?	256
Smartyプラグイン	257
PHP4環境を残す方法	261