



PHP入門

日本PostgreSQLユーザ会(JPUG)名古屋支部
設立記念セミナー

日本PostgreSQLユーザ会

桑村 潤

<JuK@PostgreSQL.jp>

2005-07-02

紹介

- 講師: 桑村潤

(株)計算力学研究センター 技術開発部 主任、
JPUG理事、日本PHPユーザ会会員、
Plamo Linux Project WebDB担当

「PHP5徹底攻略エキスパート編」、「PHP5徹底攻略」、
「PostgreSQL for Windows 徹底活用ガイド」共著、
「KERBEROS ネットワーク認証システム」翻訳、
「例題によるLinuxプログラミング」監訳等

概要

- PHPについて
- Windows版のインストール
- PHP開発環境(IDE)
- DBへのアクセス
- XMLのパーズ
- テンプレートの利用
- セキュリティ上の注意
- キャッシュとエンコード

PHPについて

- PHP: ' PHP Hypertext Preprocesser'
 - <http://www.php.net>
 - Webサイト開発用言語 (サーバサイドスクリプト)
- 10年の歴史
 - (8.Jun.1995: comp.infosystem.www.authoring.cgi にて
Personal Home Page Tools version 1.0 公開)
- 短期開発向けの特徴

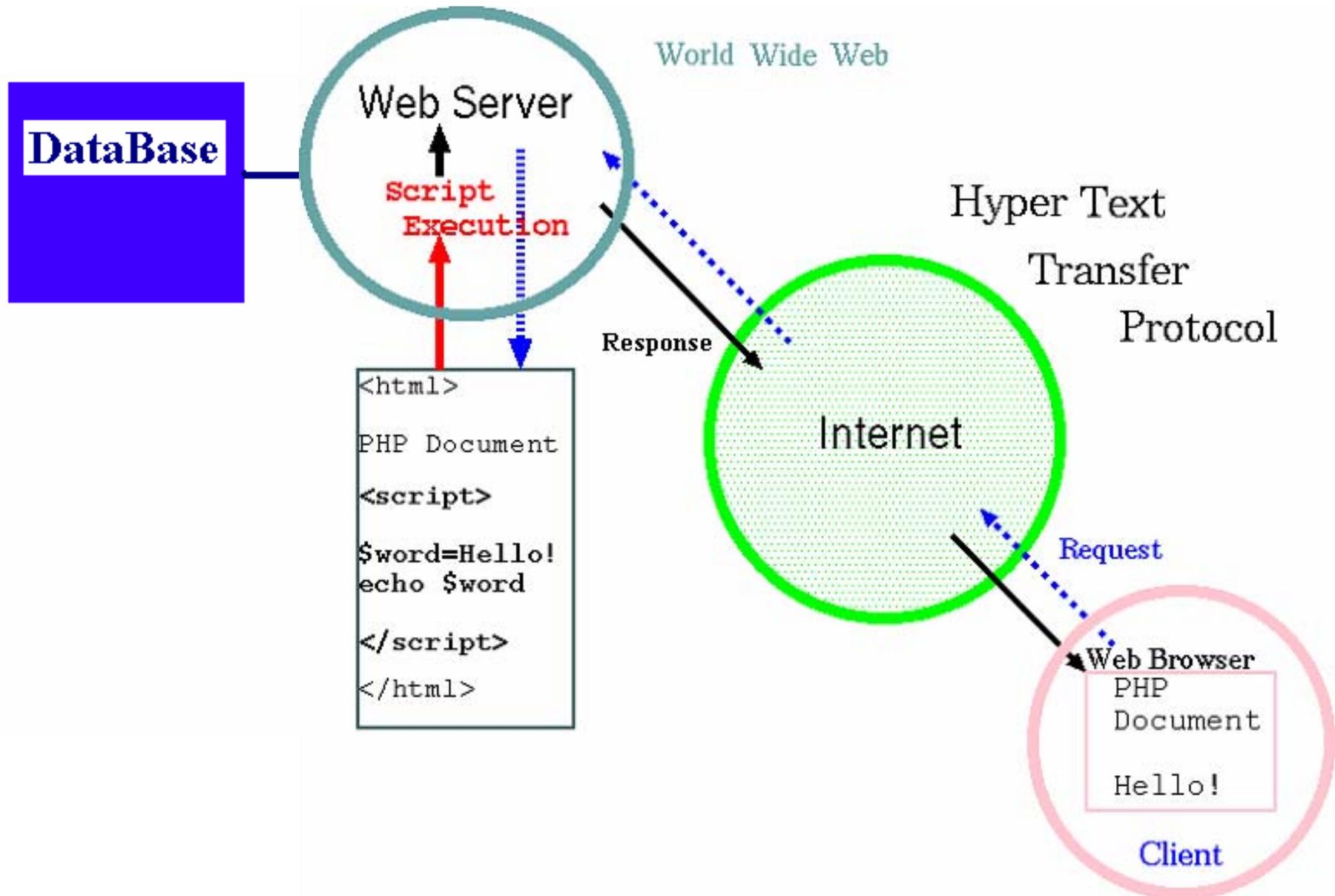
PHPの歴史

- 派生もとのPHP/FIは ‘Personal Home Page Construction Kit/Form Interpreter’ – 1995年
 - <http://www.zend.com/zend/hof/rasmus.php>
- PHP/FI-2 までは Rasmus Lerdorf 作 – 1997年
- PHP3からZeev Suraski(Zend社)らが開発 – 1997年
- PHP4はZend Engine(強力なパーサ)搭載 – 1999年
 - <http://www.zend.com>
- PHP5はZend Engine2オブジェクト指向強化 – 2004年

PHPの特徴

- サーバサイドスクリプト
 - Webサーバ上で実行。(CLI版もある)
- HTML埋め込み可能
 - HTMLの中に処理を書くことも可能
- 言語としての特徴
 - C言語、Perl言語のような記述可。オブジェクト指向化
- PHPの活用
 - PHPの特徴をWebサイト開発に活かせる

サーバサイドスクリプト



PHPの埋め込み例

PHPプログラムをHTMLに埋め込んだ例:

```
<html>
```

```
<body>
```

只今の時間:

```
<?php
```

```
    echo date( "Y-m-d(D) h:ia" , time() );
```

```
?>
```

```
<br />
```

```
</body>
```

```
</html>
```

出力:

只今の時間: 2005-06-11(Sat) 05:27pm

PHP言語の特徴

- C言語、C++言語に似た構文と関数
- Perlに似た変数や配列と関数
- オブジェクト指向プログラミング可能(クラス)
- マルチバイト文字対応
- 豊富なサポート関数群
 - DBMSインターフェース
 - イメージ処理関数(GD)
 - XML処理関数
- PEAR拡張クラス
(PEAR: PHP Extension and Application Repository)
 - DB共通インターフェース
 - 認証クラス
 - テンプレートクラス

PHPの活用

- 既存のWebページを簡単に活性化
 - 日付、ファイル更新日、条件分岐
- 既存のCGI(C, FORTRAN, Perl)を移植
 - C, FORTRAN => 変数名に \$を付ける
 - Perl => とりあえず動かしてみる
- 長いプログラムはサブルーチン化
 - Include, require 命令を使う
- 有用ルーチンはオブジェクトにして再利用
 - クラスにまとめる

インストール (Windows)

- Apach 2.0.x
- PHP 5.0.x (Apache DSO)
- PostgreSQL 8.0.x

参照: <http://www.postgresql.jp/niigata>
(ただし、Apache1.3+PHP4+PostgreSQL8 β)

Apacheのインストール

- Apacheダウンロードサイト
 - <http://www.apache.org/dist/httpd/binaries/win32/>
- Windows版Apacheインストーラパッケージ
 - apache_2.0.54-win32-x86-no_ssl.msi

※ CGI (php.exeを起動) よりもDSO (モジュール組み込み) の方がレスポンスが速い。

CGI: Common Gateway Interface

DSO: Dynamic Shared Object

PHPのインストール

- PHPダウンロードサイト
 - <http://www.php.net/downloads.php/>
 - Windows版PHPバイナリパッケージ
 - php-5.0.4-Win32.zip(PEARの必要があれば)
 - pecl-5.0.4-Win32.zip(必要があれば)
 - PHP日本語パッチダウンロードサイト
 - http://www.geocities.jp/rui_hirokawa/php/
 - Windows版PHP 日本語バイナリパッチ
 - php5.0.4-mb10a.lzh
- (マルチバイト対応、pgsql, mysql, pdo,gd2等のDLLを含む)

Apache2の設定 (DSO版PHP5)

- Apache ¥ conf ¥ httpd.conf

```
DirectoryIndex index.html index.html.var index.php index.php.var
```

```
# PHP5(DSO)  
LoadModule php5_module "C:/php5/php5apache2.dll"  
AddType application/x-httpd-php .php  
AddType application/x-httpd-php-source .phps
```

- その他、サーバ設定

- ServerAdmin, ServerName, DocumentRootなど

PHP5の設定

- Apache¥php5ts.dll

- PHP5 ¥ にパスを通す。または、php5ts.dll を Apache ¥ へコピー (あるいは、c: ¥ WINDOWS ¥ system32 ¥ へ)

- c:¥WINDOWS¥php.ini

- PHP5 ¥ php.ini-recomendedをc: ¥ WINDOWS ¥ php.iniにコピー

```
extension_dir = "c:/php5/ext"  
extension=php_pgsqll.dll
```

- そのほか、mbstring.* など適宜設定を行う

- コマンドラインからのCLI版起動テスト

- PHP4 ¥ cli ¥ php.exe -I

- PEAR

- php-5.0.4-Win32.zipを展開してできたPEARフォルダをPHP5¥PEARへ

Apache+PHP5の起動

- Path環境変数にPHP5のパスを追加
 - スタート->コントロールパネル->システム->詳細設定->環境変数

```
PHP5R=C:¥php5
PATH=%PHP5R%;%PATH%
```

- コマンドラインからApacheを起動確認

➤ Apache¥bin¥Apache.exe

➤ ^C

➤ tail Apache/log/error_log

```
[Sun Jun 12 13:22:30 2005] [notice] Apache/2.0.54 (Win32) PHP/5.0.4 configured
-- resuming normal operations
```

- PHPファイルのパーステスト

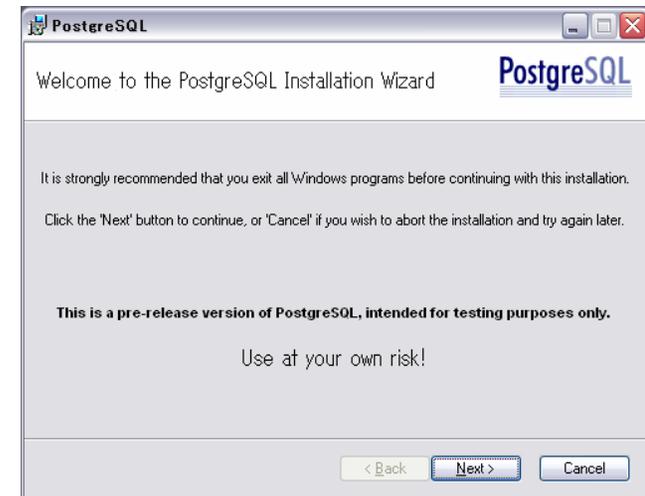
- Apache¥htdocs¥phpinfo.php を次の内容で作成

```
<?php
    phpinfo( );
?>
```

- Webブラウザにて <http://localhost/phpinfo.php> にアクセス

PostgreSQLのインストール

- pginstallerダウンロードサイト
 - <http://www.postgresql.org/ftp/binary/v8.0.3/win32/>
- Windows版インストーラ
 - [postgresql-8.0.3-ja.zip](#)
 - ダウンロードした.zipファイルを展開して.msi を取り出す
 - [postgresql-8.0-ja.msi](#)
- インストーラ(.msi)を実行
 - ダイアログに従って完了
 - メニューからpostmaster起動



*pgInstaller*に含まれるツール

- **Npgsql**
Microsoft .Netプロバイダー
(Global Assembly Cache(GAC)にインストールされる)
- **JDBC**
PostgreSQL JDBC ドライバー。
(*jdbc* ㄥサブディレクトリを *CLASSPATH* 環境変数に要追加)
- **psqlODBC**
PostgreSQL ODBCドライバ
- **pgAdmin III**
PostgreSQL管理GUI

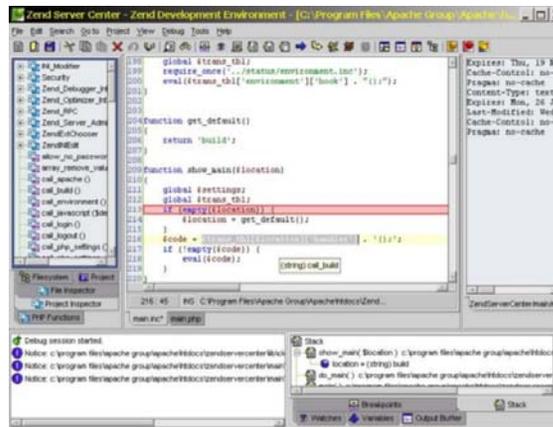
PHP開発環境 (IDE)

- コマーシャル版IDE
- オープンソース版IDE

≡ コマーシャル版IDE

- Zend Studio

- <http://www.zend.co.jp/>



Price(v.4.0):\$299
日本語版(v.3.5)
価格:45,000円
オンラインマニュアル
プロジェクト
リモートデバッグ
PHP5対応

- ActiveState Komodo

- <http://www.activestate.com/Products/Komodo/>
Price: \$295(Pro 3.0), \$29.95(Personal 3.0)

- Nusphere phpEd

- <http://www.nusphere.com/>
Price: \$299(Additional \$50)

☚ オープンソース版IDE

- Eclipse+TruStudio(本格的IDE)
 - <http://trustudio.japansite.org/>
- DBG+DDD(Unix互換GNU Debugger)
 - <http://www.gnu.org/software/ddd/>
- DBG+SE(Windows)
 - <http://dd.cron.ru/dbg/>
- PHP Edit(PHPで書かれている)
 - <http://www.waterproof.fr/>
- IDE.PHP
 - <http://www.ekenberg.se/php/ide/>
- Emacs + php-mode (筆者の環境)
HTMLテキスト編集とプログラム開発の同時進行



PHPからのDBアクセス (*PostgreSQL*の例)

- PostgreSQL関数(pg_*)
 - 処理速度が速い
 - 細かいチューニングが可能
 - 他のDBMSへの移植性は高くない
- PEARのDBクラス(DB.php)
 - コーディングが比較的簡単
 - 細かなチューニングは難しい
 - 他のDBMSへの移植性が高い

PostgreSQL関数でのアクセス

```
<?php
    $sql = "SELECT * FROM pg_am";
    $conn = pg_connect("host=localhost dbname=template1 user=postgres
password=passwd");
    if ($conn) {
        $result = pg_query($conn, $sql);
        if ($result) {
            for ($i=0; $i < pg_numrows($result); $i++) {
                $name = pg_result($result, $i, 'amname ');
                print "Name: $name<br>¥n";
            }
            pg_freeresult($result);
        } else {
            echo "問合せ失敗！ (" . pg_errormessage($conn) . ")<br>¥n";
        }
        pg_close($conn);
    } else {
        echo "接続に失敗！<br>¥n";
    }
?>
```

PEAR DBクラスでのアクセス

```
<?php
require 'DB.php';
$sql = "SELECT * FROM pg_am";
$db =
    DB::connect('pgsql://postgres:passwd@localhost/template1');
if (DB::isError($db)) {
    die ("接続失敗: ".$db->getMessage());
}
$result = $db->query($sql);
if (DB::isError($result)) {
    die ("問合せ失敗: ".$result->getMessage());
} else {
    while ($row = $result->fetchRow()) {
        print "Name: $row[0]<br>¥ n";
    }
}
$db->freeResult($result);
$db->disconnect();
?>
```

PostgreSQLのための設定

- PostgreSQLの設定(postgresql.conf)
 - max_connections = <postgres最大接続数>
 - shared_buffers = <共有メモリ上のバッファ数>
 - work_mem = <ソートに使うバッファ数>
- PHPの設定(phi.ini)
 - pgsql.max_links = <プロセス当りの最大接続数>
 - pgsql.max_persistent = <接続の内の最大持続的接続数>
- Apacheの設定(httpd.conf)
 - MaxClients = <httpd子プロセスの最大数>
- 注意点
 - max_connections > pgsql.max_links * MaxClients
 - shared_buffers > max_connections * 2 (8KB)



文字(文字列)を扱うための PostgreSQL関数

- PostgreSQLクライアントのエンコーディングを取得

```
string pg_client_encoding ( [resource connection])
```

- PostgreSQLクライアントのエンコーディングを設定

```
int pg_set_client_encoding ([resource connection, string encoding])
```

- テキスト/文字型用の文字列をエスケープする

```
string pg_escape_string ( string data)
```

三 日本語文字コードの扱い

- 処理の箇所により文字コードが異なる可能性
 - サーバが送り出す(ブラウザが表示できる)文字コード
 - ブラウザがサーバへ送る文字コード
 - PHPプログラムが書かれた文字コード
 - サーバのPHPモジュールが処理に使う文字コード
 - データベースとのインターフェースで使う文字コード
- mbstring機能の設定(`php.ini`)
 - `mbstring.language = Japanese`
 - `mbstring.internal_encoding = EUC_JP`
 - `mbstring.http_input = auto`
 - `mbstring.http_output = SJIS`
 - `output_handler = mb_output_handler`

XMLの取り扱い

- XMLの普及
 - 業務でもXMLの電子データは当たり前
- PHPでのXML
 - Expat, XSLT, DOMXMLなどの関数群
- SimpleXMLによるXMLのパーース
 - 最も単純なXMLパーサー

PHPのXML関連の関数群

- Expat XMLパーサ
 - 最も単純なパーサにはexpatを使用
 - <http://sourceforge.net/projects/expat/>
 - パーサの定義が必要
- XSL(Extensible Stylesheet Language, XSL Transformations)
 - XMLドキュメントを他のXMLドキュメントに変換する言語
 - GNOME XSLTを使用(旧Sabrotron版XSLTはPECLへ)
 - <http://www.xmlsoft.org/>
- SimpleXML
 - XMLを簡単にオブジェクトに変換
 - DOMとのインターフェースもあり
- DOM (旧DOMXMLはPECLへ)
 - DOM(Document Object Model)APIでXMLドキュメント処理
 - GNOME XMLライブラリを使用
 - <http://www.xmlsoft.org/>

SimpleXMLによるパース例

```
<?php
// SimpleXML を使った XML ファイル読み込み
function read_config($config)
{
    $param=array();
    // XMLファイルのパースし読み込む
    $conf = 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 ( $conf->children() as $name => $node) {
        $value = ''. $node.'';
        // EUC-JP に変換
        $code = mb_detect_encoding($value, "auto");
        $param[$name] = mb_convert_encoding( $value, 'EUC-JP', $code);
    }
    return $param;
}

$param = read_config("param.xml");
echo "<pre>"; var_export($param); echo "</pre>";
?>
```

≡ テンプレートの利用

- 表示デザインとプログラムロジックを分離
- 簡易的にはEval()関数で実現可能

```
<?php
    $contents = file("テンプレートファイル名");
    eval("echo ¥"$contents¥";");
?>
```

- テンプレートのクラスの利用
 - Smarty
 - <http://smarty.php.net>
 - PEAR HTML_Template_IT (Integrated Template)
 - pear install HTML_Template_IT

PEAR Integrated Templateの例

- HTML_Template_IT を使うPHPコード例

```
<?php
require_once 'HTML/Template/IT.php'
$tpl = new HTML_Template_IT(.);
$tpl->loadTemplatefile('テンプレートファイル', true, true);
$tpl->setVariable('NOW' , date('Y-m-d(D) h:i'));
$tpl->show();
?>
```

- テンプレートファイルの例

```
<html>
<body>
<table>
<tr><th>ただいまの時刻</th><td> {NOW} </td></tr>
</table>
</body>
</html>
```

≡ Webプログラムのセキュリティ対策

- クロスサイトスクリプティング対策
 - 不必要なHTMLタグの入力をさせない
 - htmlspecialchars()で特殊文字を無効に
 - Strip_tags()でタグの削除
- SQLインジェクション対策
 - SQL文中にコマンドを挿入させない
 - addslashes()で「'」や「;」をエスケープする
- <FORM>からの変数や、クッキーの値をチェック
 - register_globals = off (php.ini)
 - 入力変数(\$_POST, \$_GET, \$_COOKIE)をチェックしてから使う

≡ 商用配布ソフトの開発

- キャッシュモジュール
 - 高速化
- エンコーダー、コンパイラー
 - ソースを中間コードにして見えなくする
- APC1.0キャッシュの例(参考)

商用エンコード/キャッシュ モジュール

- Zend Encoder 3.0

- <http://www.zend.com/store/products/zend-encoder.php>

Perpetual (lifetime) License \$2400

日本語版価格: 400,000円

- Zend Accelerator 2.0

- <http://www.zend.com/store/products/zend-accelerator.php>

日本語版価格: 300,000円

- PHTML Encoder

- <http://www.rsoftlab.com/phpenc.php4>

Windows+Linux(Source):\$550

日本語は？

- IonCube Encoder

- <http://www.ioncube.com/encoder/>

Linux, FreeBSD or Windows \$349

日本語版価格: 59,800円

フリーのエンコード/ キャッシュモジュール

- PECL APC(Alternative PHP Cache)2.0
 - <http://pecl.php.net/apc>
(PHP License)
- eAccelerator(Turck MMCache)
 - <http://eaccelerator.net/HomeUK/>
(GPL)
- PHP Accelerator(ionCube)
 - <http://www.php-accelerator.co.uk/>
 - <http://asial.co.jp/products/ioncube/>
バイナリのみフリー
(性能は良いらしくyahoo.comで使われている)
- PEAR bcompiler
 - <http://pear.php.net/manual/en/pecl.bcompiler.php>
エンコードのみ

APC1.0使用時のアクセステスト (参考)

- **Apache benchmark(ab)**

- 同時に 1000 リクエストを 1000 回行なうテスト
`ab -n 1000 -c 1000 http://.../imagelist.php`

簡単なイメージ表示を含むコンテンツ、
APCを使わない状態(`apc.mode=off`)で17〜20秒
それぞれ5回くらいずつ計測(おおざっぱ)

<code>apc.mode</code>	Time(sec)
off	17〜20
mmap	13〜17
shm	8〜12 (オブジェクトファイル使用)

まとめ

- PHPで簡単にダイナミックなページ
- 好みの開発環境
- 開発効率化のためにクラスを利用
- セキュリティの考慮
- 商用アプリケーションのための工夫
- 日本PostgreSQLユーザ会でも使用

≡ 備考

- PHPのその他の話題
 - 認証システムの利用
 - セッション管理
 - 効率的なデバッグ
- PHPによる開発事例
 - Yahoo サイト
 - ショッピングサイト(楽○)
 - ソーシャルネットワーキングサイト GREE.JP
 - 大学業務システム(早稲○)
 - ブログサイト ブログ人
 - ブログツールNucleus CMS
 - CMS Drupal
 - グループウェア La'cooda Wiz
 - Wiki Pukiwiki
 - DB管理ツール phpPgAdmin

参考文献

- 「PostgreSQL徹底活用ガイドfor Windows」
インプレス、齊藤・石川・桑村・大谷・多度・望月 著、ISBN4-8443-2099-8
- 「PHP x PostgreSQL で作る最強Webシステム」
技術評論社、石井達夫著、ISBN4-7741-1647-5
- 「PHP4徹底攻略 改訂版」
ソフトバンクパブリッシング、堀田・石井・廣川 著、ISBN4-7973-2097-4
- 「PHP4徹底攻略 実戦編」
ソフトバンクパブリッシング、廣川・桑村・小山 著、ISBN4-7973-1519-9
- 「はじめてのPHP言語プログラミング入門」
技術評論社、大垣靖男 著、ISBN4-7741-2286-6
- 「PHP5徹底攻略 エキスパート編」
ソフトバンクパブリッシング、廣川・桑村 著
- “Learning PHP 5”
O’ Reilly, David Sklar, ISBN0-596-00560-1