

PHP マンモス本ダイジェスト

日本 PostgreSQL ユーザー会 (JPUG) 四国支部設立準備セミナー

日本ポストgresユーザー会

桑村 潤

<JuK@PostgreSQL.jp>

2002-11-04

2003-11-09 改訂

≡ はじめに

- PHP マンモス本：「PHP 徹底攻略」シリーズ (SOFTBANK) からかいつまんで PHP を紹介、本の内用を網羅するわけではない
- PHP を使った迅速なプログラム開発
- PHP 初心者、他言語のプログラマ向け
- PHP の有効性

予定

- 簡単な紹介かちょっとしたテクニックまで
- 各テーマ 10 分くらいずつ
- 講師：桑村潤 JPUG 分科会担当理事、

Plamo Linux WebDB 関連メンテナ

「PHP4 徹底攻略実戦編」執筆、

「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 の大雑把
- PHP 開発環境 Emacs
- XML パースの第一歩
- APC でキャッシュとエンコード

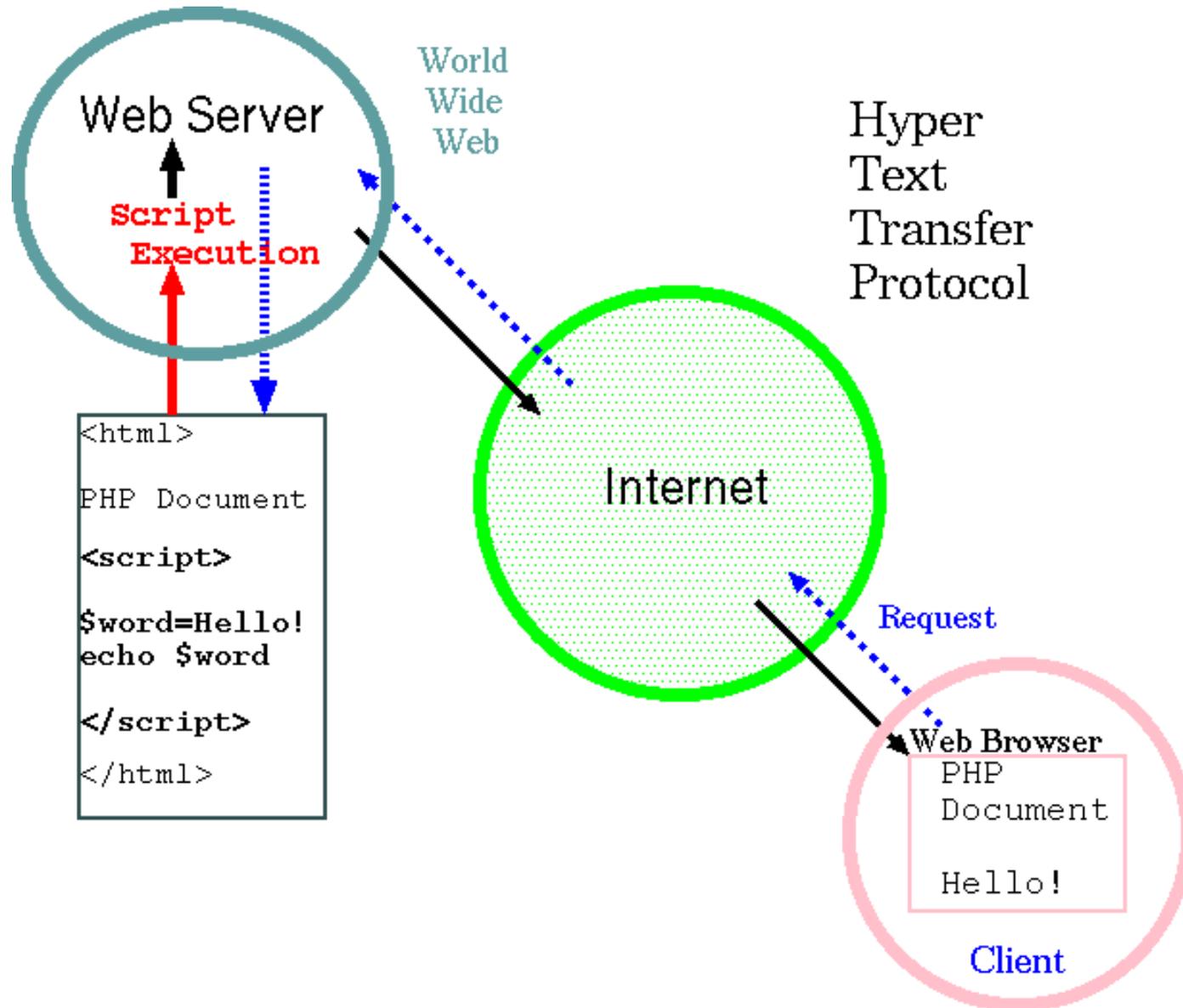
専門用語

- PHP は 'PHP:Hypertext Preprocessor'
 - <http://www.php.net>
- 派生もとの PHP/FI は 'Personal Home Page Construction Kit/Form Interpreter'
 - <http://www.zend.com/zend/hof/rasmus.php>
- PHP/FI は Rasmus Lerdorf 作
- PHP3 から Zeev Suraski (Zend 社) らが開発
- PHP4 は Zend Engine (強力なパーサ) 搭載
 - <http://www.zend.com>

PHP の特徴

- サーバサイドスクリプト
- HTML 埋め込み
- 言語としての特徴
- PHP の特徴を活かす

サーバーサイドスクリプト



≡ PHP の開始と終了 (タグ)

- `<? ?>`

 - `<? echo "PHP でこんにちは。
¥n" ; ?>`

- `<?php ?>`

 - `<?php`

 - `echo "PHP でこんにちは。
¥n" ;`

 - `?>`

- `<script language="php" >` `</script>`

 - `<script language="php" >`

 - `echo "PHP でこんにちは。
¥n" ;`

 - `</script>`

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
- DBMS インターフェース
- イメージ処理関数 (GD)
- XML 処理関数
- マルチバイト文字対応
- PEAR 拡張クラス

≡ PHP の特徴を活かす

- 既存の Web ページを活性化
 - 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 関数を使う
- 有用ルーチンはオブジェクト指向で再利用
 - クラスにまとめる

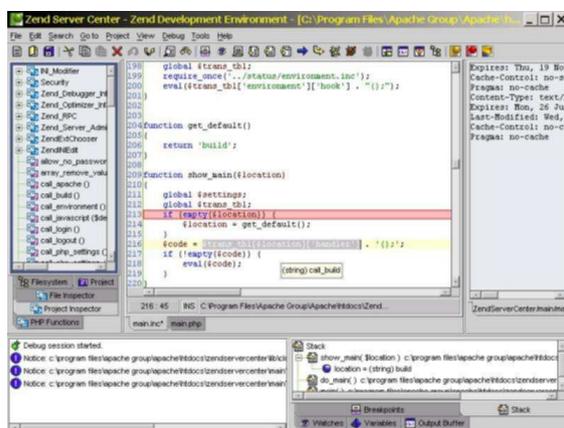
PHP 開発環境

- コマーシャル版 IDE
- オープンソース版 IDE
- Emacs での開発例を紹介

三 商用の開発環境

- Zend Studio

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



Price:\$195

\$245 (support, upgrade)

日本語版価格：54,000円

オンラインマニュアル

プロジェクト

リモートデバグ

- Komodo

- <http://www.activestate.com/Products/Komodo/>

Price: \$295 (Commercial), \$29.95 (Educational)

- Nusphere PHPed IDE (DBG)

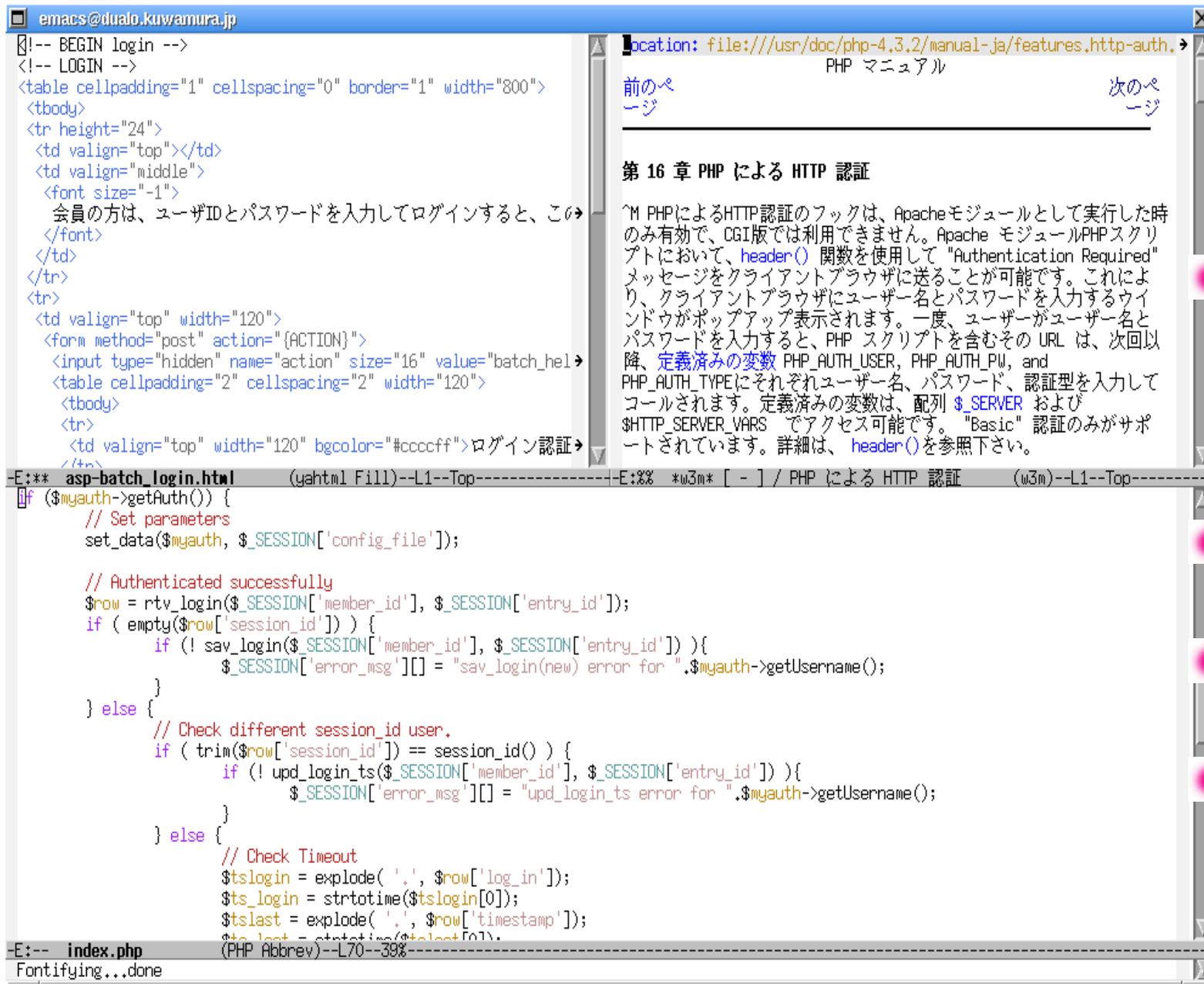
- <http://www.nusphere.com/>

Price: \$299.

☚ オープンな開発環境

- Eclipse+WebStudio (本格的 IDE)
 - <http://webstudio.japansite.org/>
- DBG+DDD (Unix 互換 GNU Debugger)
 - <http://dd.cron.ru/dbg/>
 - <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.phpedit.com>
- 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

開発環境 (GNU Emacs)



The screenshot shows the GNU Emacs editor interface. The left pane displays HTML code for a login form, including a table with a form and a login button. The right pane shows a page from the PHP manual, specifically Chapter 16 on HTTP authentication. The status bar at the bottom indicates the current file is 'asp_batch_login.html' and the cursor is at line 170, column 398.

```
emacs@dualo.kuwamura.jp
|!- BEGIN login -->
<!-- LOGIN -->
<table cellpadding="1" cellspacing="0" border="1" width="800">
<tbody>
<tr height="24">
<td valign="top"></td>
<td valign="middle">
<font size="-1">
  会員の方は、ユーザIDとパスワードを入力してログインすると、この
</font>
</td>
</tr>
<tr>
<td valign="top" width="120">
<form method="post" action="{ACTION}">
<input type="hidden" name="action" size="16" value="batch_hel
<table cellpadding="2" cellspacing="2" width="120">
<tbody>
<tr>
<td valign="top" width="120" bgcolor="#ccccff">ログイン認証
</td>
</tr>
</tbody>
</table>
</form>
</td>
</tr>
</tbody>
</table>
E: ** asp_batch_login.html (yahtml Fill)--L1--Top-----E:% *w3m* [ - ] / PHP による HTTP 認証 (w3m)--L1--Top-----
|f ($myauth->getAuth()) {
  // Set parameters
  set_data($myauth, $_SESSION['config_file']);

  // Authenticated successfully
  $row = rtv_login($_SESSION['member_id'], $_SESSION['entry_id']);
  if ( empty($row['session_id']) ) {
    if ( ! sav_login($_SESSION['member_id'], $_SESSION['entry_id']) ){
      $_SESSION['error_msg'][] = "sav_login(new) error for " . $myauth->getUsername();
    }
  } else {
    // Check different session_id user.
    if ( trim($row['session_id']) == session_id() ) {
      if ( ! upd_login_ts($_SESSION['member_id'], $_SESSION['entry_id']) ){
        $_SESSION['error_msg'][] = "upd_login_ts error for " . $myauth->getUsername();
      }
    } else {
      // Check Timeout
      $tslogin = explode( '.', $row['log_in']);
      $ts_login = strtotime($tslogin[0]);
      $tslast = explode( '.', $row['timestamp']);
      $ts_last = strtotime($tslast[0]);
    }
  }
}
E:-- index.php (PHP Abbrev)--L70--398-----
Fontifying...done
```

● emacs-w3m
(マニュアル表示)

● php-mode
(プログラム編集)

● etags
(関数検索)

● ps-print
(カラー印刷)

Emacs (php-mode. el, ps-print. el)

- php-mode. el (1. 0. 2)

- <http://sourceforge.net/projects/php-mode/>

```
(autoload 'php-mode "php-mode-102" "PHP モード" t)
(add-hook 'php-mode-user-hook 'turn-on-font-lock)
(setq auto-mode-alist (append
  (list ('"\.php$" . php-mode)) auto-mode-alist))
```

- ps-print. el (6. 5. 5)

- <http://www.cpqd.com.br/~vinicius/emacs/>

```
(setq ps-multibyte-buffer 'non-latin-printer)
(require 'ps-mule)
(defalias 'ps-mule-header-string-charsets 'ignore)
(add-hook 'ps-print-hook
  '(lambda () (setq ps-printer-name "ps")
    (load "time-stamp")
    (setq ps-right-header
      (list "/pagenumberstring load"
        'time-stamp-yyyy/mm/dd 'time-stamp-hh:mm:ss))))
```

Emacs (w3m. el)

- emacs-w3m. el

- <http://emacs-w3m.namazu.org/>

```
(autoload 'w3m "w3m" "Interface for w3m on Emacs." t)
(autoload 'w3m-find-file
          "w3m" "w3m interface function for local file." t)
(autoload 'w3m-search
          "w3m-search" "Search QUERY using SEARCH-ENGINE." t)
(autoload 'w3m-weather
          "w3m-weather" "Display weather report." t)
(autoload 'w3m-antenna
          "w3m-antenna" "Report change of WEB sites." t)
(autoload 'w3m-namazu
          "w3m-namazu" "Search files with Namazu." t)
(setq w3m-namazu-index-alist
      '(("JF" "/usr/share/namazu/templatel")))
(eval-after-load "w3m"
  '(setq w3m-command-arguments
        (nconc w3m-command-arguments
               ('("-o" "http_proxy=http://gw:8080/"))))
)
(setq w3m-no-proxy-domains '("localhost" "local.domain.jp"))
```

Emacs コマンド

- インデント揃え
 - `^I`
- カラープリント
 - `ps-print-buffer-with-faces`
 - `ps-print-region-with-faces`
- マニュアル表示
 - `w3m-find-file`
- ウェブサイト表示
 - `W3m-browse-url`
- PHP/PostgreSQL 日本語 マニュアル
検索サービスのサイト
 - <http://www.net-newbie.com/>

Emacs TAGS を活用

- 関数のタグを `etags` コマンドで作成
 - `etags *.php`
- Emacs でファイルを開いてタグを利用
 - `M-x visit-tags-table`
 - `M-[TAB]` で関数名の補完
 - `M-.` で関数の定義部へ移動
 - `M-*` で定義部から移動前の場所へ戻る

XML のパース

- Expat による XML のパース
- 業務で XML のパースは避けて通れない
- 最も単純な例でハンドラを理解

≡ PHP の XML 関連の関数群

- XML パーサ

- 最も単純なパーサは expat を使用
- Apache-1.3.7 以降に組み込み
 - <http://sourceforge.net/projects/expat/>

- XSLT (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
- Sabrotron と expat を使用
 - <http://www.gingerall.com/>

- DOMXML

- 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/>

≡ 単純な XML パーサ

● パーサの作成

- `xml_parser_create` -- XML パーサを作成する
- `xml_parser_set_option`
-- XML パーサのオプションを設定する
- `xml_set_element_handler`
-- `start` および `end` 要素のハンドラを設定する
- `xml_set_character_data_handler`
-- 文字データハンドラを設定する
- `xml_set_processing_instruction_handler`
-- 処理命令 (PI) 用ハンドラを設定する
- `xml_set_default_handler`
-- デフォルトのハンドラを設定する
- `xml_set_external_entity_ref_handler`
-- 外部リファレンスハンドラを設定する



XML ハンドラの登録

```
function new_xml_parser($file) {
    global $parser_file;

    $xml_parser = xml_parser_create();
    xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, 1);
    xml_set_element_handler($xml_parser, "startElement", "endElement");
    xml_set_character_data_handler($xml_parser, "characterData");
    xml_set_processing_instruction_handler($xml_parser, "PIHandler");
    xml_set_default_handler($xml_parser, "defaultHandler");
    xml_set_external_entity_ref_handler($xml_parser, "externalEntityRefHandler");
    $parser_file[$xml_parser] = $file;
    return array($xml_parser, $fp);
}
```

```
function startElement($parser, $name, $attrs){
    echo "<font color= orange >sE</font>";
    print "&lt;<font color='\#0000cc\'>$name</font>";
    if (sizeof($attrs)) {
        while (list($k, $v) = each($attrs)) {
            print "<font color='\#009900\'>$k</font>=\<font
                color='\#990000\'>$v</font\>";
        }
    }
    print "&gt;";
}
```

```
function endElement($parser, $name){
    echo "<font color='orange' >eE</font>";
    print "&lt;/<font color='\#0000cc\'>$name</font>&gt;";
}
```

```
function characterData($parser, $data) {
    echo "<font color='orange' >cD</font>";
    print "<b>$data</b>";
}
```

```
function defaultHandler($parser, $data) {
    echo "<font color='orange' >dH</font>";
    if (substr($data, 0, 1) == "&" && substr($data, -1, 1) == ";") {
        printf("DH<font color='\#aa00aa\'>%s</font>",
            htmlspecialchars($data));
    } else {
        printf('<font size="-1\'>%s</font>',
            htmlspecialchars($data));
    }
}
```

オレンジ色で文字を表示
(実際には処理を書く)

XML パーステスト

- 入力 XML データ

```
<?xml version="1.0" encoding="ISO-2022-JP" ?>
<my_data>
<my_0020 label="Study Date">19980126</my_0020>
<my_0022 label="Acquisition Date">20041223</my_0022>
<my_0050 label="Accession Number" />
<my_0060 label="Modality">CT</my_0060>
</my_data>
```

- 出力ブラウザ表示

```
dH<?xml version="1.0" encoding="ISO-2022-JP" ?>dH
sE<MY_DATA>cD
sE<MY_0020 LABEL="Study Date">cD19980126eE</MY_0020>cD
sE<MY_0022 LABEL="Acquisition Date">cD20041223eE</MY_0022>cD
sE<MY_0050 LABEL="Accession Number">eE</MY_0050>cD
sE<MY_0060 LABEL="Modality">cDCTeE</MY_0060>cD
eE</MY_DATA>dH

parse complete
```

商用配布ソフトの開発

- キャッシュモジュールによる高速化
- ソースを中間コードにして見えなくする
- APC キャッシュ

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>
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.rssoftlab.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

- APC (Alternative PHP Cache)
 - <http://apc.communityconnect.com/>
10%位改善 (QPL -> PHP License)
- Turck MMCache
 - <http://turck-mmcache.sourceforge.net/>
- AfterBurner Cache
 - <http://afterburner.bware.it>
キャッシュのみ (LGPL)
- PHP Accelerator (ionCube)
 - <http://www.php-accelerator.co.uk/>
性能は良いらしいがライセンス不明バイナリのみフリー
(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

APC (Alternative PHP Cache)

- APC Cache ML

- <http://lists.communityconnect.com/mailman/listinfo/apc-cache>

- PHP-4.2.3 には CVS 版 (1.0) を使用

- <http://apc.communityconnect.com/sources/apc-cvs.tar.gz>

- Apache DSO 版へ組み込み

```
tar xvfz apc-cvs.tar.gz
cd apc
phpize
./configure --enable-apc
make
make install
```

≡ APC によるキャッシュ

- 2種類のキャッシュ方法
 - 共有メモリー (shm)
 - メモリー内部にキャッシュ
 - メモリーマップファイル (mmap)
 - ファイルにキャッシュしてその写しをメモリー内部に

- php.ini で指定

```
[APC]
```

```
zend_extension="/usr/lib/php/extensions/no-debug-non-zts-  
20020429/php_apc.so"
```

```
apc.mode = shm
```

```
;apc.mode = mmap
```

```
;off
```

≡ APC のキャッシュモード

● 共有メモリ (shm)

- オブジェクト毎の有効期限設定可能
- Webサーバ全体のキャッシュ統計が可能
- SystemV セマフォを使うので漏洩に注意が必要

● メモリマップ (mmap)

- 内部のキャッシュの働きがわりやすい
- キャッシュされたオブジェクトがファイルになるのでコンパイラとしても利用可能
- 沢山のファイルを開く
- キャッシュ統計はホスト毎ではなくプロセス毎
- キャッシュ管理が複雑

APC 使用時のアクセステスト

- 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)

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

≡ APC によるエンコード

- まずメモリマップ (MMAP) モードで全てのプログラムにアクセスし、オブジェクトのファイルを生成
 - php.ini の設定
 - apc.mode = mmap
 - apc.cache_dir = オブジェクトファイルの生成ディレクトリ
 - hore.php = 生成 => hore.php.apc
- 次に、共有メモリ (SHM) モードで運用
 - hore.php.apc = コピー => hore.php
 - もちろん apc モジュールは必要
 - php.ini で細かい設定を行う
 - apc.mode = shm
 - apc.cache_rt ロック方式 {safe|fast}
 - apc.check_mtime ファイル修正時の自動キャッシュ更新 (既定値 0)
 - apc.check_compiled_source = 1
コンパイル済の mmap 形式ファイルの使用

APC の php. ini 設定

- `apc. mode`
キャッシュ動作モード {off|shm|mmap}
- `apc. cachedir` (mmap)
コンパイル済みファイルの生成場所
- `apc. cache_rt` (shm)
ロック方式 {safe|fast}
- `apc. check_mtime` (shm)
ファイル修正時の自動キャッシュ更新 (既定値 0)
- `apc. hash_buckets` (shm)
キャッシュインデクスのバケツ数 (既定値 1024)
- `apc. shm_segment_size` (shm)
SHMセグメントサイズ (既定値 33554431)
- `apc. ttl`
キャッシュの有効期限 (既定値 0 は無限)
- `apc. regex`
キャッシュしないファイルの正規表現指定 ('*.inc')
- `apc. relative_includes`
相対パスでのインクルードを可能にする (既定値 1)
- `apc. check_compiled_source`
コンパイル済みの mmap 形式ファイルの使用 (既定値 0)

まとめ

- PHP で簡単にトランジェントなページ
- 自分好みの開発ツール
- 商用アプリケーションのための工夫
- 日本ポストgresユーザー会でも使用

備考

- 赤／青マンモス本のその他の話題
 - DBMS アクセス
 - Web サイトプログラムのセキュリティ
 - セッション管理
 - 認証システムの利用
 - テンプレートの利用
 - マルチバイトコードの扱い方
 - 効率的なデバッグ
- JPUG 予定：事例紹介セミナー（冬？）
- 新しいテーマ：PHP と讃岐うどんの関係

PHP と讃岐うどん

讃岐うどんは、西暦800年頃空海が中国より伝えたといわれています。麺はその100年くらい前から伝わっていたそうですが、主に宮廷で食されるものとして民衆の口には入らなかったようです。香川県でも当時の国の機関である国分寺くらいに専属の麺職人がいたかどうかというくらいだったようです。鎮護国家と衆生救済を説く空海は小麦の生産と最新の切り麺の技術を伝え、これが民衆に広まってうどんとなったということです。即身成仏の真言密教において生きていくうちに美味しいものを食べなければ救済されたとは言えないのです。

この光景は、伽藍とバザールのひとこまを思い出させるのではないのでしょうか？すなわち、誰にでも口にできるうどんはオープンソースと言えます。また、麺の食べ方にはいろいろあり、うどんはその麺食文化のひとつの形です。まさに、スクリプト言語全体を麺と例えれば、PHPがそのうちのひとつであるかのようです。面白いことに中国から伝わった麺にはラーメンもあります。材料は小麦でない麺には蕎麦もあります。

食も言語も文化の象徴としてとらえることができるのであれば、讃岐うどんは食のオープンソース、麺食文化のPHPといったところでしょうか。