

PHP5

ThinkIT PHP5 のオブジェクト指向

<http://www.thinkit.co.jp/free/marugoto/1/1/1/1.html>

ソースからのインストール

php-4.4.2 をソースからインストールする。

```
#!/configure ¥
--prefix=/usr/local/php4 ¥
--without-pgsql ¥
--with-mysql=/usr/local/mysql ¥
--with-apxs=/usr/local/apache/bin/apxs ¥
--enable-track-vars ¥
--enable-mbstring ¥
--enable-mbregex ¥
--enable-zend-multibyte ¥
--with-gd ¥
--with-freetype-dir=/usr/lib ¥
--with-jpeg-dir=/usr/lib ¥
--with-png-dir=/usr/lib ¥
--with-zlib-dir=/usr/lib ¥
--enable-gd-native-ttf ¥
--enable-gd-jis-conv
```

足らなかったパッケージは yum でインストール。

```
# yum install libjpeg-devel ¥
libpng-devel ¥
freetype-devel ¥
flex ¥
bison
```

configure 中に bison は 1.28 推奨と警告ができるが、パッケージには 2.0-6 しかなかったのでそのまま続行(汗)

make と make install。時間は 30 分ぐらい。

```
# make
# make install
```

php の解凍フォルダより ini のコピー。

```
# cp /usr/local/src/php-4.4.2/php.ini-dist /usr/local/php4/lib/php.ini
# vi /usr/local/php4/lib/php.ini
```

mbstring 関係のセミコロンを外す。

つづいて httpd.conf の設定。

```
LoadModule php4_module libexec/libphp4.so
AddType application/x-httpd-php .php

DirectoryIndex index.html index.cgi index.php
```

apache の再起動。

```
#/usr/local/apache/bin/apachectl restart
```

php の情報を phpinfo() で確認し、mbstring、mysql、gd のあたりが設定されているか確認する。

php-4.4.4

うちの環境では

```
/usr/local/apache2  
/usr/local/mysql5
```

php-4.4.2 をソースからインストールする。

```
# cd /usr/local/src  
# wget http://www.php.net/get/php-4.4.4.tar.gz/from/jp2.php.net/mirror  
# chmod 755 php-4.4.4.tar.gz  
# tar -zxvf php-4.4.4.tar.gz  
# ./configure ¥  
--prefix=/usr/local/php444 ¥  
--without-pgsql ¥  
--with-mysql=/usr/local/mysql5 ¥  
--with-apxs2=/usr/local/apache2/bin/apxs ¥  
--enable-track-vars ¥  
--enable-mbstring ¥  
--enable-mbregex ¥  
--enable-zend-multibyte ¥  
--with-gd ¥  
--with-freetype-dir=/usr/lib ¥  
--with-jpeg-dir=/usr/lib ¥  
--with-png-dir=/usr/lib ¥  
--with-zlib-dir=/usr/lib ¥  
--enable-gd-native-ttf ¥  
--enable-gd-jis-conv ¥  
  
make  
make install
```

したあとに、httpd.conf に勝手に

```
LoadModule php4_module libexec/libphp4.so
```

が記述される。

すでにこの記述がある場合は二重になってしまうので、チェックして削除する。

php-5.2.0

libxml2 が必要。FedoraCore4 では libxml2 自体は rpm パッケージでインストールされているが、libxml2-devel が入っていないため configure で失敗する。

さきに yum で libxml2-devel をインストールしておくこと。

```
# ./configure ¥  
--prefix=/usr/local/php520 ¥  
--without-pgsql ¥  
--with-mysql=/usr/local/mysql ¥  
--with-apxs2=/usr/local/apache/bin/apxs ¥  
--enable-mbstring ¥  
--enable-zend-multibyte ¥
```

```

--with-gd ¥
--with-freetype-dir=/usr/lib ¥
--with-jpeg-dir=/usr/lib ¥
--with-png-dir=/usr/lib ¥
--with-zlib-dir=/usr/lib ¥
--enable-gd-native-ttf ¥
--enable-gd-jis-conv ¥
--enable-sqlite-utf8 ¥
--enable-memory-limit ¥
--with-mysqli=/usr/local/mysql/bin/mysql_config

```

configure の最後にワーニング表示がでている場合は、どこかタイプミスしている。

--enable-track-vars は無い？みたい。sqlite はデフォで組み込まれる模様。ただし utf-8 に対応させるのは別途指定。

PHP-5.2.1

PHP-5.2.0 には結構バグがあるみたいで update する。memory-limit は早い段階で呼び出さないといけないみたいなので記述の順番を変更。

```

# ./configure ¥
--prefix=/usr/local/php521 ¥
--enable-memory-limit ¥
--with-apxs2=/usr/local/apache/bin/apxs ¥
--without-pgsql ¥
--with-mysql=/usr/local/mysql ¥
--with-mysqli=/usr/local/mysql/bin/mysql_config ¥
--enable-mbstring ¥
--enable-zend-multibyte ¥
--with-gd ¥
--with-freetype-dir=/usr/lib ¥
--with-jpeg-dir=/usr/lib ¥
--with-png-dir=/usr/lib ¥
--with-zlib-dir=/usr/lib ¥
--enable-gd-native-ttf ¥
--enable-gd-jis-conv ¥
--enable-sqlite-utf8

```

configure で のようなワーニングが出た。

```

configure: warning: You will need re2c 0.9.11 or later if you want to regenerate PHP parsers.
configure: warning: lemon versions supported for regeneration of libsqlite parsers: 1.0 (found: none).

```

5.2.0 のときも出ていたのかは不明。

とりあえずそのまま make もできて、make install しようとする

```

# make install
Installing PHP SAPI module:      apache2handler
/usr/local/apache224/build/inststdso.sh SH_LIBTOOL=/usr/local/apr224/build-1/libtool' libphp5.la
/usr/local/apache224/modules
/usr/local/apr224/build-1/libtool --mode=install cp libphp5.la /usr/local/apache224/modules/
cp .libs/libphp5.a /usr/local/apache224/modules/libphp5.a
cp .libs/libphp5.lai /usr/local/apache224/modules/libphp5.la
libtool: install: warning: remember to run `libtool --finish /usr/local/src/php-5.2.1/libs'
chmod 755 /usr/local/apache224/modules/libphp5.so

```

までできて libphp5.so が not found で止まってしまう。

たにかに /usr/local/apache/modules には libphp5.a と libphp5.la があるのみ。

ググったところ、re2c は実際には gcc に含まれるため特に必要はないことと、lemon は libsqlite に関するものなので libphp5.so には関係なさそうだった。
そもそもワーニングだし。

を参考に re2c と lemon を入れることにした。

Web、只今勉強中

<http://d.hatena.ne.jp/kazukichop/comment?date=20061120>

```
wget http://nchc.dl.sourceforge.net/sourceforge/re2c/re2c-0.11.0.tar.gz
tar xzf re2c-0.11.0.tar.gz
cd re2c-0.11.0
./configure
make
make install
```

```
wget http://www.sqlite.org/cvstrac/getfile/sqlite/tool/lemon.c
gcc -o lemon lemon.c
mv lemon /usr/local/bin
```

configure ではワーニングがでなくなり、make も完了。
そして make install では ... やっぱり libphp5.so が作れてない(汗)

で結局のところ、make clean して make して make install していたんですが、make distclean して make して make install したらよくなった模様？
make clean ではなにか残骸？が消えていなかったのか ...

PHP-5.2.6

5.2.1 と同じだが、--enable-memory-limit がデフォルトで有効になっているため必要なくなった。
というか、指定すると configure でエラーになる。

```
# ./configure ¥
--prefix=/usr/local/php526 ¥
--with-apxs2=/usr/local/apache/bin/apxs ¥
--without-pgsql ¥
--with-mysql=/usr/local/mysql ¥
--with-mysqli=/usr/local/mysql/bin/mysqli_config ¥
--with-pdo-mysql=/usr/local/mysql ¥
--enable-mbstring ¥
--enable-zend-multibyte ¥
--with-gd ¥
--with-freetype-dir=/usr/lib ¥
--with-jpeg-dir=/usr/lib ¥
--with-png-dir=/usr/lib ¥
--with-zlib-dir=/usr/lib ¥
--enable-gd-native-ttf ¥
--enable-gd-jis-conv ¥
--enable-sqlite-utf8 ¥
--with-pdo-sqlite
```

--enable-zend-multibyte でコンパイルした PHP ならスクリプトが UTF-8 の BOM ありでも OK

PHP-5.4.4

ひさびさに、最新版ソースからテスト。

/root にダウンロードしたら、解凍し、ディレクトリに入る。

```
# tar zxvfp php-5.4.4.tar.gz
# cd php-5.4.4
```

あまりにご無沙汰なので、まずは prefix と with-apxs2 だけ指定して、デフォルトの設定がどうなるのかチェックしてみる。

```
# ./configure ¥
--prefix=/usr/local/php544 ¥
--with-apxs2=/usr/sbin/apxs ¥
```

早速エラー（汗）。足りないものを yum でインストール。

```
yum install libxml2-devel httpd-devel
```

この頃は ~っと yum(rpm)でインストールしてたので、ソースのコンパイルでは devel 系のパッケージが必要なことを忘れてました（汗）

で、晴れて

```
make
```

数十分かかる。

make install する前に、現行の libphp5.so が上書きされるので、バックアップしといたほうがいいのかも。

よければ

```
make install
```

これにより、アパッチの設定ファイル

```
/etc/httpd/conf/httpd.conf
```

には、

```
LoadModule php5_module /usr/lib/httpd/modules/libphp5.so
```

が書き加えられるが、先にインストールしていた rpm の php 設定が

```
/etc/httpd/conf.d/php.conf
```

にあり、こちらと重なるので httpd.conf 側はコメントアウト。

php.ini は勝手に用意されないなので、ソースを解凍した中にある php.ini-development もしくは

php.ini-production のいずれかを選んで /usr/local/php544/lib に配置。

```
# cd php-5.4.4
cp php.ini-production /usr/local/php544/lib/php.ini
```

みたいにして httpd を再起動、その後 phpinfo を確認してみる。

で、話を最初に戻して、prefix と with-apxs2 だけの phpinfo は

```
Configuration File (php.ini) Path /usr/local/php544/lib
apache2handler
ctype
date
dom
ereg
fileinfo
filter
hash
iconv
json
libxml
pcre
PDO
pdo_sqlite
Phar
posix
Reflection
session
SimpleXML
SPL
sqlite3
standard
tokenizer
xml
xmlreader
xmlwriter
```

みたいな感じ。

mbstring や gd、MySQL はやっぱり見当たらず。改めて configure からやり直しね。
php.ini の配置は /etc になるようにする。

```
# ./configure ¥
> --prefix=/usr/local/php544 ¥
> --with-apxs2=/usr/sbin/apxs ¥
> --with-config-file-path=/etc
> --with-mysql ¥
> --with-mysqli ¥
> --with-pdo-mysql ¥
> --enable-mbstring ¥
> --enable-zend-multibyte ¥
> --with-gd ¥
> --with-freetype-dir ¥
> --with-jpeg-dir ¥
> --with-png-dir ¥
> --with-zlib-dir ¥
> --enable-gd-native-ttf ¥
> --enable-gd-jis-conv ¥
> --enable-sqlite-utf8 ¥
```

jpeglib.h が無いと出たので

```
# yum install libjpeg-devel
```

png.h が無いと出たので

```
# yum install libpng-devel
```

freetype.h が無いと出たので

```
# yum install freetype-devel
```

configure は終わったが、最後にワーニングが出た。

```
configure: WARNING: unrecognized options: --enable-zend-multibyte,  
--enable-sqlite-utf8
```

enable-zend-multibyte は PHP5.4 より標準で利用可能、php.ini で

```
zend.multibyte = On
```

とすると利用できるそう。

enable-sqlite-utf8 は

PHP マニュアル - sqlite_libencoding

<http://uk.php.net/manual/ja/function.sqlite-libencoding.php>

で変更になったみたい。「UTF-8 マルチバイト文字は現在は正しく動かないから外す」ってことかな？

終わったら、

```
# make clean  
# make  
# make install
```

あとは最初にやったときと同じく httpd.conf を修正し、

```
/etc/init.d/httpd restart
```

phpinfo() で追加した分が確認できれば OK。

date でワーニングが出ている場合は php.ini で

```
date.timezone = Asia/Tokyo
```

PHP-5.4.13

Fedora12 に PHP-5.4.13 を入れようとしたときの記録。

ppc (mac mini G4) を正式にサポートしているのは Fedora の場合、Fedora12 までらしい。

で、Fedora12 だと php の rpm が php-5.3.3 で、ZendFramework2 は 5.3.4 以降じゃないと locale でエ

ラーがでた。

ということで Fedora12 に、PHP 最新版をソースからインストールするわけだが、例によって

```
# wget http://www.php.net/get/php-5.4.13.tar.gz/from/jp1.php.net/mirror
# tar zxvfp php-5.4.13.tar.gz
# cd php-5.4.13
# ./configure ..... 略
```

みたいな感じで、まずはワーニングとかエラーを確認しながら make に必要なパッケージをインストールしていく。

その中で Fedora12 のパッケージだとバージョンが低くてダメだったのが のヤツ。

re2c scanner generator

http://sourceforge.jp/projects/sfnet_re2c/

rpm のほうが無難？な気がするので src.rpm をゲットし、リビルドすることに。

```
# rpmbuild --rebuild re2c-0.13.5-1.src.rpm
RPM build errors:
user marcus does not exist - using root
group marcus does not exist - using root
user marcus does not exist - using root
group devel does not exist - using root
Bad exit status from /var/tmp/rpm-tmp.UdPR0t (%build)
```

ユーザーとグループを追加しないとリビルドできないみたい。

```
# useradd -M -s /sbin/nologin marcus
# groupadd devel
```

気を取り直して

```
# rpmbuild --rebuild re2c-0.13.5-1.src.rpm
```

やっぱりエラー。

どうやら gcc とかがそもそも入ってないっぽい(汗)。『gcc とかって何が必要？』と考えるのもメドイのでグループインストールをかます。

```
# yum groupinstall "Development tools"
```

もういっちょトライ。

```
# rpmbuild --rebuild re2c-0.13.5-1.src.rpm
```

今度は無事にコンパイルできた。

カレントディレクトリに rpmbuild が出来ているので

```
# cd rpmbuild/RPMS/ppc
# rpm -Uvh re2c-0.13.5-1.ppc.rpm
```


改めて php のディレクトリに戻り configure を試す。

```
# ./configure ¥
> --prefix=/usr/local/php-5.4.13 ¥
> --with-apxs2=/usr/sbin/apxs ¥
> --with-config-file-path=/etc ¥
> --with-mysql ¥
> --with-mysqli ¥
> --with-pdo-mysql ¥
> --enable-mbstring ¥
> --enable-zend-multibyte ¥
> --with-gd ¥
> --with-freetype-dir ¥
> --with-jpeg-dir ¥
> --with-png-dir ¥
> --with-zlib-dir ¥
> --enable-gd-native-ttf ¥
> --enable-gd-jis-conv ¥
> --enable-sqlite-utf8
```

php-5.4.4 のときに試した --enable-zend-multibyte と --enable-sqlite-utf8 は、組み込んでみたがワーニングは出ない模様。効いているんだかないないんだかまではチェックしてません（汗）

とりあえず OK そうなので

```
# make clean
# make
# make install
```

みたいな。

いままで気にしたことなかった（気づいてなかっただけ？）けど、make install 中の表示の中に

```
libtool: install: warning: remember to run `libtool --finish /root/php-5.4.13/libs'
```

ってのがあったので、とりあえずやっといた。

```
# libtool --finish /root/php-5.4.13/libs
```

php が無事インストールされれば httpd.conf に loadModule が追加されているので、その下に AddType を追記しておく。

```
LoadModule php5_module          /usr/lib/httpd/modules/libphp5.so
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

httpd を再起動し、phpinfo() で状態をチェックできれば OK と思う。

make や build の残骸は、必要無ければ削除しておく。

```
# rm -rf php-5.4.13
# rm -rf rpmbuild
# userdel marcus
# groupdel devel
```

PHP-5.5.0

いつのまにやら 5.5.0 ですね。Fedora12 ~ 5.4.13 のときの準備でそのままコンパイルできた。

```
# wget http://jp1.php.net/get/php-5.5.0.tar.gz/from/this/mirror
# tar zxvf php-5.5.0.tar.gz
# cd php-5.5.0
# ./configure ¥
> --prefix=/usr/local/php-5.5.0 ¥
> --with-apxs2=/usr/sbin/apxs ¥
> --with-config-file-path=/etc ¥
> --with-mysql ¥
> --with-mysqli ¥
> --with-pdo-mysql ¥
> --enable-mbstring ¥
> --with-gd ¥
> --with-freetype-dir ¥
> --with-jpeg-dir ¥
> --with-png-dir ¥
> --with-zlib-dir ¥
> --enable-gd-native-ttf ¥
> --enable-gd-jis-conv ¥
```

php-5.5.0 で --enable-zend-multibyte と --enable-sqlite-utf8 は、有効となっている模様。

```
# make
# make test
# make install
# libtool --finish /root/php-5.5.0/libs
```

make test したら PEAR 系で「バージョン古いよ」警告がちらほら。とりあえず、いまは無視。シンボリックリンクを直す。

```
# ln -sf /usr/local/php-5.5.0/bin/* /usr/bin
```

コンソール、または phpinfo() で切り替わったのを確認する。

```
# php -v
PHP 5.5.0 (cli) (built: Jul 1 2013 07:25:59)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.5.0-dev, Copyright (c) 1998-2013 Zend Technologies
```

php.ini の設定

```
[mbstring]
mbstring.language = Japanese
mbstring.internal_encoding = EUC-JP
mbstring.http_input = auto
mbstring.http_output = SJIS
mbstring.encoding_translation = Off
mbstring.detect_order = auto
mbstring.substitute_character = none;
mbstring.func_overload = 0
```

スクリプト内部で下記のように設定することも可能。

ただし、ini_set では設定できないものもある。

```
mb_language('Japanese');
ini_set('mbstring.detect_order', 'auto');
ini_set('mbstring.http_input', 'auto');
ini_set('mbstring.http_output', 'pass');
ini_set('mbstring.internal_encoding', 'UTF-8');
ini_set('mbstring.script_encoding', 'UTF-8');
```

```
ini_set('mbstring.substitute_character', 'none');
mb_regex_encoding('UTF-8');
```

.htaccess で指定する場合は

```
php_value mbstring.language Japanese
php_value mbstring.http_input EUC-JP
php_value mbstring.http_output EUC-JP
php_flag auto_detect_line_endings on
php_value output_handler mb_output_handler
php_value default_charset EUC-JP
php_value mbstring.internal_encoding EUC-JP
php_flag mbstring.encoding_translation off
php_value mbstring.detect_order auto
php_value mbstring.substitute_character none
php_flag magic_quotes_gpc off
```

値に対しては php_value で、On、Off に対しては php_flag を使用するらしい。

文字コードの指定は

```
UTF-8
EUC-JP
SJIS
ISO-2022-JP (日本語メール)
```

mbstring.internal_encoding

内部エンコーディングの設定。PHP の内部で文字データを扱う際の文字コードを設定しておく。データベースと連携する場合、データベースと同じ文字コードにしておく、いちいち変換処理をする必要がなくなる。

Shift-JIS には問題点があるので、内部エンコーディングには EUC-JP か UTF-8 を使用するようになる。

```
mbstring.internal_encoding = EUC-JP
```

mbstring.script_encoding

PHP スクリプトの文字コードを設定する。

この設定がない場合は mbstring.internal_encoding の設定が利用される。

```
mbstring.script_encoding = EUC-JP
```

mbstring.encoding_translation

古いバージョンのブラウザでは常に Shift-JIS でデータを送信してくるブラウザもある。

mbstring.encoding_translation が on ならば、Web サーバが受け取るデータ（入力データ）を自動判別し、内部エンコーディングに変換する。

```
mbstring.encoding_translation = On
```

参考

マルチバイト文字列関数

<http://www.php.net/manual/ja/ref.mbstring.php>

PHP を実行するディレクトリを制限する

php の初期設定のままでは、どのディレクトリに php を置いても実行してしまうので、制限する方法。

Apache モジュールとしてインストール

<http://php.planetmirror.com/manual/ja/security.apache.php>

open_basedir

<http://php.planetmirror.com/manual/ja/features.safe-mode.php#ini.open-basedir>

php の設定

- 1.php.ini
- 2.htpd.conf
- 3.htaccess
- 4.Windows レジストリ
- 5.PHP の ini_set()

定数	設定可能
<u>PHP_INI_ALL</u>	1 ~ 5
<u>PHP_INI_PERDIR</u>	1 ~ 3
<u>PHP_INI_SYSTEM</u>	1 ~ 2

php.ini ディレクティブ - Manual

<http://jp.php.net/manual/ja/ini.php>

サーバモジュール版では Web サーバの起動時に読み込まれる。
CGI 版ではスクリプトが実行される度に読み込まれる。

実行時設定

<http://jp.php.net/manual/ja/configuration.php>

php.ini の記述

ディレクティブ名 = 設定値

htpd.conf または .htaccess の記述

php_value ディレクティブ名 値
php_flag ディレクティブ名 on/off

PHP 資料

PHP : Hypertext Preprocessro

<http://jp.php.net/>

PHP マニュアル

<http://jp.php.net/manual/ja/>