

PHPの基本的活用法 インストールとDB連携

株式会社システムジェイ
取締役副社長 柏岡秀男

Kashioka@systemj.com

Agenda

- PHPの歴史
- PHPの動作の仕組み
- CGIとの比較
- インストールTips
- PHPコーディング最初の一步
- サンプルアプリ解説
- まとめ

PHPの歴史

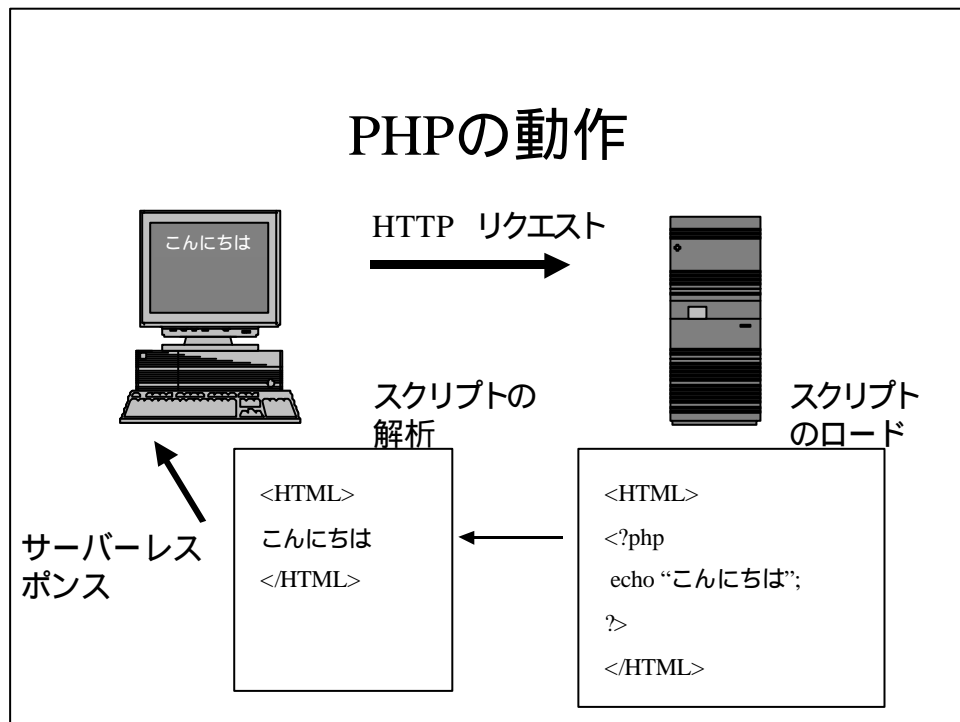
- 1994年誕生
 - PHP/FI(Personal Home Page/Form Interpreter)
- 1997年 PHP3リリース
- 1999年 国際化版ベータリリース
 - 最新版php-3.0.15-i18n-ja
- 2000年 PHP4リリース

PHPの特徴

- PHPはサーバサイドスクリプト言語
 - 実行はサーバ
 - 表示はクライアント
- DBとの連携
 - メジャーなDBに対応
- HTMLとの連携
 - HTMLも作れる
- 開発が簡単
 - エラー内容がブラウザに表示できる。
 - スクリプトなのでコンパイルがいらない。

PHPの動作

- どのように実行されるのか？



CGIの場合

- リクエストがある度にサーバでプロセスとランタイムをロードし生成してしまうため、サーバに大きな負荷を与えてしまう
- ループしたりすると大変

アクセスが集中する

サーバの負荷が高くなる

サーバダウンの危険もある



クライアント

ブラウザからURLを
要求

ホームページを表示

PHPでカウンタを表示する場合



サーバー

PHPファイルを読み
込みPHPの処理を
行いクライアントに
HTML形式で送る

(ここでカウンタ部
分も生成済み)

PHPの場合

- WWWサーバにモジュールとして組み込むためWWWサーバのサイズが大きくなるがWWWサーバの一部として動作。
- タイムアウトの設定等可能
- DSOの場合はさらに動的にモジュールを読み込むため余分なメモリの消費がない。

PHPのインストール

- インストールの形態
 - CGI
 - Apacheモジュール
 - ApacheDSO (Dynamic Shared Object)

PHP国際化対応版

- 国際化対応版を使用する利点
 - スクリプトファイル HTTP出力とで個別の文字コードを使用することができる。
 - POST/GET/COOKIE で受け取った値の文字コードを推定することができる。
 - 文字列に対する文字コード変換機能 (PHP3関数)。
 - メール送信。本文はJIS、SubjectヘッダはMIME/Base64/JISで。
 - HTTP出力のContent-Typeがtext/htmlの場合、適切なcharsetが自動的に指定される。
 - 安定した文字コード変換。
 - マルチバイト正規表現

国際化対応版ページより

CGI版PHPのインストール

```
# tar xvzf php-3.0.15-i18n-ja.tar.gz
# cd php-3.0.15-i18n-ja
# ./configure --enable-i18n --enable-mbregex
# make
```

- デフォルトではCGI版が生成される。
- セキュリティー上Webツリーの外に置くことが有効
 - 実行時、他のCGIスクリプト同様
#/usr/local/bin/php がファイルの先頭に必要
- 詳しくはマニュアルのセキュリティーの章参照

Apacheモジュール版インストール

```
# tar xvzf php-3.0.15-i18n-ja.tar.gz
# tar xvzf apache_1.3.x.tar.gz
# cd apache_1.3.x % ./configure
# cd ../php-3.0.15-i18n-ja
# ./configure --with-apache=../apache_1.3.x --enable-i18n --enable-
  mbregex
# make % make install % cd ../apache_1.3.x
# ./configure --activate-module=src/modules/php3/libphp3.a
# make
# make install
```

- Apacheのモジュールとして取り込まれる
- Apacheのリコンパイルが必要

ApacheDSO版インストール

```
% tar xvzf apache_1.3.x.tar.gz
% cd apache-1.3.x
% ./configure --enable-shared=max
% make
% make install
% cd php-3.0.15-i18n-ja
% ./configure --with-apxs=/usr/local/apache/bin/apxs --enable-i18n • --
  enable-mbregex
% make
% make install
```

- Jserve等との共存も可能
- PHP自体のバージョンアップも簡単
- モジュールが動的に読み込まれる

オプション指定

- 使用可能DB
 - Oracle,PostgreSQL,Sybase,Informix,mSQL等
- PHPのコンパイル時にオプション指定を行う。
 - --with-pgsql=DIR
 - --with-oracle=DIR
 - --with-mysql=DIR
 - DIRはDBをインストールしたディレクトリ
 - この指定によりPHPの個別のDB関数の使用が可能になる。

PHPプログラミング

PHPスクリプトの開始

```
<? echo (“こんにちは n”); ?>  
<?php echo(“こんにちは n”); ?>  
<script language=“php”>  
    echo (“こんにちは”);  
</script>  
<% echo (“こんにちは”); %>
```

最後のASP風はasp tagsの設定が必要

変数の定義

- 変数定義時に明示的な型定義を必要としません。
- 変数の実行時に定義されます。
\$var = “0”; 文字列として定義
\$var += 1; 整数として再定義
\$var = \$var + 1.3 倍精度として再定義
- 強制的に定義することも可能(settype)

簡単なコマンド

- if
- 命令の条件実行を行う
- ```
if ($a > $b) {
 print “aはbよりも大きい”;
} else {
 print “aはbよりも大きくない”;
}
```

## 簡単なコマンド

- for
- **条件により繰り返す。**
- for (式1; 式2; 式3) 文
- ```
for ($i = 1; $i <= 10; $i++) {  
    print $i;  
}
```

変数の受け渡し

- フォームが PHP スクリプトに投稿された時、渡された変数は PHP により自動的に PHP で使用可能になる。

input.html

```
<form action="test.php3" method="post">
Name: <input type="text"
name="name"><br>
<input type="submit">
</form>
```



test.php3

```
<?
echo "$name";
?>
```

その他の便利なところ

- Apacheの生成する変数や環境変数にすぐアクセスできる
 - SERVER_NAME等
- 変数名を動的に定義
 - \$a = "test"; // \$a と \$testが定義される。
 - DBのフィールド名の変数が簡単に定義できる。

その他の便利なところ

- クッキーをサポート
SetCookie ("MyCookie[]", "Testing",
time()+3600);
- その他にも色々
マニュアルを見てください

PHPサンプルソース解説

- 前提条件
- PostgreSQLインストール済み
- Sampleデータベース
- Customerテーブル作成

ex.

ID	name	email	tel
----	------	-------	-----

スクリプトの流れ

1. データベースへの接続
2. SQL(参照)の実行
3. 行数、列数の取得
4. 表も見出しを作成
5. データ一覧を作成
6. 結果の開放
7. データベースの開放

postgreSQL関数

- pg_Connect -- コネクションを開く
- pg_exec -- クエリーを実行する
- pg_NumRows -- 行数を返す
- pg_NumFields -- フィールド数を返す
- pg_FieldName -- フィールドの名前を返す
- pg_Result -- 結果IDから実際の値を返す
- pg_FreeResult -- メモリを開放する

```
1:<html>
2:<body>
3:<?PHP
4://データベースの接続
5: if (@pg_connect( "", "", "sample") == false) {
6:  echo "データベースの接続に失敗しました。 ";
7:  exit;
8: }
```

```
9://SQL文作成
10:$sql = "select * from customer";
11://SQLの実行
12:$result = pg_exec($sql);
13: if ($result == false) {
14:  echo "SQLの実行に失敗しました。 ";
15:  echo "$sql";
16:  exit;
17: }
```

```
18://行と列の取得
```

```
19:$rows = pg_numrows($result);
```

```
20:$fields = pg_numfields($result);
```

```
21:
```

```
22://フィールド名を見出しに設定
```

```
23:echo "<table border>¥n";
```

```
24: echo "<tr>¥n";
```

```
25:  for ($i=0;$i<$fields;$i++) {
```

```
26:   $name = pg_fieldname($result,$i);
```

```
27:   echo "<th>$name</th>¥n";
```

```
28: }
```

```
29:echo "</tr¥n>";
```

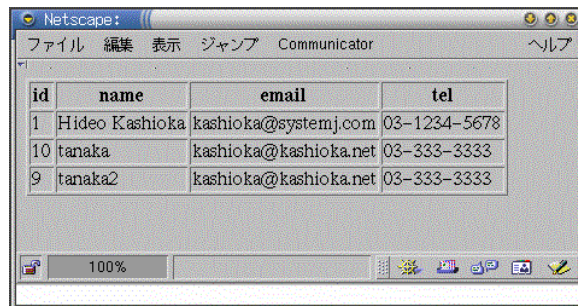
```
30:
```



```
31://顧客データベースから抜き出したデータを
32://一行づつ表として表示する
33:for ($j = 0;$j < $rows;$j++) {
34: echo "<tr>¥n";
35: for ($k = 0;$k<$fields;$k++) {
36:  $target = pg_result($result,$j,$k);
37:  echo "<td>$target</td>¥n";
38: }
39: echo "</tr>¥n";
40:}
```

```
41:echo "</table>¥n";
42:
43://結果の解放
44:pg_freeresult($result);
45:
46://データベースを閉じる
47:pg_close();
48:?>
49:</body>
50:</html>
```

実行結果



id	name	email	tel
1	Hideo Kashioka	kashioka@systemj.com	03-1234-5678
10	tanaka	kashioka@kashioka.net	03-333-3333
9	tanaka2	kashioka@kashioka.net	03-333-3333

ブラウザのソース表示をしてみると

```
<html><body>
<table border>
<tr>
<th>id</th>
<th>name</th>
<th>email</th>
<th>tel</th>
</tr><tr>
<td>1</td>
<td>Hideo Kashioka</td>
<td>kashioka@systemj.com</td>
<td>03-1234-5678</td>
</tr>
<tr>
<td>10</td>
<td>tanaka</td>
<td>kashioka@kashioka.net</td>
<td>03-333-3333</td>
</tr>
<tr>
<td>9</td>
<td>tanaka2</td>
<td>kashioka@kashioka.net</td>
<td>03-333-3333</td>
</tr>
</table>
</body></html>
```

普通のHTMLファイルとして
ブラウザは受け取る

サンプルの改造

- 一覧にリンクをつけて参照や更新
Koushin.php3?id=\$id
- 入力のチェック
Javascript・正規化
- 検索画面
- HTMLのデザイン
- 1 頁の表示件数

まとめ

- 安く
 - ライセンス料
 - ハード費用
- 早く
 - 軽いホームページ
 - スクリプトなのでデバッグも簡単

結論

どんどん使いましょう

連絡先

- PHPユーザ会
 - <http://www.php.gr.jp>
- 株式会社システムジェイ
 - <http://www.systemj.com>
- お気軽に
 - kashioka@systemj.com

PHPの基本的活用法 インストールとDB連携

株式会社システムジェイ
取締役副社長 柏岡秀男

Kashioka@systemj.com