

データベースゼミ3回目

指導 谷村 吉田, 若森, 西村

1 PHP の言語仕様

PHP とは, HTML 埋め込み型のサーバサイドスクリプトです. これを用いると WWW と DB を関係させたアプリケーションを比較的簡単に作成することが出来ます.

1.1 HTML 埋め込み型言語

PHP スクリプトでは, HTML 文書の中にスクリプト言語を埋め込むことが出来ます.

```
<HTML>
<BODY>
    <? echo("ここは PHP コードの部分です.\n");?>
</BODY>
</HTML>
```

また, ファイル全体を PHP スクリプトにし, 必要に応じて HTML の出力を行うこともできます. このように, HTML 中にスクリプトを埋め込む方法は, 以下の 3 種類あります. これらの 3 種類を混在させることも可能です.

- <? ~ ?> で囲む
- <?php ~ ?> で囲む
- <?script language='PHP'> ~ </script> で囲む

例えば, <?php ~ ?> の場合, 次のように表します.

```
<?php    echo("ここは PHP コードの部分です.\n");    ?>
```

このように表すことが出来ますが, 文の要素間にあるホワイトスペース (空白, タブおよび改行) は無視されるので, この場合次のように書き換えることも出来ます.

```
<?php
    echo("ここは PHP コードの部分です.\n");
?>
```

他の 2 種の場合も同様の形式で書くことが出来ます.

1.2 コメントの挿入

ステートメント¹の外側に、C 言語または、Java 形式のコメントを書くことが出来ます。

```
echo("処理 1"); /* C 言語形式のコメント */  
echo("処理 2"); // Java 形式のコメント
```

1.3 式

PHP の文法において、もっとも重要な要素は「式」です。すべての式は値を持ちます。

定数

定数には、数値定数と文字定数、および文字列定数があります。数値定数は、算術表記で表されるもので、文字定数は'A'や'l'などのように 1 バイト文字をシングルクォートでくくったものです。

これに対して、文字列定数は"文字列"や'文字列'と言ったものです。ダブルクォート(")でくくった文字列定数の場合、その文字列中に変数名が含まれていれば、その変数を値に展開してから評価します。一方、シングルクォーテーションでくくったものはそのままのかたちで評価され、変数展開は行われません。

変数

変数はシンボルの前に\$ (ドル記号) をつけて表します。例えば、\$a = 5 は \$a という変数に 5 という整数値を代入することを表しますが、同時に 5 という値を持つ式でもあります。

関数

ユーザ定義関数を作ることが出来ます。関数名の命名規則は変数と同様ですが、先頭に\$をつけません。逆に言えば、先頭に\$のつかないシンボルを見つけると PHP はそれを組み込み済みの予約語、シンボル定数、組み込み関数、またはユーザ関数の何れかであると見なします。

1.4 演算子

C や perl から取り組んだ、多くの演算子がサポートされています。たとえば、算術演算子の加算の場合、\$a+\$b という形式に表されます。同様の形式で、算術演算子、比較演算子、論理演算子が表されます。代入演算子についても、ほぼ C や perl と同じです。左辺は変数でなければなりません。

\$a += \$b	\$a = \$a + \$b	\$a に \$b の値を代入
\$a % \$b	\$a = \$a % \$b	\$a / \$b の剰余を \$a に代入
\$a .= \$b	\$a = \$a \$b	連結代入で \$a に \$b の値を連結

1.5 変数と配列

PHP の場合、整数、倍精度実数、文字列、配列、オブジェクトの型をサポートしますが、プログラマは、それぞれの型を明示的に宣言する必要はありません。その文脈に応じて、実行時に PHP が型を決定します。また、変数を初期化する時は、単に値を代入だけです。一般のコンパイラ言語のように、変数を使用するに先立って変数の定義（領域の確保）を行う必要はありません。

¹ 処理の単位。一つ以上の式から構成され、セミコロンで終わる。

```
$names[]="Yoshida";      //$names[0]="Yoshida"
$names[]="Nishimura";    //$names[1]="Nishimura"
    または ,
$names=array("Yoshida" ,"Nishimura");
```

上にように、値の連続的代入または、array() 関数のいずれかを用いて配列の初期化を行います。

1.6 制御構文

制御構文に関しては、ほぼ C 言語と同様で、if - else 文、while 文、For 文、Break、Switch など同様の形式で使うことが出来ます。

2 データベースとの関係

前回までのゼミでは、PostgreSQL の単独の使い方を説明しました。次に、PHP から PostgreSQL へのアクセスを行います。

2.1 準備

まず、ここで使用するテーブル名を ex1 とします。このテーブルの持つ属性 (カラム) id と name です。テーブルを示すと、以下のようになります。

```
      id|name
-----+-----
16960034|KyumaAkihiro
16970001|AnanHidehumi
16970005|ArugaHiroshi
16970013|GotoYoshiyuki
```

デフォルト状態では、アクセスできるユーザは権限を持つユーザのみになるので、SQL コマンドで、"www-data" ユーザのアクセスを許可します。grant を使って次のように与えます。

```
semi=>grant all on ex1 to "www-data";
```

これで、アクセスが許可されます。

2.2 PHP でテーブル内容を表示

ex1 テーブルを PHP を使って表示します。次のような PHP のプログラムを使用すると、ex1 のテーブルが表示されます。

```
<html>
<head><title>Example 1</title></head>
<body>
```

```

<?php
    dl("pgsql.so");
// データベースに接続する
    if (@pg_connect("", "", "semi") == false) {
        print("データベースに接続できませんでした。");
        exit;
    }
// SQL の文の生成
    $sql = "select id as id , name as name from ex1 order by id";
    $result = pg_exec($sql);          // select を実行
    if ($result == false) {
        printf("SQL: \"$sql\"の実行に失敗しました。");
        exit;
    }
    $rows = pg_numrows($result);      // 行数を取得
    $columns = pg_numfields($result); // 列数を取得
// 表を書く
    print("<table border>\n");
    for ($j = 0; $j < $rows; $j++) {
        if ($j == 0) {
            print("<tr>");
            for ($i = 0; $i < $columns; $i++) {
                $str = pg_fieldname($result, $i); // 列名の取り出し
                print("<th>$str</th>");
            }
            print("</tr>\n");
        }
        print("<tr>");
        for ($i = 0; $i < $columns; $i++) {
            $str = pg_result($result, $j, $i); // データの取り出し
            print("<td>$str</td>");
        }
        print("</tr>\n");
    }
    pg_freeresult($result);           // 検索結果の解放
    pg_close();                       // データベースとの接続切断
    print("</table>\n");
?>
</body>
</html>

```

pg_ で始まる関数が PostgreSQL にアクセスするための関数です。上に示した,ex1.php3 のプログラムで使われた関数を示します。

表 1:

• pg_connect("ホスト名","ポート番号","データベース名")	PostgreSQL に接続する。 ホスト名,ポート番号は通常"" (空文字列)で与える。
• pg_exec(SQL 文)	引数で与えられた SQL 文を実行する。
• pg_numrows()	pg_exec() の結果を引数とし, 検索された行数を返す。
• pg_numfields()	pg_exec() の結果を引数とし, 列数を返す。
• pg_fieldsname()	pg_exec() の結果と列番号 (0 から数えます) を引数とし, 対応する列の名前を返す。
• pg_result(pg_exec() の結果, 行, 列)	pg_exec() の結果から指定行, 列のデータを返す。
• pg_freeresult()	pg_exec() の作成したメモリ中の検索結果を解放。
• pg_close()	pg_connect() で確立したデータベースとの接続を切断する。
• \$PHP_SELF	PHP が自動的にセットする変数で, 実行中の自分自身のスクリプトの URL が設定される。

3 フォームについて

PHP スクリプトを書き始める前にまず HTML のフォームについて説明します。フォームとはブラウザから Web サーバへデータを送るための手段を提供するための HTML のタグです。一般的な WebDB はフォームをとおして, DB に検索条件のデータを渡しています。

3.1 フォームの開始

フォームタグは以下の書式で始めます。

```
<FORM [METHOD = POST | GET] ACTION = "URL" | "mailto:メールアドレス">
```

このタグから次に見つかった</FORM>タグまでが1つのフォームとしてブラウザに認識されます。このフォームで入力された値が変数 = 値のリストとしてサーバ側に送信されます。

METHOD は変数をサーバに渡すための方式を指定し, 省略するとGET となります。GET は変数リストを URL の末尾に?変数名 1 = 値 1&変数名 2 = 値 2&... の形式で追加する方式です。POST は変数リストを標準入力経由で渡す方式です。

ACTION は変数リストをどこに渡すかを指定します。通常は URL として CGI プログラム名や PHP スクリプト名を指定します。

3.2 入力エリアやボタンの表示

データを入力する 1 行のテキストボックスやチェックボックス，ラジオボタンなどを表示するためには以下のような書式を用います．

```
<INPUT TYPE = "表示形式" NAME = "フィールド名">
```

TYPE 属性によって，<INPUT>タグの表示，非表示や実行形式を選択します．NAME 属性は入力したデータを引き渡す時のフィールド名を指定しています．この他にもテキストエリアの大きさを指定したりする属性がありますが，以下の Sample を見て参考にしてください．

```
<FORM ACTION = "cgi-bin/bbs.cgi" METHOD = "POST">
テキストボックス<INPUT NAME = "name"><P>
サイズ指定のテキストボックス<INPUT NAME = "name"><P>
チェックボックス<INPUT TYPE = "checkbox" NAME = " cbox " VALUE = "default">
送信ボタン<INPUT TYPE = "submit" VALUE = "送信">
リセットボタン<INPUT TYPE = "reset" VALUE = "リセット"><P>
</FORM>
```

このフォームはcgi-bin/bbs.cgi に入力データを渡します．このファイルを IE で見た場合，図 1 のようになります．

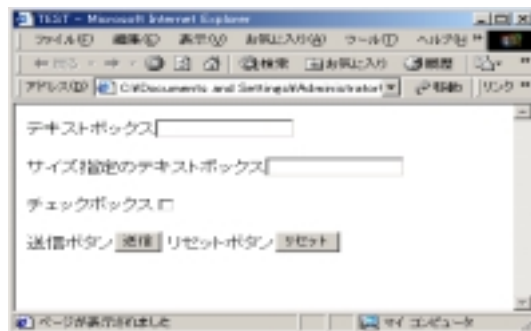


図 1: フォームの画面

TYPE 属性がsubmit は送信実行ボタンを表示します．このボタンは ACTION 属性で設定した動作を実行します．この例では，bbs.cgi にフォームデータを転送しています．

このようにしてフォームを作成し，PHP スクリプトへデータをわたし，WEB データベースを利用します．

4 練習問題

以下に 1.2.2 で使用した DB にアクセスできる PHP コードを埋め込んだ HTML ソースを表示します．

1. ソースの SQL の生成部分 (15 行目から) を書き換えて，name だけを表示するものを作成してください．
2. フォームの生成部分 (8 行目から)，SQL の生成部分 (15 行目から) を書き換えて，"id > ~" 型

の検索の行えるものを作成してください。

3. 同様に, "name" に含まれる文字列での検索を行えるものを作成して下さい。

```
1 <html>
2 <head><title>DB Example</title></head>
3 <body>
4 <?php
5 dl("pgsql.so");
6 @pg_connect("", "", "semi");
7
8 function printForm(){                // 検索フォームの生成
9     global $PHP_SELF;
10    print("<form action=\"".$PHP_SELF.\" method=\"post\">\n");
11    print("<input type=\"submit\" name=\"check\" value=\"Search\">\n");
12    print("</form>\n");
13 }
14
15 function makeSQL() {                // SQL 文の生成
16     $sql = "select * from ex1";      ==
17
```

```
42     }
43     pg_freeresult($result);          // 検索結果の解放
44     print("</table><BR>\n");
45 }
46 if($check)                          // 検索結果の表示
47     doSelect(makeSQL("id"));
48 else                                // 検索フォーム
49     printForm();
50 pg_close();
51 ?>
52 </body>
53 </html>
```