

データベース S 演習資料

第 2 回 PHP による Web インターフェース

九州工業大学 情報工学部 システム創成情報工学科
講義担当：尾下真樹

1. Web と PHP と PostgreSQL

ウェブページは、HTML (Hyper-Text Markup Language) という言語を使用して記述する。HTML は、通常のテキストファイルにタグを付け加えて、リンクや文章の構造の情報などを追加したものである。通常は HTML を使って記述されたウェブページをサーバに用意しておき、そのページの URL を指定することでページが表示される。しかし、この方法では、あらかじめ作成してあるページしか表示できない。CGI や PHP などの技術を使用すると、掲示板などの動的に生成されるページを実現することができる。CGI は、perl や C 言語などのプログラムを作成しておき、そのプログラムを使って動的にウェブページを生成する技術である。

一方、PHP では、HTML で記述されたウェブページの一部に PHP を使ったスクリプトを記述することができ、ウェブページをクライアントに転送する際にサーバ上でスクリプトが実行され、動的にページを生成される。そのため、ページ全部をプログラムから出力する必要がある CGI と比較して、PHP では必要な箇所のみを動的に生成すれば良いので記述が簡単になる。

PHP にはいろいろな関数が用意されており、それらを利用することで、PostgreSQL サーバと通信し、SQL によるクエリ（問い合わせ）を行って結果を取得するようなページなども、比較的簡単に作成できる。ちなみに、同じスクリプト言語でも、JavaScript などは、サーバ上ではなくクライアントのブラウザ上で実行されるものであり、役割が異なる。

2. HTML の書き方

HTML は、一般的には、".html", ".htm" などを拡張子とするテキストファイルとして作成する。HTML は、基本的には通常のテキストファイルと同じような感じで、テキストエディタを使って記述することができる。

<A> ~ のように、開始タグと終了タグを用いてテキストの一部を囲むことによって、そのテキストの属性や配置などを指定する。タグは何重でも入れ子にできる。また、開始タグによっては、のようにタグの属性が指定できる。

以下、最小限のタグと、その主要な属性を紹介する。タグや属性は、大文字と小文字は区別しないのでどちらで書いても構わない。

なお、HTML では、改行や空白は無視される（HTML 上で改行したり空白が空けられたりしていても、ブラウザ上では連続して表示される）ので、もし行を空けたい場合は、明示的に改行タグを入れる必要がある。

```
<HTML></HTML>, <HEAD></HEAD>, <BODY></BODY>
```

HTML ファイルは、<HTML> <HEAD> ヘッダ部分 </HEAD> <BODY> 本文 </BODY> </HTML> という構成をとる。ヘッダ部分には、ページのタイトルや製作者の情報などを記述する。本文には、ブラウザ上に表示されるメインの HTML テキストを記述する。

<TITLE></TITLE>

ヘッダ部分に記述して、そのページのタイトルを指定する。タイトルは、通常、ブラウザのタイトルバーに表示される。分かりやすいように適切なタイトルをつけておくことが期待される。

タグで囲まれた範囲がハイパーリンクとなる。ブラウザ上でその範囲をクリックすると、*url* で記述されたウェブページにジャンプする。

url は相対指定もでき、*url* にファイル名だけを書けば、現在の HTML ファイルと同一ディレクトリから指定したファイルが探されて表示される。ページ全体を引越する場合などを考えて、外部へのリンク以外は、なるべく相対指定で記述して置いたほうが良い。

改行。

<HR>

水平な区切り線を入れる。

<TABLE></TABLE>

表を作成する。表の中身は、下記の<TR><TD>タグを使用して、各行・セルを記述する。例えば、2行×3列の表は下記のように書ける。

```
<TABLE>
  <TR>
    <TD> 1行1列目</TD>
    <TD> 1行2列目</TD>
    <TD> 1行3列目</TD>
  </TR>
  <TR>
    <TD> 2行1列目</TD>
    <TD> 2行2列目</TD>
    <TD> 2行3列目</TD>
  </TR>
</TABLE>
```

この例のように、適宜インデントなどを入れると見やすくなる。

<TR></TR>

表の1行を表す。

<TD></TD>, <TH></TH>

表の1つのセルを表す。<TH>は中身が強調表示されるので、見出しなどに使える。

```
<UL></UL>, <LI>
```

箇条書き。の内側に、それぞれの項目をで記述する。入れ子にすることも可能。
 の代わりに にすると、番号付き箇条書きになる。

その他、HTML の詳しい文法については、適当な本やウェブページを参照すること。

例えば、ウェブページでは、下記の URL のサイトなどが参考になる。

<http://www.tohoho-web.com/www.htm>

3. PHP の書き方

PHP は、HTML の内部に混在させて記述することができる。PHP を含む HTML は、拡張子を “.php” “.phtml” などとする。全 HTML ファイルに対して PHP の処理を行うと効率が悪いので、ウェブサーバは、あらかじめ指定された “.php” “.phtml” などのファイルに対してのみ PHP プログラムの実行を行うようになっている。どの拡張子のファイルを対象とするかは、サーバの設定によって異なる。本演習では、PHP プログラムを含む HTML の拡張子は、 “.php” とすることにしている。

HTML 内で PHP の範囲を指定するための方法としては、以下のようないくつかの書き方があり、どの書き方を使っても構わない。本資料のサンプルコードでは、一番上の書き方を採用している。

```
<?php ... ?>
```

```
<script language="PHP"> ... </script>
```

```
<% ... %> (初期設定ではこの方法は使えない)
```

```
<? ... > (初期設定ではこの方法は使えない。古い書き方なので推奨されていない)
```

3.1. 変数

PHP は、Java や C/C++ などと基本的な書き方は同じなので、これらのプログラミング言語に慣れていれば、PHP プログラムを書くのはそれほど難しくはない。ただし、変数の扱いについてはかなり異なっている。まず、PHP では、変数をあらかじめ宣言する必要はない。初めてその変数を使うときに、その変数が自動的に定義される。また、変数名は必ず \$ で始める決まりになっている。\$ で始まっている単語があれば、PHP はそれを変数として扱う。

なお、PHP では変数の型を明示的に指定する必要はない。型は代入する値の種類に応じて自動的に適切な型に変換される。Java や C/C++ のようにキャストを使って望む型に変換することもできる。

3.2. 演算子

基本的な演算子は他の言語と同じ。

ただし、文字同士の結合をするための演算子として、「.」（ピリオド）が用意されている点が異なる。

```
$x = "1234" . "5678"; // 変数 X には、文字列型の "12345678" が入る
```

Java と同じように「+」を使って連結しようとする、それぞれの文字が一度数値型にキャストされて、数値同士として足し算が行われてしまうので注意すること。

```
$x = "1234" + "5678"; // 変数 X には、整数型の (6912) が入る
```

3.3. テキスト出力

PHP でテキスト出力処理を記述すると、その出力結果のテキストが **HTML** に埋め込まれる。

```
void print( 文字列 );
```

指定した文字列を単純に出力する。

```
void printf( 書式付文字列, 値 1, 値 2, ... );
```

書式付きで文字列を出力する。C の同名の関数と同様の機能を持つ。

書式文字列には文字列のどの場所に文字や数値を埋め込むかを指定しておき、後ろの 値 1, 値 2, ... がそこに埋め込まれて出力される。

例えば、

`printf("変数 x の値は%d です。", $x);` のように記述すると、変数 `x` の値が `%d` に埋め込まれて表示される。実数型の場合は `%f`、文字列型の場合は `%s` で指定できる。また、`%12s` のように最大の桁数を指定することも可能。

書式文字列に複数の埋め込みを記述した場合は、左から順番に値 1, 値 2, ... に対応して表示される。

```
string sprintf( 書式付文字列, 値 1, 値 2, ... );
```

書式付きで文字列を作成する。`printf` 同様の形式で文字列を作成し、その文字列を返す。クエリなどで動的に文字列を生成する必要のある時に役に立つ。

4. PHP からの PostgreSQL の操作

PHP は非常に高機能で、いろいろな関数ライブラリを持っている。その中に、PostgreSQL を操作するためのライブラリも含まれる。PostgreSQL 関連の関数は、関数名が全て `pg_` で始まるようになっているので分かりやすい。以下、基本的な関数の使い方を示す。

```
resource pg_connect( "host=ホスト名 dbname=データベース名 user=ユーザ名 ..." );
```

PostgreSQL サーバのデータベースに接続する。引数には、接続の設定をテキスト形式で記述する。設定は、「オプション=値」の形式を空白で区切って指定する。

主なオプションには、`host`, `port`, `dbname`, `user`, `password` などがある。

`host` には PostgreSQL サーバの動いているコンピュータの名前を指定する。ウェブサーバと PostgreSQL サーバが同一のコンピュータの場合は省略できる。

`dbname` には、接続するデータベースの名前を指定する。必須。

接続が成功すると、`pg_connect()` は接続の識別番号を返す。この識別番号を記録しておいて、以後の通信では、この識別番号を指定する。

```
resource pg_query( string query );
```

問い合わせを発行し、結果の格納されている領域の識別番号を返す。問い合わせの結果には、この識別番号を指定してアクセスできる。

```
int pg_num_rows( resourcece result );
```

問い合わせの結果のレコード数（行数）を返す。

```
int pg_num_fields( resourcece result );
```

問い合わせの結果の属性数（列数）を返す。

```
mixed pg_fetch_result( resourcece result, int i, int j );
```

問い合わせの結果から i 行 j 列目のデータを取得する。戻り値の型は、入っているデータの型により異なる。

```
bool pg_free_result( 識別番号 );
```

問い合わせの結果が格納されたメモリを開放する。スクリプトが終了したら自動的に全領域は開放されるので、書かなくても構わない。

```
bool pg_close( 識別番号 );
```

PostgreSQL サーバとの接続を閉じる。

5. PHP によるインターフェース作成演習（1）

5.1. データベースの準備

今回の演習では、前回作成したデータベースに、ウェブ経由でアクセスすることができるような、PHP を含むウェブページを作成する。

まずは、準備段階として、前回の資料の方法に従って、データベースを構築する（前回の課題が完了していれば、このステップは不要）。ファイル `employee_table.txt` と `employee_data.txt` を使って、テーブル `employee` を構築する。また、`department.txt` を使って、テーブル `department` を構築する。これらのファイルは、講義のウェブページからダウンロードできる。

ここまでの手順で、以下のようなテーブル `employee` と `department` ができているはずである。

```
dbname=# select * from employee;
 id | dept_no | name | age
-----+-----+-----+-----
 0001 | 01 | 尾下 真樹 | 27
 0002 | 02 | 下戸 彩 | 17
 0003 | 03 | 本村 拓哉 | 30
 0004 | 01 | 宇田 ヒカル | 20
 0005 | 01 | 織口 裕二 | 35
 0006 | 02 | 松浦 亜矢 | 17
 0007 | 03 | 山田 一郎 | 30
(7 rows)
```

```
dbname=# select * from department;
 dept_no | name
-----+-----
    01   | 開発
    02   | 営業
    03   | 総務
(3 rows)
```

次に、これらのテーブルに、PHP プログラムからアクセスできるように設定を変更する。テーブルは、初期状態では、テーブルを作成したユーザしかアクセスすることができない。PHP プログラムは、ウェブサーバの子プロセス用のユーザ **www** の権限で実行される（ユーザ **www** がデータベースを操作することになる）ので、そのままでは自分の作成したテーブルにアクセスできない。そこで、**psql** 上から、**GRANT** コマンドを使用してユーザ **www** にテーブルへのアクセス権限を与える。

```
grant operation on tname to username;
```

テーブルの操作権限を与える。

tname には、テーブルの名前を指定する。**username** には、権限を与えるユーザ名を指定する。

operation には、ユーザに許す操作の種類を指定する。**ALL** を指定すると、あらゆる操作を許す。**SELECT**（表示）、**INSERT**（追加）、**UPDATE**（変更・削除）など個別の操作のみを許可することもできる。

以下に、上記の2つのテーブルにユーザ **www** によるアクセス権限を設定する操作の例を示す。

```
username@pcXX ~
# psql dbname -h popuradb.ces.kyutech.ac.jp
Welcome to psql 7.3.2, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit

dbname=# grant ALL on employee to www;
GRANT
dbname=# grant ALL on department to www;
GRANT
```

なお、今回の演習環境では、ウェブサーバのプロセスを実行するときのユーザ名は **www** であるが、ウェブサーバの設定によってこのユーザ名は異なるので、他の環境を使用するときには注意する必要がある。

5.2. ウェブページの準備

次に、自分の作成した **html** ファイルがウェブサーバ経由で表示できることを確認する。

CL の環境では、ホームディレクトリ（**¥Z**）に **public_html** というディレクトリを作成して、そこにファイ

ルを置くと、<http://popuradb.ces.kyutech.ac.jp/~ユーザ名/ファイル名> という URL でアクセスできるようになっている。

popuradb.ces.kyutech.ac.jp は、学科の演習用のウェブサーバの名前である。今回は、データベースサーバと同じコンピュータでウェブサーバが動いている。なお、popuradb.ces.kyutech.ac.jp には、学科外からはアクセスできない設定になっているので、注意すること。

上記のように、ホームディレクトリ（ $\$Z$ ）に `public_html` というディレクトリを作成して、サンプルの `html` ファイル（`menu.html`）を置いてみる。サンプルの `html` ファイルは、講義のページに置いてあるものを利用して構わない（`emacs` などを使って自分で作成しても良い）。

`html` ファイルを置いたら、ウェブブラウザを起動して、

<http://popuradb.ces.kyutech.ac.jp/~自分のユーザ名/menu.html>

という URL を入力し、ページが正しく表示されることを確認せよ。

5.3. テーブルの一覧表示

テーブル `employee` の内容を一覧表示するための PHP ページを作成する。

`employee_list.php` というファイル名でサンプルの PHP が用意されているので、このファイルをもとに作成する。

まず、`employee_list.php` をダウンロードして、`menu.html` と同じディレクトリに置く。`employee_list.php` の中で、データベースに接続する部分の処理はユーザごとに異なるので、そこだけ `emacs` などを使用して書き換える。具体的には、16 行目の `pg_connect` 関数の引数の文字列の中の `your_db_name` を、自分のデータベースの名前（自分のアカウント名）に書き換える。

ファイルの修正が終わったら、ブラウザに `employee_list.php` の URL を入力するか、最初にコピーした `menu.html` から「従業員の一覧表示」をクリックして、`employee` の一覧が表示されることを確認してみよ。

5.4. テーブルの一覧表示の SQL の修正

PHP (+SQL) プログラミングの練習として、従業員一覧が部門ごとに表示されるように（部門でソートして表示されるように）、`employee_list.php` を修正し、動作確認をしてみよ。

どのようにプログラムを修正したら良いか、自分で考えて、部門ごとに表示されるようにすること。