

データベース 演習資料

第 2 回 PHP による Web インターフェース開発演習 (1)

九州工業大学 情報工学部
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を作成するときと同様に、テキストエディタを使って作成できる。

<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、タグの属性を指定できる。

以下、HTML ファイルを記述する上で必要となる最小限のタグと、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 ファイル中で改行や空白があっても、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。ページ全体を引越し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。

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。全ての 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” などの拡張子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、サーバの設定により異なる。本演習では、PHP プログラムを含む HTML の拡張子は、 “.php” とする。

HTML 内の PHP プログラム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、どの書き方を用いても構わない。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++ などと基本的な制御構文等は同じであるため、これらのプログラミング言語に慣れていれば、PHP プログラムを書くことはそれほど難しくない。ただし、変数の扱いについてはかなり異なっているため、PHP での変数の扱い方を理解する必要がある。

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 Java や C/C++ などのプログラミング言語と同じである。

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 言語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 は非常に高機能であり、さまざまな関数ライブラリを備えている。その中に、PostgreSQL を操作するためのライブラリも含まれている。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ため、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である場合は省略できる。

`dbname` には、接続するデータベースの名前を指定する。必ず指定する必要がある。

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( resource connection, string query );
```

問い合わせを発行して、結果が格納されている領域の識別番号を返す。1 つ目の引数には、`pg_connect` 関数の戻り値である接続の識別番号を渡す。2 つ目の引数には、SQL 文を表す文字列を渡す。問い合わせの結果は、この識別番号を引数として渡して下記の関数を呼び出すことで、アクセスできる。

```
int pg_num_rows( resource result );
```

問い合わせの結果のレコード数（行数）を返す。

```
int pg_num_fields( resource result );
```

問い合わせの結果の属性数（列数）を返す。

```
mixed pg_fetch_result( resource 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( resource result );
```

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、必ずしも実行しなくとも構わない。

```
bool pg_close( resource connection );
```

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。

前提として、前回までの演習により、下記のような **employee** と **department** の2つのテーブルが作成できているものとする。（**employee** テーブルには、各自が追加したデータも含まれているはずである。）前回の演習が終わっていない場合は、前回の演習を終わらせる。

```
dbname=# select * from employee;
 id | dept_no | name   | age
-----+-----+-----+-----
0001 | 01      | 織田 信長 | 48
0002 | 02      | 豊臣 秀吉 | 45
0003 | 03      | 徳川 家康 | 39
0004 | 01      | 柴田 勝家 | 60
0005 | 02      | 伊達 政宗 | 15
0006 | 03      | 上杉 景勝 | 26
0007 | 01      | 島津 家久 | 35
(7 行)
```

```

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設定となっている。PHP プログラムは、ウェブサーバの子プロセス用のユーザ `apache` の権限で実行される（ユーザ `apache` がデータベースを操作することになる）が、そのままでは自分が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 `apache` 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 `apache` にアクセス権限を設定する操作の例を示す。

```

$ psql -h db.tom.ai.kyutech.ac.jp -U username dbname
Password for user username: ??????
psql (9.5.19, server 9.2.24)
Type "help" for help.

```

```

dbname=# grant ALL on employee to apache;
GRANT
dbname=# grant ALL on department to apache;
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 `apache` 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（ウェブサーバへのファイルのアップロード）

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。

ウェブサーバの自分のホームディレクトリに `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://db.tom.ai.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。

`db.tom.ai.kyutech.ac.jp` は、本演習用の学科のウェブサーバの名前である。今回の演習環境では、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、ウェブサーバへのファイル送受信は、学内のコンピュータからしか行えない設定となっている。そのため、前回の演習と同様、**飯塚キャンパスのネットワークから接続するか、大学外から飯塚キャンパスの VPN を経由して接続する必要がある。**

クライアント端末からウェブサーバに SCP 接続することで、クライアント端末とウェブサーバの間でファイルの送受信を行い、ウェブサーバの自分のホームディレクトリにファイルを置くことができる。ウェブサーバのホームディレクトリに `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`) をアップロードする。サンプルの `html` ファイルは、Moolde の本講義のコースに置いてあるものを利用して構わない。

`html` ファイルをアップロードしたら、ウェブブラウザを起動して、

```
http://db.tom.ai.kyutech.ac.jp/~自分のユーザ名/menu.html
```

のように URL を入力し、アップロードしたファイルが正しく表示されることを確認する。

5.2.1. BYOD 端末の仮想環境 (Ubuntu 環境) でのファイルのアップロード

SCP 接続によるファイル転送を使うことで、本演習で使用するウェブサーバ (`db.tom.ai.kyutech.ac.jp`) に接続して、ウェブサーバに用意されている各自のホームディレクトリに、ファイルを転送できる。

BYOD 端末の仮想環境 (Ubuntu 環境) には、SCP 接続を用いてファイル転送を行う `sftp` コマンドがあらかじめインストールされている。以下のようにユーザ名 (九工大 ID) とホスト名を指定して `sftp` を起動し、リモートサーバに接続することで、ファイルを送受信できる。

```
$ sftp username@db.tom.ai.kyutech.ac.jp
Password: ????????
```

接続時にはパスワードの入力が必要になるので、各自の**九工大 ID のパスワード**を入力する。`psql` を使ってデータベースサーバに接続するときには使用する PostgreSQL ユーザのパスワードとは異なるので、注意する。

`sftp` では、`cd` (リモートディレクトリ移動)、`lcd` (ローカルディレクトリ移動)、`ls` (リモートファイル一覧表示)、`lls` (ローカルファイル一覧表示)、`put` (ファイルのアップロード)、`get` (ファイルのダウンロード)、`exit` (`sftp` を終了) などのコマンドが利用できる。より詳しい使い方を知りたい場合は、`help` コマンドで表示できる説明や `man` プログラムで表示できるマニュアル等を参照する。

5.2.2. BYOD 端末の仮想環境 (Ubuntu 環境) でのファイルのアップロードの例

以下に、クライアント端末の `/home/user/database` フォルダに置かれているファイル (`menu.html`) を、ウェブサーバの `/home/ユーザ名/public_html` にアップロードするときの、ターミナルからの一連の入力の例を示す。

```
$ cd database
$ sftp username@db.tom.ai.kyutech.ac.jp
Password: ????????
sftp> mkdir public_html
sftp> cd public_html
sftp> put menu.html
sftp> ls
```

`/home/user/database` 以外の場所にファイルを置いている場合は、最初の `cd` コマンドの移動先を適宜変更する。`/home/user/` にファイルを置いている場合は、`cd` コマンドでの移動は必要ない。

上記の例では省略されているが、最初に接続するときには、確認のメッセージが表示されるので、表示されている指示通り、`yes` と入力してリターンキーを押す。

パスワード入力時は、他の多くのプログラムと同様、セキュリティのためのキー入力の内容は画面に表示されないが、入力が行われているので、パスワードとエンターキーを入力する。

ディレクトリの作成 (`mkdir public_html`) は、最初に一度だけ行えば良い。

最後の `ls` コマンドで、ファイルの一覧を表示して、正しくアップロードされていることを確認する。

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 PostgreSQL ユーザ名）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、前回の演習でデータを登録した `employee` の一覧が正しく表示されることを確認せよ。

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` を変更し、正しく表示されることを確認する。

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変更する。

付録 1. sftp のエラーへの対処

sftp を実行したときに表示される主なエラーと対処方法は、下記の通り。

Connection timed out （もしくはプログラム実行後に処理が進まない）

飯塚キャンパス内のネットワークや飯塚キャンパスの VPN を経由して接続されていない。

学外から演習を行うときには、飯塚キャンパスの VPN に接続する必要がある。戸畑キャンパスの VPN への接続は不可。

VPN 接続について不明な点があれば、情報基盤センターのオンラインガイド

<https://onlineguide.isc.kyutech.ac.jp/guide2020/index.php/home/vpn>

を参照するか、情報基盤センターの相談窓口で質問すること。

Are you sure you want to continue connecting (yes/no)?

初めて接続する場合は、確認のメッセージが表示されるので、表示されている指示通り、`yes` と入力してリターンキーを押す。

Permission denied, please try again

ユーザ名またはパスワードが間違っている。パスワードは、PostgreSQL ユーザのパスワードではなく、九工大 ID のパスワードを入力する必要がある。

No such file or directory

アップロードしようとしているファイルが、クライアント側のカレントディレクトリに存在しない。

sftp プロンプトから、`lpwd` のコマンドを実行して、現在のクライアント側のカレントディレクトリを確認する。`lcd` コマンドで、クライアント側のカレントディレクトリを移動できる。

付録 2. html・php のエラーへの対処

ウェブブラウザで html・php のページを表示するときに生じる主なエラーと対処方法は、下記の通り。

サイトにアクセスできません、と表示される

URL が間違っているか、飯塚キャンパス内のネットワークや飯塚キャンパスの VPN を経由して接続されていない。後者の場合は、上記の付録 1 の説明を参照する。

Not Found The requested URL ??? was not found on this server.

URL が間違っているか、表示しようとしているファイルがウェブサーバのホームディレクトリの `public_html` の下に正しくアップロードされていない。

sftp でウェブサーバに接続し、以下の通りコマンドを入力して、`public_html` の下にファイルが存在することを確認する。

```
sftp> cd public_html  
sftp> ls
```

`employee_list.php` にアクセスした時に、「データベース接続処理でエラーが発生しました。」と表示される `employee_list.php` 内の接続先の自分のデータベース名を記述する箇所が、正しく修正されていない可能性が高い。

5.4 節の説明に従って、正しく修正したファイルをアップロードする。

`employee_list.php` にアクセスした時に、「クエリー実行処理でエラーが発生しました。」と表示される

自分のデータベースの 2 つのテーブルに、ユーザ `apache` へのアクセス権限が設定されていない可能性が高い。

5.1 節の説明に従って、2 つのテーブルにアクセス権限を設定する。