



中村研究室ゼミ

PHPとデータベースの連携

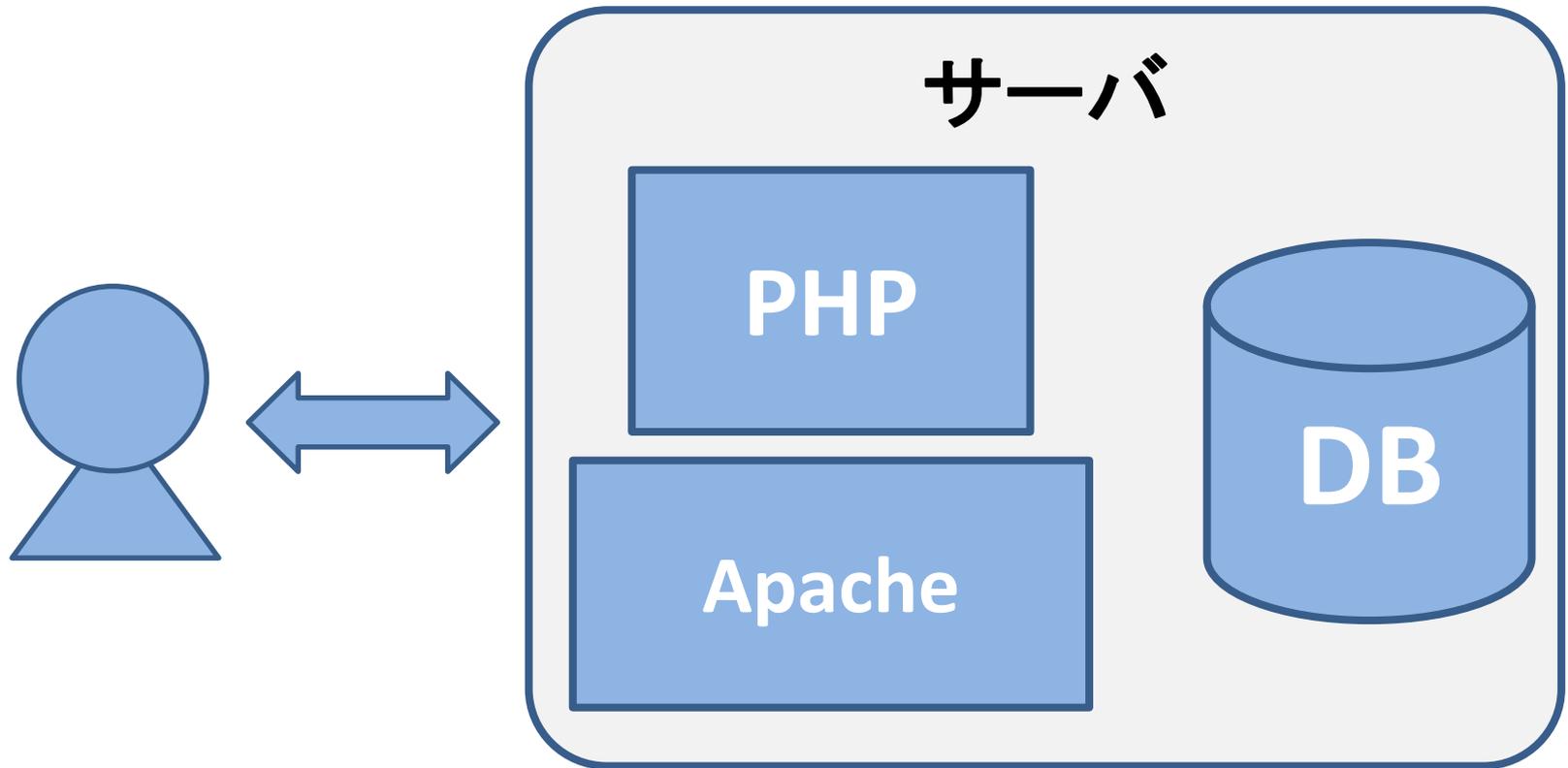
中村 聡史

Web情報システム

- 膨大な情報をいかにして管理するか？
- 膨大な問い合わせをいかにして捌くか？
- ユーザの問い合わせをどう処理するか？



PHPとMySQL





PHPとMySQL

- PHPを利用してMySQLのデータベースを利用
 - 手軽にデータベースを利用可能

```
<?php
$con = mysql_connect("localhost", "ユーザ名", "パスワード") or die;
mysql_select_db( "weather_db" );
$results=mysql_query( "select day,weather from weather_table" );
while( $line = mysql_fetch_array( $results ) ){
    echo $line["day"];
    echo "/";
    echo $line["weather"];
    echo "<br>";
}
mysql_close( $con );
?>
```



PHP と MySQL

- MySQLに接続: `mysql_connect`
- MySQLから切断: `mysql_close`
- データベースを選択: `mysql_select_db`
- SQLを送信: `mysql_query`
- 結果を1行ずつ取得: `mysql_fetch_array`



mysql_connect

- MySQL のサーバに接続する際に使用
- `mysql_connect("サーバのアドレス", "アクセスユーザ名", "アクセスパスワード");`
 - PHPのプログラムから見た時のサーバのアドレスを記入することに注意！ localhostとは自身のサーバ

(例)

```
mysql_connect( "localhost", "root", "" );;
```

```
mysql_connect( "snakamura.org", "nakamura", "satoshi" );
```



mysql_close

- MySQL のサーバとの接続を切断する
- mysql_connect とセットで利用される
- mysql_connet(接続情報);

(例)

```
$con = mysql_connect( "localhost", "root", "" );  
:  
mysql_close( $con );
```



mysql_select_db

- 使用するデータベースを選択する
- 「USE weather_db」と同じ意味

- (例)

```
mysql_select_db( "weather_db" );
```

```
mysql_select_db( "nakamura_db" );
```



mysql_query

- SQL を送信する
- `mysql_query("SQLの内容");`
(注) "" の中に「"」を書く場合は「¥"」とするか「'」を使う
- (例)
`mysql_query("INSERT INTO CUSTOMER
(ID, NAME, AGE) VALUES (5, '中村聡史', 34);");`
`mysql_query("SELECT * FROM weather_table;");`
`mysql_query("SELECT count(*) FROM
weather_table WHERE weather = 'Fine';");`



結果をどう取得するか？

- SELECT の結果は表の形になっている. それをどう取得するか？

No	姓	名	性別	誕生日	所属	...
00001	中村	聡史	男	*****	*****	...
00002	工藤	北斗	男	*****	*****	...
00003	吉田	万里子	女	*****	*****	...
00004	力石	優武	男	*****	*****	...
00005	黒川	恵輔	男	*****	*****	...
00006	山本	黎	女	*****	*****	...
:	:	:	:	:	:	

1行ずつ取得して、それを配列に入れていく



mysql_fetch_array

- 結果を1行ずつ配列として取得する
- `mysql_fetch_array(全ての結果);`

(例)

```
$results = mysql_query( "select day,weather from  
weather_table" );
```

```
$line = mysql_fetch_array( $results );
```

```
echo $line["day"];
```

```
echo $line["weather"];
```

```
$line = mysql_fetch_array( $results );
```

```
echo $line["day"];
```

```
echo $line["weather"];
```



mysql_fetch_array と繰り返し

- mysql_fetch_array は結果の件数分実行しなければならない
- 件数がわからない場合にどう記述する？

```
5 $kekka = mysql_fetch_query( $ret );  
6 echo $kekka[0];  
7 echo $kekka[1];  
8 echo $kekka[2];  
9 $kekka = mysql_fetch_query( $ret );  
10 echo $kekka[0];  
11 echo $kekka[1];  
12 echo $kekka[2];  
13 $kekka = mysql_fetch_query( $ret );  
14 echo $kekka[0];  
15 echo $kekka[1];  
16 echo $kekka[2];  
17 $kekka = mysql_fetch_query( $ret );  
18 echo $kekka[0];  
19 echo $kekka[1];  
20 echo $kekka[2];  
21 $kekka = mysql_fetch_query( $ret );  
22 echo $kekka[0];  
23 echo $kekka[1];  
24 echo $kekka[2];  
25 $kekka = mysql_fetch_query( $ret );  
26 echo $kekka[0];  
27 echo $kekka[1];  
28 echo $kekka[2];  
29 $kekka = mysql_fetch_query( $ret );  
30 echo $kekka[0];  
31 echo $kekka[1];  
32 echo $kekka[2];  
33 $kekka = mysql_fetch_query( $ret );  
34 echo $kekka[0];  
35 echo $kekka[1];  
36 echo $kekka[2];  
37
```

繰り返しにより最後まで取得！

```
40 echo $kekka[2];  
49 $kekka = mysql_fetch_query( $ret );  
50 echo $kekka[0];
```



mysql_fetch_array

- 結果を1行ずつ配列として取得する
- `mysql_fetch_array(結果);`

(例)

```
$results = mysql_query( "select year,month,day from  
weather_table" );  
while( $line = mysql_fetch_array( $results ) ){  
    echo $line["year"];  
    echo $line["month"];  
    echo $line["day"];  
}
```



演習

- 天気データベースに接続し、結果を表示する下記のようなPHPのプログラムを書いてみましょう

```
<?php
$con = mysql_connect("localhost", "", "");
mysql_select_db( "weather_db" );
$results=mysql_query( "select year,month,day,weather from weather_table" );
while( $line = mysql_fetch_array( $results ) ){
    echo $line["year"];
    echo "/";
    echo $line["month"];
    echo "/";
    echo $line["day"];
    echo ": ";
    echo $line["weather"];
    echo "<br>";
}
mysql_close( $con );
?>
```



演習

- 様々なSQLを実際に試してみましよう
- また、結果をTABLEタグなどを用いて整形して表示しましよう
- 文字化けする場合は下記をmysql_connectの後に挿入し、ブラウザの文字コードをUnicodeにしましよう

```
mysql_query("SET NAMES utf8", $con);
```



演習

- 天気データベースに接続し，下記の結果をテーブルなどで提示せよ（グラフで図示してもよい）
 - 北海道 (pref=1) と，東京 (pref=13)，沖縄 (pref=47) の，最高気温平均，最低気温平均を計算してテーブルなどで表示せよ
 - 晴れの日が多い順に，晴れの数と県名を並べて表示せよ
 - 県ごとの最高気温と最低気温の差の平均を求め，その差が少ない順に並べて県名と一緒に提示せよ



PHPで関数はどうする？

- Processingにあった関数を，PHPではどう書く？
 - function に変更されるだけ

```
int getNumberOfDivisor( int num ){  
    int i=1;  
    int count=0;  
    while( i<=num ){  
        if( num%i == 0 ){  
            count++;  
        }  
        i++;  
    }  
    return count;  
}  
  
println( getNumberOfDivisor( 1000 ) );
```

```
function getNumberOfDivisor( $num ){  
    $i=1;  
    $count=0;  
    while( $i<=$num ){  
        if( $num%$i == 0 ){  
            $count++;  
        }  
        $i++;  
    }  
    return $count;  
}  
  
echo getNumberOfDivisor( 1000 );
```



グローバル変数の注意点

- PHPではグローバル変数を関数内で普通に使うことはできない！

```
<?php
$number = 10;

function showNumber(){
    echo $number;
}
?>
```

こういう使い方は駄目！
基本的には引数渡し！
またはglobalと明示的に！

```
<?php
$number = 10;
showNumber( $number );

function showNumber( $num ){
    echo $num;
}
?>
```

```
<?php
$number = 10;

function showNumber(){
    global $number;
    echo $number;
}
?>
```



なるべく関数化を！

- データベースへの接続を1つの関数に
- データベースからの切断を1つの関数に
- データベースとのやりとりを1つの関数に
など



演習

- 作ろうと思っているWebシステムのデータベースを設計し、PHPのプログラムを書いて、そこからデータを取得してみましよう
- また、データの挿入をしてみましよう

演習

- MySQLとPHPを利用した掲示板を作ろう
 - 掲示板用のデータベースを作成
 - Message-ID, Name, Message
 - HTMLまたはPHPで入力フォームを用意
 - INPUT TYPE
 - POSTでメッセージを受けて、その情報をデータベースに格納し、投稿された全データをテーブル形式で表示せよ



手順(1)

- データベース(BBS_DB)とテーブル(BBS_TABLE)を作成
 - ID: 整数値 (PRIMARY KEY)
 - Name: TEXT (utf8_unicode_ci)
 - Message: TEXT (utf8_unicode_ci)
 - ほかに日時やメールアドレスなどを入れても可

ID	Name	Message
1	くどう	ちわっすくどーです
2	わたなべ	なじよしたい？
3	くどう	！？



手順(2)

- HTMLを利用して入力フォームを作成する
 - FORMタグ
 - methodはPOSTにしましょう
 - actionで送信先を指定しましょう
 - INPUTタグ
 - type=textで1行のテキスト, type=submitで送信ボタン
 - nameでその入力ボックスの名前を指定
 - name="naamae", name="message" など
 - 複数行入力したい場合は, TEXTAREAタグを利用
 - `<textarea col=20 row=3 name="message"></textarea>`
 - name="message"



手順(3)

- データを受け取るPHPを作成する
 - `$_POST["hoge"]` で受け取る
 - (例) `$_POST["name"]` や `$_POST["message"]`
 - まずは受け取った内容を表示するPHPを作成しよう
- 受け取ったデータをBBS_TABLEに挿入する際のSQLを作成する
 - `INSERT INTO ~ ~`



手順(4)

- データベースに接続する
`$con = mysql_connect("localhost", "", "");`
- データベース内の利用するテーブルを指定
(例) `mysql_select_db("user_db");`
- 作成したSQLを利用してデータベースへ挿入してみましよう
 - `mysql_query` を利用



手順(5)

- データベース内のメッセージを取得するプログラムを書いてみましょう
 - `mysql_fetch_query`
 - 表示順は新しい順にしましょう
- メッセージ一覧を, TABLEタグを利用して整形してみましょう
- メッセージを削除できるようにしてみましょう
 - `delete from bbs_table where id=$delete_id;`
- メッセージを検索できるようにしてみましょう
 - ... where message like "%中村%"



参考(削除:DELETE)

- メッセージの削除をどうやって実装するか？
 - 掲示板の場合, IDがテーブル内の書き込みを一意に識別するものであるため, そのIDを使って削除!
- WHEREでIDが一致した時に削除するようにプログラムを作成
- 削除するためには下記をPHPに伝える必要あり
 - 削除したい対象のID
 - 削除したいという行為

どうやって実装するか？



参考 (削除 : DELETE)

- GETで `bbs.php?act=delete&id=5` と送ると, 5番のIDを削除するとかどうか?
 - Aタグなどでリンクとして埋め込む!
 - `$_GET["act"]`がdeleteで, `$_GET["id"]`が5なら削除というプログラムを内部に書く
- `<input type="hidden" name="" value="">` を使うと, GET/POSTで情報送信することが可能
 - `<input type="hidden" name="act" value="delete">`
 - `<input type="hidden" name="id" value="5">`
 - などのように送って, プログラムで削除する!



参考 (LIKE: 検索)

- WHERE で文字列の部分マッチをしたい場合は LIKE を使う。また、前方、後方に他の文字を許容する場合には「%」をつける

```
SELECT name from nkmr-lab WHERE name LIKE "%哉"
```

– 最後に「哉」がついているnameを取って来い

- 和田拓哉, 菅澤卓哉, 三輪聡哉

```
SELECT name from nkmr-lab WHERE name LIKE "%川|%"
```

- 黒川恵輔, 川村涼太

```
SELECT name from nkmr-lab WHERE name LIKE "川|%"
```

- 川村涼太



宿題

- 先週作成した家計簿のテーブルを利用し, PHPでデータの登録, 一覧化(可能なら編集や削除など)が可能な家計簿のシステムを作しましょう
- 個人個人で興味のある何らかのデータを格納するテーブル群を作成し, さらにデータを挿入せよ. また, そこからデータを取り出す仕組みを作しましょう

関係ないけれど...

- プログラミング演習Iの実世界の模倣で作ったプログラムをWebにおいてみましょう！
- Processing.js を使うことで、ProcessingのプログラムをWebで簡単に再現できます！
 - JavaScriptでProcessingを再現する仕組み



← → ↻ ☆ ABP  ☰

 Processing.js
a port of the Processing Visualization Language

home download reference exhibition learning articles community blog [view source](#)

Processing.js 1.4.7 released!
Since our last release, 1.4.1, Processing.js has been given a bit of a source code overhaul, culminating in a new release: 1.4.7! This release adds in new

Downloads

- [Download the latest version of Processing.js!](#)

手順

- 作成したプログラム (*.pde) をWebサーバにアップロード (例: imitate.pde)
- processing.min.js を下記よりダウンロードして, Webサーバにアップロード
- 下記のコードをHTML内に書き込む!

<HEAD> と </HEAD> の間に下記を挿入

```
<script src="processing.min.js"></script>
```

BODY内の適当な場所に下記を挿入 (width, heightは適当に決めるように)

```
<canvas width="940px" height="78px" data-processing-sources="imitate.pde"></canvas>
```