

## 付録 A

# 名簿データベースのソースコード

### A.1 show\_all.php

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <meta charset="UTF-8" />
5:     <title>名簿データベース：全データ表示</title>
6:   </head>
7:   <body>
8:     <?php
9:       $dbname = "../..../unix10/meibo.db";
10:      $table = "meibo_table";
11:
12:      $dsn = "sqlite:" . $dbname;
13:
14:      try
15:      {
16:          $connect = new PDO($dsn, null, null);
17:      }
18:      catch (PDOException $error)
19:      {
20:          echo '接続に失敗しました: ' . $error->getMessag
21:          ge();
22:      }
23:      $sql = "select * from " . $table;
24:
25:      $query_ret = $connect->query($sql);
26:
27:      while($data = $query_ret->fetch(PDO::FETCH_ASSOC))
28:      {
29:          echo "id: " . $data['id'] . " name: " . $data['name'] .
30:          " name_yomi: " . $data['name_yomi'] . "<br>¥n";
31:      }
```

```
31:
32:     unset($connect);
33:
34: ?>
35:     <a href="index.html">戻る</a>
36: </body>
37: </html>
```

## A.2 insert.php

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <meta charset="UTF-8" />
5:     <title>名簿データベース：データ追加</title>
6:   </head>
7:   <body>
8:     <?php
9:       $dbname = "../..../unix10/meibo.db";
10:      $table = "meibo_table";
11:
12:      $dsn = "sqlite:" . $dbname;
13:
14:      try
15:      {
16:          $connect = new PDO($dsn, null, null);
17:      }
18:      catch (PDOException $error)
19:      {
20:          echo '接続に失敗しました: ' . $error->getMessa
21: ge();
22:          exit;
23:      }
24:
25:      $sql = "select max(id) from " . $table;
26:      $query_ret = $connect->query($sql);
27:
28:      $next_id = $query_ret->fetchColumn() + 1;
29:
30:      // if _GET['id'] != null
31:      if(isset($_GET['id']))
32:      {
33:          if(($_GET['id'] >= 1) && ($_GET['name'] != "") && ($_GE
34: T['name_yomi'] != ""))
35:          {
36:              $sql = "insert into " . $table . " values(" . $_GET[
37: 'id'] . ", '" . $_GET['name'] . "', '" . $_GET['name_yomi
38: ']' . "'))";
```

```
35:         $query_ret = $connect->query($sql);
36:         if($query_ret == FALSE)
37:         {
38:             echo "データ登録失敗!: " . $sql . "<br>¥n"
;
39:         }
40:         else
41:         {
42:             echo "データ追加成功!: " . $sql . "<br>¥n"
;
43:         }
44:     }
45: }
46:
47: unset($connect);
48:
49: ?>
50: <form>
51:     id : <input type="text" name="id" value="<?php echo $next_
id ?>"><br>
52:     Name: <input type="text" name="name" width="32"><br>
53:     Name_yomi: <input type="text" name="name_yomi" width="64">
<br>
54:     <input type="submit" value="追加"> <input type="reset" v
alue="消去">
55: </form>
56:
57:     <a href="index.html">戻る</a>
58: </body>
59: </html>
```

## A.3 delete\_update.php

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <meta charset="UTF-8" />
5:     <title>名簿データベース：データ削除・更新</title>
6:   </head>
7:   <body>
8:     <?php
9:     $dbname = "../..../unix10/meibo.db";
10:    $table = "meibo_table";
11:
12:    $dsn = "sqlite:" . $dbname;
13:
14:    try
15:    {
```

```
16:     $connect = new PDO($dsn, null, null);
17: }
18: catch (PDOException $error)
19: {
20:     echo '接続に失敗しました: ' . $error->getMessage();
21:     exit;
22: }
23:
24: $sql = "select * from " . $table;
25: $query_ret = $connect->query($sql);
26:
27: echo "<table>\n";
28: while($data = $query_ret->fetch(PDO::FETCH_ASSOC))
29: {
30:     echo "<tr>";
31:     echo "<td><form action=¥\"delete_exec.php¥\"><input type=
¥\"submit¥\" value=¥\"削除¥\"><input type=¥\"hidden¥\" name=¥
¥\"id¥\" value=¥\"\" . $data['id'] . \"¥\"></form></td> ";
32:     echo "<td><form action=¥\"update_exec.php¥\"><input type=
¥\"submit¥\" value=¥\"更新¥\"><input type=¥\"hidden¥\" name=¥
¥\"id¥\" value=¥\"\" . $data['id'] . \"¥\"><input type=¥\"hidden¥
¥\" name=¥\"name¥\" value=¥\"\" . $data['name'] . \"¥\"><input ty
¥\"pe=¥\"hidden¥\" name=¥\"name_yomi¥\" value=¥\"\" . $data['name_
¥\"yomi'] . \"¥\"></form></td>";
33:
34:     echo "<td>id: \" . $data['id'] . \" name: \" . $data['name
¥\"'] . \" name_yomi: \" . $data['name_yomi'] . \"</td>";
35:     echo "</tr>\n";
36: }
37: echo "</table>\n";
38:
39: unset($connect);
40: ?>
41: <a href="index.html">戻る</a>
42: </body>
43: </html>
```

## A.4 delete\_exec.php

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4:     <meta charset="UTF-8" />
5:     <title>名簿データベース：データ削除実行</title>
6: </head>
7: <body>
8: <?php
```

```
9:     $dbname = "../../unix10/meibo.db";
10:     $table = "meibo_table";
11:
12:     $dsn = "sqlite:" . $dbname;
13:
14:     try
15:     {
16:         $connect = new PDO($dsn, null, null);
17:     }
18:     catch (PDOException $error)
19:     {
20:         echo '接続に失敗しました: ' . $error->getMessag
    ge();
21:         exit;
22:     }
23:
24:     // if $_GET['id'] != null
25:     if($_GET['id'] >= 1)
26:     {
27:         $sql = "delete from " . $table . " where id = " . $_GET
    ['id'];
28:         $query_ret = $connect->query($sql);
29:         if($query_ret == FALSE)
30:         {
31:             echo "削除失敗!: " . $sql . "<br>¥n";
32:         }
33:         else
34:         {
35:             echo "削除成功!: " . $sql . "<br>¥n";
36:         }
37:     }
38:
39:     unset($connect);
40:
41: ?>
42:
43:     <a href="index.html">戻る</a>
44: </body>
45: </html>
```

## A.5 update\_exec.php

```
1:<!DOCTYPE html>
2:<html>
3: <head>
4:     <meta charset="UTF-8" />
5:     <title>update_exec.php</title>
6: </head>
```

```
7: <body>
8:   <h1>データ更新実行(update_exec.php)</h1>
9:   <?php
10:     $dbname = "../..../unix10/meibo.db";
11:     $table = "meibo_table";
12:     $dsn = "sqlite:" . $dbname;
13:     try {
14:       $connect = new PDO($dsn, null, null);
15:     }
16:     catch(PDOException $error)
17:     {
18:       echo "接続に失敗しました：" . $error->getMessage();
19:       exit;
20:     }
21:
22:     if($_GET['id'] >= 1)
23:     {
24:       if(isset($_GET['update_exec']))
25:       {
26:         if($_GET['update_exec'] == "on")
27:         {
28:           $sql = "UPDATE " . $table . " SET name = '" . $_GET['name'] . "',
name_yomi = '" . $_GET['name_yomi'] . "' WHERE id = " . $_GET['id'];
29:           $query_ret = $connect->query($sql);
30:           if($query_ret == FALSE)
31:           {
32:             echo "更新失敗!：" . $sql . "<br />";
33:           }
34:           else
35:           {
36:             echo "更新成功!：" . $sql . "<br />";
37:           }
38:
39:         }
40:       }
41:     }
42:
43:     echo "<form>";
44:     echo "<input type='hidden' name='update_exec' value='on'>";
45:     echo "<input type='text' name='id' value='" . $_GET['id'] . "'>";
46:     echo "<input type='text' name='name' value='" . $_GET['name'] . "'>";
47:     echo "<input type='text' name='name_yomi' value='" . $_GET['name_y
omi'] . "'>";
48:     echo "<input type='submit' value='更新'>";
49:     echo "</form>";
50:
51:     unset($connect);
52:   ?>
```

```
53: <p><a href="index.html">メニューに戻る</a></p>  
54: </body>  
55:</html>
```



## 付録 B

# Web サーバが保持しているアクセスデータ一覧

内容	key 値	例
Web サーバのアドレス (FQDN or IP アドレス)	HTTP_HOST	133.88.120.197
参照元 URL	HTTP_REFERER	http://133.88.120.197/ \~tkouya/webhpclinux/index.html
ブラウザ名称	HTTP_USER_AGENT	Mozilla/5.0(Windows; U;WindowsNT6.1;en-US) AppleWebKit/534.13(KHTML, likeGecko)Chrome/9.0.597. 94Safari/534.13
受信可能なファイルエンコード形式	HTTP_ACCEPT_ENCODING	gzip, deflate, sdch
受信可能な国別コード	HTTP_ACCEPT_LANGUAGE	ja, en-US;q=0.8, en;q=0.6
受信可能な文字コード形式	HTTP_ACCEPT_CHARSET	Shift_JIS, utf-8;q=0.7, *;q=0.3

内容	key 値	例
サーバ側のパス設定	PATH	/sbin:/usr/sbin:/bin:/usr/bin
サーバ署名	SERVER_SIGNATURE	<address>Apache/2.2.3(CentOS)Serverat133.88.120.197Port80</address>
サーバソフトウェア	SERVER_SOFTWARE	Apache/2.2.3(CentOS)
サーバ名	SERVER_NAME	133.88.120.197
サーバのアドレス	SERVER_ADDR	133.88.120.197
サーバの受信ポート番号 (通常 TCP80 番ポート)	SERVER_PORT	80
クライアントマシンアドレス	REMOTE_ADDR	133.88.121.79
サーバ側のトップ Web ページ格納用ディレクトリ	DOCUMENT_ROOT	/home/tkouya/unix
サーバ管理者 (メールアドレス)	SERVER_ADMIN	tkouya@cs.sist.ac.jp
実行スクリプト名	SCRIPT_FILENAME	/home/tkouya/public_html/webhpclinux/var_dump_server.php
クライアントのポート番号	REMOTE_PORT	46391
CGI 形式	GATEWAY_INTERFACE	CGI/1.1
プロトコル名	SERVER_PROTOCOL	HTTP/1.0
アクセス方法 (通常アクセスは GET もしくは POST)	REQUEST_METHOD	GET
クエリ文字列	QUERY_STRING	test=3&test2=4
アクセス URI(URL)	REQUEST_URI	/~tkouya/webhpclinux/var_dump_server.php
アクセス時刻	REQUEST_TIME	1297337285

## 付録 C

# アクセスリストの全ソーススクリプト

### C.1 access\_list.php

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <meta charset="UTF-8" />
5:     <title>Access List to this page</title>
6:   </head>
7:   <body>
8:     <h1>アクセスリスト</h1>
9:
10:    <h2>アクセス情報 (データベース格納分のみ)</h2>
11:    <?php
12:      // アクセス元の情報
13:      // $_SERVER['HTTP_REFERER'] ... リンク元
14:      // $_SERVER['HTTP_USER_AGENT'] ... ブラウザ名
15:      // $_SERVER['REMOTE_ADDR'] ... アクセス元(クライアント)アドレス
16:      // $_SERVER['REQUEST_TIME'] ... アクセス日時
17:
18:      // $_SERVER定義チェック(未定義の場合もあり)
19:      $referer = '';
20:      $user_agent = '';
21:      $remote_address = '';
22:      $request_time = time(); // アクセス時間
23:      if(isset($_SERVER['HTTP_REFERER']))
24:        $referer = $_SERVER['HTTP_REFERER'];
25:      if(isset($_SERVER['HTTP_USER_AGENT']))
26:        $user_agent = $_SERVER['HTTP_USER_AGENT'];
27:      if(isset($_SERVER['REMOTE_ADDRESS']))
28:        $remote_address = $_SERVER['REMOTE_ADDRESS'];
29:      if(isset($_SERVER['HTTP_USER_AGENT']))
```

```
30:     $request_time = $_SERVER['REQUEST_TIME'];
31:
32:     echo '$_SERVER[' . '¥' . 'HTTP_REFERER' . '¥' . ']' => ' . $referer .
    "<br />¥n";
33:     echo '$_SERVER[' . '¥' . 'HTTP_USER_AGENT' . '¥' . ']' => ' . $user_a
    gent . "<br />¥n";
34:     echo '$_SERVER[' . '¥' . 'REMOTE_ADDR' . '¥' . ']' => ' . $remote_add
    ress . "<br />¥n";
35:     echo '$_SERVER[' . '¥' . 'REQUEST_TIME' . '¥' . ']' => ' . $request_t
    ime . ' => ' . date('Y-m-d H:i:s', $request_time) . "<br />¥n";
36: ?>
37:
38: <h2>グラフ</h2>
39:
40: <div><h3>ブラウザ数</h3>
41: 
42: </div>
43: <div><h3>時間ごとのアクセス数</h3>
44: 
45: </div>
46:
47: <h2>アクセス一覧</h2>
48: <?php
49:     // データベースに接続
50:     $dbname = "../unix10/access_list.db";
51:     $table = "access_list_all";
52:     $dsn = "sqlite: " . $dbname;
53:     try
54:     {
55:         $connect = new PDO($dsn, null, null);
56:     }
57:     catch (PDOException $error)
58:     {
59:         echo '接続に失敗しました: ' . $error->getMessage();
60:         exit;
61:     }
62:
63:     // 最大id設定
64:     $sql = "SELECT max(id) FROM " . $table;
65:     $query_ret = $connect->query($sql);
66:     $data = $query_ret->fetch(); // $data[0], $data[1], ...
67:     $next_id = $data[0] + 1;
68:
69:     echo "アクセス総数: " . ($next_id - 1) . "<br />¥n";
70:
71:     // データベース書き込み (追記のみ)
72:     $sql = "INSERT INTO " . $table . " VALUES (" . $next_id . ", '" .
    $referer . "', '" . $user_agent . "', '" . $remote_address . "', " .
    $request_time . ")";
73:     $query_ret = $connect->query($sql);
```

```
74:
75: // 全データ表示:時刻の新しい順
76: // $sql = "SELECT * FROM " . $table . " ORDER by request_time DESC";
77: $sql = "SELECT * FROM " . $table . " ORDER by request_time DESC LI
MIT 3";
78: $query_ret = $connect->query($sql);
79:
80: while($data = $query_ret->fetch(PDO::FETCH_ASSOC))
81: {
82:     echo "<p><table style=¥\"border: solid¥\">¥n";
83:     echo "<tr><td style=¥\"border: solid¥\">ID</td><td style=¥\"border:
solid¥\">" . $data['id'] . "</td></tr>¥n";
84:     echo "<tr><td style=¥\"border: solid¥\">REFERFER</td><td style=¥\"b
order: solid¥\">" . $data['referer'] . "</td></tr>¥n";
85:     echo "<tr><td style=¥\"border: solid¥\">USER_AGENT</td><td style=¥
\"border: solid¥\">" . $data['user_agent'] . "</td></tr>¥n";
86:     echo "<tr><td style=¥\"border: solid¥\">REMOTE_ADDR</td><td style=
¥\"border: solid¥\">" . $data['remote_address'] . "</td></tr>¥n";
87:     echo "<tr><td style=¥\"border: solid¥\">REQUEST_TIME</td><td style=
¥\"border: solid¥\">" . date('Y-M-d H:i:s', $data['request_time']) . "</t
d></tr>¥n";
88:     echo "</table></p>¥n";
89: }
90:
91: // データベース切断
92: unset($connect);
93: ?>
94: <hr />
95: <a href="index.html">トップに戻る</a>
96: </html>
```

## C.2 access\_list\_graph\_24hours.php

```
1: <?php
2: // データベースに接続
3: $dbname = "../..../unix10/access_list.db";
4: $table = "access_list_all";
5: $dsn = "sqlite:" . $dbname;
6: try
7: {
8:     $connect = new PDO($dsn, null, null);
9: }
10: catch (PDOException $error)
11: {
12:     echo '接続に失敗しました: ' . $error->getMessage();
13:     exit;
14: }
15:
```

```
16: // アクセス時間のみ全取得
17: $sql = "SELECT request_time FROM " . $table;
18: $query_ret = $connect->query($sql);
19:
20: // $time_name : 0, 1, ..., 23
21: $hour_name = array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23);
22:
23: //初期化
24: $data = array(); // グラフデータ
25:
26: for($i = 0; $i < count($hour_name); $i++)
27:     $data[$i] = array($hour_name[$i], 0);
28:
29: while($request_time = $query_ret->fetch())
30: {
31:     // 時間のみ取り出し
32:     $hour = date('H', $request_time[0]);
33:
34:     // カウント
35:     for($i = 0; $i < count($hour_name); $i++)
36:     {
37:         if($hour == $hour_name[$i])
38:             $data[$i][1]++;
39:     }
40: }
41:
42: // データベース切断
43: unset($connect);
44:
45: //グラフ描画
46: require_once 'phpplot.php'; // PHPlotクラス
47:
48: // グラフ描画(棒グラフ)
49: // 320 x 240 pixels
50: $graph = new PHPlot(480, 240);
51: // グラフの枠線を直線で描く -> 'plain'
52: $graph->SetImageBorderType('plain');
53:
54: // 日本語フォント設定
55: $graph->SetTTFPath('./'); // PHPlotスクリプトと同じ位置
56: // IPAゴシック8~18ポイントに設定
57: $graph->SetFontTTF('title', 'ipaexg.ttf', 18); // タイトル
58: $graph->SetFontTTF('x_label', 'ipaexg.ttf', 8); // x軸ラベル
59: $graph->SetFontTTF('y_label', 'ipaexg.ttf', 8); // y軸ラベル
60: $graph->SetFontTTF('x_title', 'ipaexg.ttf', 10); // x軸タイトル
61: $graph->SetFontTTF('y_title', 'ipaexg.ttf', 10); // y軸タイトル
62:
63: // 棒グラフを描画 -> bars
64: $graph->SetPlotType('bars');
```

```
65: // データ形式 -> array("データラベル", 値)
66: $graph->SetDataType('text-data');
67: // 描画すべきデータ -> $data
68: $graph->SetDataValues($data);
69:
70: // グラフタイトル
71: $graph->SetTitle("時間ごとのアクセス数");
72:
73: // 影なし:
74: $graph->SetShading(0);
75:
76: // グラフ描画実行
77: $graph->DrawGraph();
78: ?>
```

### C.3 access\_list\_graph\_browser.php

```
1: <?php
2: // データベースに接続
3: $dbname = "../.././unix10/access_list.db";
4: $table = "access_list_all";
5: $dsn = "sqlite:" . $dbname;
6: try
7: {
8:     $connect = new PDO($dsn, null, null);
9: }
10: catch (PDOException $error)
11: {
12:     echo '接続に失敗しました: ' . $error->getMessage();
13:     exit;
14: }
15:
16: // ブラウザデータのみ全取得
17: $sql = "SELECT user_agent FROM " . $table;
18: $query_ret = $connect->query($sql);
19:
20: // $user_agent_name : ブラウザを特定するための文字列
21: $user_agent_name = array('MSIE', 'Firefox', 'WebKit');
22:
23: //初期化
24: $data = array(); // グラフデータ
25:
26: for($i = 0; $i < count($user_agent_name); $i++)
27:     $data[$i] = array($user_agent_name[$i], 0);
28:
29: while($str_user_agent = $query_ret->fetch())
30: {
31:     // ブラウザ数カウント
```

```
32:     for($i = 0; $i < count($user_agent_name); $i++)
33:     {
34:         if(strpos($str_user_agent[0], $user_agent_name[$i]) != false)
35:             $data[$i][1]++;
36:     }
37: }
38:
39: // データベース切断
40: unset($connect);
41:
42: //グラフ描画
43: require_once 'phplot.php'; // PHPlotクラス
44:
45: // グラフ描画(棒グラフ)
46: // 320 x 240 pixels
47: $graph = new PHPlot(320, 240);
48: // グラフの枠線を直線で描く -> 'plain'
49: $graph->SetImageBorderType('plain');
50:
51: // 日本語フォント設定
52: $graph->SetTTFPath('./'); // PHPlotスクリプトと同じ位置
53: // IPAゴシック8~18ポイントに設定
54: $graph->SetFontTTF('title', 'ipaexg.ttf', 18); // タイトル
55: $graph->SetFontTTF('x_label', 'ipaexg.ttf', 8); // x軸ラベル
56: $graph->SetFontTTF('y_label', 'ipaexg.ttf', 8); // y軸ラベル
57: $graph->SetFontTTF('x_title', 'ipaexg.ttf', 10); // x軸タイトル
58: $graph->SetFontTTF('y_title', 'ipaexg.ttf', 10); // y軸タイトル
59:
60: // 棒グラフを描画 -> bars
61: $graph->SetPlotType('bars');
62: // データ形式 -> array("データラベル", 値)
63: $graph->SetDataType('text-data');
64: // 描画すべきデータ -> $data
65: $graph->SetDataValues($data);
66:
67: // グラフタイトル
68: $graph->SetTitle("ブラウザ数");
69:
70: // 影なし:
71: $graph->SetShading(0);
72:
73: // グラフ描画実行
74: $graph->DrawGraph();
75: ?>
```