

DTTP Hypertext Preprocessor

Part-1



ステップアップ編

第①部

DTTP

4.2.8 検索結果の表示

最後に検索結果を出力する画面です。ある条件によりDB検索を行った結果は、一般にその件数を予測できません。クライアント側に対して大量のデータを一度に返してしまうのは、ネットワーク負荷や画面の応答速度の面からも避けるべきです。出力データ件数が多い可能性がある場合、ページ制御を行うことを考慮する必要があります。

zip_display()に渡す引数は、検索結果全体を一定の行数で区切り、その中の何ページ目を表示するのか指示する数値です。先頭のページなら0になります。

zip_display.inc

```
1  <?php
2  //
3  //  検索結果の表示
4  //  引数：ページ番号
5  //
6  function    zip_display($page)
7  {
8      if (DEBUG) {
9          print "zip_display($page)<br>\n";
10     }
11     $type    = $_POST['type'];
12     $zip7    = $_POST['zip7'];
13     $pref_j  = $_POST['pref_j'];
14     $city_j  = $_POST['city_j'];
15     $count   = $_POST['count'];
16     $db      = sqlite_open(DB) or die(DB . " がオープンできません。 \n");
17     $select_phrase = "SELECT zip7,pref_j,city_j,town_j FROM zip ";
18     if ($type == 1) {          //  郵便番号で検索
19         $where_clause = "WHERE zip7 LIKE '$zip7%'";
20     } else {                  //  住所で検索
21         $where_clause = "WHERE 0=0";
22         if (strlen($pref_j) > 0) {
23             $where_clause .= " AND pref_j = '$pref_j'";
24         }
25         if (strlen($city_j) > 0) {
26             $where_clause .= " AND city_j = '$city_j'";
27         }
28     }
29     //
30     //  新規条件指定時（セッションデータとして保存）
```

```

31 //
32 if ($page == 0) {
33     $sql = "SELECT COUNT(*) FROM zip " . $where_clause;
34     $ar = sqlite_array_query($db, $sql, SQLITE_NUM);
35     $count = $ar[0][0]; // 新しい条件に該当する件数
36     $_SESSION['type'] = $type;
37     $_SESSION['zip7'] = $zip7;
38     $_SESSION['pref_j'] = $pref_j;
39     $_SESSION['city_j'] = $city_j;
40     $_SESSION['count'] = $count;
41 if (DEBUG) {
42     print "\$_SESSION[] = "; print_r($_SESSION); print "<br>\n";
43 }
44 }
45 $sql = $select_phrase . $where_clause
46     . " LIMIT " . RECORD_PER_PAGE
47     . " OFFSET " . $page * RECORD_PER_PAGE;
48 //
49 // ページ制御部分の表示
50 //
51 $pages = $count / RECORD_PER_PAGE; // ページ数
52 if ($pages > 1) {
53     print "検索結果ページ：";
54     for ($i=0; $i<$pages; $i++) {
55         if ($i == $page) {
56             printf(" %d\n", $i+1);
57         } else {
58             printf(" <a href=\"{%$_SERVER['PHP_SELF']}?page=%d\">%d</a>\n",
59                 $i, $i+1);
60         }
61     }
62     if ($page+1 < $pages) {
63         printf(
64             " <a href=\"{%$_SERVER['PHP_SELF']}?page=%d\">次ページ</a><br>\n",
65             $page+1);
66     }
67 }
68
69 if (DEBUG) { print "$sql<br>\n"; }
70
71 $res = sqlite_query($db, $sql); // SQL の発行
72 $rows = sqlite_num_rows($res); // 結果の行数

```

```

73     if ($rows > 0) {
74         print "<table border>\n"
75         . "<tr><th> 郵便番号 <th> 都道府県 <th> 市区町村 <th> 町域 </tr>\n";
76         for ($i=0; $i<$rows; $i++) {
77             $row = sqlite_fetch_array($res, SQLITE_ASSOC);
78             printf("<tr><td>%s<td>%s<td>%s<td>%s</tr>\n",
79                 $row['zip7'], $row['pref_j'], $row['city_j'], $row['town_j']);
80         }
81         print("</table>\n");
82     } else {
83         print "該当するデータはありません。 \n";
84     }
85     sqlite_close($db);
86     print "</body></html>";
87 } // zip_display();
88 ?>

```

16行目でデータベースを開き、17行目ではレコードを抽出するためのSQL文の先頭部分を作って\$select_phraseにセットしておきます。18～28行目では、SELECTで指定するWHERE句だけを、別途\$where_clauseにセットします。

32～44行目では、\$where_clauseで抽出されるレコード全体の件数を求めて、現在の抽出条件とともにセッション変数に保存します。この処理は当該スクリプトが呼び出されるたびに行っても正しく動作しますが、速度的に不利なので、先頭ページ表示時のみで行われるように条件を限定しています。

45～47行目で実際に抽出をかけるためのSQL文を生成します。この関数の引数である\$pageが絞り込み条件として使われています。WHERE句による抽出されたレコードの集合(レコードセット)の中からOFFSETで指定したレコード番号(先頭は0)から数えて、LIMITで指定したレコード数分だけのレコードを取り出しています。

次に、51行目では検索結果が全部で何ページあるのかを算出しています。これが1ページに収まるのであれば問題ありませんが、2ページ以上にまたがる場合、図4-7のように、検索結果が全部で何ページあるのか、また現在何ページ目を表示中なのかという、ページ制御用のナビゲータを表示します(52～67行目)。

やっと準備完了です。71行目でSQLを発行し、次の行で取り出した行数を取得します。73～84行で、行数分だけループしながらHTMLの表を構成しています。