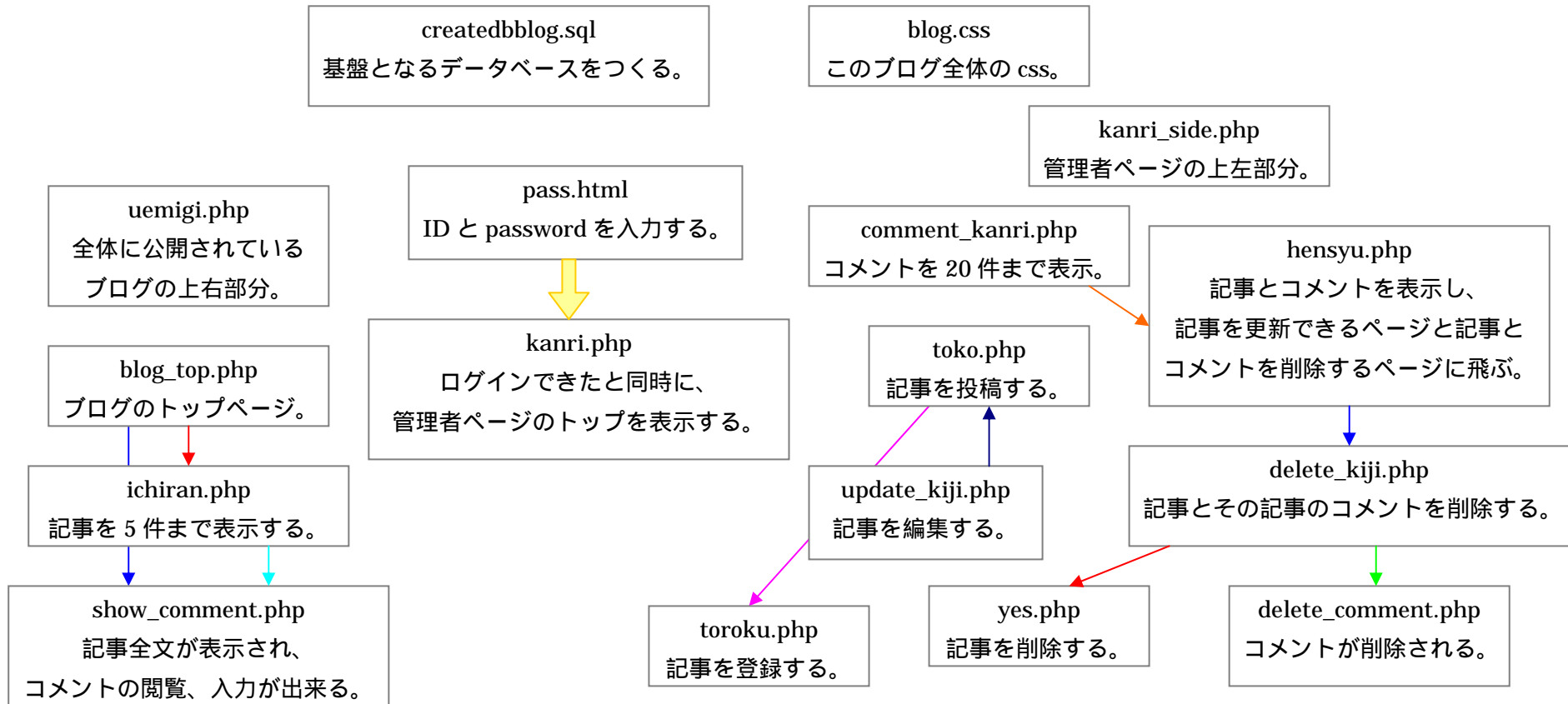


福田ゼミの3回生が後期にBLOGを制作する際に役に立つマニュアルを制作しました。	2
CREATEDBBLOG.SQL	3
BLOG.CSS	4
UEMIGI.PHP	6
KANRI_SIDE.PHP	8
BLOG_TOP.PHP	11
TOKO.PHP	13
TOROKU.PHP	15
ICHIRAN.PHP	17
SHOW_COMMENT.PHP	20
COMMENT_KANRI.PHP	26
HENSYU.PHP	29
DELETE_COMMENT.PHP	32
DELETE_KJI.PHP	33
YES.PHP	36
UPDATE_KJI.PHP	37
PASS.HTML	39
KANRI.PHP	40

福田ゼミの3回生が後期にblogを制作する際に役に立つマニュアルを制作しました。



createdblog.sql

基盤となるデータベースを作る。

`#drop database blog;` 下の内容を変更した場合に必要。変更したら # を除いて保存して、`mysql u root p < createdblog.sql` を二回実行する。

`create database blog;` 新しいデータベースを作る。

`use blog;` 使用するデータベースを選ぶ。

`CREATE TABLE kiji (` データベースに新しいテーブルを作る。

`id INTEGER UNIQUE PRIMARY KEY NOT NULL AUTO_INCREMENT,`

//INTEGER (整数) UNIQUE (重複なし) PRIMARY KEY (メインになる id) NOT NULL (空欄なし) AUTO_INCREMENT (自動的に連番を振る)

`title varchar(30),` varchar は最大の大きさとして 1 から 255 までの文字数を指定できる。

`kiji text,`

`hiduke date`

`);`

`CREATE TABLE comment (`

`id INTEGER UNIQUE PRIMARY KEY NOT NULL AUTO_INCREMENT,`

`kiji_id integer,` integer は整数。

`name varchar(255),`

`comment text,`

`hiduke date,`

`url varchar(255)`

`);`

blog.css

ブログの全体の css。

```
#title{ width: 700px;text-align: center;height:40px;background-color: #b0c4de }
#profile{ background-color: #ffccd0;font-size:12pt;width: 200px;float: right;height:100% }
#kanri{ background-color: #ffccd0;font-size:12pt;width: 200px;float: left;height:100% }
#menu{ width: 500px;float: left }
#footer{ clear: both;text-align: center;width: 700px}

body{ width:700px;margin-top:20px;margin-left:10%;margin-left:10%; background-color: #ffffe5 }

p.ichiran{ font-size:20pt;color:#0000ff;font-weight:bold }
p.sakujo{ font-size:17pt;color:#ff0000;font-weight:bold;text-align:center }
p.toko{ font-size:20pt;color:#ff0000;font-weight:bold }
p.make{ font-size: 20pt;font-weight: bold }
p.top{ text-align: center; }

span.srashu{ font-size:24pt;font-weight:bold }
a.sakujo{ text-decoration:none;font-size:25pt;font-weight:bold }
a.migi:link{ color:#696969;font-size:15pt;font-weight:bold }
a.migi:visited{ color:#696969;font-size:15pt;font-weight:bold }
a.migi:active{ color:#696969;font-size:15pt;font-weight:bold }
a.kids:link{ color:#ffff00;text-decoration:none;font-size:40pt;font-weight:bold }
a.kids:visited{ color:#ffff00;text-decoration:none;font-size:40pt;font-weight:bold }
a.kids:active{ color:#008000;text-decoration:none;font-size:45pt;font-weight:bold }
a.toko:link{ color:#ff0000;font-size:24pt;font-weight:bold;text-decoration:none }
a.toko:visited{ color:#ff0000;font-size:24pt;font-weight:bold;text-decoration:none }
```

```

a.toko:active{ color:#e0ffff;font-size:26pt;font-weight:bold;text-decoration:none }
a.ichiran:link{ color:#0000ff;font-size:24pt;font-weight:bold;text-decoration:none }
a.ichiran:visited{ color:#0000ff;font-size:24pt;font-weight:bold;text-decoration:none }
a.ichiran:active{ color:#e0ffff;font-size:26pt;font-weight:bold;text-decoration:none }

table.kiji{ border:3px;margin:2px;border-style:dotted;border-color:#32cd32;width:480px }
table.kiji{ border:3px;margin:2px 3px 5px;border-style:dotted;border-color:#9932cc;width:480px }
table.kijikanri{ border:1px;border-color:#000000;border-style:solid;margin:2px 2%;width:480px }
table.com{ border:3px;margin:2px;border-style:dotted;border-color:#ffa500;width:480px }
table.kkiji{ border:3px;margin:2px;border-style:dotted;border-color:#32cd32;width:480px }
table.pass{ border:3px;margin:2px;border-style:dotted;border-color:#000000;width:400px;height:120px }
table.login{  }
table.toko{ border:none;width:480px }
table.sakujo{ border:none;width:450px;margin-left:5%;margin-right:5% }
table.nextback{ border:3px;margin:2px;border-style:none;width:480px }

td.sakujoi{ color:#696969;text-align:center;font-size:15pt }
td.sakujoy{ color:#ff0000;text-align:center;text-decoration:none;font-size:20pt;font-weight:bold }
td.sakujon{ color:#0000ff;text-decoration:none;font-size:20pt;font-weight:bold }
td.toroku{ width:50px;text-align:center }
tr{ height:1.7em }

li.sankaku{ list-style-image:url(attention.gif) }
li.armadillo{ list-style-image:url(armadillo.gif) }

h3{ color:darkblue }

```

uemigi.php

全体に公開されているブログの上部と右側。

```
<div id="title">
<h1><a class="kids" href="blog_top.php">Yumi Kids</a></h1>
</div>
<br>
<div id="profile">
<p></p>
<p class="make">創造者:yumi</p>
<p>大阪府出身 / 21 歳 A 型</p>

<p><a class="migi" href="ichiran.php">記事一覧</a></p>

<h3>最新の記事(5 件)</h3>
<ul>
<?php
$sql = "select * from kiji order by id DESC limit 5";
$data = mysql_query($sql, $connect);
while ($record = mysql_fetch_array($data)){
    $id = $record['id'];
    $title = $record['title'];
    $date = $record['hiduke'];
    printf("<li class='sankaku'"><a href='show_comment.php?id=%d'">%s(%s)" , $id , $title , $date);
}
?>
</ul>
```

投稿された最新の 5 件の記事を
記事のタイトル (投稿した日付)
この形式で新しいものから表示する。

```
<h3>最近のコメント(7件)</h3>
<ul>
<?php
$ssql_comment = "select * from comment order by id DESC limit 7";
$data_comment = mysql_query($ssql_comment, $connect);
while ($comment_record = mysql_fetch_array($data_comment)){
    $comment_id = $comment_record['id'];
    $comment_name = $comment_record['name'];
    $comment_date = $comment_record['hiduke'];
    $ssql = "select * from kiji where id='$skiji_id'";
    $data = mysql_query($ssql, $connect);
    if ($skiji_record = mysql_fetch_array($data))
        $skiji_title = $skiji_record['title'];
    printf("<li class=%¥"sankaku¥" ><a href=%¥"show_comment.php?id=%d#com¥" >%s:%s(%s)</a></li>", $id, $comment_name, $skiji_title,
    $comment_date);
}
?>
</ul>

<p><a class="migi" href="pass.html">管理者ページ</a></p>
</div>
```

投稿された最新の7件のコメントを
投稿者：記事のタイトル(コメントした日付)
この形式で新しいものから表示する。

kanri_side.php

管理者ページの上部和左側。

```
<div id="title">
<h1><a class="kids" href="kanri.php">Yumi Kids</a></h1>
</div>
<br>
<div id="kanri">
<p></p>
<p class="make">創造者:yumi</p>
<p>大阪府出身 / 21 歳 A 型</p>

<p><a class="migi" href="toko.php">新規登録画面</a></p>
<p><a class="migi" href="kiji_kanri.php">記事の管理</a></p>
<p><a class="migi" href="comment_kanri.php">コメントの管理</a></p>
<p><a class="migi" href="kanri.php">管理者ページ TOP</a></p>
<p><a class="migi" href="blog_top.php">BLOG TOP</a></p>

</div>
```


ここでは繰り返し使われている HTML 文書や PHP 文書を、 ~ にまとめました。
番号の下に記述してある文書を、番号が記載されているところに挿入する。

//今の HTML の最新バージョンは 4.01 です。最初に DOCTYPE をこのように記述します。

```
<!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional/EN"  
"http://www.w3.org/TR/html14/loose.dtd">
```

```
<?php
```

```
//データベースに接続する。MySQL に接続する。
```

```
$connect = mysql_connect("localhost", "root", "semi");
```

```
//もし接続できなかったら、エラーメッセージを出して終了する。
```

```
If (!connect) {
```

```
    Die("database connection fail");
```

```
}
```

```
//データベースを選ぶ。
```

```
Mysql_select_db("blog", $connect);
```

```
?>
```

```
<html>
```

```
<head>
```

```
//ページに使われているドキュメントのタイプ、文字コードを設定する。
```

```
//HTML 文書は " text/html "。さらに " ; " で区切って " charset= " で使われている文字コードを記入する。UNIX での標準コードは EUC-JP。
```

```
<meta http-equiv="Content-Type" content="text/html; charset=EUC-JP" >
```

```
<link rel="stylesheet" href="blog.css" type="text/css" title="基本 CSS" media="screen, print" >
<title></title>
</head>
<body>

</body>
</html>
```

blog_top.php

ブログのトップページ。最新記事 3 件まで表示される。

```
<?php
```

```
//組み込み関数。指定されたファイルのデータを挿入する。
```

```
require("uemigi.php");
```

```
//Mysql サーバーへ送信する query を作成。
```

```
$sql = "select * from kiji order by id DESC limit 3";
```

```
//作成した mysql query を実際に送信する。
```

```
$data = mysql_query($sql, $connect);
```

```
//結果の行を取得する。
```

```
while ($record = mysql_fetch_array($data)){
```

```
    $id = $record['id'];
```

```
    $title = $record['title'];
```

```
    $kiji = $record['kiji'];
```

```
    $date = $record['hiduke'];
```

```
    $sql_comment = "select * from comment where kiji_id = '$id'";
```

```
    $data_comment = mysql_query($sql_comment, $connect);
```

```
    if ($data_comment) {
```

```
        $num_comment = mysql_num_rows($data_comment);
```

```
    } else {
```

```
        $num_comment = 0;
```

```
    }
```

```
    print("<table class=¥"kiji¥">");
```

```
    printf("<tr><td><img src=¥"attentions.gif¥"></td><td colspan=¥"2¥">タイトル : %s</td></tr>", $title);
```

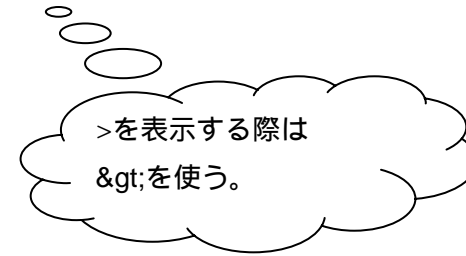
全てのフィールドの kiji テーブルから id を取り出して、降順に並べる。
ASC は結果が昇順、DESC は降順に並ぶ。
DESC limit 3 で、limit で取り出される記事が 3 つまでに制限される。

¥ = バックスラッシュ

```

printf("<tr><td></td><td colspan=%¥"2¥">%s...</td></tr>", nl2br(get3lines($kiji));
printf("<tr><td></td><td>【%s】", $date);
printf("<a href=%¥"show_comment.php?id=%d#com¥">コメント(%d)</a></td>", $id, $num_comment);
printf("<td align=%¥"right¥"><a href=%¥"show_comment.php?id=%d¥">read more&gt;&gt;</a></td></tr>", $id);
print("</table><br><br>¥n");
}
mysql_close($connect);
?>

```



上記のように入力すると、
記事は最新の三件のみ、内容は上から三行まで表示される。



toko.php

記事を投稿するページ。

```
<?php
require("kanri_side.php");

$today = getdate();
$date = sprintf("%d-%d-%d", $today["year"], $today["mon"], $today["mday"]);
?>
```

//フォームデータの送信先になる URL と、送信方法を指定する。

```
<form action="toroku.php" method="POST">
```

```
<table class="kiji">
```

```
<tr><td class="toroku">日付 : </td><td><?php echo $date;?></td></tr>
```

//<input>タグの type 属性を "text" に設定し、フォーム内に一行のテキスト入力フィールドを作成。

//name 属性は必須。テキスト入力フィールドの表示幅は一度に表示される文字数により size 属性で指定する。

```
<tr><td class="toroku">題名 : </td><td><input type="text" name="title" size="40"></td></tr>
```

//textarea はフォーム内に複数行のテキスト入力フィールドを作成するためのタグ。

//cols は、複数行のテキスト入力フィールドの幅を定義するもの。rows は、複数行のテキスト入力フィールドの行数を定義するもの。

```
<tr><td valign="top">本文:</td><td><textarea name="honbun" cols="40" rows="7 "></textarea></td></tr>
```

//<input>タグの type 属性を "submit" に設定すると、フォーム内にフォームデータを送信するための送信ボタンを作成できる。

//<input>タグの type 属性を "reset" に設定すると、フォーム内にリセットボタンを作成できる。

The screenshot shows a web page titled "Yumi Kids". On the left is a vertical navigation menu with links: "umi", "歳A型", "録画面", "管理", "の管理", and "ページ". The main content area is a form for posting an article. It has a date field showing "日付: 2007-10-22". Below that is a "題名:" (Title) label followed by a text input field. Underneath is a "本文:" (Body) label followed by a multi-line text area. At the bottom of the form are two buttons: "投稿" (Submit) and "クリア" (Clear). A red arrow points from the PHP code's title input field to the "題名:" field in the form. A cyan arrow points from the PHP code's submit and clear buttons to the "投稿" and "クリア" buttons in the form. A cyan text box with the text "送信ボタンとリセットボタン" (Submit and Reset buttons) is located near the buttons, with a cyan arrow pointing to the "クリア" button.

//ボタンのラベルを任意のものに変更したい場合には value 属性を指定。

```
<tr><td></td><td align="center"><input type="submit" value="投稿"> <input type="reset" value="クリア"></td></tr>
</table>
```

//入力されていないけど、必要なデータは hidden で送信する。

```
<input type="hidden" name="date" value="<?php echo $date;?>">
<input type="hidden" name="update" value="shinki">
```

```
</form>
```

toroku.php

記事を登録するページ。

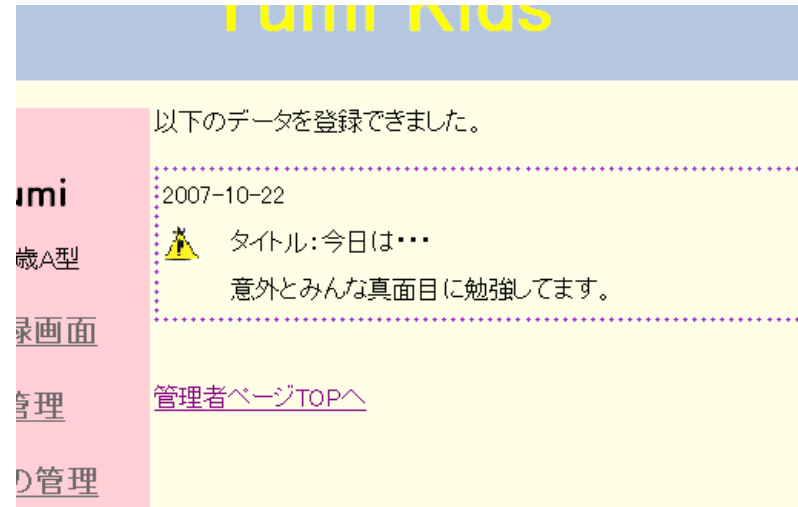
```
<?php
require("kanri_side.php");

//フォームから送信されたデータを受信する。
$update = $_POST['update'];
$date = $_POST['date'];
$title = $_POST['title'];
$kiji = $_POST['honbun'];
if ($update == "update") {
    $id = $_POST['id'];
```

//登録データの変更を複数の条件を与えて一括で変更することも、where 句を使って特定の行や列を指定して、その部分のみを変更することも可能です。

```
    $sql = "update kiji set title='$title', kiji='$kiji', hiduke='$date' where id=$id";
} else {
    $sql = "insert into kiji (title,hiduke,kiji) values('$title','$date','$kiji)";
}
```

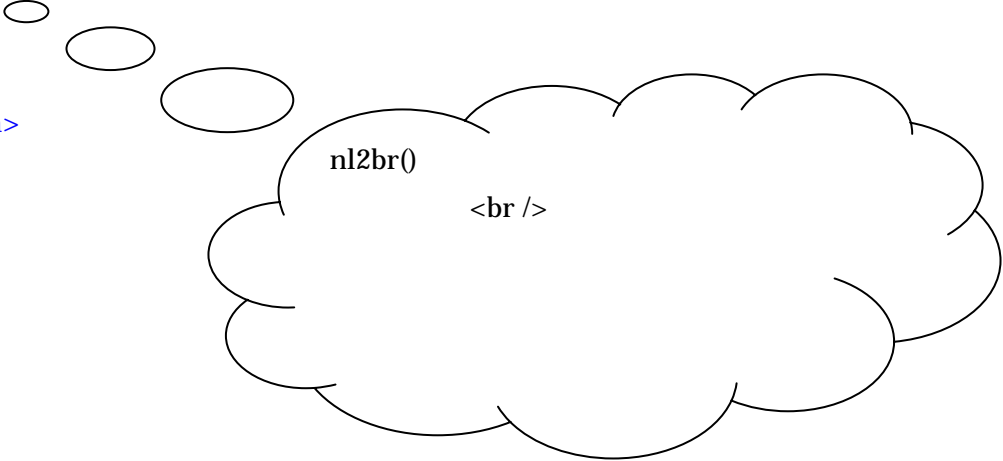
```
$data = mysql_query($sql, $connect);
if ($data == False){
    if ($update == "shinki") {
        die("登録できませんでした。");
    } else {
        die("更新できませんでした。");
    }
}
```



```
}  
} else{  
    if ($update == "shinki") {  
        print("<p>以下のデータを登録できました。 </p>");  
    } else {  
        print("<p>以下のデータを更新しました。 </p>");  
    }  
}  
mysql_close($connect);
```

//ここで登録できたデータを表示する。

```
print("<table class=¥"kiji¥">");  
printf("<tr><td colspan=¥"2¥">%s</td>", $date);  
printf("<tr><td><img src=¥"attentions.gif¥"></td><td>タイトル : %s</td></tr>", $title);  
printf("<tr><td></td><td>%s</td></tr>", nl2br($kiji));  
print("</table><br><br>¥n");  
?>  
<a href="kanri.php">管理者ページ TOP へ</a>
```



nl2br()関数は、
改行文字を
タグに置き換える。
つまりこれを入れないと入力した通りに
改行されない。

ichiran.php

記事を新しいものから順番に表示する。1 ページに 5 件まで表示し、5 件以上記事があれば、次のページや前のページへのリンクが表示される。

```
<?php
```

```
function get3lines($s) {  
    $lines = explode("¥n", $s, 4);  
    $num_lines = min(3, count($lines));  
    $threelines = array_splice($lines, 0, $num_lines);  
    return implode("¥n", $threelines);  
}
```

記事を三行まで取り出す関数。
中身は難しいのでそのまま使いましょう。

```
<?php
```

```
require("uemigi.php");
```

```
?>
```

```
<p class="ichiran">記事一覧</p>
```

```
<?php
```

```
$page = (int)$GET['page'];
```

```
    if ($page == False) {
```

```
        $page = 0;
```

```
    }
```

```
$start = $page * 5;
```

```
$sql = "select * from kiji order by id DESC limit $start, 5";
```

```
$data = mysql_query($sql, $connect);
```

```
$data_num = mysql_num_rows($data);
```

記事を 5 件まで取り出す。

The screenshot shows a web page with a yellow background and a blue header. The main content area is titled "記事一覧" (Article List) and contains three article entries. Each entry has a title, a snippet of text, and a "read more" link. The sidebar on the right contains navigation links and a list of recent articles.

//表示しているページの1番小さいidと表示されているデータの件数を足して、そのページの1番大きいidを\$lastに代入する。

```
$last = $start + (int)$data_num;
```

```
while ($record = mysql_fetch_array($data)){
```

```
    $id = $record['id'];
```

```
    $title = $record['title'];
```

```
    $kiji = $record['kiji'];
```

```
    $date = $record['hiduke'];
```

```
    $sql_comment = "select * from comment where kiji_id = '$id'";
```

```
    $data_comment = mysql_query($sql_comment, $connect);
```

```
    if ($data_comment) {
```

```
        $num_comment = mysql_num_rows($data_comment);
```

```
    } else {
```

```
        $num_comment = 0;
```

```
    }
```

```
    print("<table class='kiji'>");
```

```
    printf("<tr><td><img src='attentions.gif'></td><td colspan='2'>タイトル : %s</td></tr>", $title);
```

```
    printf("<tr><td></td><td colspan='2'>%s...</td></tr>", nl2br(get3lines($kiji));
```

```
    printf("<tr><td></td><td>【%s】", $date);
```

```
    printf("<a href='show_comment.php?id=%d#com'>コメント(%d)</a></td>", $id, $num_comment);
```

```
    printf("<td align='right'><a href='show_comment.php?id=%d'>read more>></a></td></tr>", $id);
```

```
print("</table><br><br>");
```

```
}
```

```
$sql_all = "select * from kiji order by id DESC";
```

```
$data_all = mysql_query($sql_all, $connect);
```

```

//全部で何件データがあったか数える。
$data_all_num = mysql_num_rows($data_all);

$next_page = $page + 1;
$prev_page = $page - 1;

print("<table class=¥\"nextback¥\">");
//$start が 0 じゃなかったら前の 5 件のリンクを表示。
if ($start != 0) {
    printf("<tr><td align=¥\"left¥\"><a href=¥\"ichiran.php?page=%d¥\"> 前の 5 件</a></td>", $prev_page);
} else {
    print("<tr><td align=¥\"left¥\"> 前の 5 件</td>");
}

//全てのデータが$last より大きかったら次の 5 件のリンクを表示。
if ($data_all_num > $last) {
    printf(" <td align=¥\"right¥\"><a href=¥\"ichiran.php?page=%d¥\">次の 5 件</a></td></tr>", $next_page);
} else {
    print("<td align=¥\"right¥\">次の 5 件</td></tr>");
}
mysql_close($connect);
?>
</table>

```

show_comment.php

記事全文が表示され、その記事に対するコメントの閲覧、投稿ができる画面。

```
<a name="kiji"> </a>
<?php
require("uemigi.php");
?>
<table class="kiji">
<?php
if ($_GET['id']) {
    $id = $_GET['id'];
} else {
    $id = $_POST['id'];
    $comment_name = $_POST['name'];
    $comment_honbun = $_POST['honbun'];
    $comment_date = $_POST['date'];
    $comment_url = $_POST['url'];
```

送信先のフォームデータを送るメソッド(方法)が POST なので、\$_POST でデータを受け取る。
この POST は下記の form から送信されたデータを受け取っている。

```
// insert into のすぐ後ろのカッコの中には、データを入力する列名を入力する順番に記述する。
```

```
// values の後ろのカッコには、実際に入力するデータを記述します。insert into の後ろのカッコ内で指定した列の順番に対応するようにデータは記述されなければなりません。また、それぞれの列のデータ型でなければならない。
```

```
$sql = "insert into comment (kiji_id, name, comment, hiduke, url) values
('{$id}','{$comment_name}','{$comment_honbun}','{$comment_date}','{$comment_url}');
```

```
$data = mysql_query($sql, $connect);
if ($data == FALSE) {
```

The screenshot shows a web browser window with the URL `http://192.168.1.101/show_comment.php?id=10000`. The page has a yellow background and a blue header with the text "Yumi Kids". The main content area is divided into two columns. The left column contains a post with a title "ほうほう" and a date "2007-10-22". The text of the post is: "しゃぼん玉の会に顔をだしてきた。はたから見るとたそがれてる人みたいだった。でかいしゃぼん玉ができたらしいけど、私の頭に当たって消えたって怒られた。そんなの知らない。しゃぼん玉マシンがうまく扱えなくて切なかった。". The right column contains a sidebar with a "創造者:" section listing "大阪府出身/" and a "記事:" section with a link to "ほうほう (2007-10-22)". Below the sidebar is a section for "この記事に対するコメント" (Comments on this article) with two comments: "楽しそう♪" by "ゆみ" and "次は参加希望!!!!!!" by "だんで".

```

        echo "コメント・エラー";
    }
}
$page = (int)$_GET['id'];
if ($page == False) {
    $page = 0;
}

$sql = "select * from kiji where id='$id'";
$data = mysql_query($sql, $connect);
// fetch_array で 1 件のレコードを取り出す。
if ($record = mysql_fetch_array($data)){
    $id = $record['id'];
    $title = $record['title'];
    $kiji = $record['kiji'];
    $date = $record['hiduke'];
    printf("<tr><td>タイトル : %s</td></tr>", $title);
    printf("<tr><td>%s</td></tr>", nl2br($kiji));
    printf("<tr><td align='right'>%s</td></tr>", $date);
}
print("</table><br><hr>");
?>

```

ここに"com"という名前を付けておくことで、直接この位置にジャンプすることが出来る。
 例えばコメント(%d) この様に入力する。

```

<h3><a name="com">この記事に対するコメント</a></h3>
<?php
$si = 0;

```

```

$Sql = "select * from kiji where id='$id'";
$data = mysql_query($Sql, $connect);
if ($record = mysql_fetch_array($data)){
    $title = $record['title'];
    $kiji = $record['kiji'];
    $date = $record['hiduke'];
    $Sql_comment = "select * from comment where kiji_id='$id'";
    $data_comment = mysql_query($Sql_comment, $connect);
    while ($comment_record = mysql_fetch_array($data_comment)) {
        $comment_id = $comment_record['id'];
        $comment_name = $comment_record['name'];
        $comment_kiji = $comment_record['comment'];
        $comment_date = $comment_record['hiduke'];
        $comment_url = $comment_record['url'];
        $i += 1;
        print("<table class='com'>");
        printf("<tr><td>%s</td></tr>", nl2br($comment_kiji));
        printf("<tr><td align='right'>%s ", $comment_name);
        printf(" | %s | ", $comment_date);
        if ($comment_url != "") {
            printf("<a href='%s'> URL</a></td></tr>", $comment_url);
        } else {
            printf(" URL</td></tr>");
        }
        print("</table><br><br>");
    }
}

```

```
}  
?>  
<hr>
```

コメント入力画面と閲覧画面を同じページに表示したいので、送信先と受信先を同じファイルにする。
ここで入力された内容を受け取るタグは同じファイルの上部に記載している。

```
<h3>コメントを書く</h3>  
<form action="show_comment.php" method="POST">  
<table class="toko">  
  <tr><td align="right">名前:</td><td><input type="text" name="name" size="20"></td></tr>  
  <tr><td align="right">URL:</td><td><input type="text" name="url" size="35"></td></tr>  
<tr><td valign="top">コメント:</td><td><textarea name="honbun" cols="40" rows="7 "></textarea></td></tr>
```

//リセットボタンが押された場合にはフォーム内のすべての要素がリセットされ初期値に戻ります。

```
<tr><td></td><td align="center"><input type="submit" value="投稿"> <input type="reset" value="クリア"></td></tr>  
</table>
```

```
<?php  
$today = getdate();  
$date = sprintf("%d-%d-%d", $today["year"], $today["mon"], $today["mday"]);  
?>
```

```
<input type="hidden" name="date" value="<?php echo $date;?>">  
<input type="hidden" name="id" value="<?php echo $id;?>">  
</form>
```

```
<table class="nextback">  
<?php
```

```

// kiji テーブルの id の最大値を求める。
$sql = "select max(id) from kiji";
$data = mysql_query($sql, $connect);
$record = mysql_fetch_array($data, $connect);
$id_last = $record[0];

// kiji テーブルの id の最小値を求める。
$sql = "select min(id) from kiji";
$data = mysql_query($sql, $connect);
$record = mysql_fetch_array($data, $connect);
$id_first = $record[0];

$n_page = $page + 1;
$b_page = $page - 1;

//現在の id が 1 番小さい id より大きい場合、一つ前の id へのリンクを表示。
if ($id_first < $id){
    printf("<tr><td align=%¥"left¥"><a href=%¥"show_comment.php?id=%d¥">前の記事</a></td>", $b_page);
//1 番小さい id だった場合は、リンクを表示しない。
} else {
    print("<tr><td align=%¥"left¥">前の記事</td>");
}
printf("<td align=%¥"center¥"><a href=%¥"show_comment.php?id=%d#kiji¥">このページのトップ</a></td>", $id);

//現在の id が 1 番大きい id より小さい場合、一つ後の id へのリンクを表示。
if ($id_last > $id){

```

ここをクリックすると”kiji”という名前が付いている位置に画面が移動する。
この場合はページの上にある。


```
    printf("<td align=%¥"right¥"><a href = ¥"show_comment.php?id=%d¥">次の記事</a></td></tr>", $n_page);  
//1 番大きい id だった場合は、リンクを表示しない。  
} else {  
    print("<td align=%¥"right¥">次の記事</td></tr>");  
}  
mysql_close($connect);  
?>  
</table>
```

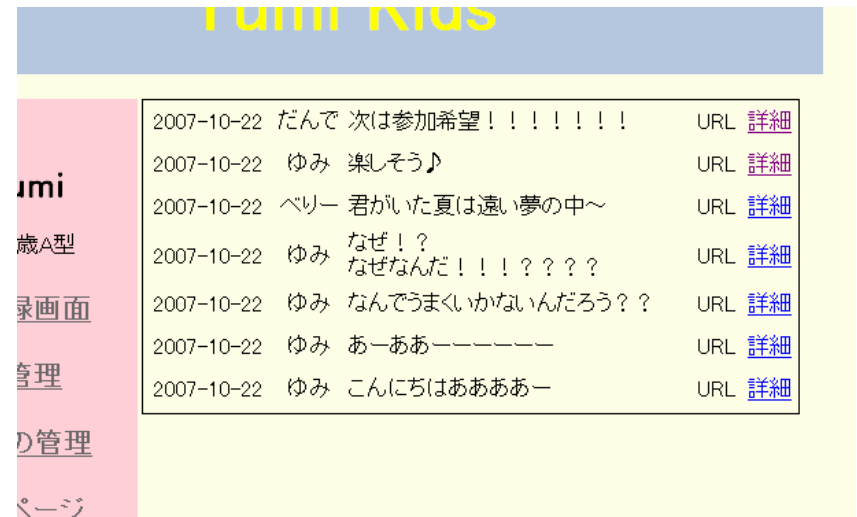
comment_kanri.php

全てのコメントが新しいものから 20 件まで表示される。20 件ごとにページができる。
詳細をクリックすると記事とその記事に対するコメント (hensyu.php) が表示される。

```
<?php
require("kanri_side.php");
?>
<table class="kijikanri">
<tr><th width="100px">日付</th><th width="120px">名前</th><th width="160px">本文</th><th width="50px">URL</th><th width="50px">編集
</th></tr>
```

```
<?php
//ページ数を確認 (カウント)
$page = (int)$_GET['page'];
//失敗したら
if ($page == False) {
//ページ数に 0 を代入
    $page = 0;
}
//1 ページのコメント数を 20 件までに制限する。
$limit = $page * 20;

$i = 0;
$sql_comment = "select * from comment order by id DESC limit $limit, 20";
$data_comment = mysql_query($sql_comment, $connect);
$data_num = mysql_num_rows($data_comment);
```



日付	名前	本文	URL	編集
2007-10-22	だん	次は参加希望！！！！！！	URL	詳細
2007-10-22	ゆみ	楽しそう♪	URL	詳細
2007-10-22	ベリー	君がいた夏は遠い夢の中～	URL	詳細
2007-10-22	ゆみ	なぜ！？ なぜなんだ！！！！？？？	URL	詳細
2007-10-22	ゆみ	なんでうまくいかないんだろう？？	URL	詳細
2007-10-22	ゆみ	あーあー————	URL	詳細
2007-10-22	ゆみ	こんにちはあああー	URL	詳細

```

$last = $limit + (int)$data_num;

while ($comment_record = mysql_fetch_array($data_comment)){
    $comment_id = $comment_record['id'];
    $kiji_id = $comment_record['kiji_id'];
    $comment_name = $comment_record['name'];
    $comment_kiji = $comment_record['comment'];
    $comment_date = $comment_record['hiduke'];
    $comment_url = $comment_record['url'];
    $i += 1;
    $sql = "select * from kiji where id='$kiji_id'";
    $data = mysql_query($sql, $connect);
    if ($record = mysql_fetch_array($data)) {
        $id = $record['id'];
    }
    printf("<tr><td align=¥"center¥">%s</td>", $comment_date);
    printf("<td align=¥"center¥">%s</td>", $comment_name);
    printf("<td>%s</td>", nl2br($comment_kiji));
    if ($comment_url != ""){
        printf("<td align=¥"center¥"><a href=¥"%s¥"URL</a></td>", $comment_url);
    } else {
        print("<td align=¥"center¥">URL</td>");
    }
    printf("<td align=¥"center¥"><a href=¥"hensyu.php?id=%d¥">詳細</a></td></tr>", $id);
}
print("</table>");

```

```
$sql_all = "select * from comment order by id DESC";
$data_all = mysql_query($sql_all, $connect);
$data_all_num = mysql_num_rows($data_all);

$next_page = $page + 1;
$prev_page = $page - 1;
print("<table class='nextback' align='right'>");

if ($limit != 0) {
    printf("<tr><td align='left'><a href='comment_kanri.php?page=%d'>前の 20 件</a></td>", $prev_page);
}
if ($data_all_num > $limit) {
    printf("<td align='right'><a href='comment_kanri.php?page=%d'>次の 20 件</a></td></tr>", $next_page);
}
mysql_close($connect);
?>
```

hensyu.php

記事とその記事に対するコメントが表示され、記事を更新できるページ (update_kiji.php) と記事やコメントを削除するページ (delete_kiji.php や delete_comment.php) に飛ぶ。

```
<?php
require("kanri_side.php");
?>
<table align="right" class="kiji">
<?php
if ($_GET['id']) {
    $id = $_GET['id'];
} else {
    $id = $_POST['id'];
    $comment_name = $_POST['name'];
    $comment_honbun = $_POST['honbun'];
    $comment_date = $_POST['date'];
    $comment_url = $_POST['url'];
    $sql = "insert into comment (kiji_id, name, comment, hiduke, url) values
('{$id}', '{$comment_name}', '{$comment_honbun}', '{$comment_date}', '{$comment_url}')";
    $data = mysql_query($sql, $connect);
    if ($data == FALSE) {
        echo "コメント・エラー";
    }
}

$sql = "select * from kiji where id='{$id}'";
```

```

$data = mysql_query($sql, $connect);
while ($record = mysql_fetch_array($data)){
    $id = $record['id'];
    $title = $record['title'];
    $kiji = $record['kiji'];
    $date = $record['hiduke'];
    printf("<tr><td>タイトル : %s</td><td></td></tr>", $title);
    printf("<tr><td colspan=2>%s</td></tr>", nl2br($kiji));
    printf("<tr><td>【%s】 </td>", $date);
    printf("<td align=right>[<a href=update_kiji.php?id=%d>更新</a>]", $id);
    printf("<a href=delete_kiji.php?id=%d>削除</a></td></tr>", $id);
}
print("</table><br>");
?>
<br><br>

```

<h3>この記事に対するコメント</h3>

<?php

\$i = 0;

\$sql = "select * from kiji where id='\$id'";

\$data = mysql_query(\$sql, \$connect);

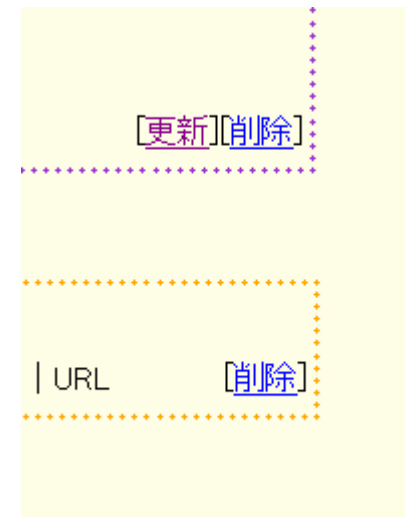
if (\$record = mysql_fetch_array(\$data)){

 \$title = \$record['title'];

 \$kiji = \$record['kiji'];

 \$date = \$record['hiduke'];

 \$sql_comment = "select * from comment where kiji_id='\$id'";



```

Sdata_comment = mysql_query($sql_comment, $connect);
while ($comment_record = mysql_fetch_array($data_comment)) {
    $comment_id = $comment_record['id'];
    $comment_name = $comment_record['name'];
    $comment_kiji = $comment_record['comment'];
    $comment_date = $comment_record['hiduke'];
    $comment_url = $comment_record['url'];
    $i += 1;
    print("<table align=¥"right¥" class=¥"com¥">");
    printf("<tr><td colspan=¥"2¥">%s</td></tr>", nl2br($comment_kiji));
    printf("<tr><td align=¥"right¥">%s ", $comment_name);
    printf(" | %s | ", $comment_date);
    if ($comment_url != "") {
        printf("<a href=¥"%s¥"> URL</a></td>", $comment_url);
    } else {
        printf(" URL</td>");
    }
    printf("<td align=¥"right¥">[<a href=¥"delete_comment.php?id=%d&com_id=%d¥">削除</a>]</td></tr>", $id, $comment_id);
print("</table><br><br>¥n");
}
}
mysql_close($connect);
?>

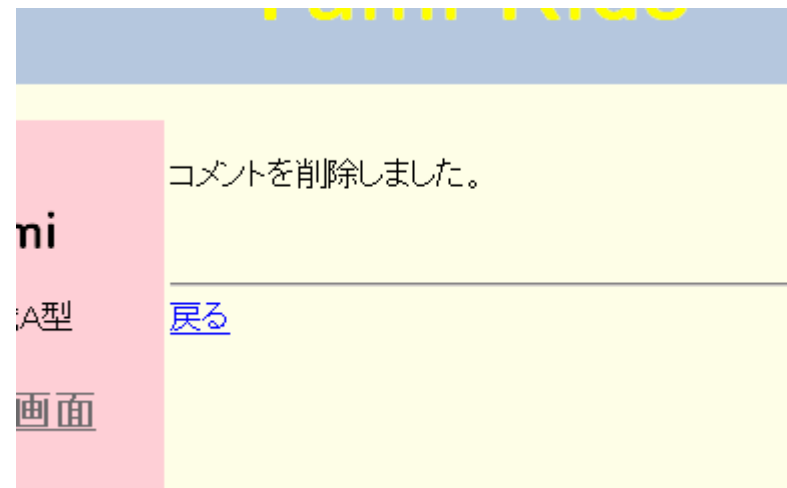
```

delete_comment.php

コメントが削除されるページ。削除されていれば下のような画面が表示される。

```
<?php
require("kanri_side.php");
?>
<br>
<?php
$com_id = $_GET['com_id'];
$id = $_GET['id'];

// SQL では delete 文を用いることでデータの削除を行いません。where 句で条件を指定すると、削除すべきデータを特定することができます。
$sql = "delete from comment where id = '$com_id'";
$data = mysql_query($sql, $connect);
if ($data == FALSE) {
    echo("<p>コメント・エラー</p>");
} else {
    echo("<p>コメントを削除しました。</p>");
}
print("<br><hr>¥n");
printf("<a href=¥\"comment_kanri.php?id=%d¥\">戻る</a>", $id);
mysql_close($connect);
?>
```



delete_kiji.php

記事とその記事のコメント全てを削除するか確認するページ。

```
<?php
require("kanri_side.php");
?>
<table class="kiji">
<?php
if ($_GET['id']) {
    $id = $_GET['id'];
} else {
    $id = $_POST['id'];
    $comment_name = $_POST['name'];
    $comment_honbun = $_POST['honbun'];
    $comment_date = $_POST['date'];
    $comment_url = $_POST['url'];
    $sql = "insert into comment (kiji_id, name, kiji, hiduke, url) values
('{$id}', '{$comment_name}', '{$comment_honbun}', '{$comment_date}', '{$comment_url}')";
    $data = mysql_query($sql, $connect);
    if ($data == FALSE) {
        echo "コメント・エラー";
    }
}

$sql = "select * from kiji where id='{$id}'";
$data = mysql_query($sql, $connect);
```

The screenshot shows a web page with a header "Yumi Kids" in yellow. On the left is a pink sidebar with navigation links: "yumi", "/21歳A型", "登録画面", "の管理", and "の管理". The main content area has a yellow background. A comment by "yumi" is shown with a title "タイトル:なんでだろお??", content "なんじゃこりゃ-----!!!!!!!!!!!!", and date "【2007-10-22】". Below the comment is a confirmation message: "この記事とコメントを削除してもよろしいですか?". There are two buttons: "YES" in blue and "NO" in purple.

```

while ($record = mysql_fetch_array($data)){
    $id = $record['id'];
    $title = $record['title'];
    $kiji = $record['kiji'];
    $date = $record['hiduke'];
    $sql_comment = "select * from comment where kiji_id = '$id'";
    $data_comment = mysql_query($sql_comment, $connect);
    if ($data_comment) {
//ここで数を数えて、あったらその数を代入。select 文によって返された行の数を知るには mysql num rows()を用います。
        $num_comment = mysql_num_rows($data_comment);
    } else {
//なかったら 0 を代入
        $num_comment = 0;
    }

    printf("<tr><td>タイトル : %s</td><td></td></tr>", $title);
    printf("<tr><td colspan=¥"2¥">%s</td></tr>", nl2br($kiji));
    printf("<tr><td>【%s】 </td>", $date);
    printf("<td><font color=¥"blue¥">コメント(%d)</td></tr>", $num_comment);
}
print("</table><br>¥n");

$sql = "select * from kiji order by id";
print("<table class=¥"sakujo¥">");
print("<tr><td colspan=¥"2¥" class=¥"sakujo¥">この記事を削除してもよろしいですか？ </td></tr>");
printf("<tr><td class=¥"sakujoy¥"><a href=¥"yes.php?id=%d¥">YES</a></td>", $id);

```

```
printf("<td class=%s"sakujon%><a href=%s"hensyu.php?id=%d%>NO</a></td></tr>", $id);  
print("</table>");  
mysql_close($connect);  
?>
```

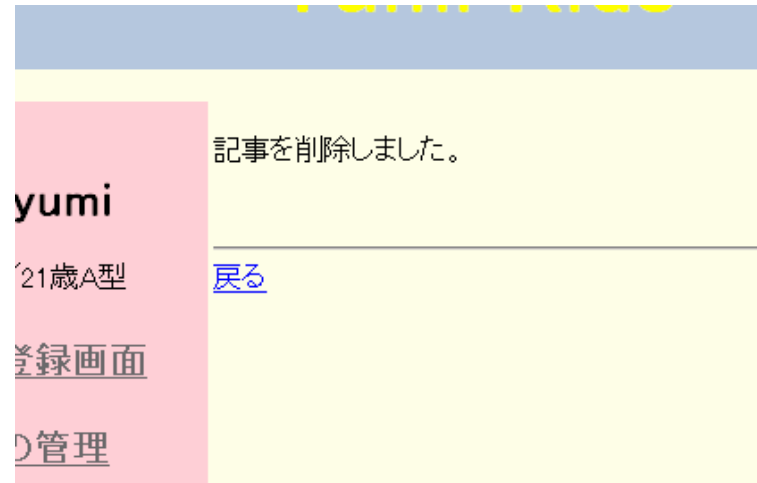
yes.php

記事を削除するページ。削除されていれば右下の画面が表示される。

```
<?php
require("kanri_side.php");

//記事の id を取得。
$Sid = $_GET['id'];
//記事を削除する。
$sql = "delete from kiji where id = '$Sid'";
$data = mysql_query($sql, $connect);

//削除する記事のコメントを削除する。
$sql_com = "delete from comment where kiji_id = '$Sid'";
$data_com = mysql_query($sql_com, $connect);
if ($data == FALSE) {
    echo("<p>記事エラー</p>");
} else {
    echo("<p>記事を削除しました。 </p>");
}
print("<br><hr>¥n");
printf("<a href='¥\"kanri.php?id=%d¥\">戻る</a>", $Sid);
mysql_close($connect);
?>
```



update_kiji.php

記事を編集するページ。前回の記事が入力された状態で表示される。

```
<?php
```

```
require("kanri_side.php");
```

```
//日付を画像のように表示したい時はこのように入力する。
```

```
$today = getdate();
```

```
$date = sprintf("%d-%d-%d", $today["year"], $today["mon"], $today["mday"]);
```

```
$id = $_GET['id'];
```

```
$sql = "select * from kiji where id='$id'";
```

```
$data = mysql_query($sql, $connect);
```

```
if ($data) {
```

```
    $record = mysql_fetch_array($data);
```

```
    $title = $record['title'];
```

```
    $kiji = $record['kiji'];
```

```
    $date = $record['hiduke'];
```

```
} else {
```

```
    printf("$sql¥n");
```

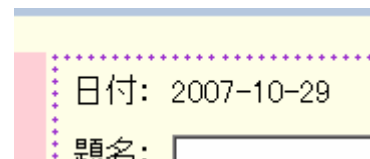
```
    die("交信エラー");
```

```
}
```

```
?>
```

```
<form action="toroku.php" method="POST">
```

```
<table class="kiji">
```



日付: 2007-10-29
題名:

GET か POST の二択。

POST は大量のデータを転送するときや、セキュリティ上データを露出しない方が望ましいデータの送信などに使われる。

POST はフォームから送信できる文字数に制限がないのに対して、GET は上限が決まっている。

```
<tr><td class="toroku">日付 : </td><td><?php echo $date;?></td></tr>
```

//value は送信される内容を指定する。ここでは前回入力した内容が入力された状態で表示される。

```
<tr><td class="toroku">題名 : </td><td><input type="text" name="title" value="<?php echo $title;?>" size="40"></td></tr>  
<tr><td valign="top">本文:</td><td><textarea name="honbun" cols="40" rows="7"><?php echo $kiji;?></textarea></td></tr>  
<tr><td></td><td align="center"><input type="submit" value="投稿"> <input type="reset" value="クリア"></td></tr>  
</table>
```

```
<input type="hidden" name="date" value="<?php echo $date;?>">  
//id を使って呼び出すから、新規以外の場合は大抵 id が必要。  
<input type="hidden" name="id" value="<?php echo $id;?>">  
<input type="hidden" name="update" value="update">  
</form>
```

The screenshot shows a web page with a header 'Yumi Kids' in yellow text on a blue background. Below the header is a form area with a light yellow background. On the left side of the form area, there is a vertical pink sidebar with the text 'Yumi', '歳A型', '録画面', '管理', 'の管理', and 'ページ'. The main form area contains the following elements: '日付: 2007-10-22', '題名: なんてだろお??' (with a text input field), '本文: なんじゃこりゃ-----!!!!!!!!!!' (with a text area), and two buttons at the bottom: '投稿' (Post) and 'クリア' (Clear). The form is enclosed in a dotted purple border.

pass.html

ID と password を入力するページ。ここから管理者ページに飛ぶ。

```
<h3>ID とパスワードを入力してください。 </h3>
```

```
<form action="kanri.php" method="POST">
```

```
<table class="pass">
```

```
<tr><td align="right">ID:</td><td><input type="text" name="logid" size="20"></td></tr>
```

//<input>タグの type 属性を "password" に設定したもので、主にパスワード入力用のためマスクされたテキストフィールドを作成する場合に使われる。

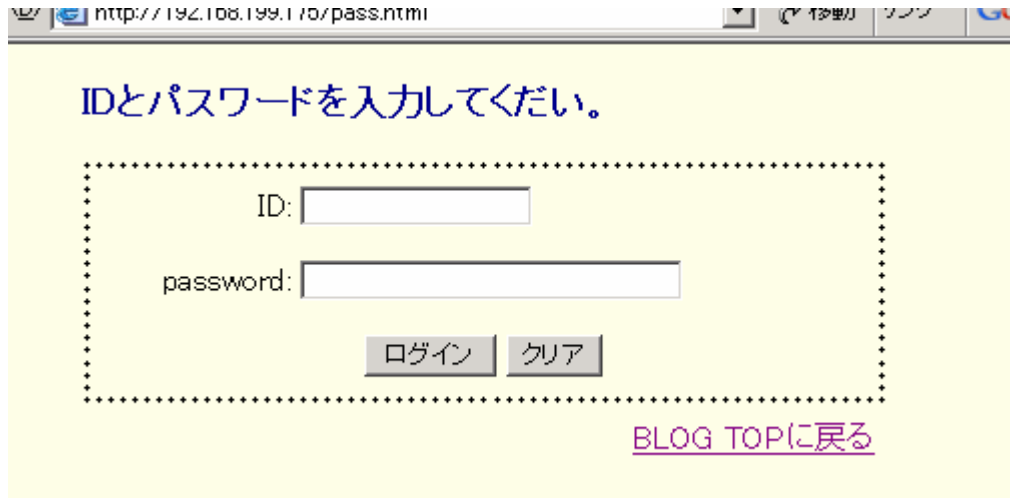
//一行のテキスト入力フィールドと同様の働きをするが、ブラウザ上では文字は (*) 等に表示されます。

```
<tr><td align="right">password:</td><td><input type="password" name="pass" size="35"></td></tr>
```

```
<tr><td colspan="2" align="center"><input type="submit" value="ログイン"> <input type="reset" value="クリア"></td></tr>
```

```
</table>
```

```
<table width=400px><tr><td align="right"><a href="blog_top.php">BLOG TOP に戻る</a></td></tr></table>
```



kanri.php

ログインできたと同時に、管理者ページのトップを表示する。ログインできなければ、下のような画面が表示される。

```
<?php
```

```
//ここで LOGID と PASS を定義しておく。
```

```
define("LOGID","yumi");
```

```
define("PASS","sensei");
```

```
$logid = $_POST['logid'];
```

```
$pass = $_POST['pass'];
```

```
// logid と pass が空でなければ
```

```
if ($logid != "" && $pass != "") {
```

```
// cookie に保存する。ただし、保存したものは1時間経つとリセットされる。
```

```
    setcookie("logid", $logid, time()+3600);
```

```
    setcookie("pass", $pass, time()+3600);
```

```
}
```

```
//cookie に保存されている logid と pass、もしくは入力された logid と pass が定義したものに一致しなければ終了。
```

```
if (!(LOGID == $_COOKIE['logid'] && PASS == $_COOKIE['pass']) && !(LOGID == $logid && PASS == $pass)) {
```

```
    echo "<p>ID かパスワードが違います</p>";
```

```
    echo "<p><a href=¥\"blog_top.php¥\">BLOG TOP に戻る</a></p>";
```

```
    echo "<p><a href=¥\"pass.html¥\">もう一度入力</a></p>";
```

```
    exit();
```

```
}
```

```
require("kanri_side.php");
```



```
mysql_close($connect);
```

```
?>
```