



G's ACADEMY
TOKYO

SESSION & LOGIN



前回の課題

～ブックマークアプリにUSER管理画面を作る～

- ・ ユーザーを追加 (登録画面、登録処理)
- ・ ユーザー一覧表示 (一覧画面、[更新/削除リンク](#))
- ・ ユーザーを変更 ([更新画面](#)、[更新処理](#))
- ・ ユーザーを削除 ([削除処理](#))

できる人は！

管理ユーザー：管理画面

◇管理ユーザーテーブルを作成

- **DB名:** `gs_db`
- **Table名:** `gs_user_table`
- **Field名:**
 - `id:` `int(12)` `AUTO INCREMENT PRIMARY KEY`
 - `name :` `var_char(64)`
 - `lid:` `var_char(128)`
 - `lpw:` `var_char(64)`
 - `kanri_flg:` `int(1)` ※0=一般, 1=管理者
 - `life_flg:` `int(1)` ※0=使用中, 1=使用しなくなった

※Fieldの右にあるのは、データ型(Type)です。

<http://mysql.akarukutanoshiku.com/category5/entry21.html>

本日の授業内容

アジェンダ

- DB準備
- SESSION
- LOGIN機能
 - * password_hash / password_verify
- 管理処理

今日の授業準備

- 配布サンプル
htdocs/以下に”php04”フォルダを置く
- ” gs_db4 ”データベースを作成する
配布サンプル” SQL ”フォルダ内の
ファイルを2つをデータ「インポート」
 - gs_an_table.sql
 - gs_user_table.sql
- XAMPP使用の人→funcs.phpのパスワード設定

SESSION

■SESSION: 動作確認

SESSIONを使用したい場合には必ずファイルの最初に、

```
<?php  
session_start();
```

を記述する。

SESSION変数は「サーバー側に変数を保持」することが可能になる
※サーバー側においてるので、送信しなくても変数値を他ページを共有可能！！

◇サンプル

以下ファイルを順番に作ります。

1. session01.php
SESSION変数をセット
2. session02.php
SESSION変数をインクリメント (+1)

■ SESSION: IDの取得と表示

`session_start()`; すると、
そのサーバーにアクセスしてるクライアントにユニークIDを付与します。
`session_id()`; 関数で各ブラウザに割り振られたSESSION_IDを取得することができる記述しましょう！！

◇ `sessionid.php` (作って確認しましょう！)

```
<?php
```

```
//session_id を表示して確認しましょう！
```

```
session_start();
```

```
$sid = session_id();
```

```
echo $sid;
```

```
?>
```

SESSION IDはブラウザのどこに保存されてるの？

sessionid.phpをChromeブラウザで表示しましょう！

◇操作&確認方法：

Chrome ブラウザ → 右クリック → 検証 → Application → Cookies (localhost)

※session_idの英数字は全員違います。

The screenshot shows the Chrome DevTools Application tab with the 'Cookies' section expanded for 'http://localhost'. A red box highlights the 'PHPSESSID' cookie with the value '0lhtsc6t4ju9qdbh998hejt36s'. A red arrow points from this value to a separate box on the left containing the same session ID. The left box is labeled 'データ登録' (Data Registration). The 'Application' tab is selected, and the 'Cookies' section is expanded. The 'http://localhost' address bar is also highlighted.

データ登録

0lhtsc6t4ju9qdbh998hejt36s

Application

Filter

Name	Value
PHPSESSID	0lhtsc6t4ju9qdbh998hejt36s

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

http://localhost

■SESSION: IDの取得とSESSION変数へ値代入

```
<?php
session_start();
$cid = session_id();
$_SESSION["name"]="やまざき";
$_SESSION["num"]=1000;
$_SESSION["value"]=100;
?>
```

A diagram on a blue parallelogram background. On the left, the text `session_id()` is written. Three arrows point from the parentheses of `session_id()` to the following three lines of text: `$_SESSION["name"]`, `$_SESSION["num"]`, and `$_SESSION["auth"]`.

◇サーバー側の変数保持：

... /アプリケーション/MAMP/tmp/php/ 以下にファイルが作成されます。

ファイル名：session idsess *****

■SESSION:別ページでSESSION変数を取得表示

```
<?php
```

```
//1. 必ず”session_start();”関数を最初に行う!!
```

```
session_start();
```

```
//2. SESSION変数に値を代入!!
```

```
$_SESSION["name"]="やまざき";
```

```
$_SESSION["num"]=1000;
```

```
//3. SESSION変数に預けた値を表示
```

```
echo $_SESSION["name"];
```

```
?>
```

ユーザーデータベース

アンケートシステムのDB構築

◇テーブル作成

前回作ってます

- **DB名:** `gs_db`
- **Table名:** `gs_an_table`
- **Field名:**
 - `id:` `int(12)` `AUTO INCREMENT PRIMARY KEY`
 - `name :` `var_char(64)`
 - `email :` `var_char(128)`
 - `age:` `int(3)`
 - `naiyou :` `text`
 - `indate:` `datetime`

※フォームの項目を増やした場合こちらのFieldも増やしましょう。

※Fieldの右にあるのは、データ型(Type)です。

<http://mysql.akarukutanoshiku.com/category5/entry21.html>

アンケートシステムのDB構築

◇ユーザテーブルを作成

今日使います！

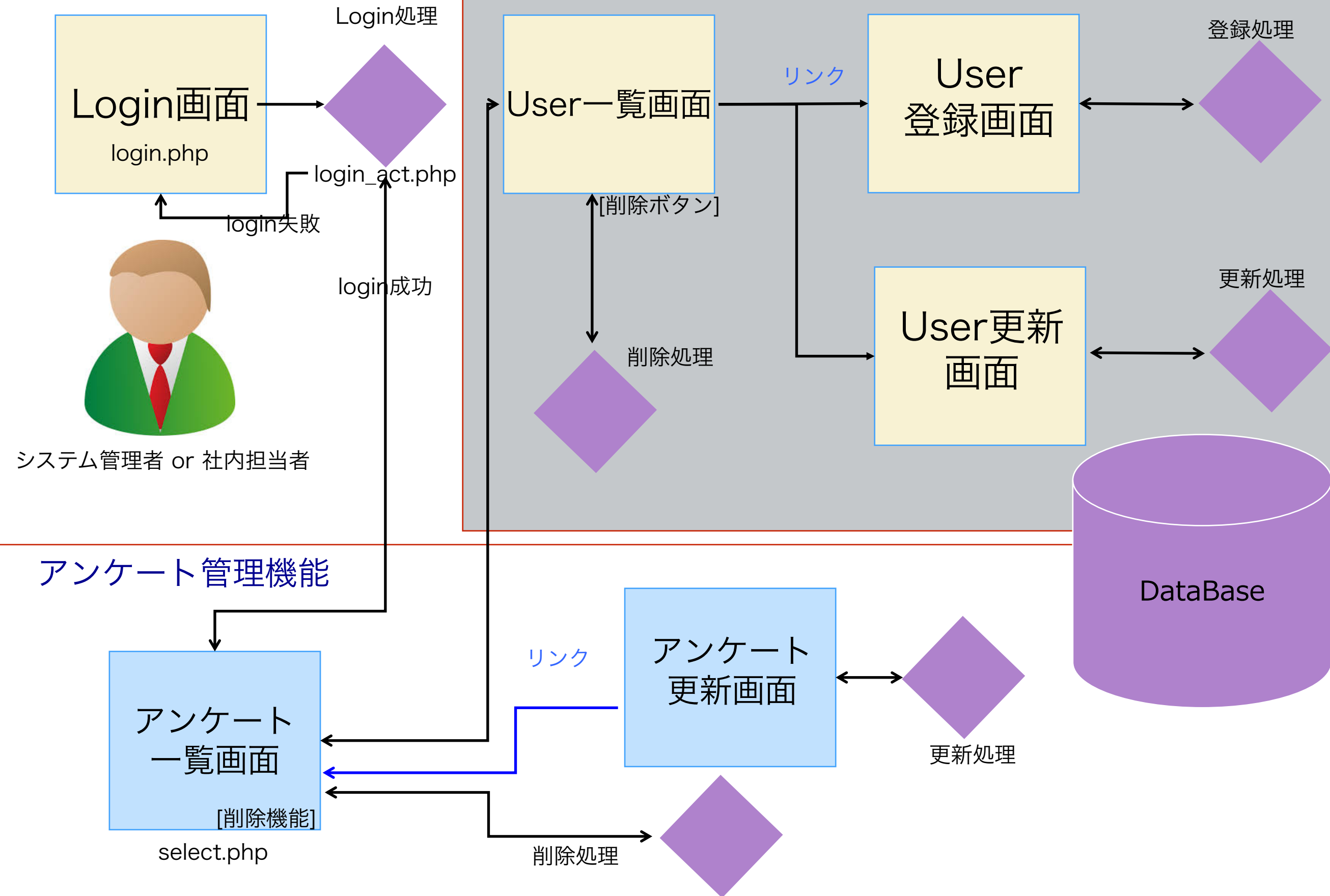
- **DB名:** `gs_db`
- **Table名:** `gs_user_table`
- **Field名:**
 - `id:` `int(12)` `AUTO INCREMENT PRIMARY KEY`
 - `name :` `var_char(64)`
 - `lid:` `var_char(128)`
 - `lpw:` `var_char(255)`
 - `kanri_flg:` `int(1)` ※0=一般者, 1=管理者
 - `life_flg:` `int(1)` ※0=使用中, 1=使用しなくなった

※Fieldの右にあるのは、データ型(Type)です。

<http://mysql.akarukutanoshiku.com/category5/entry21.html>

SYSTEM完成図

管理システム完成図



【再度確認】 認証機能の流れ

認証機能 (スクラッチ)

OKの場合 : LOGIN画面 → 認証処理 → 一覧画面
login.php → login_act.php → select.php

NGの場合 : LOGIN画面 → 認証処理 → 一覧画面
login.php → login_act.php → login.php

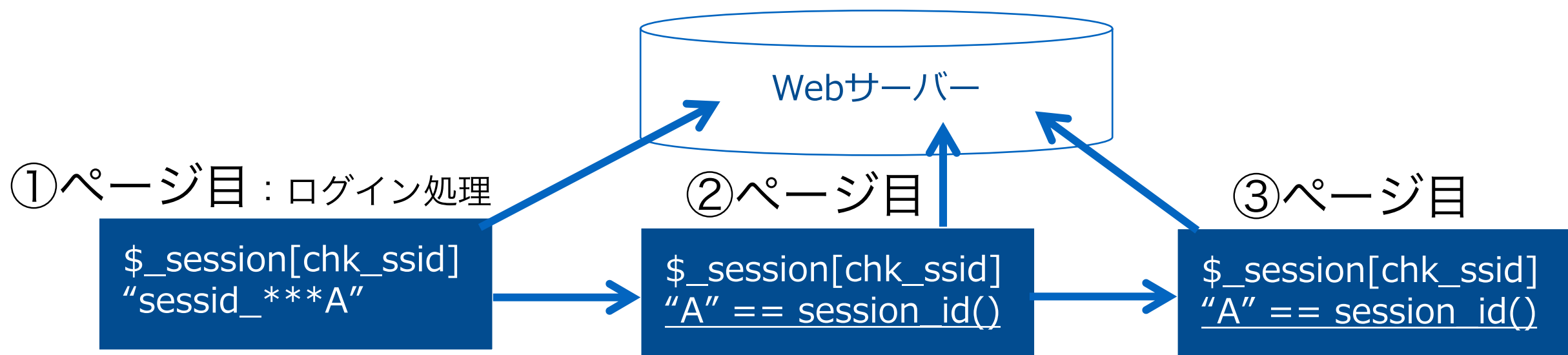
※PointはID&PasswdをSELECT文でユーザーの有無確認、SESSION使用

■SESSION: チェック

// 2. セッションチェック

```
if(  
    !isset($_SESSION["chk_ssid"]) ||  
    $_SESSION["chk_ssid"] != session_id()  
)  
{  
    exit("LOGIN ERROR");  
}  
else{  
    session_regenerate_id(true),  
    $_SESSION["chk_ssid"] = session_id();  
}
```

ここ超大事!!!
ページ遷移する毎にセッション
IDを自動で新しく発行してくれ
ます!!



```
<?php
//必ずsession_startは最初に記述
session_start();

//現在のセッションIDを取得
$sold_sessionid = session_id();

//新しいセッションIDを発行 (前のSESSION IDは無効)
session_regenerate_id( true ); //trueが大事！

//新しいセッションIDを取得
$new_sessionid = session_id();

//旧セッションIDと新セッションIDを表示
echo "古いセッション: $sold_sessionid<br />";
echo "新しいセッション: $new_sessionid<br />";
?>
```

関数化

SESSIONチェック まわりの処理を関数化！

LOGOUT処理

■ ログアウト処理

logout.php

```
<?php
session_start();

//SESSION初期化
$_SESSION = array();

//Cookieに保存してたSessionIDの保存期間を過去にして破棄
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), '', time()-42000, '/');
}

//SESSION削除
session_destroy();
header("Location: login.php");
exit();

?>
```


課題発表

【課題1】 ログイン認証&認証チェック機能を付ける

- ブックマークアプリとユーザー管理機能を合体

ブックマークアプリに”ユーザー管理機能”のリンクを作成。

[\[ブックマーク登録 | ブックマーク表示 | ユーザー登録 | ユーザー表示\]](#)

[※USER管理機能を課題で作ってる人限定です。](#)

- ログイン認証

login.phpを作成。認証後は「ブックマークアプリ」一覧表示画面に遷移すること。

- 認証チェック

”ブックマークアプリ”&”ユーザー管理機能”には、sessionをチェックするロジックを記述、認証チェックをおこなう。

今日の授業内容を「ブックマークアプリ」にも同じことをするだけです。

【課題】 ログイン認証を必要としない画面を追加

- ログインしてなくても見れるページを2ページ作成
- ログイン認証を必要としない
ブックマークアプリ「一覧表示画面」を作成
※select.php の認証しなくても見れるバージョン, 一部を隠す
- ログイン認証を必要としない
ブックマークアプリ「詳細画面」を作成
※detail.php の認証しなくても見れるバージョン, 一部を隠す

◇管理FLGでの表示の違いを作しましょう

◆ kanri_flg = 1

ユーザー登録・変更・表示のメニューを表示
リンク例)

[\[ブックマーク登録 | ブックマーク表示 | ユーザー登録 | ユーザー表示\]](#)

◆ kanri_flg = 0

ユーザー関連のメニューは非表示
リンク例)

[\[ブックマーク登録 | ブックマーク表示\]](#)

パスワードのhash化

課題：できる人は！！
(プロトタイプでは必要ないけど)

～ 2つの関数のみ使用 ～

◇パスワードハッシュ作成

password_hash("登録する文字",
~~*PASSWORD_DEFAULT*~~);

※ DB:Passwordカラム型をvarchar(255)に変更!

※ ユーザー登録時に使用 (ハッシュ化してDBに登録しておくため)

<http://php.net/manual/ja/function.password-hash.php>

◇パスワードのマッチ チェック

password_verify("パスワード入力値", "DB値");

※ LOGIN認証時に使用する (ハッシュ文字と入力文字を比較して判定する関数)

<http://php.net/manual/ja/function.password-verify.php>

Passwordハッシュ化：事前準備

データベース >> gs_user_table >> lpwカラムを変更
varchar(255) に！



サーバ: localhost » データベース: gs_db » テーブル: gs_user_table

表示 構造 SQL 検索 挿入 エクスポート

名前	データ型 ?	長さ/値 ?	デフ:
lpw	VARCHAR	255	な

1. テストデータのパスワードをハッシュ化

①hash.phpを作成 (テストデータ作成用)

```
1 <?php
2
3 //パスワード作る場合
4 //ユーザー管理画面の登録する前に以下処理が必要になる
5 $pw = password_hash("test", PASSWORD_DEFAULT);
6 echo $pw;
7
8 ?>
```

②ブラウザで表示 → 文字をコピー

localhost/gs_js/lab4/PHP/php04/hash.php

\$2y\$10\$jmot7MkoGd4R7Z0bNCFwVeTWweYUWOPViFc9N1vqtNydyrLPIX3Cu

③DBの"gs_user_table" のパスワードを変更

id	name	lid	lpw
1	TEST	test	\$2y\$10\$jmot7MkoGd4R7Z0bNCFwVeTWweYUWOPViFc9N1vqtNydyrLPIX3Cu

user登録処理にも"password_hash()"を使い
パスワードをハッシュ化して登録させます。

ハッシュ化する前はtest

2. "login_act.php"の一部を修正

①SQLとbindValueを修正

修正

```
$sql = "SELECT * FROM gs_user_table WHERE lid=:id";  
$stmt = $pdo->prepare($sql);  
$stmt->bindValue(':id', $lid);  
$res = $stmt->execute();
```

②password_verify関数を使ってパスワードを比較

```
if( password_verify($lpw, $val["lpw"]) ){  
    $_SESSION["chk_ssid"] = session_id();  
    $_SESSION["kanri_flg"] = $val['kanri_flg'];  
    $_SESSION["name"] = $val['name'];  
    header("Location: select.php");  
}else{  
    //logout処理を経由して全画面へ  
    header("Location: login.php");  
}
```