



**G's ACADEMY**  
**TOKYO**

# SESSION & LOGIN



# 本日の授業内容

# アジェンダ

- DB準備（前回のgs\_db3を使用！！）
- SESSION
- LOGIN機能
  - \* password\_hash / password\_verify
- 管理処理

# 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。

0lhtsc6t4ju9qdbh998hejt36s

データ登録

1 Application

Name	Value
PHPSESSID	0lhtsc6t4ju9qdbh998hejt36s

2 http://localhost

3

## ■SESSION：IDの取得とSESSION変数へ値代入

```
<?php
session_start();
$sid = 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 this text to the right, where the following session variables are listed: `$_SESSION["name"]`, `$_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();"関数を最初に実行！！
```

```
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！

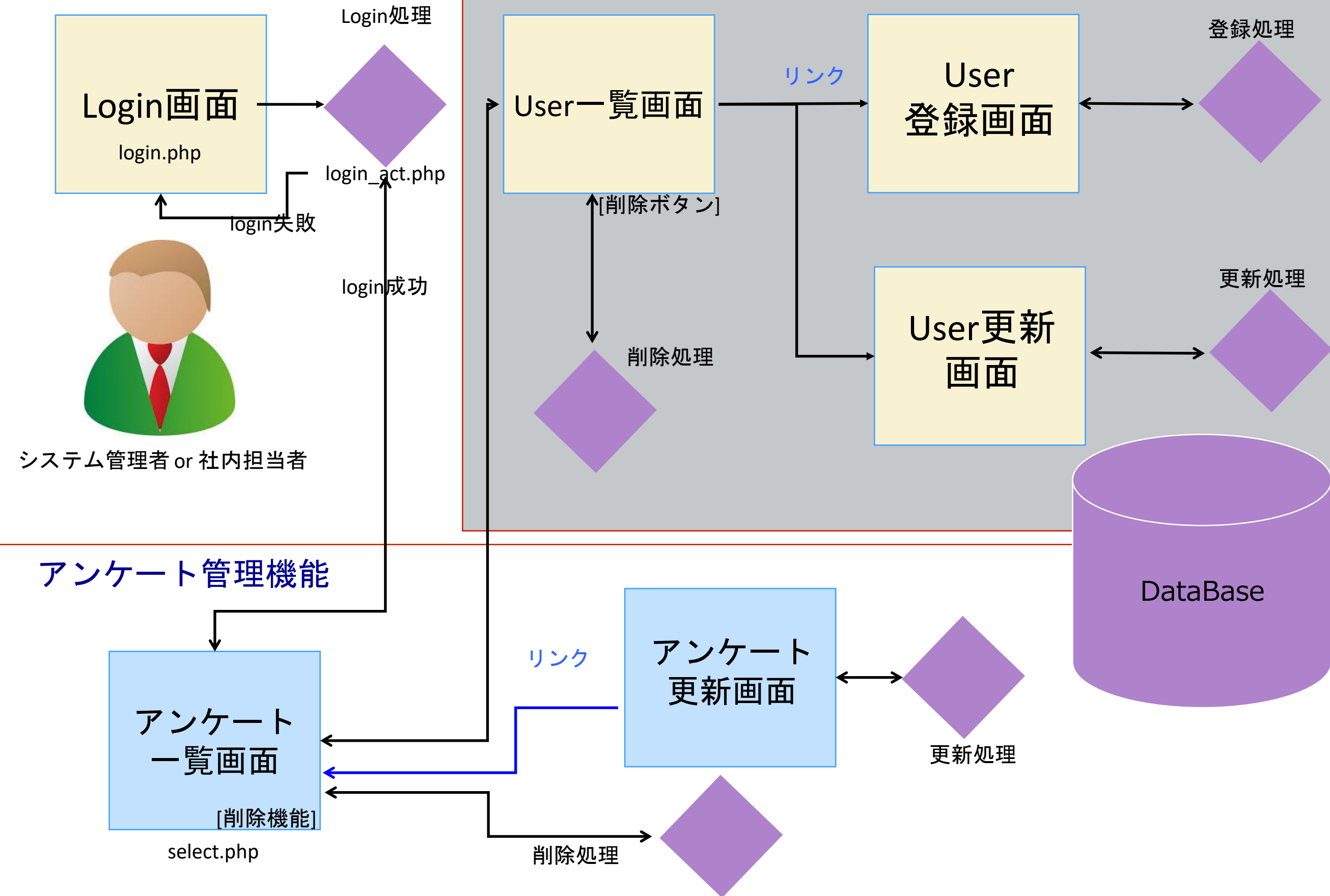
- **DB名:** `gs_db`
- **Table名:** `gs_user_table`
- **Field名:**
  - `id:` `int(12)` `AUTO INCREMENT PRIMARY KEY`
  - `name :` `var_char(64)`
  - `lid:` `var_char(128)`
  - `lpw:` `var_char(255)`
  - `kanri_flg:` `int(1)` ※0=一般者, 1=管理者
  - `life_flg:` `int(1)` ※0=使用中, 1=使用しなくなった

※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

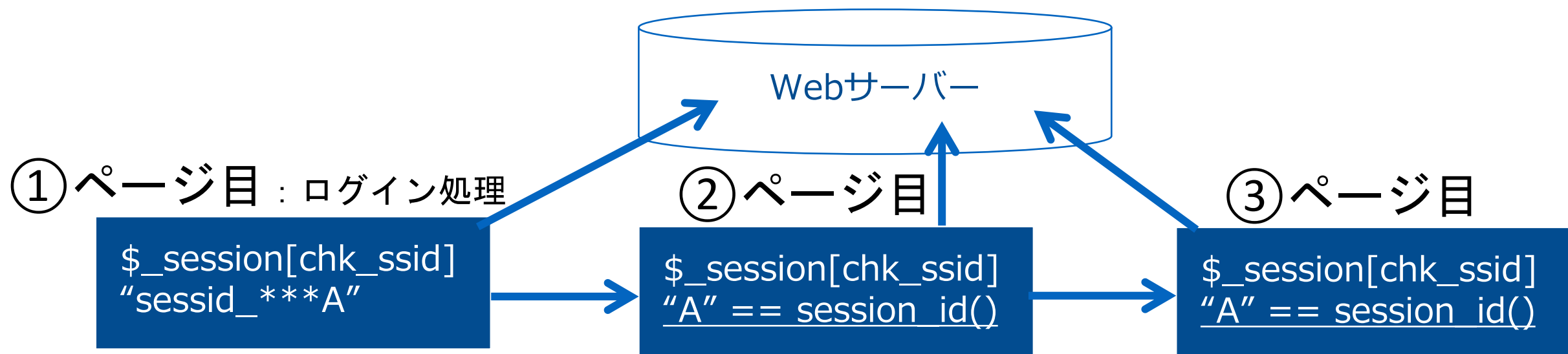
※PointはID&PasswdをSELECT文でユーザーの有無確認、SESSION使用

## ■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

# 前回から追加

## ～ USER管理画面 ～

- 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](#))

# 【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。



# ◇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](#)