

パスワードのhash化

パスワード保護を強化

◇ password_hash

パスワードハッシュを作る

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

※(DB:Passwordカラム型をvarchar(255)に変更)

◇ password_verify

パスワードがハッシュにマッチするかどうかを調べる

```
password_verify( 'パスワード入力値', 'DB値');
```

詳しくはこちらを参考に

password_hash

パスワードハッシュを作る

(DB:Passwordカラム型をvarchar(255)に変更)

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

password_verify

パスワードがハッシュにマッチするかどうかを調べる

<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 |

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

③gs_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

"login_act.php"の一部を修正

①SQLとbindValueを修正

```
$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");  
}
```