第８回　データベース接続（SELECT）

８－１．データベース接続

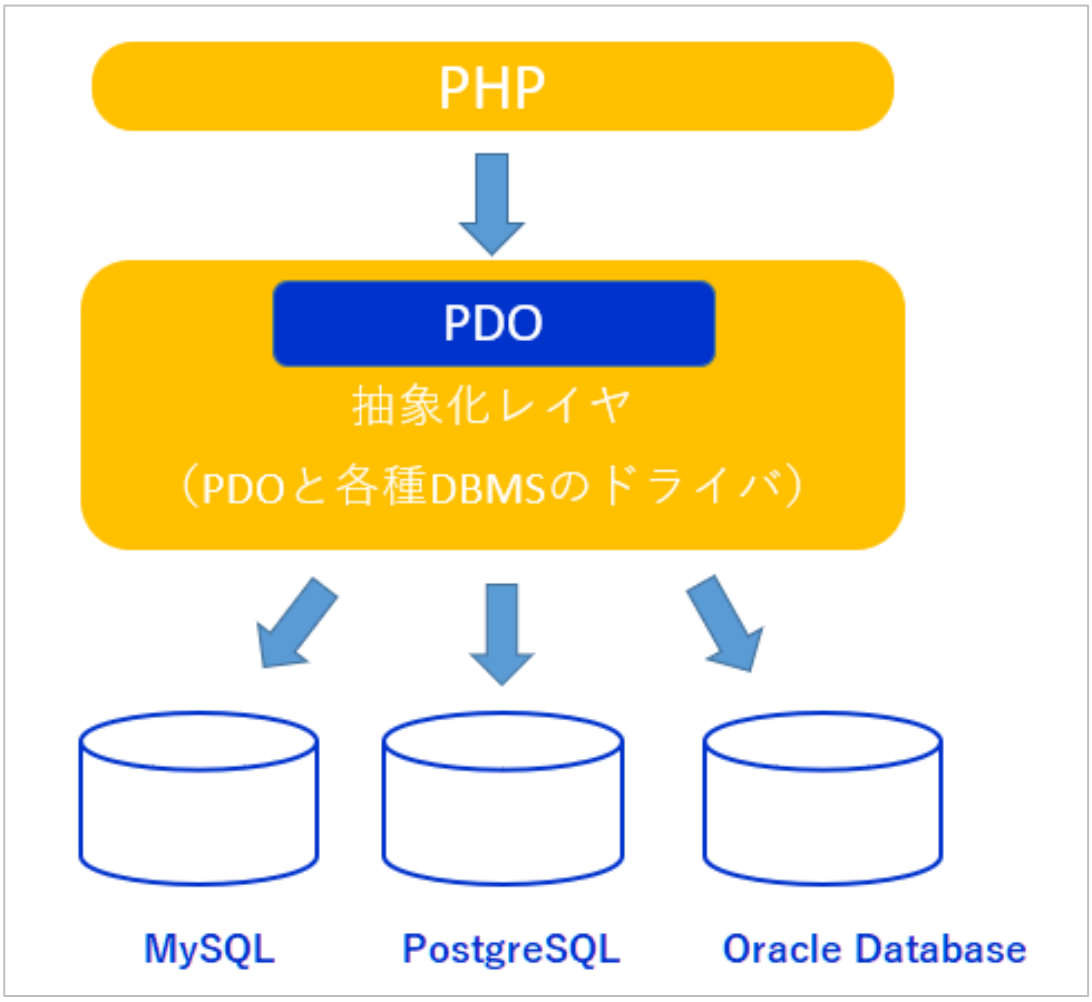
８－１－１．PDO とは

PHP から SQL を実行するために、PDO（PHP Data Object）を使用します。

PDO とは、PHP の拡張モジュールです。データベース製品の違いを吸収するための統一されたインターフェース（PDO クラス）を提供します。

従来は様々なデータベース（postgreSQL, Microsoft SQL Server, Oracle Database）などにより、プログラムの書き方が異なりました。そのため、利用するデータベースを変更すると、PHP プログラムも大幅に修正する必要がありました。

このような手間を省くため、PHP とデータベース（正確には DBMS）の間に抽象化レイヤを挟んで、各種 DBMS の違いを抽象化レイヤで吸収し、DBMS が異なっても同じ処理をできるようにするというのが、PDO の役割です。抽象化レイヤには、PDO と PDO が使用する各種ドライバが含まれています。



８－１－２．データベースへの接続

データベース処理は下記のような手順になります。Sample コードと併せて確認していきましょう。

① PDO クラスをインスタンス化する

② PDO の動作オプションを指定する

③ SQL 文の準備と実行

④ SQL 実行結果の処理

⑤ PDO オブジェクトを破棄

★Sample コード（SELECT　＊条件指定なしの場合）

Sample のデータベース情報**（皆さんが使用する DB とは違うものを使用しています）**

・ホスト名　　：　localhost

・DB 名　　：　malldb

・DB ユーザー　：　malluser

・DB パスワード　：　mall

・テーブル　　：products

```php
$dsn = "mysql:host=localhost;dbname=malldb;charset=utf8mb4";

$db = new PDO($dsn, "malluser", "mall");
$db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);

$sql = "SELECT * FROM products";
$stmt = $db->prepare($sql);
$stmt->execute();

$result = [];
while($rows = $stmt->fetch(PDO::FETCH_ASSOC)){
  $result[] = $rows;
}

$stmt = null;
$db = null;
```

③ ② ⑤ ① ④

① PDO インスタンス化

■構文

A B C

new PDO([接続先 DB 情報], [DB のログイン ID], [DB のパスワード])

コンストラクタの引数は、３つです。

A

接続先 DB 情報を表す文字列を、DSN（Data Source Name）といいます。

データベースにより書式が異なりますが、MySQL の場合は下記のようになります。

mysql:host=[接続先ホスト名または IP アドレス]; dbname=[接続先 DB 名];

charset=[接続時の文字エンコーディング名]

＊緑色部分には環境に合わせた値を記載します。

B

DB のログイン ID を記載します。

C

DB のログイン ID を記載します。

② PDO の動作オプションを指定

■構文（PDO::setAttribute メソッド）

```
$インスタンス変数->setAttribute([オプション種別], [オプション値])
```

■setAttribute メソッドで指定できるオプション定数

オプション種別	指定方法	意味	デフォルト値
PDO:ATTR_CASE	定数で指定	SELECT 結果のカラム名について大文字／小文字の扱いを指定	PDO::CASE_NATURAL
PDO::ATTR_ERRMODE	定数で指定	エラー通知方法を指定	PDO::ERRMODE_SILENT
PDO::ATTR_ORACLE_NULLS	定数で指定	NULL と空文字の変換方法を指定。Oracle 以外でも使用可能	PDO::NULL_NATURAL
PDO::ATTR_AUTOCOMMIT	真偽値で指定	自動コミットするか否かを指定	true
PDO::ATTR_EMULATE_PREPARES	真偽値で指定	プリペアドステートメントのエミュレーションを有効にするか否かを指定	true
PDO::ATTR_DEFAULT_FETCH_MODE	定数で指定	SELECT 結果の PHO 変数へのマッピング方法を指定	PDO::FETCH_BOTH

■PDO：：ATTR_CASE に指定できる値

定数名	意味
PDO:CASE_LOWER	小文字に変換する
PDO::CASE_NATURAL	変換しない
PDO::CASE_UPPER	大文字に変換する

■PDO::ATTR_ERRMODE に指定できる値

定数名	意味
PDO::ERRMODE_SILENT	エラーを出力しない。PDO::errCode メソッドでエラーコード取得のみ可能
PDO::ERRMODE_WARNING	E_WARNING 定数レベルのエラーを出力
PDO::ERRMODE_EXCEPTION	例外（おもにＰＤＯException）をスローする

■PDO::ATTR_ORACLE_NULLS に指定できる値

定数名	意味
PDO::NULL_NATURAL	変換しない
PDO::NULL_EMPTY_STRING	空文字を NULL に変換する
PDO::NULL_TO_STRING	NULL を空文字に変換する

■PDO::ATTR_DEFAULT_FETCH_MODE に指定できる値

定数名	意味	結果セットの変数の参照例
PDO::FETCH_ASSOC	カラム名をキーとした連想配列を返す	$result['column']
PDO::FETCH_NUM	0 で始まる配列を返す	$result[0]
PDO::FETCH_BOTH	FETCH_ASSOC と FETCH_NUM をミックスした配列を返す	$result[0] / $result['column']

PDO::FETCH_OBJ	カラム名をプロパティに持つインスタンスを返す	$result->column

③ SQL 文の準備と実行

■構文（PDO::prepare メソッド） SQL の準備

```
$ PDO ステートメント変数 ＝$インスタンス変数->prepare([SQL])
```

PHP の変数値を入れる箇所は、プレースホルダとして設定しておきます。そのまま文字列に変数名を連結しないようにしてください。

例：＊赤字の箇所がプレースホルダです。冒頭にコロンを付けることでプレースホルダを表します。

```
$ PDO ステートメント変数 ＝$インスタンス変数->prepare('SELECT * FROM PRODUCT WHERE NAME = :name');
```

プレースホルダとは、仮に確保された場所のことで、後ほど何らかの値をセットすることになります。

■構文　プレースホルダに値をバインドする

```
$PDO ステートメント変数->bindParam([プレースホルダ名], [バインドする値], [値のデータ型])
```

■値のデータ型に指定できる定数

定数名	意味
PDO::PARAM_BOOL	真偽値型
PDO::PARAM_NULL	NULL
PDO::PARAM_INT	整数型
PDO::PARAM_STR	文字列型、または小数型

PDO::PARAM_LOB	ラージオブジェクト型

■構文（PDOStatement::execute メソッド） SQL の実行

```
$ PDO ステートメント変数->execute( )
```

④ SQL 実行結果の処理

SELECT の場合、実行結果を fetch メソッドまたは fetchAll メソッドにて取得します。

SQL を実行すると、結果セット（メモリ上に作成された仮想的なテーブル）が作成されます。フェッチとは結果セットから 1 レコードっまたは全レコード分を取り出して PHP の変数に割り当てることです。

■構文（PDOStatement::fetch メソッド）

```
$ PDO ステートメント変数->fetch([フェッチモード定数])

                または

$ PDO ステートメント変数->fetchAll([フェッチモード定数])
```

*フェッチモード定数に指定できる値は、「■PDO::ATTR_DEFAULT_FETCH_MODE に指定できる値」と同じです。

fetch メソッドの方が PHP のメモリ使用量を大幅に抑えられます。

Sample コードでは、1 件ずつ取得した結果を、変数$result に代入しています。

⑤ PDO オブジェクトを破棄

DB の処理が終了したら、接続を閉じましょう。接続の順番は以下の通りです。

1. PDO ステートメント変数を閉じるには null を代入します。

2．　　PDO インスタンス変数を閉じるには null を代入します。

＊明示的にこれを行わなかった場合は、スクリプトの終了時に自動的に 接続が閉じられます。