

php でデータベースに対して処理する場合、**PDO**と言う**クラス**を使用します。このクラスを使用する事によって、**MySQL** のみでは無く他の を利用する場合にコードの移行が容易になります。

利用するには、クラスである為、**new** **キーワード**を使用して を実行してインスタンスを生成して変数にセットして使用します。

```
try {
    $pdo = new PDO( $GLOBALS["connect_string"], $GLOBALS["user"], $GLOBALS["password"] );
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch (PDOException $e) {
    print "エラー : {$e->getMessage()}";
    exit();
}
```

PDO の標準的な利用方法は、まず用意した SQL 文字列を メソッドに渡して**実行の準備**をします。この際、SQL 文字列の中に**名前付きパラメータ**を用いて、SQLの内容を変数によって置換する準備をしておきます。

```
SELECT * FROM 社員マスタ WHERE 社員コード = :score
```

その後、 メソッドを使用してSQL 文を完成させ、 メソッドで SQL文を実行します。SQL 文が **SELECT** によりデータの読み出しの場合は、さらに メソッドによって列名による連想配列を行単位で取得して使用し、SQL が更新の場合は、SQL を実行する事によって更新が完了します。

SQL が更新の場合、PHP の変数から名前付きパラメータで設定する場合、データベースの列定義に対して一般的に 型か 型かで変換方法を決めます。但し、その二つの型以外にデータベースでは という値をセットする時に注意が必要です。

SELECT を使用する SQL 文でデータを読み出す際、 型の列には注意が必要です。**MySQL が標準で返すフォーマット**では、画面表示やコントロールがうまく行かない場合があります。

この場合、たいていにおいて ***を使わずに全ての列名を記述**し、対象の列に対してはフォーマットを設定する を使用します。そして、SQL の 句で別名を設定し、PHP 側で利用する為の正しい列名を付け直します。

データベースから画面上のコンボボックスを作る場合も注意して下さい。コンボボックスは何もしなければ一覧の先頭が選択されて表示されます。画面表示時に正しくコンボボックスの内容を選択する為には、コンボボックスの一覧を読み込むループ時に、HTML の 属性を設置して正しく選択できるようにしてください。