

■PHPのPDO(PHP Data Objects)関数一覧(MySQLクラス)

※「chk」欄は、PHPのMySQL連携プログラミングで、利用頻度と処理上の必要度から判断した優先度です。

作成:2015.10.24/修正2016.06.29 yoshi

No.	関数	機能	chk	サンプル他
PDO クラス \$db->xxx() 形式				
1	PDO::beginTransaction	トランザクションを開始する	○	<code>\$db->beginTransaction();</code>
2	PDO::commit	トランザクションをコミットする	○	<code>\$db->commit();</code>
3	PDO::_construct	データベースへの接続を表す PDO インスタンスを生成する	◎	<code>\$db = new PDO('mysql:dbname=testdb;host=localhost','root','admin');</code>
4	PDO::errorCode	データベースハンドラにおける直近の操作に関連する SQLSTATE を取得する	△	<code>echo "DB実行エラー:". \$db->errorCode();</code>
5	PDO::errorInfo	データベースハンドラにおける直近の操作に関連するエラー情報(配列)を取得する (エラー情報の配列は、[0]:SQLSTATE エラーコード、[1]:ドライバ固有のエラーコード、[2]:ライバ固有のエラーメッセージ)	△	<code>print_r(\$db->errorInfo());</code>
6	PDO::exec	SQL ステートメント(更新、削除)を実行し、作用した行数を返す(SELECT文は、結果を返さない)	△	<code>\$count = \$db->exec("DELETE FROM persons WHERE seq_num = 5");</code>
7	PDO::getAttribute	データベース接続の属性を取得する	△	<code>\$status = \$db->getAttribute(PDO::ATTR_AUTOCOMMIT);</code>
8	PDO::getAvailableDrivers	利用可能な PDO ドライバの配列を返す	△	<code>print_r(PDO::getAvailableDrivers());</code>
9	PDO::inTransaction	ドライバ内で、現在トランザクションがアクティブになっているかどうかを調べる。現在アクティブな場合に TRUE、そうでないときに FALSE。(PHP5.3.3以降)	×	<code>var_dump(\$db->inTransaction());</code>
10	PDO::lastInsertId	最後に挿入された行の ID あるいはシーケンスの値を返す (注)commitの後に\$db->lastInsertId();とやると、0しか返ってきません。	○	<code>\$lastid = \$db->lastInsertId();</code>
11	PDO::prepare	文を実行する準備を行い、文オブジェクトを返す	◎	<code>\$sth = \$db->prepare('SELECT * FROM persons WHERE name=?');</code>
12	PDO::query	prepareを使わずにSQL文を実行 SQL ステートメントを実行し、結果セットを PDOStatement オブジェクトとして返す fetch()メソッドのオプション定数(PDO::FETCH_ASSOC他)が利用可能	△	<code>\$data = \$db->query("SELECT * FROM persons WHERE name='まきこ'", PDO::FETCH_ASSOC); foreach(\$data as \$row){ print_r(\$row); }</code>
13	PDO::quote	クエリ用の文字列をクオートする。入力文字列のまわりに必要であれば引用符を付け、入力文字列にあるシングルクオートをエスケープします。	△	<code>\$name = 'まきこ'; \$query = "SELECT * FROM persons WHERE name=" . \$db->quote(\$name);</code>
14	PDO::rollBack	トランザクションをロールバックする	○	<code>\$db->rollBack();</code>
15	PDO::setAttribute	属性を設定する	△	<code>\$db->setAttribute(PDO::ATTR_EMULATE_PREPARES, true);</code>
PDOStatement クラス \$sth->xxx() 形式				
16	PDOStatement::bindColumn	カラムを PHP 変数にバインドする	△	<code>\$sth->bindColumn(1, \$name); /* カラム番号によってバインド */ \$sth->bindColumn(name, \$name); /* カラム名によってバインド */</code>
17	PDOStatement::bindParam	指定された変数名にパラメータをバインドする	○	<code>\$sth = \$db->prepare("INSERT INTO persons (name,birth) VALUES(?,?); \$sth->bindParam(1, \$name, PDO::PARAM_STR); \$sth->bindParam(2, \$birth, PDO::PARAM_STR);</code>
18	PDOStatement::bindValue	値をパラメータにバインドする	△	<code>\$sth = \$db->prepare('SELECT name,num,kubun FROM xxx WHERE num < ? AND kubun = ?'); \$sth->bindValue(1, \$num, PDO::PARAM_INT); \$sth->bindValue(2, \$kubun, PDO::PARAM_STR);</code>
19	PDOStatement::closeCursor	他のSQL ステートメントを発行できるようにサーバーへの接続を解放し、ステートメントは再実行可能な状態のまま残されます。	△	<code>\$sth->closeCursor();</code>
20	PDOStatement::columnCount	結果セット中のカラム数を返す	△	<code>\$count = \$sth->columnCount();</code>
21	PDOStatement::debugDumpParams	プリペアドステートメントに含まれるSQLクエリ、パラメータ関連情報、キーの名前あるいは位置他を標準出力(debug用に有用)	○	<code>echo '<pre>'; \$sth->debugDumpParams(); echo '<pre>';</code>
22	PDOStatement::errorCode	文ハンドラにおける直近の操作に関連する SQLSTATE を取得する	○	<code>echo "DBステートメントエラー:". \$sth->errorCode();</code>
23	PDOStatement::errorInfo	文ハンドラにおける直近の操作に関連する拡張エラー情報を取得する	○	<code>print_r(\$sth->errorInfo());</code>
24	PDOStatement::execute	プリペアドステートメントを実行する	◎	<code>\$sth->execute(array('まきこ'));</code>
25	PDOStatement::fetch	結果セットから次の行を取得する	○	<code>while(\$row = \$sth->fetch(PDO::FETCH_ASSOC)) { ... }</code>

26	PDOStatement::fetchAll	全ての結果行を含む配列を返す	◎	<code>\$data = \$sth->fetchAll();</code>
27	PDOStatement::fetchColumn	結果セットの次行から単一カラムを返す。 引数に行から処理したい 0 で始まるカラム番号を指定します。何も値が与えられない場合、最初のカラムをフェッチ	△	<code>\$sth = \$db->prepare("SELECT name, kubun FROM xxx"); \$sth->execute(); \$result1 = \$sth->fetchColumn(); // nameのカラム \$result2 = \$sth->fetchColumn(1); // kubunのカラム</code>
28	PDOStatement::fetchObject	次の行を取得し、それをオブジェクトとして返す	△	
29	PDOStatement::getAttribute	文の属性を取得する	△	
30	PDOStatement::getColumnMeta	結果セットのカラムに対するメタデータを返す。この関数は、実験的 なもの	×	
31	PDOStatement::nextRowset	複数の行セットを返す文ハンドラで次の行セットに移動する。 1つ以上の行セットを返すスタアドプロシージャをサポートしていて、2番目以降の PDOStatement オブジェクトに関連する行セットにアクセスすることができます。 それぞれの行セットは、前の行セットと異なるカラムセットを含むことができます。	△	<code>\$sth = \$db->query(' xxx '); do { \$rowset = \$stmt->fetchAll(PDO::FETCH_NUM); if(\$rowset) {print_r(\$rowset);} } while (\$sth->nextRowset());</code>
32	PDOStatement::rowCount	直近の SQL ステートメントによって作用した行数を返す	○	<code>\$count = \$sth->rowCount();</code>
33	PDOStatement::setAttribute	文の属性を設定する	△	
34	PDOStatement::setFetchMode	この文に対するデフォルトのフェッチモードを設定する	△	<code>\$sth = \$db->query(' xxx '); \$result = \$stmt->setFetchMode(PDO::FETCH_NUM); while (\$row = \$stmt->fetch()) { print_r(\$rowset); }</code>

PDO ドライバ

35	PDO_MYSQL	MySQL 4.1 以降に存在するプリペアドステートメントを ネイティブにサポートしているという利点があります。	◎	<code>\$db = new PDO("mysql:dbname=testdb;host=xxx;port=xxx","username","password");</code>
36	PDO_OCI	Oracle データベース 12c, 11g, 10g, 9i 及び 8i にアクセス、SQL 及び PL/SQL 文をサポートします。	△	<code>\$db = new PDO("oci:dbname=id","username","password");</code>
37	PDO_SQLSRV	MS SQL Server (SQL Server 2005 以降) および SQL Azure データベースにアクセスできる (Windows 版では PHP 5.3 以降、SQLSRV という別の MS SQL 接続用拡張モジュールが必要)	△	
38	PDO_PGSQL	PHP から PostgreSQL データベースへのアクセスを可能	△	
39	PDO_ODBC	ODBC ドライバあるいは IBM DB2 Call Level Interface (DB2 CLI) ライブラリを使用した データベースへのアクセスが可能。 Windows では、PDO_ODBC はデフォルトで PHP コアに組み込み。Windows ODBC ドライバマネージャに対してリンクしていて、システムの DSN に登録されているあらゆるデータベースに対して PHP から接続することができます。 Microsoft SQL Server データベース接続の際は、このドライバを推奨	○	<code>\$db = new PDO("odbc:MSSQLServer","username","password");</code>
40	PDO_IBM	DB2 Call Level Interface (DB2 CLI) を使用した IBM DB2 Universal Database、IBM Cloudscape および Apache Derby データベースへのアクセスが可能	△	

【MySQLのオプション指定:定数】

PDO::MYSQL_ATTR_LOCAL_INFILE	LOAD LOCAL INFILE を有効にします。	インスタンス作成のdriver_options配列内のみ
PDO::MYSQL_ATTR_INIT_COMMAND	MySQL サーバーへの接続時に実行するコマンドを指定します。再接続の際には自動的に再実行されます。	インスタンス作成のdriver_options配列内のみ array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8", ...)
PDO::MYSQL_ATTR_READ_DEFAULT_FILE	my.cnf ではなく、指定した名前のファイルからオプションを読み込みます。 (mysqlndを使っている場合は利用不可)	
PDO::MYSQL_ATTR_READ_DEFAULT_GROUP	my.cnfあるいは別のファイル (MYSQL_READ_DEFAULT_FILE で指定したもの) の中の、指定した名前のグループからオプションを読み込みます。(mysqlndを使っている場合は利用不可)	
PDO::MYSQL_ATTR_MAX_BUFFER_SIZE	バッファの最大サイズ。デフォルトは 1 MiB です。(mysqlndを指定してコンパイルした場合は利用不可)	
PDO::MYSQL_ATTR_DIRECT_QUERY	プリペアドステートメントではなく、直接クエリを実行します。	
PDO::MYSQL_ATTR_FOUND_ROWS	変更された行数ではなく、見つかった (マッチした) 行数を返します。	
PDO::MYSQL_ATTR_IGNORE_SPACE	関数名の後に続く空白を許可します。すべての関数名を予約語にします。	

他

【fetch又はfetchallの「fetch_style」のオプション定数】

PDO::FETCH_ASSOC	連想配列形式の結果セットを返す。		
PDO::FETCH_NUM	添字配列の結果セットを返す。		
PDO::FETCH_BOTH	添字配列と連想配列形式の両方の結果セットを返す。(デフォルト)		
PDO::FETCH_OBJ	結果セットに返された際のカラム名と同名のプロパティを有する匿名のオブジェクトを返す。		
PDO::FETCH_BOUND	TRUEを返し、結果セットのカラムの値を PDOStatement::bindColumn()メソッドでバインドされたPHP変数に代入。		
PDO::FETCH_CLASS	結果セットのカラムがクラス内の名前付けされたプロパティにマッピングされている、要求されたクラスの新規インスタンスを返す。fetch_styleに PDO::FETCH_CLASSSTYPEが含まれている場合 (例: PDO::FETCH_CLASS PDO::FETCH_CLASSSTYPE) は、最初のカラムの値から クラス名を決定。		
PDO::FETCH_INTO	結果セットのカラムがクラス内の名前付けされたプロパティにマッピングされている要求された既存インスタンスを更新。		
PDO::FETCH_LAZY	PDO::FETCH_BOTHとPDO::FETCH_OBJの組合せで、オブジェクト変数名を作成。		
PDO::FETCH_NAMED	PDO::FETCH_ASSOCと同じ形式の配列を返します。ただし、同じ名前のカラムが複数あった場合は、そのキーが指す値は、同じ名前のカラムのすべての値を含む配列になる。		

【データベースハンドルの属性を設定の定数】

… ■PDO::setAttribute用

\$db->setAttribute (\$attribute , \$value);

PDO::ATTR_CASE	強制的にカラム名を指定したケースにする		
PDO::CASE_LOWER	強制的にカラム名を小文字にする		
PDO::CASE_NATURAL	データベースドライバによって返されるカラム名をそのままにする		
PDO::CASE_UPPER	強制的にカラム名を大文字にする		
PDO::ATTR_ERRMODE	エラーレポート		
PDO::ERRMODE_SILENT	エラーコードのみ設定する		
PDO::ERRMODE_WARNING	E_WARNING を発生させる		
PDO::ERRMODE_EXCEPTION	例外を投げる		
PDO::ATTR_ORACLE_NULLS	NULLと空文字列の変換(Oracle だけでなく、全てのドライバで利用可能)	○	\$db->setAttribute(PDO::ATTR_ORACLE_NULLS, PDO::NULL_EMPTY_STRING); ※PDO::NULL_EMPTY_STRING は、「NULL は空文字に変換される」
PDO::NULL_NATURAL	変換しない		
PDO::NULL_EMPTY_STRING	空文字は NULL に変換される		
PDO::NULL_TO_STRING	NULL は空文字に変換される		
PDO::ATTR_STRINGIFY_FETCHES	フェッチする際、数値を文字列に変換する。boolを必要とする		
PDO::ATTR_STATEMENT_CLASS	PDOStatement に由来するユーザーが提供するステートメントクラスを設定する。永続的な PDO インスタンスは使用できず、array(string classname, array(mixed constructor_args))を必要とする。		
PDO::ATTR_TIMEOUT	タイムアウト秒数を指定する。		
PDO::ATTR_AUTOCOMMIT	それぞれの単一文で自動コミットするかどうか。(MySQL,OCI,Firebirdで利用可能)	○	\$db->setAttribute(PDO::ATTR_AUTOCOMMIT, false); //自動コミットをOFF
PDO::ATTR_EMULATE_PREPARES	プリペアドステートメントのエミュレーションを有効または無効にする。		
PDO::ATTR_EMULATE_PREPARES	常に プリペアドステートメントをエミュレートする (TRUEの場合) か、ネイティブのプリペアドステートメントを使おうとする (FALSEの場合) かを設定できる。		
PDO::MYSQL_ATTR_USE_BUFFERED_QUERY	バッファされたクエリを使用する。(MySQLで利用可能)		
PDO::ATTR_DEFAULT_FETCH_MODE	デフォルトのフェッチモードを設定する。		

… ■PDO::getAttribute用

\$db->getAttribute (\$attribute);

PDO::ATTR_AUTOCOMMIT	トランザクションを開始できるように オートコミットを無効 (FALSE) 又は有効	○	TRUE: 1 / FALSE: 0
PDO::ATTR_CASE	強制的にカラム名を指定したケースの状態		ON: 1 / OFF: 0
PDO::ATTR_CLIENT_VERSION	DBクライアントのバージョン		例: → "mysqlnd 5.0.10"
PDO::ATTR_CONNECTION_STATUS	DB接続情報を取得	△	\$db->getAttribute(constant("PDO::ATTR_CONNECTION_STATUS")) 例: → "localhost via TCP/IP"
PDO::ATTR_DRIVER_NAME	利用DBドライバ		例: → "mysql"
PDO::ATTR_ERRMODE	エラーレポート		ON: 1 / OFF: 0
PDO::ATTR_ORACLE_NULLS	空文字をSQLのNULL 値に変換 (Oracle だけでなく、全てのドライバで利用可能)	○	ON: 1 / OFF: 0
PDO::ATTR_PERSISTENT	スクリプトが終了してもデータベースへの接続を維持し、次回に再利用		ON: 1 / OFF: 0
PDO::ATTR_PREFETCH	独自アプリケーションにおけるメモリ使用量に対する速度のバランスを 調整するためのプリフェッチサイズ		【MySQLドライバでは、未搭載機能Fatal ERROR】

PDO::ATTR_SERVER_INFO	データベースサーバーのThread数、オープンテーブル数、query平均速度等	△	例: → " Uptime: 39392 Threads: 2 Questions: 7 Slow queries: 0 Opens: 34 Flush tables: 1 Open tables: 0 Queries per second avg: 0.000 "
PDO::ATTR_SERVER_VERSION	データベースのバージョン		例: → "5.5.35"
PDO::ATTR_TIMEOUT	タイムアウト秒数		【MySQLドライバーでは、未搭載機能Fatal ERROR】