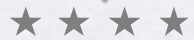


---

# MYSQL

DBの基本 #2



# MySQLのデータ型

<https://dev.mysql.com/doc/refman/5.6/ja/data-types.html>

型	特徴
整数型	TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
浮動小数点数型	FLOAT, DOUBLE
日付型・時刻型	DATE, DATETIME, TIMESTAMP, TIME, YEAR
CHAR型・VARCHAR型	CHAR, VARCHAR
BINARY型・VARBINARY型	BINARY, VARBINARY
BLOB型・TEXT型	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT...
ENUM型	文字列定数をリストする
SET型	文字列定数をリストする

# テーブル構造の変更

<https://dev.mysql.com/doc/refman/5.6/ja/alter-table.html>

コマンド	働き
ALTER TABLE ... MODIFY ...	カラムの定義変更
ALTER TABLE ... ADD ...	カラム（列）の追加
ALTER TABLE ... CHANGE ...	カラムの名前と定義、変更
ALTER TABLE ... DROP ...	カラムを削除



# PHPからMySQLを利用

```
<?php
    phpinfo();
?>
```

## mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.12-dev - 20150407 - \$Id: b5c5906d452ec590732a93b051f3827e02749b83 \$
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	

## mysqlnd

Dir	mysqlnd	enabled
mysqli.allow_local_infile	Version	mysqlnd 5.0.12-dev - 20150407 - \$Id: b5c5906d452ec590732a93b051f3827e02749b83 \$
mysqli.allow_persistent	Compression	supported
mysqli.default_host	core SSL	supported
mysqli.default_port	extended SSL	supported
mysqli.default_pw	Command buffer size	4096
	Read buffer size	32768

## PDO

PDO support	enabled
PDO drivers	mysql

## pdo\_mysql

PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.12-dev - 20150407 - \$Id: b5c5906d452ec590732a93b051f3827e02749b83 \$

Directive	Local Value	Master Value
pdo_mysql.default_socket	/var/run/mysqld/mysqld.sock	/var/run/mysqld/mysqld.sock

# PHPからMySQLを利用(1)

PHP5.5以降非推奨  
以後利用しない

```
<?php
$username="xxxxx";
$password="yyyyy";
$dbname  ="zzzzz";

$sp=mysql_connect("localhost",$username,$password)
    or die("DB接続不可");
mysql_select_db($dbname,$sp);          //useと同等

$sql="select * from tb1";
$result=mysql_query($sql);

while($kekka=mysql_fetch_array($result)){    //実際の取だし
    var_dump($kekka);
    echo "<br>\n";
}
mysql_close($sp);
?>
```

# PHPからMySQLを利用(2)

## mysqliを利用

```
<?php
$username="xxxxx";
$password="yyyyy";
$dbname  ="zzzzz";

$mysqli = new mysqli('localhost', $username, $password);
if ($mysqli -> connect_errno) {
    print('<p>データベースへの接続に失敗しました。</p>' . $mysqli -> connect_error);
    exit();
}
$mysqli -> select_db($dbname);
$mysqli -> set_charset("utf-8");

$sql = "SELECT * FROM tb1";
$result = $mysqli -> query($sql);

while ($kekka = $result -> fetch_array()) {
    var_dump($kekka);
    echo "<br>\n";
}
$result -> free();
$mysqli -> close();
?>
```



# PHPからMySQLを利用(3)

## PDOを利用

```
<?php
$username="xxxxx";
$password="yyyyy";
$dbname  ="zzzzz";

try{
    $pdo = new PDO("mysql:host=localhost;dbname=$dbname;charset=utf8",$username,$password);
    $sql = "SELECT * FROM tb1";
    $result = $pdo -> query($sql);

    // while($kekka=$result -> fetch(PDO::FETCH_ASSOC)){
    foreach ($result as $kekka){
        var_dump($kekka);
        echo "<br>\n";
    }
} catch(PDOException $e){
    print("Error:" . $e->getMessage());
    die();
}
$result = null;
$pdo = null;
?>
```

# PDOStatement

\$pdo->query() の戻り値はPDOStatement

<http://php.net/manual/ja/class.pdostatement.php>

配列/ポインタのどちらにも似た形式で利用できる。

## 配列風

```
foreach ($result as $kekka){  
    var_dump($kekka);  
    echo "<br>\n";  
}
```

## ポインタ風

```
while($kekka=$result -> fetch()){  
    var_dump($kekka);  
    echo "<br>\n";  
}
```



# PDOStatement

- [PDOStatement::bindColumn](#) – カラムを PHP 変数にバインドする
- [PDOStatement::bindParam](#) – 指定された変数名にパラメータをバインドする
- [PDOStatement::bindValue](#) – 値をパラメータにバインドする
- [PDOStatement::closeCursor](#) – カーソルを閉じてステートメントを再実行できるようにする
- [PDOStatement::columnCount](#) – 結果セット中のカラム数を返す
- [PDOStatement::debugDumpParams](#) – SQL プリペアドコマンドを出力する
- [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
- [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
- [PDOStatement::execute](#) – プリペアドステートメントを実行する
- [PDOStatement::fetch](#) – 結果セットから次の行を取得する
- [PDOStatement::fetchAll](#) – 全ての結果行を含む配列を返す
- [PDOStatement::fetchColumn](#) – 結果セットの次行から単一カラムを返す
- [PDOStatement::fetchObject](#) – 次の行を取得し、それをオブジェクトとして返す
- [PDOStatement::getAttribute](#) – 文の属性を取得する
- [PDOStatement::getColumnMeta](#) – 結果セットのカラムに対するメタデータを返す
- [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
- [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
- [PDOStatement::setAttribute](#) – 文の属性を設定する
- [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

# 課題

- 自分のdbにあるテーブルtb1に含まれるデータを表形式にして、全て出力しなさい。
- プログラム: public\_html/sql02/select.php
- テーブルの出力部分は関数にしてみよう。

※ query()の結果は、数字配列・連想配列の両方が返ってくる  
fetch(オプション)により、片方だけ取り出すこともできる  
PDO::FETCH\_NUM or PDO::FETCH\_ASSOC