# What is PyMySQL

PyMySQL is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2.0 and contains a pure-Python MySQL client library. The goal of PyMySQL is to be a drop-in replacement for MySQLdb.

Now, let's start step by step procedure to connect Python with MySQL Database.

# Steps to connect Python with MySQL

Our first step is to install the PyMySQL module in our system.

## Step 1: Install pymysql module

Before we install, let's check the requirement of the pymysql package.

The last stable release is available on PyPI and can be installed with pip.

Type the following command in your terminal.

```
python3 -m pip install PyMySQL
```

## Step 2: Create a Database and Table in MySQL

When we connect the Python to MySQL Database, we have to provide the following things in the argument.

1. Host Name/servername
2. Username
3. Password
4. Database Name

## Step 3: Import pymysql package

Create a Python file called **app.py** and import pymysql package.

```
# app.py
```

```
import pymysql
```

## Step 4: Provide your MySQL Credentials

Now, write the pymysql connect() function and pass the credentials, hostname, and **database name as arguments.**

```
# app.py

db = pymysql.connect("your hostname","your username","your password","your database" )
```

My hostname is your **localhost**.

My username is **root.**

My password is **blankt.**

My database name is **college**

## Step 5: Prepare a cursor object using a cursor() method.

Next, the **db** object is used to create a **cursor** object, which in turn is used to execute **SQL queries.**

```
# app.py

cur = db.cursor()

cur.execute('SELECT * FROM tablename')
```

I have created a table named **tablename.**

We have executed the SELECT statement to fetch all the rows from the table.

## Step 6: Read Operation

READ Operation on any database means to fetch some useful information from the database.

Once the database connection is established, you are ready to make a query into this database. You can use either the **fetchone()** method to fetch a single record or **fetchall()** method to fetch multiple values from a database table.

1. **fetchone()** − It fetches the next row of a query result set. A result set is an object that is returned when a cursor object is used to query a table.
2. **fetchall()** − It fetches all the rows in a result set. If some rows have already been extracted from the result set, then it retrieves the remaining rows from the result set.
3. **rowcount** − This is a read-only attribute and returns the number of rows that were affected by an execute() method.

### Let's use cur.fetchall() function to get all the rows.

```
# app.py

for row in cur.fetchall():
    print(row)
```

It will print all the rows in the Python console. Now the last step is to close the connection.

## Step 7: Close MySQL connection

Finally, before coming out, it ensures that the database connection is closed and resources are released.

```
# app.py

db.close()
```

## Final Code to connect Python to MySQL

```
# app.py

import pymysql

db = pymysql.connect("localhost","root","","databasename" )

# prepare a cursor object using cursor() method
cur = db.cursor()
```

```
try:
  cur.execute('SELECT * FROM student')

  for row in cur.fetchall():
      print(row)
except:
   print ("Error: unable to fetch data")

db.close()
```

## Output