

Please visit website: <http://cxyroad.com>

ctor`

创建数据库连接

=====

...

```
import mysql.connector
```

```
mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin" # 数据库密码
    # database=""
)
print(mydb)
```

...

![image.png](<https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/eeb5fd74c930495c8fffc7a5a025bc92~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=1540&s=214068&e=png&b=212225>)

创建数据库

=====

创建数据库使用`CREATE DATABASE`语句，我们来创建一个名为`python-connect`的数据库，同时也可以使用`SHOW DATABASES`语句来查看数据库是否存在：

...

```
import mysql.connector
```

```
mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin" # 数据库密码
```

```
)

print(mydb)
mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE python_connect")
mycursor.execute("SHOW DATABASES")
for database in mycursor:
    print(database)

...

```

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/dadc4026c21d46bd92d3d539c1da9d1e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=1540&s=281105&e=png&b=212225)

可以看见成功创建。

创建数据表

=====

创建数据表使用 `**"CREATE TABLE"**` 语句，创建数据表前，需要确保数据库已存在，我们在我们创建的`python_connect`中进行创建一张User表，并且通过`**SHOW TABLES**`来查看：

```
...

import mysql.connector

mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin", # 数据库密码
    database="python_connect"
)

print(mydb)
mycursor = mydb.cursor()
mycursor.execute("CREATE TABLE user (Id VARCHAR(255), name VARCHAR(255))")
mycursor.execute("SHOW TABLES")
for x in mycursor:
    print(x)

```

...

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/ab9bf7e08f3243a0b256b3a228e72fdf~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=1540&s=240239&e=png&b=212225)

插入数据

=====

我们像这种表中插入一条数据。

...

```
import mysql.connector
```

```
mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin", # 数据库密码
    database="mydb")
mydb.commit()
```

```
mycursor.execute("SELECT * FROM user")
```

```
myresult = mycursor.fetchall() # fetchall() 获取所有记录
```

```
for x in myresult:
    print(x)
```

...

更新表数据

=====

数据表更新使用 `**"UPDATE"**` 语句

...

```
import mysql.connector
```

```
mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin", # 数据库密码
    database="mydb")
mydb.commit()
```

```

    database="python_connect"
)

print(mydb)
mycursor = mydb.cursor()

mycursor.execute("UPDATE user SET name = %s WHERE Id = %s",
('king', '4'))
mydb.commit()

# 第二个参数必须为元组
mycursor.execute("SELECT * FROM user where id = %s ", ('4',))
myresult = mycursor.fetchone()

```

```
print(myresult)
```

```
...
```

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/766418bfe099492da2bc03b8962325bf~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=1540&s=258091&e=png&b=212225)

删除表

```
====
```

除表使用 `**"DROP TABLE" **` 语句，`IF EXISTS` 关键字是用于判断表是否存在，只有在存在的情况才删除：

```
...
```

```
import mysql.connector
```

```

mydb = mysql.connector.connect(
    host="localhost", # 数据库主机地址
    user="root", # 数据库用户名
    passwd="kylin", # 数据库密码
    database="python_connect"
)

```

```

print(mydb)
sql = "DROP TABLE IF EXISTS user" # 删除数据表 user

mycursor.execute(sql)

```

...

总结

==

使用python操作数据库还是十分简单的。

1. ``mysql.connector.connect()``: 获取连接对象
2. ``mydb.cursor()``: 获取数据库的操作对象
3. ``mycursor.execute(sql, parameters)``: 来运行sql, 第二个参数为元组
4. 为了防止数据库查询发生 SQL 注入的攻击, 我们可以使用 ``%s`` 占位符
5. ``mycursor.executemany()``: 执行多条sql操作
6. ``mydb.commit()``: 提交事务
7. ``mycursor.fetchall()``: 执行为查询SQL后, 获取全部的记录
8. ``mycursor.fetchone()``: 执行为查询SQL后, 获取一条的记录

基本上就涉及到了这些API, 相对于Java还是少了非常多的代码, 十分方便。

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/605242c72d43464286103ee93c7ce279~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=350&h=350&s=59237&e=png&b=fcfcfc)

原文链接: <https://juejin.cn/post/7352075810935095305>