

# Verbindungsvarianten Python - MySQL

Es gibt **zwei Libraries**, mit welchen wir eine Verbindung mit einer MySQL-Datenbank realisieren können:

- mysql-connector-python
- pymysql

In den vorhergehenden Beispielen haben wir beide bereits verwendet. Nimm im Zweifelsfall das offizielle Paket, d.h. `mysql-connector-python`.

## mysql-connector-python

Dazu muss die Library **mysql-connector-python** geladen sein:

```
pip install mysql-connector-python oder conda install mysql-connector-python
```

### Verbindungsbeispiel mit mysql-connector-python

```
1  import mysql.connector
2
3  # Verbindung zur MySQL-Datenbank herstellen
4  conn = mysql.connector.connect(
5      host="localhost",
6      user="dein_benutzername",
7      password="dein_passwort",
8      database="deine_datenbank"
9  )
10
11 # Erstellen eines Cursors
12 cursor = conn.cursor()
13
14 # Eine einfache Abfrage ausführen
15 cursor.execute("SELECT DATABASE()")
16
17 # Ergebnis abrufen und ausdrucken
18 result = cursor.fetchone()
19 print(f"Verbunden mit der Datenbank: {result[0]}")
20
21 # Verbindung schließen
22 cursor.close()
23 conn.close()
```

### Eigenschaften dieser Variante:

- Offizielle Unterstützung: mysql-connector-python wird direkt von Oracle entwickelt und gepflegt, was bedeutet dass es regelmässig aktualisiert wird und Unterstützung für die neuesten MySQL-Versionen bietet.
- Da es ein offizieller Connector ist, ist es sehr gut kompatibel mit verschiedenen Versionen von MySQL.
- In manchen Szenarien kann es etwas langsamer sein als PyMySQL, insbesondere bei sehr grossen Datenmengen oder komplexen Abfragen.

## PyMySQL

Dazu muss die Library **pymysql** geladen sein:

```
pip install pymysql oder conda install pymysql `
```

#### Verbindungsbeispiel mit pymysql

```
1  import pymysql
2
3  # Verbindung zur MySQL-Datenbank herstellen
4  conn = pymysql.connect(
5      host="localhost",
6      user="dein_benutzername",
7      password="dein_passwort",
8      database="deine_datenbank"
9  )
10
11 # Erstellen eines Cursors
12 cursor = conn.cursor()
13
14 # Eine einfache Abfrage ausführen
15 cursor.execute("SELECT DATABASE()")
16
17 # Ergebnis abrufen und ausdrucken
18 result = cursor.fetchone()
19 print(f"Verbunden mit der Datenbank: {result[0]}")
20
21 # Verbindung schließen
22 cursor.close()
23 conn.close()
```

#### Eigenschaften dieser Variante:

- Da PyMySQL komplett in Python geschrieben ist, kann es einfacher zu installieren und auf verschiedenen Plattformen zu verwenden sein.
- In bestimmten Anwendungsfällen kann PyMySQL schneller sein als mysql-connector-python.
- PyMySQL ist ein Open-Source-Projekt und wird nicht offiziell von Oracle unterstützt, was bedeutet, dass es möglicherweise nicht so regelmässig aktualisiert wird.
- Könnte manchmal Probleme mit der Kompatibilität haben, insbesondere mit den neuesten MySQL-Versionen oder spezifischen MySQL-Funktionen.