# PHP und MySQL (oder MariaDB)
## Verbindung zur Datenbank herstellen

# Vorgehen

1. Zugangsdaten definieren
2. Verbindung herstellen
3. Mögliche Fehler abfangen

# Vorbemerkung

Den PHP-Code schreiben wir erst mal in die Seite, auf der das Formular eingebunden ist, z.B.:

```
<!DOCTYPE html>
<html lang="de">
…
<body>
    <h1>Schülerwettbüro</h1>
    <?php
        $username = 'root';
        …
    ?>
    <form action="index.php" method="post">
        …
        …
</body>
</html>
```

# Schritt 1:
# Zugangsdaten definieren

*benötigt:*

| | |
|---|---|
| *Hostname (Servername)* | *(meistens: localhost)* |
| *Datenbankname* | *(Beispiel: freiburgerVerkehrsDB)* |
| *Benutzername* | *(Beispiel: root)* |
| *Passwort* | *(Beispiel: root)* |

***Am einfachsten speichern wir alle Zugangsdaten in Variablen; dann haben wir das an einer Stelle und können es bequem ändern/kontrollieren.***

```
$servername = 'localhost';
$dbname   = 'meineDatenbank';
$username    = 'root';
$passwort    = ''; // bei WAMP/xampp leer, bei MAMP (macOS)
    'root'
```

# Schritt 2:
# Verbindung zum Server herstellen

*Erst mal fassen wir die bekannten Serverdaten zusammen. Wir wissen:*

1. *Es ist eine MySQL-Datenbank.*
2. *Wir kennen den Servernamen.*
3. *Wir kennen den Namen der Datenbank.*

```
$servername     = 'localhost';
$dbname         = 'meineDatenbank';
$username       = 'root';
$passwort       = '';
```

```
$serverdaten = "mysql:host=$servername;dbname=$dbname";
```

*Weitere Parameter sind möglich, bspw.*

```
$serverdaten =
    "mysql:host=$servername;dbname=$dbname;charset=utf8";
```

*(um die Zeichenkodierung festzulegen, wenn es bspw. Probleme mit Umlauten gibt)*

# Schritt 2:
# Verbindung zum Server herstellen

*Um die Verbindung zum Server zu öffnen, legen wir einfach ein neues Objekt der Klasse PDO ("PHP Data Objects") an, das (in unserem Fall) drei Parameter bekommt:*

*- Server- und DB-Daten*

*- Username*

*- Passwort*

*Das Objekt bezeichnen wir bspw. als $verbindung; oft wird auch $pdo benutzt.*

```
$servername     = 'localhost';
$dbname         = 'meineDatenbank';
$username       = 'root';
$passwort       = '';
$serverdaten    = "mysql:host=$servername;dbname=$dbname";
```

```
$verbindung =
    new PDO($serverdaten, $username, $passwort);
```

# Schritt 3:
# Mögliche Fehler abfangen

Problem: Enthält

```
$verbindung =
    new PDO($serverdaten, $username, $passwort);
```

einen Fehler (z.B. falsches Passwort), bricht das Skript ab und wir erfahren nicht, welcher Fehler es war (falsches Passwort? Falscher Benutzername? Keine Verbindung zur Datenbank? Etc.)

Deshalb müssen wir den Fehler ab"fangen" (catch).

```
try                                  { // PHP-Code }
catch(Exception $einFehler)          { // PHP-Code }
```

# Schritt 3:
# Mögliche Fehler abfangen

Wir umgeben den Verbindungsaufbau mit einem try-catch-Block: Probiere xy, wenn es einen Fehler gibt, fange den ab und mache was anderes.

```
try
{
    $verbindung = new PDO($serverdaten, $username, $passwort);
    // wenn hier ein Fehler "geworfen" wird, spuck uns keine Fehlermeldungen um die Ohren, sondern führe den catch-Block aus
}
catch(Exception $fehler)
    // $fehler ist ein Objekt der Exception-Klasse. Exceptions enthalten genauere Informationen über den Fehler.
    // eigentlich präziser: catch(PDOException $fehler)
{
    print $fehler->getMessage();
    // Ausgabe der Fehlermeldung
}
```