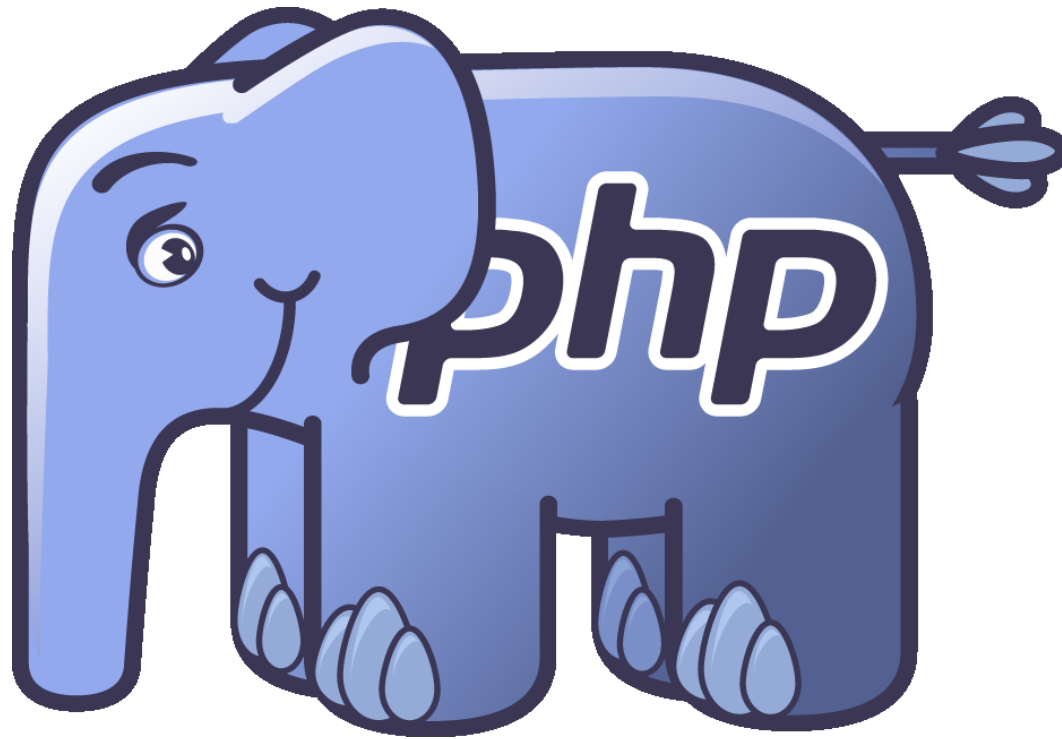


# PHP PDO



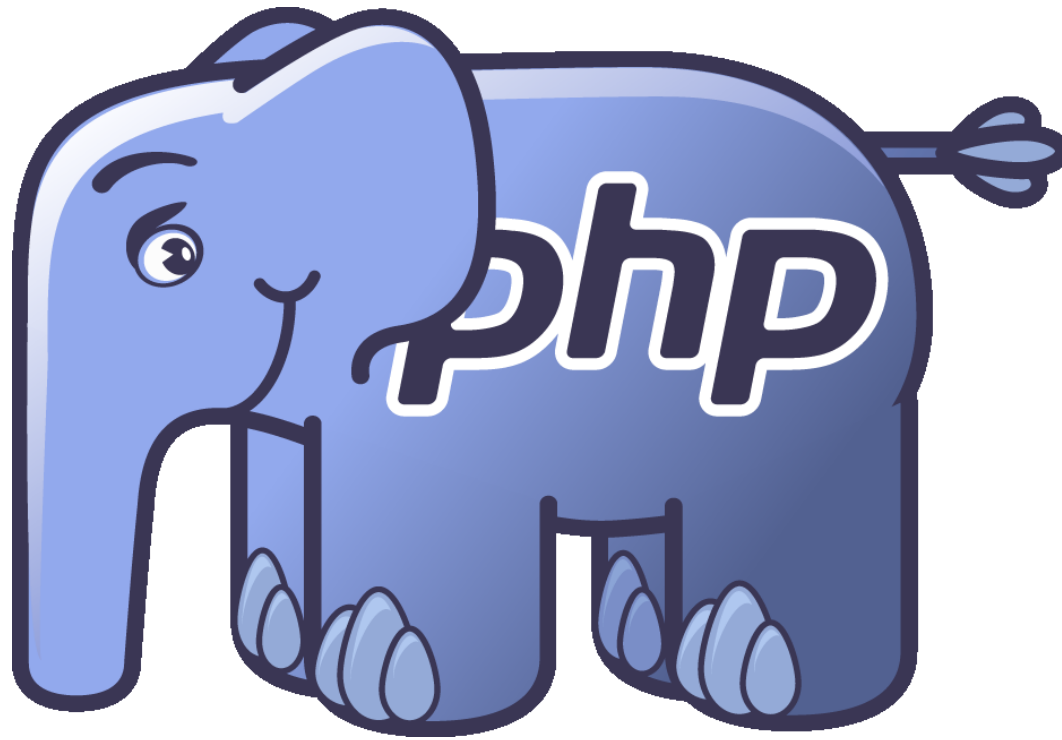
# INTRODUCTION

# PDO

- PDO : **P**HP **D**ata **O**bjects
- Extension PHP fournissant une interface pour accéder à une base de données
- Fournit une interface d'**abstraction pour l'accès aux données**
  - SQL spécifique au moteur
- Interface orientée objet

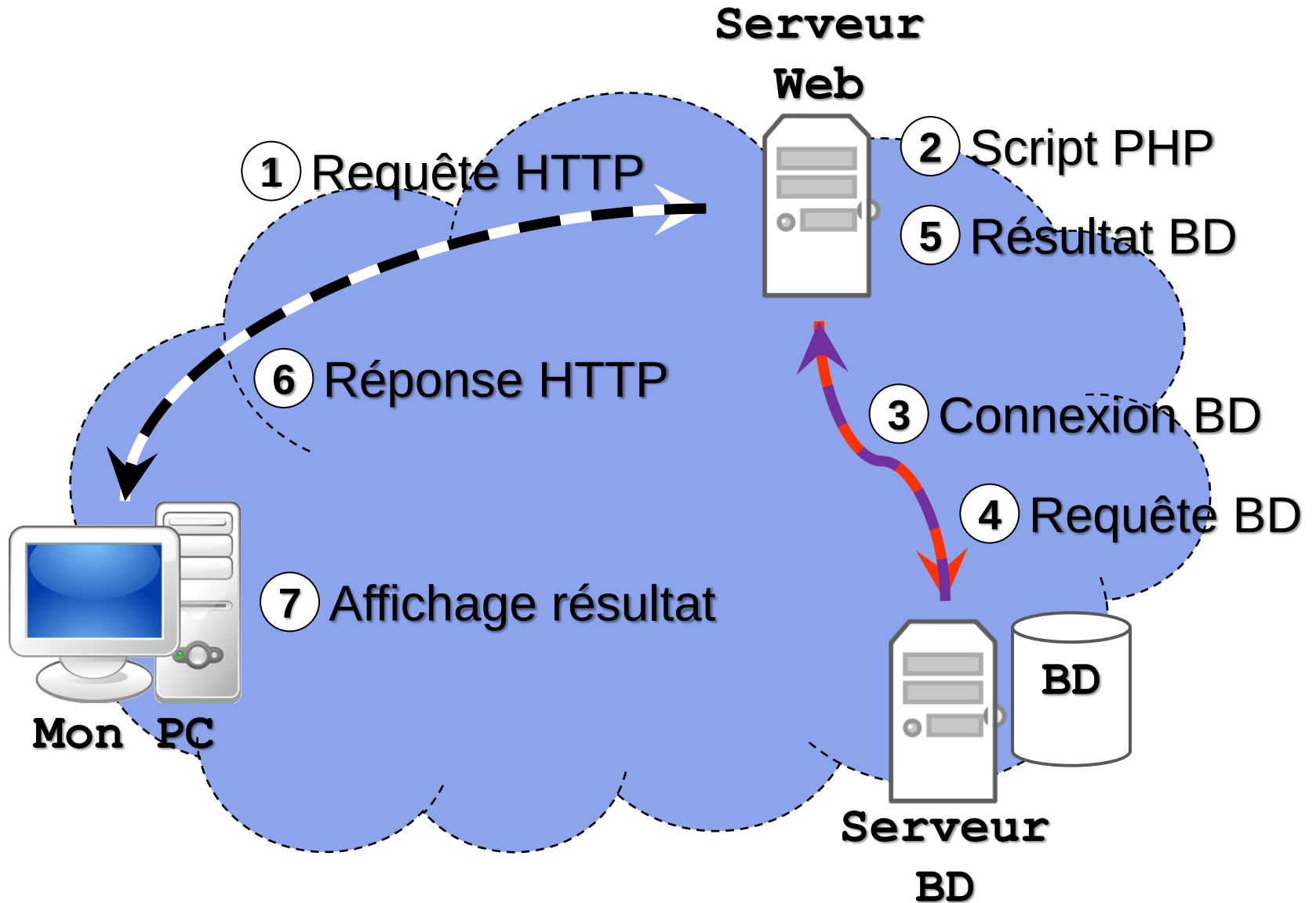
# Bases de données supportées

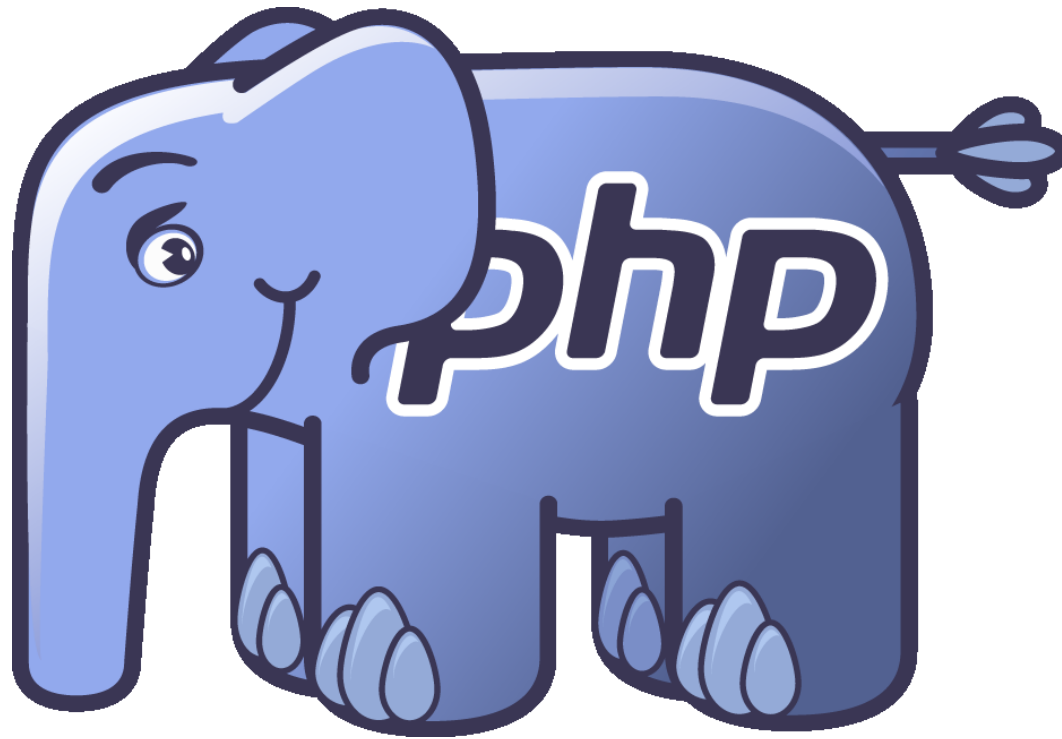
| Nom du pilote | Bases de données prises en charge         |
|---------------|-------------------------------------------|
| PDO_CUBRID    | Cubrid                                    |
| PDO_DBLIB     | FreeTDS / Microsoft SQL Server / Sybase   |
| PDO_FIREBIRD  | Firebird                                  |
| PDO_IBM       | IBM DB2                                   |
| PDO_INFORMIX  | IBM Informix Dynamic Server               |
| PDO_MYSQL     | MySQL 3.x/4.x/5.x                         |
| PDO_OCI       | <i>Oracle Call Interface</i>              |
| PDO_ODBC      | ODBC v3 (IBM DB2, unixODBC et win32 ODBC) |
| PDO_PGSQL     | PostgreSQL                                |
| PDO_SQLITE    | SQLite 3 et SQLite 2                      |
| PDO_SQLSRV    | Microsoft SQL Server / SQL Azure          |
| PDO_4D        | 4D                                        |



**ÉCHANGES DE DONNÉES**

# Échanges de données

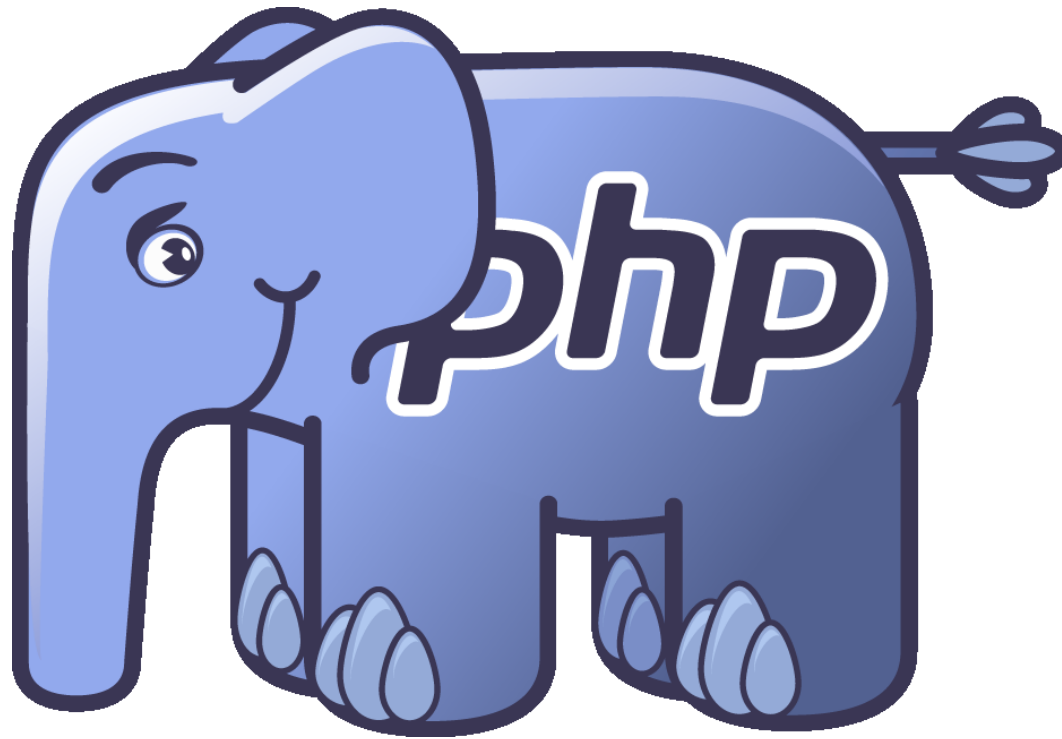




**COMMENT SE CONNECTER ?**

# Connexions et gestionnaire de connexion

- Instanciation d'un objet **PDO**
- `$dbh=new PDO(DSN [, user [, pass [, options]]]);`
- *DSN* : **D**ata **S**ource **N**ame
  - `nom_du_driver:syntaxe_spécifique_au_driver`
  - Ex : `mysql:host=localhost;dbname=ma_base`
- *user* : nom d'utilisateur, *pass* : mot de passe
- *options* : tableau associatif
  - spécifiques au driver
- Fin de connexion : `$dbh=null ;` ou `unset($dbh) ;`



# **GESTION DES ERREURS**

# Gestion des erreurs de connexion

- Connexion par construction d'un objet
- Gestion envisageable des erreurs
  - Aucune
  - Fin brutale (exit, die)
  - Exception
- En cas d'erreur de connexion
  - Objet **PDOException** lancé
  - PDOException **hérite de Exception**

# Gestion des erreurs de connexion

```
<?php
try {
    $dbh = new PDO('mysql:host=h;dbname=db',
                    $user, $pass) ;

    ...
    $dbh = null ;
}
catch (PDOException $e) {
    echo "<p>Erreur: " . $e->getMessage() ;
    die() ;
}
?>
```

# Gestion des erreurs (hormis connexion)

- **PDO::ERRMODE\_SILENT** (*par défaut*)
  - Mode silencieux, mise en place d'un code d'erreur
  - PDO : `errorCode()` / `errorInfo()`
  - PDOStatement : `errorCode()` / `errorInfo()`
- **PDO::ERRMODE\_WARNING**
  - Mise en place du code d'erreur
  - Émission d'une erreur de type `E_WARNING`
- **PDO::ERRMODE\_EXCEPTION**
  - Mise en place du code d'erreur
  - Objet `PDOException` lancé

# Gestion des erreurs (hormis connexion)

```
<?php
try {
    $dbh = new PDO('mysql:host=h;dbname=db',
                    $user, $pass) ;
    $dbh->setAttribute(PDO::ATTR_ERRMODE,
                       PDO::ERRMODE_EXCEPTION) ;
    ...
    $dbh = null ;
}
catch (PDOException $e) {
    echo "<p>Erreur: " . $e->getMessage() ;
    die() ;
}
```

# Gestion des erreurs : code d'erreur

```
<?php
    $pdo = new PDO("mysql:host=localhost") ;
    $pdostat = $pdo->query(« hiiii" ) ;
    if ($pdo->errorCode()) {
        echo "ERREUR !!\n" ;
        echo "<pre>\n" ;
        var_dump($pdo->errorInfo()) ;
        echo "</pre>\n" ;
    }
```

ERREUR !!

```
array(3) {
    [0]=> string(5) "42000"
    [1]=> int(1064)
    [2]=> string(47) "Erreur de syntaxe près de 'hiiii' à la ligne 1"
}
```

Code SQLSTATE

Code erreur spécifique  
du driver

Chaîne erreur spécifique  
au driver

# Gestion des erreurs : exceptions

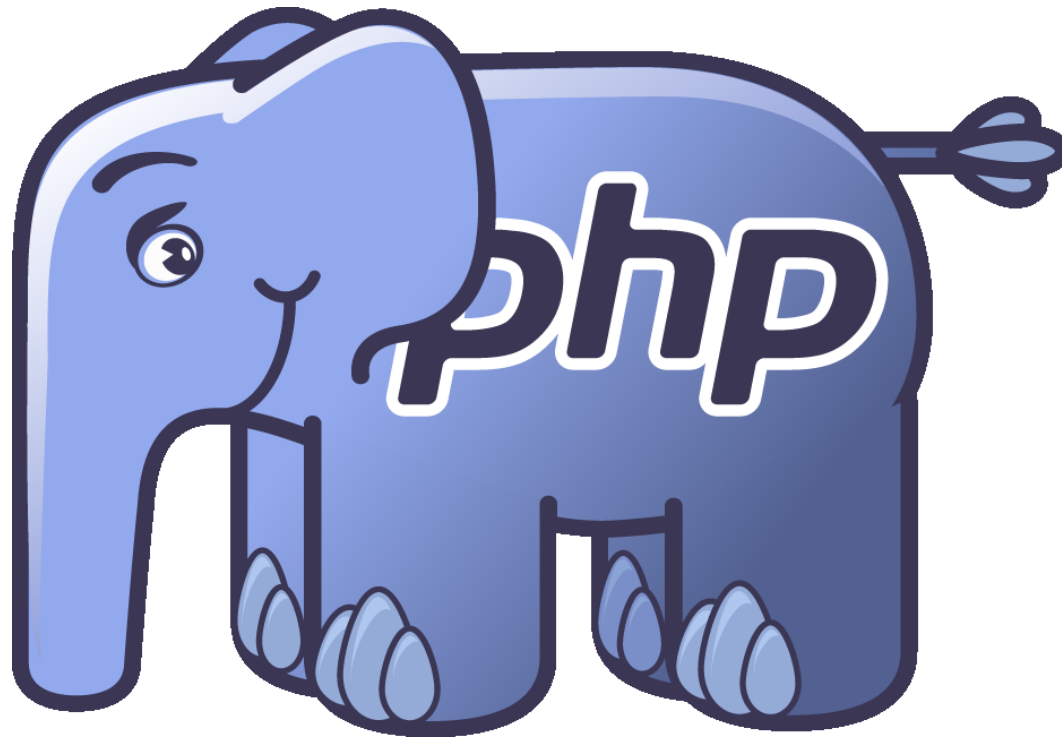
```
<?php
try {
    $pdo = new PDO("mysql:host=localhost") ;
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
                      PDO::ERRMODE_EXCEPTION) ;
    $pdostat = $pdo->query("COUCOU") ;
}
catch (Exception $e) {
    echo "<p>ERREUR : ".$e->getMessage() ;
}
```

Code SQLSTATE

Chaîne erreur spécifique au driver

Code erreur spécifique du driver

ERREUR : SQLSTATE[42000]: Syntax error or access violation: 1064  
Erreur de syntaxe près de 'COUCOU' à la ligne 1



**EFFECTUER UNE REQUÊTE**

# Exécution d'une requête

- PDOStatement **PDO::query** ( *string statement* )

Résultat de requête

Requête

```
<?php
try {
    $pdo = new PDO("mysql:host=localhost") ;
    $pdostat = $pdo->query("SELECT * FROM clients") ;
}
catch (Exception $e) {
    echo "<p>ERREUR : ".$e->getMessage() ;
}
```

**Une requête doit toujours être préparée !  
query est donc à bannir**

# Exploitation des résultats d'une requête

- Récupération des données ligne à ligne
- Une ligne peut être :
  - un tableau indexé
  - un tableau associatif
  - un tableau mixte (par défaut)
  - un objet anonyme
  - un objet d'une classe définie par l'utilisateur
- Récupération des données d'une colonne

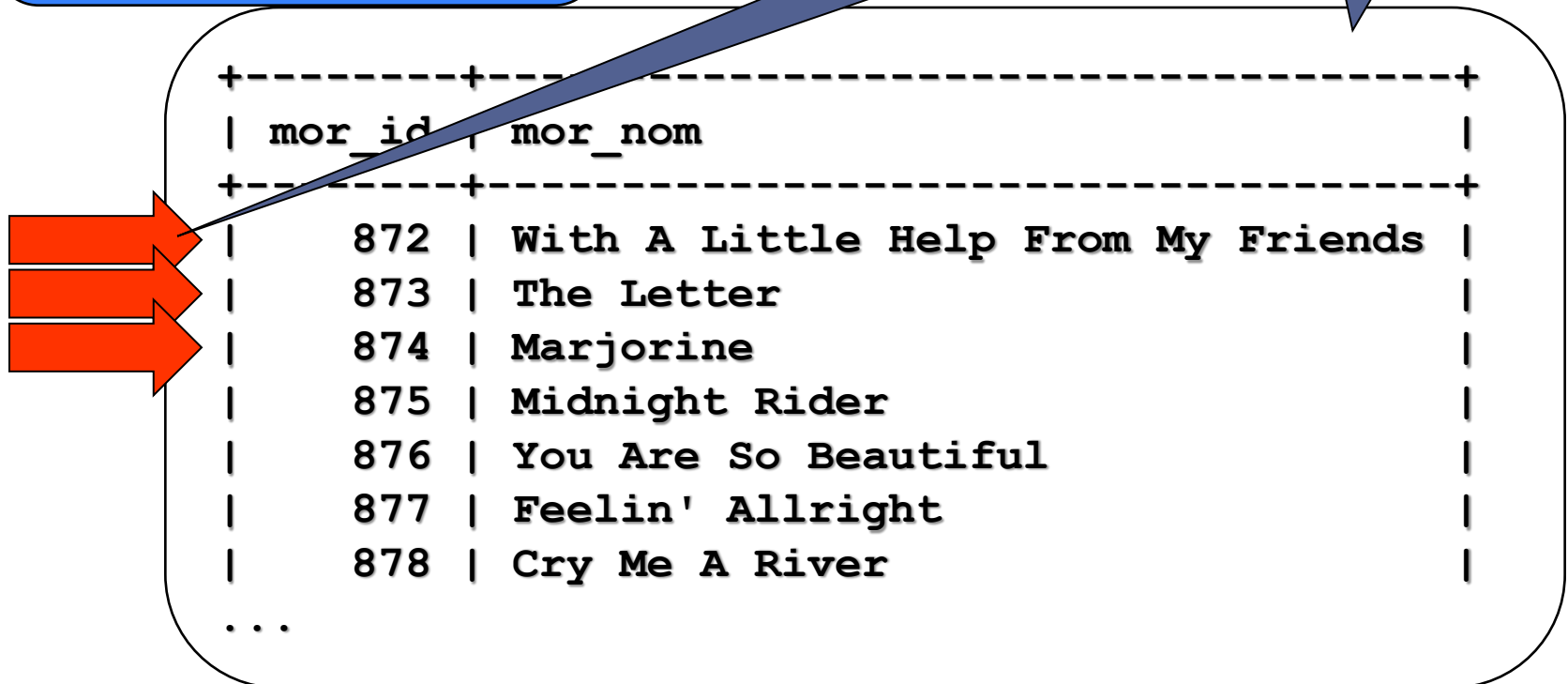
# Parcours du résultat d'une requête

- Parcourir le résultat de la requête

```
SELECT *  
FROM morceau  
ORDER BY mor_id
```

Résultat de requête

Curseur interne



| mor_id | mor_nom                            |
|--------|------------------------------------|
| 872    | With A Little Help From My Friends |
| 873    | The Letter                         |
| 874    | Marjorine                          |
| 875    | Midnight Rider                     |
| 876    | You Are So Beautiful               |
| 877    | Feelin' Allright                   |
| 878    | Cry Me A River                     |
| ...    |                                    |

# Exploitation des résultats d'une requête (1)

```
try {  
    $pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
    $pdo->setAttribute(PDO::ATTR_ERRMODE,  
                       PDO::ERRMODE_EXCEPTION) ;  
    $pdostat = $pdo->query("SELECT name FROM user") ;  
    $pdostat->setFetchMode(PDO::FETCH_ASSOC) ;  
    foreach ($pdostat as $ligne) {  
        echo "<p>" . $ligne['name'] . "\n" ;  
    }  
}  
  
catch (Exception $e) {  
    echo "<p>ERREUR : ".$e->getMessage() ;  
}
```

# Exploitation des résultats d'une requête (2)

```
try {  
    $pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
    $pdo->setAttribute(PDO::ATTR_ERRMODE,  
                        PDO::ERRMODE_EXCEPTION) ;  
    $pdostat = $pdo->query("SELECT name FROM user") ;  
    foreach ($pdostat->fetchAll(PDO::FETCH_ASSOC)  
            as $ligne) {  
        echo "<p>" . $ligne['name'] . "\n" ;  
    }  
}  
  
catch (Exception $e) {  
    echo "<p>ERREUR : ".$e->getMessage() ;  
}
```

# Exploitation des résultats d'une requête (3)

```
try {  
    $pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
    $pdo->setAttribute(PDO::ATTR_ERRMODE,  
                        PDO::ERRMODE_EXCEPTION) ;  
    $pdostat = $pdo->query("SELECT name FROM user") ;  
    while ($ligne  
           = $pdostat->fetch(PDO::FETCH_ASSOC)) {  
        echo "<p>" . $ligne['name'] . "\n" ;  
    }  
}  
  
catch (Exception $e) {  
    echo "<p>ERREUR : ".$e->getMessage() ;  
}
```

# Modes de récupération des données (1)

## ■ PDO::FETCH\_ASSOC

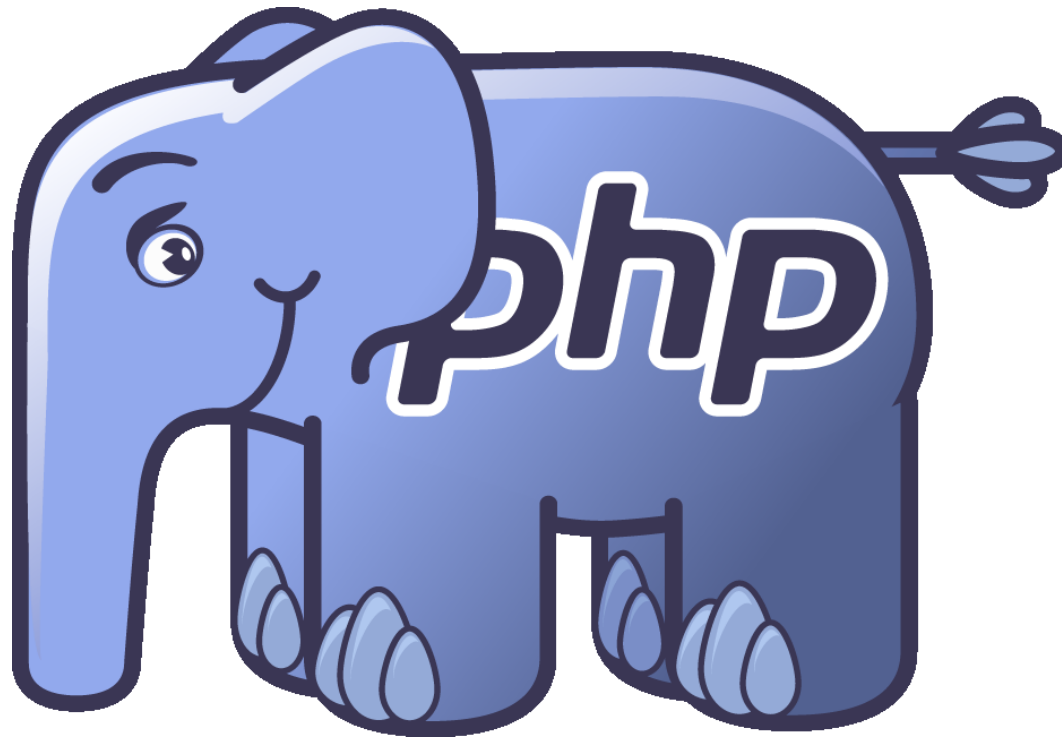
- retourner chaque ligne dans un tableau indexé par les noms des colonnes comme elles sont retournées dans le jeu de résultats correspondant. Si le jeu de résultats contient de multiples colonnes avec le même nom, PDO::FETCH\_ASSOC retourne une seule valeur par nom de colonne.

## ■ PDO::FETCH\_NUM

- retourner chaque ligne dans un tableau indexé par le numéro des colonnes comme elles sont retournées dans le jeu de résultats correspondant, en commençant à 0.

# Modes de récupération des données (2)

- **PDO::FETCH\_BOTH** (*par défaut*)
  - retourner chaque ligne dans un tableau indexé par les noms des colonnes ainsi que leurs numéros, comme elles sont retournées dans le jeu de résultats correspondant, en commençant à 0.
- **PDO::FETCH\_OBJ**
  - retourner chaque ligne dans un objet avec les noms de propriétés correspondant aux noms des colonnes comme elles sont retournées dans le jeu de résultats.



**REQUÊTES PRÉPARÉES**

# Préparation d'une requête

- PDOStatement `PDO::prepare(string statement [, array driver_options])`
  - *statement* : la requête à préparer. Peut contenir des paramètres anonymes (?) ou nommés (:nom)
  - *driver\_options* : tableau d'options du driver
  - retourne un objet `PDOStatement` qui effectuera l'association des paramètres et exécutera la requête

```
$pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
$pdostat = $pdo->prepare(  
    "SELECT * FROM user WHERE User= ?") ;
```

# Association des paramètres d'une requête

- `bool PDOStatement::bindValue(mixed parameter, mixed value [, int data_type])`
  - *parameter* : le paramètre (nom ou position [1...n])
  - *value* : sa valeur
  - *data\_type* : le type de la valeur
    - `PDO::PARAM_BOOL` booléen.
    - `PDO::PARAM_NULL` NULL SQL.
    - `PDO::PARAM_INT` INTEGER SQL.
    - `PDO::PARAM_STR` CHAR, VARCHAR ou autre chaîne.
- `bool PDOStatement::execute([array parameters])`

# Préparation puis exécution d'une requête (1)

```
$pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
$pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE EXCEPTION) ;
```

```
$pdostat = $pdo->prepare(  
    "SELECT * FROM user WHERE User=?") ;
```

```
$pdostat->bindValue(1, 'root') ;
```

```
$pdostat->execute() ;
```

```
// Utilisation du résultat
```

```
$pdostat->bindValue(1, 'cutrona') ;
```

```
$pdostat->execute() ;
```

```
// Utilisation du résultat
```

**Exécution de la requête**

# Préparation puis exécution d'une requête (2)

```
$pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
$pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE EXCEPTION) ;
```

```
$pdostat = $pdo->prepare(  
    "SELECT * FROM user WHERE User= :utilisateur") ;
```

```
$pdostat->bindValue(':utilisateur', 'root') ;
```

```
$pdostat->execute() ;
```

```
// Utilisation du résultat
```

```
$pdostat->bindValue(':utilisateur', 'cutrona') ;
```

```
$pdostat->execute() ;
```

```
// Utilisation du résultat
```

**Exécution de la requête**

# Préparation puis exécution d'une requête (3)

```
$pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
$pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE EXCEPTION) ;
```

```
$pdostat = $pdo->prepare(  
    "SELECT * FROM user WHERE User=?")
```

```
;
```

```
$pdostat->execute(array('root')) ;
```

```
// Utilisation du résultat
```

```
$pdostat->execute(array('cutrona')) ;
```

```
// Utilisation du résultat
```

**Exécution de la requête**

# Préparation puis exécution d'une requête (4)

```
$pdo=new PDO("mysql:host=localhost;dbname=mysql") ;  
$pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE EXCEPTION) ;
```

```
$pdostat = $pdo->prepare(  
    "SELECT * FROM user WHERE User= :utilisateur") ;
```

```
$pdostat->execute(  
    array(':utilisateur' => 'root')) ;
```

// Utilisation du résultat

```
$pdostat->execute(  
    array(':utilisateur' => 'cutrona')) ;
```

// Utilisation du résultat

**Exécution de la requête**

# Intérêt des requêtes préparées

- Amélioration des performances en cas d'exécutions répétées
- Émulation faite par PDO si le driver ne les supporte pas nativement
- Protection automatique des valeurs des paramètres pour **interdire les attaques par injection de code SQL**