

```

1. <?php
2. //configure data to connect to database
3. $server="localhost";
4. $admin="phpUser";
5. $password="phpUser_pwd.iaw";
6. $db="demo";
7. try{
8. //connect to database
9. $pdo=new PDO("mysql:host=$server;dbname=$db", $admin,
    $password, array(PDO::ATTR_ERRMODE =>
    PDO::ERRMODE_EXCEPTION));
10. }
11. //Test Error
12. catch(PDOException $e){
13. die("Error : ".$e->getMessage());
14. }
15. ?>

```

Communiquer avec une BD ▶ PDO ▶ Connexion et contrôle

Fichier connexion.php

```

1. <?php
2. include("connexion.php");
3. $sql="SELECT * FROM `person`";
4. //Interact with database
5. $pdoStatement=$pdo->query($sql);
6. $result = $pdoStatement->fetchAll(PDO::FETCH_ASSOC);
7. print_r($result);
8. //other solution
9. echo "<br><br>";
10. $pdoStatement = $pdo->query($sql);
11. while ($row = $pdoStatement->fetch())
12. echo "<br>\nFirst Name = ". $row['first_name']. "<br>".
    " Last Name = " . $row['last_name']. "<br>".
    " Mail = " . $row['mail']. "<br>".
    "<br>";
13.
14. if ($pdo) {
15.     $pdo = NULL ; // close the connexion
16. }
17.
18. ?>

```

Communiquer avec une BD ▶ PDO ▶ Lecture tableau

Les données sont retournées dans un tableau associatif en mémoire

```

1. <?php
2. include("connexion.php");
3. $sql="SELECT * FROM `person`";
4. $pdoStatement=$pdo->query($sql);
5. $result = $pdoStatement->fetchAll(PDO::FETCH_OBJ); //
    return array of anonymous object instance of stdClass with
    column names as properties
6. foreach ($result as $key => $value) {
7.     echo $key."<br>";
8.     print_r($value);
9.     echo "<br>";
10. }
11. ?>

```

Communiquer avec une BD ▶ PDO ▶ Lecture objet anonyme

Les données sont retournées dans des objets instances de la classe stdClass

```

1. <?php
2. include("connexion.php");
3. include("Person.class.php");
4. $sql="SELECT * FROM `person`";
5. $pdoStatement=$pdo->query($sql);
6. $pdoStatement->setFetchMode(PDO::FETCH_CLASS, 'Person');
7. while ($person = $pdoStatement->fetch())
8.     echo $person;
9. ?>

```

Fichier Person.class.php

```

1. <?php
2. class Person{
3.     public function __toString() :String
4.     {
5.         return "<br>First Name =", $this->first_name."<br>","Last Name =", $this-
            >last_name."<br>","Mail =", $this->mail."<br>";
6.     }
7. }
8. ?>

```

Communiquer avec une BD ▶ PDO ▶ Lecture objet spécifique

Les données sont retournées dans des instances de la classe spécifiée dans le code.
Les attributs sont ajoutées à la classe et ensuite le constructeur de la classe est appelée

```

1. <?php
2. include("connexion.php");
3. $sql="SELECT * FROM `person` where `mail` LIKE :domain";
4. //Interact with database
5. $pdoStatement=$pdo->prepare($sql);
6. $pdoStatement->execute(array(":domain"=>"%@imt-atlantique.fr"));
7. $result = $pdoStatement->fetchAll(PDO::FETCH_ASSOC);
8. print_r($result);
9. if ($pdo) {
10.     $pdo = NULL ; // close the connexion
11. }
12. ?>

```

Communiquer avec une BD ▶ PDO ▶ Lecture avec paramètres

```

1. $firstName="Issam";
2. $lastName="REBAI";
3. $mail="issam.rebai@imt-atlantique.fr";
4. $sql="INSERT INTO person(`lastName`,`firstName`,`mail`)
    VALUES ('$lastName','$firstName','$mail')";
5. //Interact with database
6. $nbResult=$pdo->exec($sql);
7. echo "Result of request : ".$nbResult;
8. if ($pdo) {
9.     $pdo = NULL ; // close the connexion
10. }
11. ?>

```

Communiquer avec une BD ▶ PDO ▶ Insertion

Requête construite par concaténation → Risque d'injection de code SQL

```

<?php
//....
$firstName=$_GET["firstName"];
$lastName=$_GET["lastName"];
$mail=$_GET["mail"];
$sql="INSERT INTO person(`lastName`,`firstName`,`mail`) VALUES
(?,?,?)";
$req= $pdo->prepare($sql);
//Interact with database
$nbResult=$req->execute(array($lastName, $firstName,$mail));
echo "Result of request : ".$nbResult;
if ($pdo) {
    $pdo = NULL ; // close the connexion
}
?>

```

Communiquer avec une BD ▶ PDO ▶ Insertion

Requête préparée avec marqueur ? → protection contre l'injection de code SQL

```

<?php
//....
$sql="INSERT INTO person(`lastName`,`firstName`,`mail`) VALUES
(:ln,:fn,:mail)";
$req= $pdo->prepare($sql);
//Interact with database
$nbResult=$req->execute(array('ln'=>$lastName,
    'fn'=>$firstName, 'mail'=>$mail));
?>

```

Communiquer avec une BD ▶ PDO ▶ Insertion

Requête préparée avec marqueurs nominatifs → protection contre l'injection de code SQL