



# Se connecter à une base de données MySQL en PHP

Télécharger  
le PDF du  
cours



Cours  
complet  
PHP et  
MySQL

INTRODUCTION  
AU COURS PHP ET  
MYSQL

1. Introduction au  
cours : définitions et  
rôles du PHP et du  
MySQL

2. Client et serveur :  
définitions et  
interactions

3. Mise en place de  
notre  
environnement de  
travail

4. Créer, enregistrer  
et exécuter un script  
PHP

5. Afficher un  
résultat en PHP avec

Dans ce nouveau chapitre, nous allons passer en revue les différents moyens que nous avons de nous connecter au serveur et donc à nos bases de données MySQL en PHP.

Nous discuterons des avantages et des inconvénients de telle ou telle méthode et allons également apprendre à nous connecter à nos bases de données à proprement parler.

## Se connecter à MySQL en PHP : les API proposées par le PHP

Pour pouvoir manipuler nos bases de données MySQL en PHP (sans passer par phpMyAdmin), nous allons déjà devoir nous connecter à MySQL.

Pour cela, le PHP met à notre disposition deux API (Application Programming Interface) :

- L'extension MySQLi ;
- L'extension PDO (PHP Data Objects).

Note : Auparavant, nous pouvions également utiliser l'extension MySQL. Cependant, cette extension est désormais dépréciée et a été remplacée par MySQLi (« i » signifie « improved », c'est-à-dire « amélioré » en français).

## Quelle API préférer : MySQLi ou PDO ?

Le PHP nous fournit donc deux API pour nous connecter à MySQL et manipuler nos bases de données.

Chacune de ces deux API possède des forces différentes et comme vous vous en doutez elles ne sont pas forcément interchangeables.

Il existe notamment une différence notable entre ces deux API : l'extension MySQLi ne va fonctionner qu'avec les bases de données MySQL tandis que PDO va fonctionner avec 12 systèmes de bases de données différents.

Pour cette raison, nous préférons généralement le PDO car si vous devez un jour utiliser un autre système de bases de données, le changement sera beaucoup plus simple que si vous avez tout codé en MySQLi auquel cas vous devrez réécrire le code dans son ensemble.

ement sites web 1€/mois - 100% illimité,  
e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com  
ites

7. Les types de données en PHP	
8. Opérateurs et concaténation en PHP	
LES STRUCTURES DE CONTRÔLE EN PHP	
9. Présentation des conditions et des opérateurs de comparaison	En termes de fonctionnement, MySQLi et PDO sont tous les deux orienté objet (bien que MySQLi propose également une API en procédural), et ils supportent également tous les deux les requêtes préparées qui servent à se prémunir des injections SQL (nous reparlerons de cela dans la suite du cours).
10. Les conditions if, if...else et if...elseif...else	Dans ce cours, j'utiliserai donc PDO sauf pour ce chapitre où il me semble intéressant de vous montrer les différences d'écriture pour un script de connexion à une base de données MySQL.
11. Créer des conditions robustes avec les opérateurs logiques	<h2>Connexion au serveur avec MySQLi orienté objet</h2>
12. Ecrire des conditions condensées avec les opérateurs ternaire et fusion null	Pour se connecter au serveur et accéder à nos bases de données MySQL en MySQLi orienté objet, nous allons avoir besoin de trois choses : le nom du serveur ainsi qu'un nom d'utilisateur (avec des privilèges de type administrateur) et son mot de passe.
13. L'instruction switch en PHP	Dans le cas où votre site est hébergé sur un serveur, votre hébergeur vous donnera ces différents éléments. Ici, bien évidemment, nous travaillons en local. Le nom de notre serveur est donc <b>localhost</b> .
14. Les boucles PHP et les opérateurs d'incréméntation et de décréméntation	
15. Inclure des fichiers dans un autre en PHP avec include et require	
DÉCOUVERTE DES FONCTIONS EN PHP	
16. Introduction aux fonctions PHP	Concernant les identifiants au serveur local, ils peuvent changer selon vos paramétrages et selon le système que vous utilisez. Cependant, si vous disposez des réglages par défaut, le nom d'utilisateur devrait toujours être <b>root</b> et le mot de passe associé devrait être soit <b>root</b> soit une chaîne de caractère vide.
17. Contrôler le	

ement sites web 1€/mois - 100% illimité,  
e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com  
ites

variables en PHP

20. Constantes et constantes magiques en PHP

LES VARIABLES TABLEAUX EN PHP

21. Présentation des tableaux et tableaux numérotés en PHP

22. Les tableaux associatifs en PHP

23. Les tableaux multidimensionnels en PHP

MANIPULER DES DATES EN PHP

24. Le timestamp UNIX et la date en PHP

25. Obtenir et formater une date en PHP

26. Comparer des dates et tester la validité d'une date en PHP

LES VARIABLES SUPERGLOBALES PHP

27. Les variables superglobales PHP

28. Création et gestion des cookies en PHP

29. Définir et utiliser les sessions en PHP

MANIPULER DES FICHIERS EN PHP

```
4. <title>Cours PHP / MySQL</title>
5. <meta charset="utf-8">
6. <link rel="stylesheet" href="cours.css">
7. </head>
8. <body>
9. <h1>Bases de données MySQL</h1>
10. <?php
11.     $servername = 'localhost';
12.     $username = 'root';
13.     $password = 'root';
14.
15.     //On établit la connexion
16.     $conn = new mysqli($servername, $username, $password);
17.
18.     //On vérifie la connexion
19.     if($conn->connect_error){
20.         die('Erreur : ' . $conn->connect_error);
21.     }
22.     echo 'Connexion réussie';
23.
24. </body>
25. </html>
```

Cours PHP / MySQL

localhost:8888/cours-php/cours.php

Bases de données MySQL

Connexion réussie

Pour se connecter, nous instancions la classe prédéfinie `mysqli` en passant au constructeur les informations suivantes : nom du serveur auquel on doit se connecter, nom d'utilisateur et mot de passe.

Nous stockons les informations de connexion dans un objet qu'on appelle ici `$conn`. Cet objet représente notre connexion en soi.

PHP	Informations relatives à la tentative de connexion.
32. Créer et écrire dans un fichier en PHP	Pour cela, nous utilisons la propriété <code>connect_error</code> de la classe <code>mysqli</code> qui retourne un message d'erreur relatif à l'erreur rencontrée en cas d'erreur de connexion MySQL ainsi que la fonction <code>die()</code> pour stopper l'exécution du script en cas d'erreur.
33. Autres opérations sur les fichiers en PHP	Attention : La propriété <code>connect_error</code> de <code>mysqli</code> ne fonctionne correctement que depuis la version 5.3 de PHP. Utilisez la fonction <code>mysqli_connect_error()</code> pour les versions antérieures. Notez ici qu'on aurait également pu utiliser les exceptions et des blocs <code>try</code> et <code>catch</code> pour gérer les erreurs potentielles. Je voulais juste vous présenter une autre manière de faire ici.
UTILISER LES EXPRESSIONS RÉGULIÈRES OU RATIONNELLES EN PHP	Dans le cas où la connexion réussit, on se contente d'afficher un message « connexion réussie ».
34. Introduction aux expressions rationnelles ou expressions régulières	Si vous désirez la liste complète des propriétés et méthodes de la classe <code>mysqli</code> , je vous invite à consulter la <a href="#">documentation officielle</a> .
35. Les fonctions PCRE PHP	
36. Les classes de caractères des regex	
37. Les métacaractères des regex PHP	
38. Les options des expressions régulières disponibles en PHP	
PROGRAMMATION ORIENTÉE OBJET (POO) PHP : CONCEPTS DE BASE	
39. Introduction à la programmation orientée objet PHP : classes, instances et objets	
40. Propriétés et méthodes en PHP orienté objet	
41. Les méthodes	

43. Classes étendues et héritage en PHP orienté objet	<div><h2>Bases de données MySQL</h2><p>Connexion réussie</p></div>
44. Surcharge d'éléments et opérateur de résolution de portée en PHP	<p>Ce script ressemble à priori au précédent et pourtant il est bien très différent : nous n'avons cette fois-ci plus recours à notre classe <code>mysqli</code> ni à l'orienté objet.</p> <p>A la place, nous utilisons les fonctions <code>mysqli_connect()</code> pour nous connecter à la base de données et <code>mysqli_connect_error()</code> pour obtenir des informations sur l'erreur de connexion si il y en a une.</p>
45. Les constantes de classe en PHP objet	
46. Les propriétés et méthodes statiques en PHP objet	
47. Les méthodes et les classes abstraites en PHP objet	
48. Les interfaces en PHP orienté objet	
49. Les méthodes magiques en orienté objet PHP	
PROGRAMMATION ORIENTÉE OBJET PHP : NOTIONS AVANCÉES	
50. Le chaînage de méthodes en PHP	<h2>Connexion au serveur avec PDO</h2> <p>Pour se connecter en utilisant PDO, nous allons devoir instancier la classe <code>PDO</code> en passant au constructeur la source de la base de données (serveur + nom de la base de données) ainsi qu'un nom d'utilisateur et un mot de passe.</p>
51. Les closures et les classes anonymes en PHP objet	<pre>1. &lt;!DOCTYPE html&gt; 2. &lt;html&gt; 3.   &lt;head&gt; 4.     &lt;title&gt;Cours PHP / MySQL&lt;/title&gt; 5.     &lt;meta charset="utf-8"&gt; 6.     &lt;link rel="stylesheet" href="cours.css"&gt; 7.   &lt;/head&gt; 8.   &lt;body&gt; 9.     &lt;h1&gt;Bases de données MySQL&lt;/h1&gt;</pre>
52. L'auto chargement des classes en PHP	
53. Le mot clef final en PHP objet	
54. La résolution	

# INTRODUCTION AUX BASES DE DONNÉES. AU SOL

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Cours PHP / MySQL</title>
5.         <meta charset="utf-8">
6.         <link rel="stylesheet" href="cours.css">
7.     </head>
8.     <body>
9.         <h1>Bases de données MySQL</h1>
10.        <?php
11.            $servername = 'localhost';
12.            $username = 'root';
13.            $password = 'root';
14.
15.            //On essaie de se connecter
```

6/10



le données, SSL, Bande passante illimitée, Baie de stockage SSD [hosteur.com](https://www.hosteur.com)  
ites



78. Création de jointures SQL	Pour fermer la connexion avant cela, nous allons devoir utiliser différentes méthodes selon la méthode d'ouverture choisie.
79. L'opérateur SQL UNION	Si on utilise MySQLi orienté objet, alors il faudra utiliser la méthode <code>close()</code>
80. Les opérateurs de sous requête SQL	<div></div>
GESTION DES FORMULAIRES HTML AVEC PHP	
81. Rappels sur les formulaires HTML	
82. Récupérer et manipuler les données des formulaires HTML en PHP	
83. Sécurisation et validation des formulaires en PHP	
CONCLUSION DU COURS PHP ET MYSQL	
84. Conclusion du cours complet PHP et MySQL	
	Si on utilise MySQLi procédural, on utilisera la fonction <code>mysqli_close()</code>



ement sites web 1€/mois - 100% illimité,  
e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com  
ites





Si on utilise PDO, il faudra détruire l'objet représentant la connexion et effacer toutes ses références. Nous pouvons faire cela en assignant la valeur **NULL** à la variable gérant l'objet.



ement sites web 1€/mois - 100% illimité,  
e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD [hosteur.com](#)  
ites



Vous devez vous connecter pour publier un commentaire.

[Connexion](#)

[Confidentialité](#)

[CGV](#)

[Sitemap](#)

© Pierre Giraud - Toute reproduction interdite - Mentions légales



ement sites web 1€/mois - 100% illimité,  
e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com  
ites