Les instructions préparées en PHP

1. Objectifs

- Être capable d'utiliser les API php pour accéder à une base de données MySQL

2. Présentation

- Les instructions préparées en PHP sont une méthode utilisée pour exécuter des requêtes SQL de manière sécurisée et efficace. Cette méthode implique de préparer une requête SQL avec des paramètres vides qui seront remplis avec des valeurs réelles lors de l'exécution de la requête.
- La préparation de la requête est effectuée en deux étapes.
  - Tout d'abord, la requête est envoyée au serveur de base de données qui l'analyse et l'optimise.
  - Ensuite, les paramètres sont ajoutés à la requête, ce qui permet d'éviter les injections SQL, une technique de piratage qui permet à un attaquant d'insérer du code malveillant dans une requête SQL.
- Lorsque la requête est préparée, elle est stockée dans la mémoire cache du serveur de base de données, ce qui permet d'accélérer les requêtes ultérieures. Lorsque la requête est exécutée, les paramètres sont remplis et la requête est envoyée au serveur de base de données pour être exécutée.
- Les instructions préparées en PHP sont particulièrement utiles pour les requêtes qui sont exécutées plusieurs fois avec des valeurs différentes. Elles offrent également une meilleure sécurité en empêchant les attaques par injection SQL.
- Deux manières sont offertes pour exécuter une requête:
  - **Exécution Directe**
    - cette méthode implique l'utilisation de la méthode query() ou exec() pour envoyer directement une requête SQL au serveur de base de données et récupérer le résultat.
    - La méthode **query()** est utilisée pour les requêtes de type **SELECT**,
    - La méthode **exec()** est utilisée pour les requêtes de type **INSERT**, **UPDATE** ou **DELETE**. Par exemple :

```
// Exécution directe d'une requête SELECT
$sql = "SELECT * FROM users WHERE username='john.doe'";
```

apcpedagogie.com

```
$result = $conn->query($sql);

// Exécution directe d'une requête INSERT
$sql = "INSERT INTO users (username, password) VALUES ('john.doe',
'password123')";
$conn->exec($sql);
```

- Cependant, cette méthode est vulnérable aux attaques par injection SQL si les valeurs des paramètres ne sont pas correctement échappées.

- **Utilisation de la méthode préparée**
  - cette méthode implique la préparation de la requête SQL avec des paramètres vides qui seront remplis avec des valeurs réelles lors de l'exécution de la requête. Cette méthode est plus sécurisée car elle empêche les attaques par injection SQL. Les méthodes prepare(), bind_param() et execute() sont utilisées pour préparer et exécuter une requête préparée. Par exemple :

```
// Requête préparée SELECT
$sql = "SELECT * FROM users WHERE username=? AND password=?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
$result = $stmt->get_result();

// Requête préparée INSERT
$sql = "INSERT INTO users (username, password) VALUES (?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
```

  - En utilisant des requêtes préparées, les valeurs des paramètres sont échappées automatiquement et sont donc protégées contre les attaques par injection SQL.

3. Avantages

- Les instructions préparées en PHP sont des requêtes SQL prédéfinies et précompilées qui peuvent être exécutées plusieurs fois avec des valeurs différentes. Les avantages des instructions préparées sont les suivants :

apcpedagogie.com

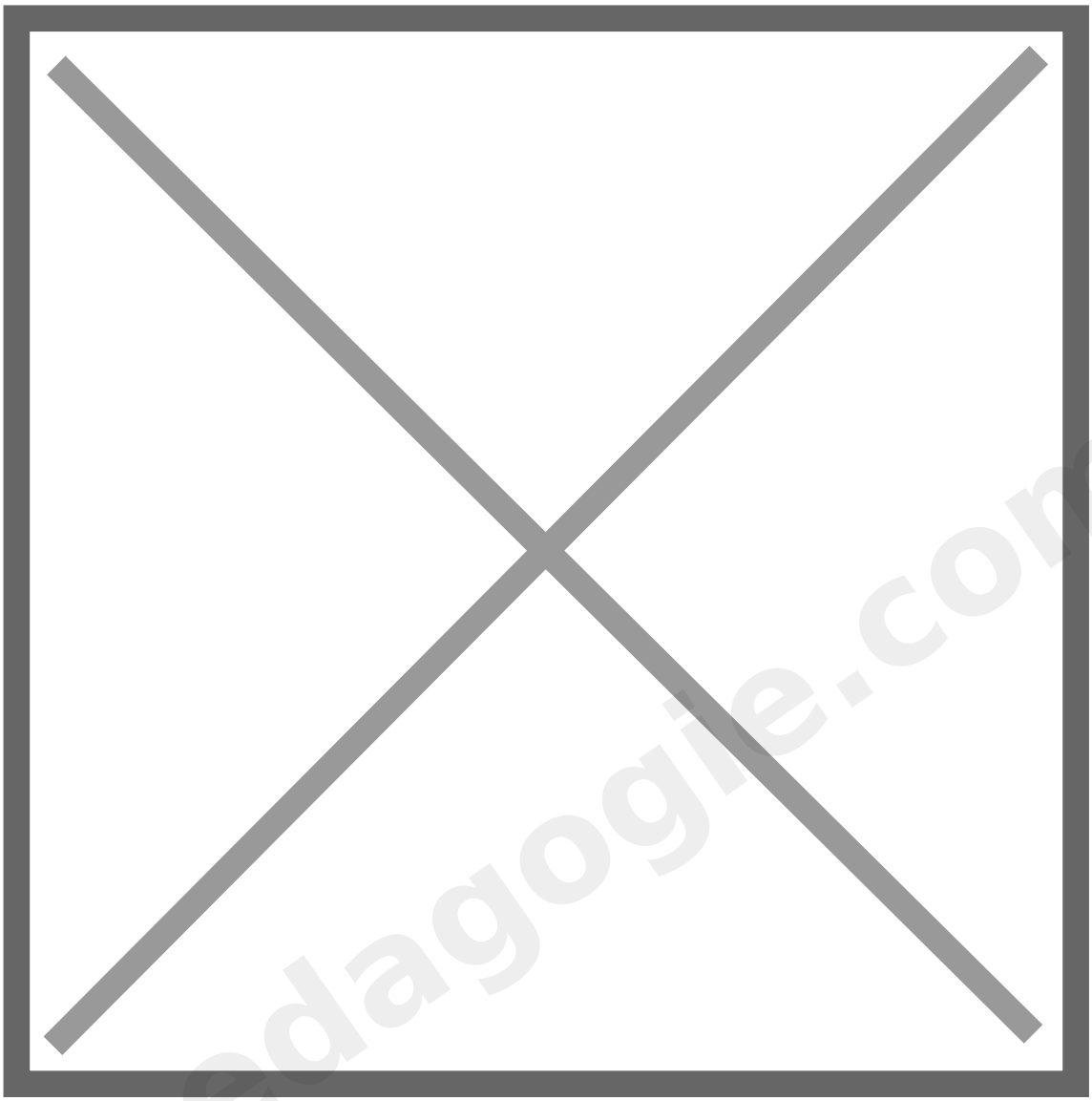
- **Sécurité :** L'utilisation d'instructions préparées permet de réduire les risques de failles de sécurité telles que les injections SQL. Les instructions préparées séparent les instructions SQL et les données, ce qui empêche les attaquants de modifier les instructions SQL.
- **Performance :** L'utilisation d'instructions préparées permet d'améliorer les performances de l'application, en particulier pour les requêtes répétitives. Les instructions préparées sont précompilées par le SGBD et optimisées pour une exécution rapide, ce qui permet de réduire les temps de réponse.
- **Lisibilité :** L'utilisation d'instructions préparées permet de rendre le code plus lisible et plus facile à maintenir. Les instructions préparées sont généralement plus claires et plus concises que les requêtes SQL inline, ce qui rend le code plus facile à comprendre pour les développeurs.
- **Flexibilité :** Les instructions préparées peuvent être utilisées pour exécuter des requêtes SQL dynamiques en fonction des données de l'utilisateur ou des paramètres de l'application. Cela permet de rendre l'application plus flexible et plus adaptable aux besoins de l'utilisateur.

> ***En résumé, les instructions préparées en PHP offrent des avantages en termes de sécurité, de performances, de lisibilité et de flexibilité pour les applications qui nécessitent l'utilisation de requêtes SQL.***

4. Utilisation

- La requête directe peut être changée de la manière suivante:

apcpedagogie.com



- PDO prend en charge les espaces réservés positionnels (`?`) et nommé (`:...`)

  Les instructions préparées en PHP

- **Les espaces réservés positionnels**

  - **Les espaces réservés positionnels** sont identifiés par un point d'interrogation (`?`) dans la requête préparée.
  - Les valeurs à lier sont ensuite fournies sous forme d'un tableau ou d'une liste dans la méthode execute() dans l'ordre dans lequel elles apparaissent dans la requête préparée.

apcpedagogie.com

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE id = ? AND status = ?');
$stmt->execute([$user_id, $status]);
```

> *Dans PDO, nous pouvons utiliser des espaces réservés positionnels et nommés. Pour les requêtes simples, personnellement, je préfère les espaces réservés positionnels, je les trouve moins verbeux, mais c'est entièrement une question de goût.*

- **Les espaces réservés nommés**

  - **Les espaces réservés nommés** sont identifiés par un nom préfixé par deux-points (`:`), suivi d'un identifiant unique. Les valeurs à lier sont ensuite fournies sous forme d'un tableau associatif dans la méthode `execute()`.

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE name = :name AND email = :email');
$stmt->execute([':name' => $name, ':email' => $email]);
```

> *L'utilisation d'espaces réservés nommés peut rendre votre code plus lisible, en particulier lorsque vous avez plusieurs paramètres à lier. De plus, cela peut aider à éviter les erreurs d'ordre de paramètres, car les valeurs sont liées en fonction de leur nom plutôt que de leur position dans la requête préparée.*

5. La démarche à suivre pour utiliser une requête préparée

- Voici les étapes générales pour utiliser une requête préparée en PHP :
  - ***1-*** **Préparer la requête** :
    - La première étape consiste à préparer la requête SQL en utilisant la méthode "prepare" de l'objet PDO. Cette méthode prend en paramètre la requête SQL avec des marqueurs de position à la place des valeurs réelles. Par exemple, une requête préparée pour insérer des données dans une table peut ressembler à ceci :

```
$stmt = $pdo->prepare('INSERT INTO users (name, email) VALUES (:name, :email)');
```

apcpedagogie.com

- ***2-* Binder les valeurs**:
  - La deuxième étape consiste à binder les valeurs réelles aux marqueurs de position dans la requête préparée en utilisant la méthode "bindValue" ou "bindParam" de l'objet PDO. Par exemple :

```
$stmt->bindValue(':name', 'John Doe');
$stmt->bindValue(':email', 'john.doe@example.com');
```

  - Vous pouvez également utiliser des tableaux pour binder plusieurs valeurs en même temps. Par exemple :

```
$data = array('name' => 'John Doe', 'email' =>
'john.doe@example.com');
$stmt->execute($data);
```

- ***3-* Exécuter la requête** :
  - La dernière étape consiste à exécuter la requête en utilisant la méthode "execute" de l'objet PDO. Cette méthode exécutera la requête avec les valeurs bindées et renverra un objet PDOStatement qui contiendra les résultats de la requête.

```
$stmt->execute();
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

  - Notez que les étapes 2 et 3 peuvent être effectuées ensemble en utilisant la méthode "execute" avec un tableau de valeurs. Par exemple :

```
$stmt->execute(array('name' => 'John Doe', 'email' =>
'john.doe@example.com'));
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

- En utilisant des requêtes préparées, vous pouvez réduire les risques de failles de sécurité telles que les injections SQL et améliorer les performances de votre application.

6. Sélection de plusieurs lignes

- La récupération de plusieurs lignes à partir d'une requête préparée serait identique à celle d'une requête sans paramètres déjà affichés:

apcpedagogie.com

Les instructions préparées en PHP

- **Liaison des paramètres**

  - Il existe plusieurs moyens de lier les paramètres avec des valeurs ou des variables. C'est à dire de remplacer les paramètres par les valeurs:
  - Soit on regroupe les valeurs dans un tableau et on passera ce tableau en paramètre à la méthode "`execute`" comme dans l'exemple ci-dessous, si on utilise des marqueur interrogatif "`?`"

```
$stmt=$pdo-> prepare("SELECT * FROM etudiants where genre
                      LIKE ? and prenomEtud like ?");
$stmt->execute(['Femme','__h%']);
```

  - Soit on regroupe les valeurs dans un tableau associatif et on passera ce tableau en paramètre à la méthode exécute comme dans l'exemple ci-dessous, si on utilise des marqueurs nommés.

7. Exercices d'application

1. **Exercice 01**

   1. **Énoncé**

      - Utiliser la base de donnée "**notes_des_étudiants**", crée dans la leçon précédente pour répondre aux questions suivantes:
        - ***1-*** Remplir avec **PhPMyAdmin** la table "**etudiants**" avec des données fictives
        - ***2-*** Utiliser la page **connexionDB.php** crée dans la leçon précédente pour ce connecter à la base de donnée.
        - ***3-*** Créer une page php dont le nom est "**affichageEtudiants.php**" (utiliser template bootstrap5)
        - ***4-*** Écrire un script php permettant d'afficher la liste des étudiants (Hommes), dans un tableau HTML, en utilisant deux méthodes différentes.
        - ***5-*** Récupérer les étudiants (Hommes) dont le prénom contient la lettre 'h' dans la deuxième position.

apcpedagogie.com

- ***6-*** Afficher la liste des étudiantes (Femmes), dans un tableau HTML , en utilisant deux méthodes différentes.
- ***7-*** Récupérer les étudiantes (Femmes) dont le prénom contient la lettre 'h' dans la deuxième position.

- **Remarque** : vous pouvez utiliser le fichier **style.css**

2. **Solution**

- Essayez de faire l'exercice de votre côté avant de regarder la solution !

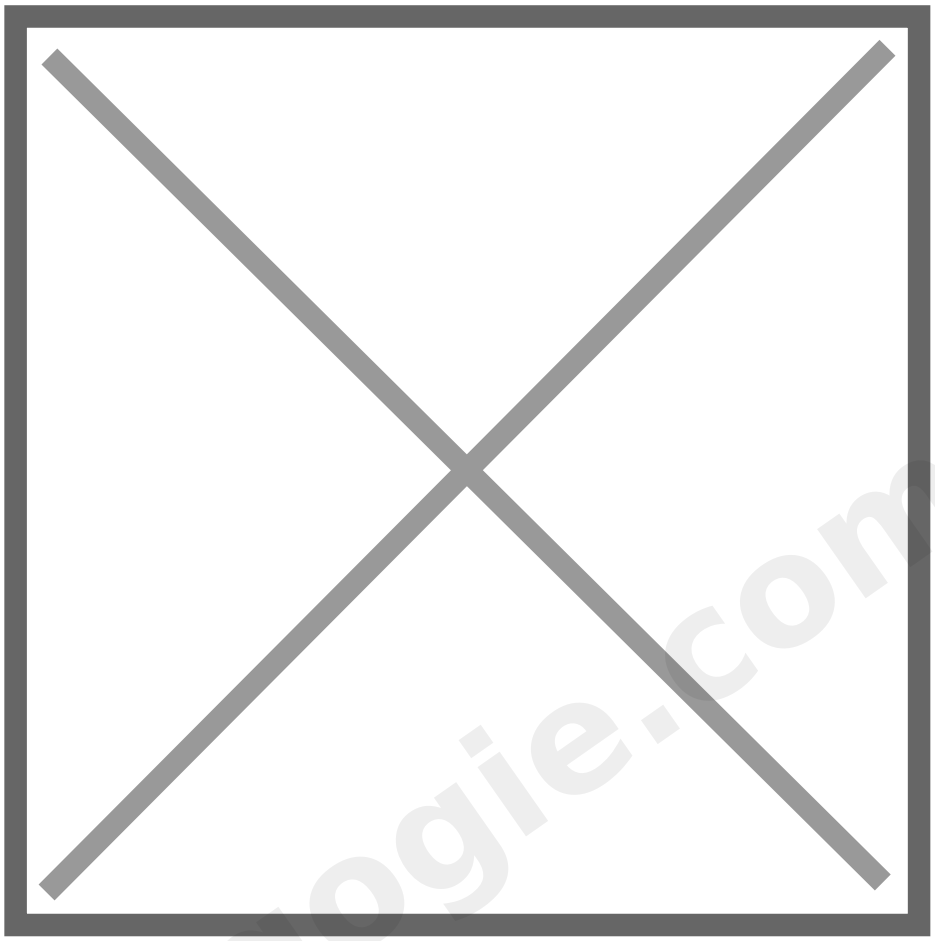
2. **Exercice 02**

1. **Énoncé**

- Soit la base de données "**gestiondesapprenants**" dont le dictionnaire de données ci-dessous.

  Version imprimable - phpMyAdmin 4.8.5
- ***1-*** Créer un script PHP nommé **connect.php** contenant uniquement les variables de connexion.
- ***2-*** Créer le squelette d'une page web classique (en-tête, corps, etc.). La sauvegarder sous le nom **connexion.php**.
- ***3-*** Inclure le fichier **connect.php** dans **connexion.php**, puis créer un nouvel objet PDO (PHP Data Object) permettant de vous connecter à MySQL en utilisant les informations saisies dans la page **connect.php**.
- ***4-*** Ajouter au script **connexion.php** des instructions de gestion d'erreur (try, catch). En cas de connexion non réussie, afficher l'erreur standard renvoyée par la méthode getMessage().
- **Affichage par ligne**
  - Créer une page PHP nommé **affichageCode.php**, la page doit correspondre au schémas suivant:

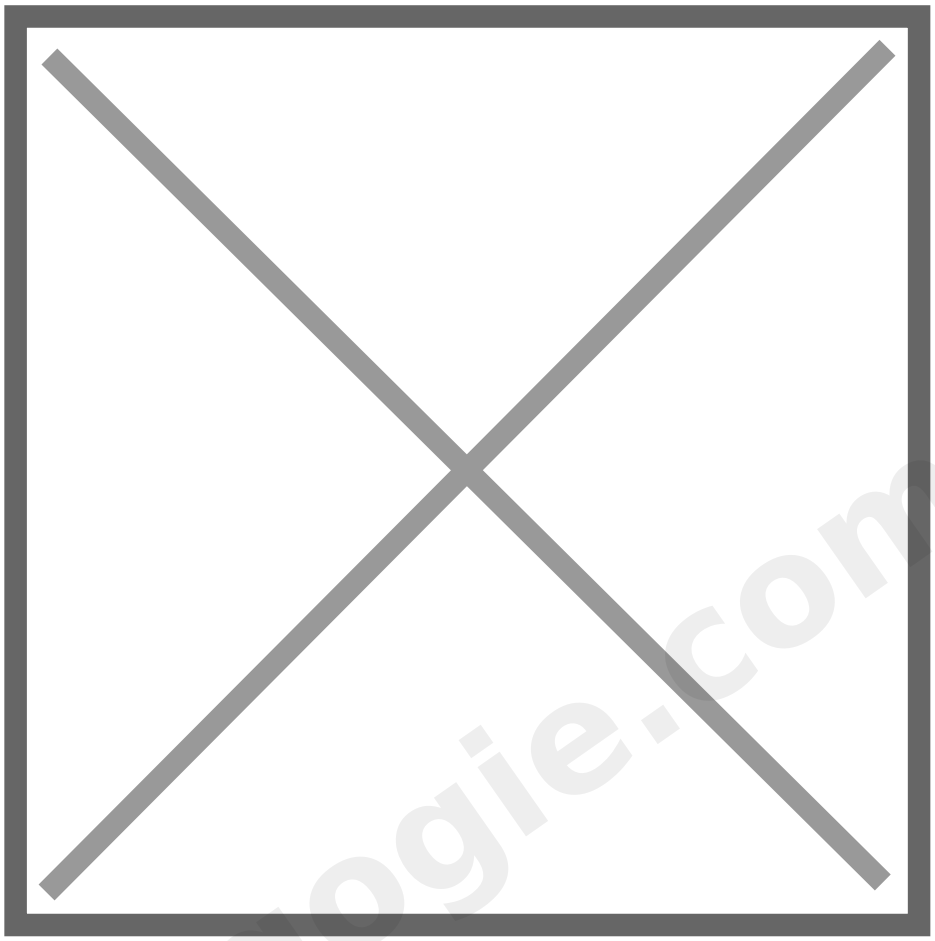
apcpedagogie.com



- Pour créer la page, utiliser le célèbre **framework CSS bootstrap4**.
- Pour les champs liées (cdeGouvernorat,cdeGroupe et cdeUser) et le champs civilité afficher les valeur réels de ces champs.

- **Affichage multiple**
  - Créer une page PHP nommé **rechercheMultiple.php**, la page doit correspondre au schémas suivant:

apcpedagogie.com



2. **Solution**

    - Essayez de faire l'exercice de votre côté avant de regarder la Solution !

3. **Exercice 03**

4. **Énoncé**

    - **Création d'une application de gestion de tâches**
        - L'objectif de cette application est de permettre aux utilisateurs de créer et de gérer des listes de tâches à faire. Les utilisateurs pourront s'inscrire, se connecter et créer des listes de tâches. Chaque liste pourra contenir plusieurs tâches, avec une description, une date d'échéance et un statut (en cours, terminé, annulé). Les utilisateurs pourront modifier ou supprimer leurs listes de tâches et marquer les tâches comme terminées ou annulées.

apcpedagogie.com

- Pour développer cette application, vous devrez utiliser PHP pour créer la logique serveur, HTML pour la structure des pages web et Bootstrap 5 pour la mise en forme des pages. Pour interagir avec la base de données, vous utiliserez PDO avec la méthode prepare() pour préparer et exécuter des requêtes SQL sécurisées.

- **Voici les fonctionnalités principales de l'application :**
  - Inscription et connexion des utilisateurs
  - Création, modification et suppression de listes de tâches
  - Ajout, modification et suppression de tâches dans une liste
  - Marquage des tâches comme terminées ou annulées
  - Affichage des listes de tâches et des tâches associées pour chaque utilisateur
  - Le développement de l'application devra être divisé en plusieurs étapes, comprenant la création de la base de données, la mise en place de la logique serveur avec PHP, la création des pages web avec HTML et Bootstrap 5, et la mise en place de la sécurité avec PDO et la méthode prepare().
  - Le code devra être propre, bien organisé et facilement compréhensible, avec des commentaires pour expliquer les parties les plus complexes. Les pages web devront être esthétiques et conviviales, avec une navigation intuitive pour les utilisateurs.

Bonne chance pour le développement de cette application !

Cours PHP

Source : https://apcpedagogie.com/php-declarations-preparees · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com