

Module « Technologies du Web »

Feuille de TP 3 MYSQL/PHP

Exercice 1 :

Testez les scripts PHP suivant pour apprendre à se connecter à un Serveur, à une base de données, créer une table et insérer des données dans cette base, et enfin afficher et exploiter ces données.

Utilisation de bases de données MySQL avec PHP

L'accès à une base de données se réalise en 3 étapes:

- La connexion
- La requête SQL
- La déconnexion

1. La connexion

Avant de faire le moindre accès à une base de données, il faut impérativement établir une connexion et choisir sa base de données

```
<?php
```

```
$serveurBD = "localhost";
```

```
$nomUtilisateur = "toto";
```

```
$motDePasse = "secret";
```

```
$baseDeDonnees = "mabase";
```

```
mysql\_connect($serveurBD,
```

```
$nomUtilisateur,
```

```
$motDePasse);
```

```
$connexionReussie = mysql_select_db($baseDeDonnees);
```

```
// Et pour mettre fin à la connexion
```

```
mysql_close(); ?>
```

- **\$serveurBD** est le nom (ou adresse IP) du serveur où se trouve la base de données MySQL
- **\$nomUtilisateur** est l'identifiant sous lequel vous souhaitez vous connecter à la base de données. Chez vous par défaut, c'est généralement "root".
- **\$motDePasse** est le mot de passe associé à l'identifiant. Chez vous c'est une chaîne vide par défaut.
- **\$baseDeDonnees** est un nom de base de données qui doit vous devrez en créer une en passant par exemple par PHPMyAdmin (A vous de choisir son nom).

Voici un exemple concret en local:

```
<?php
```

```
$serveurBD = "localhost";
```

```
$nomUtilisateur = "root";
```

```
$motDePasse = "";
```

```
$baseDeDonnees = "mabase";
```

```
$idConnexion = mysql_connect($serveurBD,$nomUtilisateur, $motDePasse);
```

```
if ($idConnexion !== FALSE) echo "Connexion au serveur réussie";  
else echo "Echec de connexion au serveur";
```

```
$connexionBase = mysql_select_db($baseDeDonnees);
```

```
if ($connexionBase) echo "Connexion à la base réussie";
```

```
else echo "Echec de connexion à la base"; // Et pour
```

```
mettre fin à la connexion mysql_close();
```

```
?>
```

On peut aussi utiliser une autre méthode pour stopper le script si la connexion ne peut pas se faire (inutile d'essayer d'accéder à la base si le serveur n'est pas accessible) :

```
<?php
    $serveurBD = "localhost";
    $nomUtilisateur = "root";
    $motDePasse = "";
    $baseDeDonnees = "mabase";
    @mysql_connect($serveurBD,
        $nomUtilisateur,
        $motDePasse)
        or die("Impossible de se connecter au serveur de bases de données.");
    @mysql_select_db($baseDeDonnees) or
die("Cette base de donnees n'existe pas");
    // Et pour mettre fin à la connexion
mysql_close();
?> or die("") permet d'afficher un texte et de stopper le script si la fonction précédente renvoie une erreur. Notez
que pour ne pas avoir à l'écran les messages d'erreur remontés par PHP, il suffit de mettre un @ devant les appels
aux fonctions, par exemple @mysql_connect()
```

2. La déconnexion

Comme vous avez pu le voir dans les exemples précédents, la déconnexion se fait par appel à la fonction *mysql_close()*.

3. Requête SQL ne retournant pas de résultat (CREATE TABLE, INSERT, UPDATE, etc.)

Pour exécuter une requête SQL nous pouvons faire appel à la fonction *mysql_query()*.

```
<?php
    $serveurBD = "localhost";
    $nomUtilisateur = "root";
    $motDePasse = "";
    $baseDeDonnees = "mabase";
    @mysql_connect($serveurBD,
        $nomUtilisateur,
        $motDePasse)
        or die("Impossible de se connecter au serveur de bases de données.");
    @mysql_select_db($baseDeDonnees) or
die("Cette base de donnees n'existe pas");
    $sql = "CREATE TABLE IF NOT EXISTS smi " .
        "(id INTEGER PRIMARY KEY AUTO_INCREMENT, " .
        "nom VARCHAR(128))";
    $retour = mysql_query($sql); if
($retour === FALSE) {
        echo "La requête CREATE TABLE a échoué.";
    } else {
        echo "La table smi a été créé.";
    }
    $sql = "INSERT INTO smi (nom) VALUES (' TP1 HTML SMI S6')";
    $retour = mysql_query($sql);
    if ($retour === FALSE) {
        echo "La requête INSERT a échoué.";
    } else {
        echo "Un nouveau nom a été ajouté dans la table smi.";
    }
    // Et pour mettre fin à la connexion
mysql_close();
?>
```

Avec ce script, nous créons une table puis nous y insérons un nom. Ajoutez autre nom de TP.

4. Requête SQL retournant des résultats (ex: SELECT)

Pour une requête de type SELECT, nous ferons également appel à la fonction *mysql_query()* mais le traitement sera légèrement différent.

```
<?php
$serveurBD = "localhost";
$nomUtilisateur = "root";
$motDePasse = "";
$baseDeDonnees = "mabase";
@mysql_connect($serveurBD,
    $nomUtilisateur,
    $motDePasse)
    or die("Impossible de se connecter au serveur de bases de données.");
@mysql_select_db($baseDeDonnees) or
die("Cette base de données n'existe pas");
$sql = "SELECT * FROM smi";
$retour = mysql_query($sql); if
($retour === FALSE) {
    echo "La requête SELECT a échoué.";
} else {
    while ($enreg = mysql\_fetch\_array($retour)) {
echo $enreg["id"]."-".$enreg["nom"]."<br />\n";
    }
}
// Et pour mettre fin à la connexion
mysql\_close();
?>
```

Exercice 2 :

- Ecrire un script PHP qui va créer la base de données suivante « **dbsmi** ».
- Ecrire un script PHP qui va (**CREATE TABLE**) la Table inscription, dans la base **dbsmi**.
- Ecrire le script PHP qui va (**INSERT, UPDATE, et AFFICHER.**) Les données du formulaire suivant dans cette table.

File:///C:/wamp/www/essai/test/PHP/FormPHP.html

Formulaire d'insertion des données Etudiants TP PHP 2014/2015

Nom
nom

Prénom
prenom

Email
email

Sexe
☐ male ☐ female

Mot de Passe

Age
AGE

Envoyer

Indications sur La démarche à suivre comme suit :

- On commence par récupérer les champs, avec if et isset
- On vérifie si les champs sont vides, avec if et empty
- connexion à la base
- on écrit la requête sql
- on insère les informations du formulaire dans la table
- on affiche le résultat pour le visiteur □ on ferme la connexion

Exercice 3 :

Bien sûr faire entrer des données dans une base de données est une très bonne chose, mais l'idéal et de tester si des valeurs de certains champs comme (NOM, Email, ETC) existe dans la table pour éviter les doublons.

- Réécrire le script PHP de l'exercice précédent pour vérifier si un tel NOM ou Email existe déjà dans la table pour éviter les doublons.

Exercice 4 :

- Ecrire le script PHP qui permet de lister et afficher les noms de toutes les Bases de Données qui existent sur le serveur MYSQL.