

SOMMAIRE

- Qu'est ce que PDO
- Les variables de sessions
- Le CRUD
- Htmlebrities
- Include
- Rappel TP2
- TP 3

QU'EST-CE QUE PDO

- PDO signifie PHP Data Objects. Il s'agit d'une interface qui permet aux scripts PHP d'interroger une base de données via des requêtes SQL.

PDO est une extension qui s'ajoute au PHP pour que ses différentes fonctionnalités soient disponibles dans le langage. On peut utiliser l'ensemble de ses fonctions pour exécuter des requêtes SQL quelque soit le SGBD. Autrement dit, si l'application Web repose sur le SGBD MySQL, on peut migrer vers le SGBD PostgreSQL.

- La déclaration de la chaîne de connexion se fait comme ceci:

```
<?php
    $pdo = new PDO("mysql:host=localhost;dbname=mabdd","user","motDePass");
?>
```
- \$pdo correspond à l'instance PDO créée en passant au constructeur les paramètres qui constituent la chaîne de connexion.

QU'EST-CE QUE PDO

Gestion des erreurs (Exception PDOException)

- Si les paramètres renseignés au constructeur de la classe PDO sont corrects et que le serveur de base de données fonctionne normalement alors l'objet \$pdo sera créé.
- Si les paramètres ne sont pas bien déclarés ou que le serveur de base de données n'est pas accessible alors une exception de type **PDOException** est lancée automatiquement. Il faut donc la rattraper (catch) comme ceci:

```
<?php
    try{
        $pdo=new PDO("mysql:host=localhost;dbname=mabase","user","1234");
    }
    catch(PDOException $e){
        echo $e->getMessage();
    }
?>
```

- Si notre projet Web est composé de plusieurs pages, elles auront besoin de dialoguer avec la base de données. Le code précédent sera donc déclaré dans chacune d'elles. on peut dans ce cas placer ce code dans un fichier à part qu'on inclura au besoin (à l'aide de **include()** ou **require()**...).

LES VARIABLES DE SESSIONS

- Une variable de session PHP est une variable stockée sur le serveur.
- C'est une variable temporaire qui a une durée limitée et est détruite à la déconnexion (fermeture du navigateur).
- Les variables de session sont **partagées** par toutes les pages PHP d'une session (accès depuis un même navigateur). Elles permettent donc le passage d'information entre pages.

Exemple d'utilisation

- Sauvegarde d'identifiants de connexion
- Sauvegarde d'un panier d'achat

LES VARIABLES DE SESSIONS

Démarrer une session

```
<?php session_start(); ?>
```

Ce code permet de charger le fichier contenant les variables de session sur le serveur, ou s'il n'existe pas de la créer.

Ce code est à placer au début de toutes les pages PHP qui souhaitent utiliser les variables de sessions, avant tout autre code PHP ou HTML.

Utiliser les variables

```
<?php
```

```
$_SESSION['variable'] = valeur ;
```

```
?>
```

Un tableau association `$_SESSION` est alors mis à disposition pour gérer des variables.

LES VARIABLES DE SESSIONS

Supprimer une variable : `unset($_SESSION['variable'])`

Supprimer toutes les variables : `session_unset()`

Supprimer la session : `session_destroy()`

CRUD

CRUD signifie: Create, Read, Update, Delete

Ce sont les fonctionnalités de base de toute application.

Ainsi nous avons la possibilité de :

Créer des données dans un formulaire HTML et les insérer en Base de données.

Lire de données depuis une base de données et les afficher à l'écran.

Mettre à jour ces données dans la base de données.

Supprimer ces données de la base de données.

HTML ENTITIES

La fonction `htmlentities()` est semblable à `htmlspecialchars()`.

Cette fonction se charge d'encoder tous les caractères qui ont un équivalent en HTML.

Ainsi, le caractère « é » deviendra « `é` » ce qui n'est pas le cas avec `htmlspecialchars()`.

HTMLENTITIES VS HTMLSPECIALCHARS

Si nous utilisons `htmlentities()` pour encoder les données avant de les enregistrer dans la base de données puis que nous appliquons à nouveau `htmlentities()` avant d'afficher les données lues dans la BD, nous verrons s'afficher à l'écran les accents encodés.

C'est pourquoi nous préférons utiliser `htmlspecialchars()`, qui laisse les caractères accentués intacts.

Rappel du TP 2

Une fois inscrit, l'utilisateur peut choisir un login et un mot de passe, une vérification sera faite sur le mot de passe en demandant de le saisir une seconde fois pour s'assurer qu'il soit identique.

Les login et mot de passe seront stockés en bdd, ainsi lorsque l'utilisateur se connecte sa session est donc active.

Créer une page de profil sur laquelle accèdera l'utilisateur après s'être authentifié.

Afficher un bouton de déconnexion, lorsque ce dernier est cliqué l'utilisateur est dirigé vers la page d'accueil.

TP 3

- Après avoir accéder à son profil, l'utilisateur souhaite ajouter une photo de profil. Avant de commencer le code écrire les étapes nécessaires à l'ajout d'une photo sur le site.
- Créer une page (Accueil) à laquelle on peut accéder seulement si nous sommes connecté. Cette page comprendra un menu de navigation (Accueil, profil et déconnexion), une image qui prendra toute la largeur de la page, un titre principal, et à la suite trois paragraphes placés côte à côte. Les boutons du menu doivent être fonctionnels. (Voir le wireframe ci-dessous).
- Aller plus loin: Remplacer l'image par un slider.

TP 3

