TD2 : PHP « avancé »
V3.3.1

Cette œuvre de Mickaël Martin Nevot est mise à disposition sous licence Creative Commons Attribution - Utilisation non commerciale - Partage dans les mêmes conditions.

Document en ligne : www.mickael-martin-nevot.com

1 Généralités

Vous visualiserez systématiquement votre travail dans différents navigateurs Web.

N'oubliez pas de faire des recherches sur le Web à chaque fois que cela est nécessaire en prenant soin de vérifier que les informations trouvées soient correctes.

Vous trouverez la boîte à outils ainsi que l'ensemble des documents nécessaires à la réalisation de ce TD sur le site Web de l'enseignant.

2 Inclusion de fichier

Dans le `TD1-2 : Premiers pas en PHP`, vous avez utilisé les fonctions `start_page(…)` et `end_page()` en les définissant dans chaque fichier.

Copiez ces fonctions dans un fichier nommé `utils.inc.php` puis éditez un nouveau fichier l'incluant (cela vous permettra d'utiliser les fonctions définies dans `utils.inc.php`) :

```
<?php
    include 'utils.inc.php';
?>
```

Remarque
L'extension `.inc.php` n'est pas obligatoire mais elle permet de reconnaître l'utilité d'un fichier avec seulement son nom.

3 Environnement

PHP permet de récupérer plusieurs types d'informations sur l'environnement, par exemple :

```
<?php
    // Correspond à l'adresse IP de celui qui visualise la page.
    echo getenv('REMOTE_ADDR');
    // Correspond à l'adresse IP du serveur.
    echo getenv('HTTP_HOST');
    // Correspond au logiciel serveur Web utilisé.
```

 (cc) BY-NC-SA

```
    echo getenv('SERVER_SOFTWARE');
?>
```

Il existe aussi une fonction `phpinfo()` qui permet de visualiser tous les renseignements sur l'environnement (cette fonction permet uniquement d'afficher les informations à l'écran et non de les stocker dans une variable). Testez-là (cette fonction divulgue des informations de sécurité stratégiques, il est donc vivement déconseillé de l'utiliser dans un contexte de production et de la réserver pour un usage de débogage) :

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

4 Formulaire

4.1 Création du formulaire

Réalisez un formulaire HTML contenant les champs suivants :

Champ	Type
Identifiant	zone de saisie textuelle (`<input type="text">`)
Civilité (sexe)	boutons radio (`<input type="radio">`)
E-mail	zone de saisie textuelle (`<input type="text">`)
Mot de passe	mot de passe (`<input type="password">`)
Vérification de mot de passe	mot de passe (`<input type="password">`)
Téléphone	zone de saisie textuelle (`<input type="text">`)
Pays	liste de choix (`<select></select>`)
Conditions générales	case à cocher (`<input type="checkbox">`)
Bouton de soumission	soumission (`<input type="submit">`)

Remplissez la liste de choix du champ de pays avec seulement quelques noms de pays.

Le bouton de soumission est pour le moment fictif (il ne fait aucune action).

4.2 Traitement du formulaire

Dans le formulaire, complétez la balise `<form></form>` de la manière suivante :

```
<form action="data-processing.php" method="post">
```

Puis modifiez le bouton de soumission en lui donnant le nom « action » et la valeur « mailer ».

Créez ensuite le fichier `data-processing.php`. Dans ce fichier récupérez les variables de votre formulaire puis testez si la valeur du bouton pressé est bien « mailer » :

```
<?php
    if($action === 'mailer')
    {

    }
    else
    {
        echo '<br><strong>Bouton non géré !</strong><br>';
    }
```

 (cc) BY-NC-SA

```
?>
```

Si l'action est celle voulue, initialisez une variable (avec l'opérateur de concaténation) qui contiendra le contenu de *l'e-mail*. N'hésitez pas à sauter des lignes ou mettre des tabulations (`\t`), par exemple :

```
<?php
    $message = 'Voici vos identifiants d\'inscription :' . "\n";
    $message .= 'Email : ' . $email . "\n";
    $message .= 'Mot de passe : ' . "\n" . $password;
?>
```

Ensuite, envoyez *l'e-mail* (avec la fonction `mail()`). Pensez qu'elle n'affiche rien à l'écran et qu'il plus convivial d'avoir un lien pour revenir au sommaire et d'afficher un message du type : `Votre mail a bien été envoyé.`

Enfin, modifier l'action du formulaire afin qu'une soumission de ce dernier recharge la page Web courante tout en conservant le fichier `data-processing.php` mais dorénavant en l'incluant (`require 'data-processing.php';`) à la page Web du formulaire et en le modifiant en conséquence.

5 Base de données

5.1 Création de la base de données

En utilisant l'application Web **phpMyAdmin**, créez une table `user` avec un champ dans la table pour chaque champ utile du formulaire, plus un champ `date`. En outre, il est nécessaire de créer un identifiant `id` de type `INT` ne pouvant être `NULL`, ayant comme attribut `AUTO_INCREMENT` et clef primaire de la table. Utilisez le type `VARCHAR` pour les champs textuels (en n'oubliant pas de fixer la taille), `INTEGER` pour les champs numériques, `INTEGER` ou `VARCHAR` pour les listes d'options ou cases à cocher et `DATE` pour les champs de date. Enfin, entrez quelques enregistrements de tests dans la base.

5.2 Communication avec MySQL

Il est utile de toujours faire apparaître les erreurs de manière à les connaître, tout particulièrement avec une procédure asynchrone comme celle d'une connexion à une base de données.

Créez un fichier `base.php` qui ouvre une connexion au serveur de la base de données (avec le nom du serveur, le nom d'utilisateur et le mot de passe) :

```
<?php
    $dbLink = mysqli_connect(dbHost, dbLogin, dbPass)
        or die('Erreur de connexion au serveur : ' . mysqli_connect_error());
?>
```

Sélectionnez ensuite votre base de données :

```
<?php
    mysqli_select_db($dbLink , dbBd)
        or die('Erreur dans la sélection de la base : ' . mysqli_error($dbLink)
);
?>
```

 (cc) BY-NC-SA

Une fois ces deux opérations réussies, vous pouvez écrire les requêtes SQL de votre choix (grâce à des chaînes de caractères) :

```
<?php
    $query = 'SELECT id, email, date FROM user';
?>
```

Envoyez une requête à la base de données en utilisant la fonction `mysql_query(…)` :

```
<?php
    if(!($dbResult = mysqli_query($dbLink, $query)))
    {
        echo 'Erreur de requête<br>';
        // Affiche le type d'erreur.
        echo 'Erreur : ' . mysqli_error($dbLink) . '<br>';
        // Affiche la requête envoyée.
        echo 'Requête : ' . $query . '<br>';
        exit();
    }
?>
```

L'instruction `$dbRow = mysqli_fetch_assoc($dbResult));` stocke la réponse récupérée sous forme de tableau associatif dans la variable `$dbRow` et renvoie vrai si il y a au moins un enregistrement retourné. Si on relance cette instruction, `$dbRow` contiendra l'enregistrement suivant de sorte qu'une boucle de ce type permette de parcourir tous les enregistrements.

Récupérez le résultat de la requête, enregistrement après enregistrement :

```
<?php
    while($dbRow = mysqli_fetch_assoc($dbResult))
    {
        …
    }
?>
```

Dans la boucle, affichez les différents champs du résultat (par exemple séparés par un saut de ligne) :

```
<?php
    echo $dbRow['id'] . '<br>';
    echo $dbRow['email'] . '<br>';
    echo $dbRow['date'] . '<br>';
    echo '<br><br>';
?>
```

Remarque

Il est possible de mettre la date au format français avec une instruction du type :

```
<?php
    echo date('d.m.Y', strtotime($dbRow['date']));
?>
```

5.3 Traitement d'un formulaire avec MySQL

En SQL, les dates sont au format ISO, soit du genre : *« 2024-10-21 »*.

Dans le fichier `data-processing.php` il faut tout d'abord se connecter au serveur puis sélectionner la base de données. Une fois ces deux opérations réussies, insérez un nouvel enregistrement en base

 (cc) BY-NC-SA

de données (la valeur de la variable `$today` n'est pas renseignée par le formulaire, vous pouvez l'obtenir avec une instruction du type : `$today = date('Y-m-d');`) :

```
<?php
    $query = 'INSERT INTO user (date, email …) VALUES (\'' . $today . '\', \''
. $email . '\', ' . … . ')';
?>
```

Envoyez la requête au serveur :

```
<?php
    if(!($dbResult = mysqli_query($dbLink, $query)))
    {
        echo 'Erreur dans requête<br >';
        // Affiche le type d'erreur.
        echo 'Erreur : ' . mysqli_error($dbLink) . '<br>';
        // Affiche la requête envoyée.
        echo 'Requête : ' . $query . '<br>';
        exit();
    }
?>
```

Aucune de ces instructions ne provoque d'affichage (même si l'opération se déroule correctement, il n'y aura que l'affichage d'une page blanche). Améliorez cette « interface » en affichant un texte à l'écran du genre :

```
Bonjour, Mickaël
Votre inscription a bien été enregistrée, merci.
```

Enfin, testez ce formulaire en vérifiant que les inscriptions sont bien enregistrées en base de données.

Remarques

Le champ `id` n'est pas renseigné dans la requête car il est de type `AUTO_INCREMENT` : il prendra donc une valeur automatiquement.

Vous avez utilisé du PHP pour obtenir la date du jour mais comme vous programmez conjointement en SQL et en PHP, vous pourriez aussi utiliser les fonctions SQL (ce qui est d'ailleurs préférable). Remplacez `$today` dans la requête par `NOW()` et constatez que le résultat est identique. Cette remarque est aussi valable pour la conversion de date au format français.

6 Page d'authentification

6.1 Mot de passe simple dans le code source

Dans un nouveau fichier `login.php`, réalisez un formulaire composé des champs `Login` et `Mot de passe` (`<input type="password">`) ainsi que d'un bouton `ok` qui doit lancer un appel au fichier `test-pass.php`.

Éditez le fichier `test-pass.php` de sorte de récupérer la valeur des champs `Login` et `Mot de passe`. Si l'ensemble est correct (avec les valeurs des identifiants de connexion dans une nouvelle table `user` de la base de données), chargez une page Web de bienvenue, sinon affichez une page Web contenant un message du genre (avec la redirection `header('Location: page2.php');`) :

 (cc) BY-NC-SA

```
Erreur de login ou de mot de passe.
```

Attention
Lorsque vous utilisez une commande de redirection, il ne doit rien y avoir avant, pas même la balise ouvrante `<html>` (ou un saut de ligne). Exemple :

```
<?php
    if (condition)
    {
        header(…);
        exit() ;
    }
    else
    {
        start_page('erreur') ;
        …
    }
?>
```

6.2 Retour au formulaire avec un message d'erreur

Pour l'instant, lorsque l'utilisateur entre des identifiants erronés, il arrive sur une page d'erreur et il doit cliquer sur le bouton précédent de son navigateur Web pour réessayer.

Renvoyez-le automatiquement sur la page de connexion avec une redirection en cas d'erreur d'authentification après avoir affecté une valeur à `$_SESSION['error']`. Puis, modifiez le fichier `login.php` en récupérant `$_SESSION['error']` et en l'affichant en cas d'erreur de connexion.

6.3 Identifiant de session

L'inconvénient des méthodes ci-dessus est que le mot de passe empêche de passer à la page « protégée » mais que si l'on connaît son URL, rien ne nous empêche d'accéder à la page directement.

Vous pouvez régler ce problème en testant le mot de passe directement dans cette page, sans page intermédiaire. Ainsi, même si l'on connaît son URL, il faut quand même connaître le mot de passe. Cette solution assez correcte possède cependant une limitation : elle n'est valable que si l'on a qu'une seule page à protéger. C'est à cela, entre autres, que sert un identifiant de session. Le serveur a la capacité de reconnaître un utilisateur, c'est ce qu'il se passe lorsque vous faites des achats sur un site marchand par exemple : vous mettez un article dans un panier, puis en continuant de parcourir le site Web, vous en mettez un autre qui s'ajoute bien dans votre panier.

Pour cela, on va utiliser une variable « superglobale » valable pour une session (un utilisateur), qui vérifiera l'identification.

Dans `test-pass.php`, commencez par créer une session :

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

Puis si le mot de passe est correct, modifiez la valeur de la variable « superglobale » `$_SESSION[…]` (variable définie à chaque fois que la fonction `session_start(…)` est appelée) :

```
<?php
    $_SESSION['suid'] = session_id();
```

```
?>
```

Dans chaque fichier où vous voulez n'autoriser que des utilisateurs authentifiés, testez l'authentification :

```
<?php
    session_start();
    if(isset($_SESSION['suid']))
    {
        die('Erreur d\'authentification');
    }
    else
    {
        …
    }
?>
```

Utilisez les liens hypertextes sous cette forme (`<?php echo SID ;?>` permet de passer en argument l'identifiant de la session) :

```
<a href="page2.php?<?php echo SID ;?>">par ici</a>
```

Modifiez les fichiers concernés par l'authentification pour qu'ils prennent en compte l'identifiant de session.