

Afin de transmettre des variables de pages en pages, plusieurs possibilités s'offrent à vous :

- les divers champs des formulaires, qu'ils soient hidden ou non.
- passer les variables directement à travers les liens.
- utiliser les cookies.
- utiliser les sessions.

Cependant, toutes ces possibilités n'offrent pas le même niveau de sécurité. En effet, certaines de ces possibilités sont vraiment pratiquées dans leurs modes d'utilisation (comme les cookies par exemple mais tout le monde n'est pas obligé d'accepter les cookies), ce qui implique, dans la majorité des cas, un bas niveau de sécurité (cas du passage des variables par les liens, ce qui implique que les variables seront visibles de tout le monde).

De même, faire dans chaque page un formulaire contenant des champs hidden permettant de faire circuler les différentes variables à travers toutes les pages du site n'est pas vraiment pratique.

C'est pourquoi, dans tous ces cas où la sécurité de vos données est primordiale, vous devrez utiliser les sessions qui vous permettront de faire circuler différentes variables (comme un mot de passe par exemple) à travers les pages de votre site, tout en étant assez confortables à l'emploi. Pour utiliser les sessions, différentes fonctions PHP s'offrent à nous. Voici déjà un petit tableau récapitulatif des précédentes fonctions liées aux sessions :

Fonction	Signification
<code>session_start</code>	Démarre une session
<code>session_register</code>	Enregistre une variable de session
<code>session_unregister</code>	Efface une variable de session
<code>session_is_registered</code>	Vérifie si une variable est déclarée pour la session en cours
<code>session_id</code>	Retourne l'id de la session en cours
<code>session_name</code>	Retourne le nom de la session en cours
<code>session_unset</code>	Detruit toutes les variables de la session en cours
<code>session_destroy</code>	Detruit la session en cours

Afin de voir concrètement comment fonctionnent les sessions, prenons alors un exemple simple :

- imaginons que notre site possède une section membre où chaque membre devra se loguer avant de pouvoir y entrer.
- de plus, on aimerait bien être sûr qu'il s'agisse toujours de ce même membre qui est connecté.

On aura alors une page contenant un formulaire permettant à notre visiteur de se connecter à une section membre (page **index.html**) :

#### exemple1.php

```
1.  <html> <head> <title>Formulaire d'identification</title> </head>
2.  <body>
3.      <form action="login.php" method="post">
4.          Votre login : <input type="text" name="login">
5.          <br />
6.          Votre mot de passé : <input type="password" name="pwd"><br />
7.          <input type="submit" value="Connexion">
8.      </form>
9.  </body>
10. </html>
```

D'après cette page, vous pouvez remarquer que lorsque le visiteur le remplira et qu'il cliquera sur le bouton de connexion, on se retrouvera au niveau de la page *login.php* avec une variable \$pseudo qui contiendra le login de notre visiteur ainsi qu'une variable \$pwd contenant son mot de passe ; variables qu'il faudra naturellement tester avant de démarrer notre session (car seuls les membres pourront accéder à notre espace membre, espace où l'on utilisera notre session).

On aura alors par exemple (page **login.php**) :

#### exemple2.php

```
11. <?php
12. // On définit un login et un mot de passe de base
13. $login_valide = "moi";
14. $pwd_valide = "lemien";
15.
16. // on teste si nos variables sont définies
17. if (isset($_POST['login']) && isset($_POST['pwd'])) {
18.     // on vérifie les informations saisies
19.     if ($login_valide == $_POST['login'] && $pwd_valide == $_POST['pwd']) {
20.         session_start ();
21.         // on enregistre les paramètres de notre visiteur comme variables
de session ($login et $pwd) (
22.             $_SESSION['login'] = $_POST['login'];
23.             $_SESSION['pwd'] = $_POST['pwd'];
24.             // on redirige notre visiteur vers une page de notre section membre
25.             header ('location: page_membre.php');
26.         }
27.         else {
28.             echo '<body onLoad="alert(\'Membre non reconnu...\')">';
29.             // puis on le redirige vers la page d'accueil
30.             echo '<meta http-equiv="refresh" content="0;URL=index.htm">';
31.         }
32.     } else {
33.         echo 'Les variables du formulaire ne sont pas déclarées.';
34.     }
35. ?>
```

Remarquez également que nous utilisons notre *session\_start* avant tout code HTML.

Voyons alors le code de la page de notre section membre, la page **page\_membre.php**. On a :

#### exemple3.php

```
36. <?php
37. // On démarre la session (ceci est indispensable dans toutes les pages de
notre section membre)
38. session_start ();
39.
40. // On récupère nos variables de session
41. if (isset($_SESSION['login']) && isset($_SESSION['pwd'])) {
```

```

42.
43.         // On teste pour voir si nos variables ont bien été enregistrées
44.         echo '<html>';
45.         echo '<head>';
46.         echo '<title>Page de notre section membre</title>';
47.         echo '</head>';
48.
49.         echo '<body>';
50.         echo 'Votre login est ' . $_SESSION['login'] . ' et votre mot de passe
est ' . $_SESSION['pwd'] . ' .';
51.         echo '<br />';
52.
53.         // On affiche un lien pour fermer notre session
54.         echo '<a href="./logout.php">Déconnection</a>';
55.     }
56.     else {
57.         echo 'Les variables ne sont pas déclarées.';
58.     }
59.     ?>

```

Voyons alors le code de la page permettant au membre de se déconnecter (la page **logout.php**). On aura alors :

#### **exemple4.php**

```

60. <?php
61. // On démarre la session
62. session_start ();
63.
64. // On détruit les variables de notre session
65. session_unset ();
66.
67. // On détruit notre session
68. session_destroy ();
69.
70. // On redirige le visiteur vers la page d'accueil
71. header ('location: index.htm');
72. ?>

```

Résumons alors tout ce que nous venons de voir :

- chaque session à un id différent (ce qui permet d'éviter la confusion entre les connexions).
- à chaque page où notre session doit être active, on doit placer un *session\_start* en tout début de page (avant tout code HTML).
- toutes les variables enregistrées au cours de notre session, seront accessibles dans les pages de notre session.
- n'oubliez JAMAIS de détruire vos variables de session lors de la déconnexion.

#### **Exercice 1 :**

Reprendre l'exercice sur la génération de graphique à partir du contenu de la BD. Ajouter une page d'authentification et empêcher l'accès 'direct' à la page générant le schéma (i.e. si un utilisateur saisi l'URL de la page générant le graphique, l'accès doit être interdit et on est renvoyé vers la page d'authentification). Sur la page proposant le graphique, ajouter un lien permettant la déconnexion.

**Remarque :** dans la vraie vie, les variables de session sont également cryptées.