1) Comment utiliser la Session ?

- Initialiser la session

- Données (Variables)

Définition :

C'est un espace de stockage que l'on va initialiser et qu'on va pouvoir stocker toutes les infos dont on va avoir besoin pour le bon fonctionnement du site.

La session c'est un tableau qui s'appelle $\_session.

Voyons maintenant un tutorial vraiment important pour la **sécurité** de vos scripts : **les sessions**.  
  
En effet, 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 pratiquent 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 vous permettant de vous familiariser avec ces différentes fonctions (que nous détaillerons bien sur dans la suite de ce tutorial) :

|  |  |
| --- | --- |
| **Fonction** | **Signification** |
| **session\_start** | Démarre une session |
| **session\_register** | Enregistre une variable de session |
| **session\_unregister** | Efface une variable de session |
| **session\_is\_registered** | Vérifie si une variable est déclarée pour la session en cours |
| **session\_id** | Retourne l'id de la session en cours |
| **session\_name** | Retourne le nom de la session en cours |
| **session\_unset** | Detruit toutes les variables de la session en cours |
| **session\_destroy** | Destruit la session en cours |

Sachez également qu'ils existent d'autres fonctions agissant sur les sessions.  
En revanche, étant donné qu'elles ne sont pas nécessaires à la compréhension de notre tutorial, nous ne les détaillerons pas ici (cf. la documentation pour de plus amples informations).  
Cependant, ces fonctions commencent toujours par **session**.  
  
  
**Attention !!!**  
  
Avant d'aller plus loin dans le tutorial, vous devez savoir que les sessions ne sont accessibles qu'à partir de **PHP 4**.  
Cependant, je vous rassure, la plupart des hébergeurs sont aujourd'hui fait évoluer leur PHP en PHP 4.  
  
  
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 logué avant de pouvoir y entrer.
* de plus, on aimerait bien être sur 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.htm**) :

exemple1.php

1. <html>
2. <head>
3. <title>Formulaire d'identification</title>
4. </head>
5. <body>
6. <form action="login.php" method="post">
7. Votre login : <input type="text" name="login">
8. <br />
9. Votre mot de passé : <input type="password" name="pwd"><br />
10. <input type="submit" value="Connexion">
11. </form>
12. </body>
13. </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 naturrellement 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   1. <?php 2. *// On définit un login et un mot de passe de base pour tester notre exemple. Cependant, vous pouvez très bien interroger votre base de données afin de savoir si le visiteur qui se connecte est bien membre de votre site* 3. $login\_valide = "moi"; 4. $pwd\_valide = "lemien"; 5. *// on teste si nos variables sont définies* 6. if ([isset](http://www.lephpfacile.com/manuel-php/function.isset.php)($\_POST['login']) && [isset](http://www.lephpfacile.com/manuel-php/function.isset.php)($\_POST['pwd'])) { 7. *// on vérifie les informations du formulaire, à savoir si le pseudo saisi est bien un pseudo autorisé, de même pour le mot de passe* 8. if ($login\_valide == $\_POST['login'] && $pwd\_valide == $\_POST['pwd']) { 9. *// dans ce cas, tout est ok, on peut démarrer notre session* 10. *// on la démarre :)* 11. [session\_start](http://www.lephpfacile.com/manuel-php/function.session-start.php) (); 12. *// on enregistre les paramètres de notre visiteur comme variables de session ($login et $pwd) (notez bien que l'on utilise pas le $ pour enregistrer ces variables)* 13. $\_SESSION['login'] = $\_POST['login']; 14. $\_SESSION['pwd'] = $\_POST['pwd']; 15. *// on redirige notre visiteur vers une page de notre section membre* 16. [header](http://www.lephpfacile.com/manuel-php/function.header.php) ('location: page\_membre.php'); 17. } 18. else { 19. *// Le visiteur n'a pas été reconnu comme étant membre de notre site. On utilise alors un petit javascript lui signalant ce fait* 20. echo '<body onLoad="alert(\'Membre non reconnu...\')">'; 21. *// puis on le redirige vers la page d'accueil* 22. echo '<meta http-equiv="refresh" content="0;URL=index.htm">'; 23. } 24. } 25. else { 26. echo 'Les variables du formulaire ne sont pas déclarées.'; 27. } 28. ?>   Remarquer également que nous utilisons notre **session\_start** avant tout code HTML. |

Remarquer 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

1. <?php
2. *// On démarre la session (ceci est indispensable dans toutes les pages de notre section membre)*
3. session\_start ();
4. *// On récupère nos variables de session*
5. if ([isset](http://www.lephpfacile.com/manuel-php/function.isset.php)($\_SESSION['login']) && [isset](http://www.lephpfacile.com/manuel-php/function.isset.php)($\_SESSION['pwd'])) {
6. *// On teste pour voir si nos variables ont bien été enregistrées*
7. echo '<html>';
8. echo '<head>';
9. echo '<title>Page de notre section membre</title>';
10. echo '</head>';
11. echo '<body>';
12. echo 'Votre login est '.$\_SESSION['login'].' et votre mot de passe est '.$\_SESSION['pwd'].'.';
13. echo '<br />';
14. *// On affiche un lien pour fermer notre session*
15. echo '<a href="./logout.php">Déconnection</a>';
16. }
17. else {
18. echo 'Les variables ne sont pas déclarées.';
19. }
20. ?>

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

exemple4.php

1. <?php
2. *// On démarre la session*
3. session\_start ();
4. *// On détruit les variables de notre session*
5. [session\_unset](http://www.lephpfacile.com/manuel-php/function.session-unset.php) ();
6. *// On détruit notre session*
7. [session\_destroy](http://www.lephpfacile.com/manuel-php/function.session-destroy.php) ();
8. *// On redirige le visiteur vers la page d'accueil*
9. [header](http://www.lephpfacile.com/manuel-php/function.header.php) ('location: index.htm');
10. ?>

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.

En respectant ces règles vous pourrez très rapidement faire vous-même votre espace membre, voir même pourquoi pas une boutique en ligne :)

- Destruction

2) Comment réceptionner un formulaire ?

HTML Page.php

Form

|  |
| --- |
|  |
|  |

OK

Selon la méthode - POST

- GET

$\_POST['Login']

$\_GET['Nom']