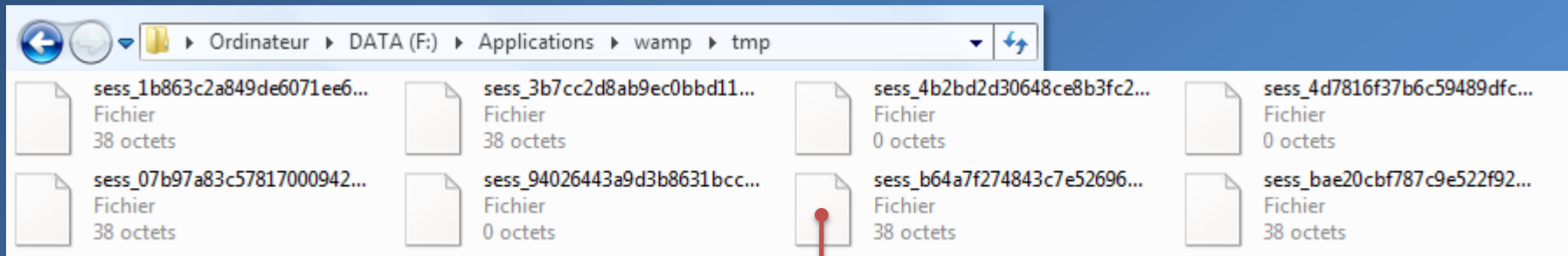


PHP

Les sessions

Les sessions

- Une session est personnelle à chaque visiteur et permet de stocker temporairement des variables.
- Les données de session sont stockées dans un fichier sur le serveur : *on ne peut pas y avoir accès du côté client par le biais d'un script JavaScript.*



```
monlogin|s:4:"titi";monpwd|s:4:"toto";
```

Enfin, une session c'est un peu comme un cookie mais coté serveur.

A quoi ça sert une session ?

- Pister un utilisateur pour établir des statistiques de navigation
- Transmettre des données de page en page comme un panier
- Stocker les données d'identification tels que le login et le mot de passe crypté

PHP – Les sessions

MÉCANISMES

Démarrer une session

- Automatiquement :
si l'option `session.auto_start` est à 1.
- Explicitement :
en utilisant l'instruction `session_start()` qui permet de créer une nouvelle session, ou restaurer une session existante.



Si les données sont transmises par le biais de cookies, `session_start()` doit être appelée avant d'afficher quoi que ce soit !!!
Attention aux pages qui commencent par une ligne vide.

Durée de vie d'une session

- La durée de vie d'une session dépend de trois paramètres :
 - La fréquence à laquelle va être vérifié sa validité

```
; Define the probability that the 'garbage collection' process is started  
; on every session initialization.  
; The probability is calculated by using gc_probability/gc_divisor,  
; e.g. 1/100 means there is a 1% chance that the GC process starts  
; on each request.  
session.gc_probability = 1  
session.gc_divisor     = 100
```

Ici 1/100 cad 1% vérification 1 fois sur 100 de la validité de la session

- Sa durée maximum d'existence (*en nombre de secondes*)

```
; After this number of seconds, stored data will be seen as 'garbage' and  
; cleaned up by the garbage collection process.  
session.gc_maxlifetime = 1440
```

- Au delà de ce temps, les données de la session seront considérées comme périmées.

Manipulation de données dans une session

- Les données d'une session sont stockée dans la super globale `$_SESSION[...]`
- Pour ajouter, tester l'existence ou supprimer une donnée dans une session, on procède donc de la même manière qu'avec un tableau associatif

```
<?php
    session_start();                //Création/Récupération de la session
    $_SESSION['nom'] = $_REQUEST['nom']; //Création de la variable "nom" dans la session

    if (!isset($_SESSION['count'])) { //Test d'existence de la variable "count"
        $_SESSION['count'] = 0;       //Création de la variable "count"
    } else $_SESSION['count']++;      //Incrémentation de la variable "count"
?>
```

```
<?php
    session_start();                //Récupération de la session et destruction de la
    unset($_SESSION['nom']);         //variable "nom" : on ne peut donc plus y accéder
?>
```

Détruire une session

- Vider le contenu de la session :
la session est toujours accessible mais elle est vide.

```
<?php
session_start(); //on démarre la session
$_SESSION=array(); // $_SESSION est un tableau vide
?>
```

- Détruire la session :
la session n'est plus accessible.

```
<?php
session_start();//on démarre la session
if (session_destroy()) echo 'session détruite';
else echo 'ERREUR : impossible de détruite la session'
?>
```

- Régénérer l'identifiant de session :
L'identifiant est modifié mais les données ne sont pas effacées.

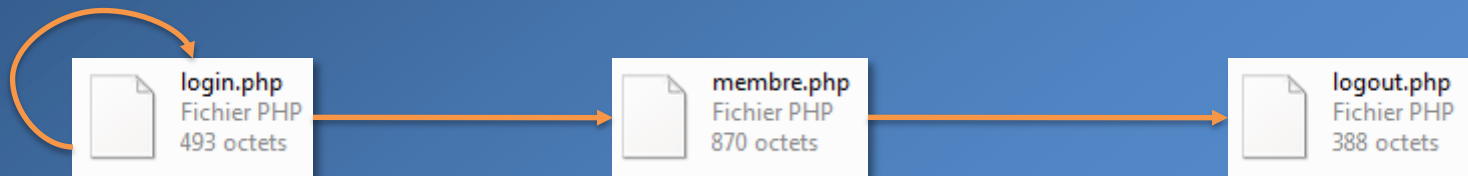
```
<?php
session_start(); //on démarre la session
session_regenerate_id(); // regenère l'id de session sans la détruire
?>
```


PHP – Les sessions

EXEMPLE

Création d'un espace membre

- Dans cet exemple, on utilise une session pour stocker les informations relatives à l'identification d'un utilisateur et les transmettre de page en page.
- L'architecture est la suivante :



login.php (page d'authentification)

```
1 <?php
2 $error='';
3 if (isset($_POST['envoyer'])) // si bouton connexion on traite le formulaire
4 {
5     $login_valid='laurent';
6     $mdp_valid='gr45tu';
7     if (($_POST['login']==$login_valid) && ($_POST['pwd']==$mdp_valid))
8     {
9         session_start(); // démarre la session
10        $_SESSION['log']=$login_valid; // création de variable de session
11        $_SESSION['mdp']=$mdp_valid;
12        // redirection vers la page membre
13        header('location:membre.php');
14    }
15    else $error= '<p style="color:red">Erreur de login ou mot de passe</p>';
16 }
17 ?>
18 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
19 <head>
20     <title>Page login</title>
21     <meta http-equiv="content-type" content="text/html; charset=utf-8" />
22 </head>
23 <body>
24 <?php // sinon on affiche le formulaire
25 echo $error;
26 echo '<form action="login.php" method="post">
27     <p>Login : <input type="text" name="login" size="35" /><br />
28     Mot de passe : <input type="password" name="pwd" size="16" /><br />
29     <input type="submit" name="envoyer" value="Connexion" /></p>
30 </form>';
31 ?>
32 </body>
33 </html>
```

Vérification du login et mot de passe si le formulaire est posté
Si l'utilisateur est valide, on le redirige vers la page membre.

Sinon on affiche le formulaire

Test du fichier login.php

```
1 <?php
2 $error='';
3 if (isset($_POST['envoyer'])) // si bouton connexion on traite le formulaire
4 {
5     $login_valid='laurent';
6     $mdp_valid='gr45tu';
7     if (($_POST['login']==$login_valid) && ($_POST['pwd']==$mdp_valid))
8     {
9         session_start();// démarre la session
10        $_SESSION['log']=$login_valid; // création de variable de session
11        $_SESSION['mdp']=$mdp_valid;
12        // redirection vers la page membre
13        header('location:membre.php');
14    }
15    else $error= '<p style="color:red">Erreur de login ou mot de passe</p>';
16 }
17 ?>
18 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
19 <head>
20     <title>Page login</title>
21     <meta http-equiv="content-type" content="text/html; charset=utf-8" />
22 </head>
23 <body>
24 <?php // sinon on affiche le formulaire
25     echo $error;
26     echo '<form action="login.php" method="post">
27         <p>Login : <input type="text" name="login" size="35" /><br />
28         Mot de passe : <input type="password" name="pwd" size="16" /><br />
29         <input type="submit" name="envoyer" value="Connexion" /></p>
30     </form>';
31 ?>
32 </body>
33 </html>
```

En cas d'erreur sur le login ou mot de passe

Erreur de login ou mot de passe

Login :

Mot de passe :

membre.php

```
1 <?php
2 //on démarre la session
3 session_start();
4
5 // traitement d'une variable de session pour savoir si l'utilisateur
6 // est connecté
7
8 if (!isset($_SESSION['log'])) // si la session n'existe pas
9     header("location:login.php"); // retour vers la page de connexion
10
11 // sinon on continue le script
12
13 .....
14 .....
15
16 // on peut placer un lien vers la déconnexion de la session
17 $sHtml.='<a href="./logout.php">Déconnexion</a>';
18
19 ?>
20
21 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
22     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
23 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
24 <head>
25     <title>Réponses</title>
26     <meta http-equiv="content-type" content="text/html; charset=utf-8" />
27     <meta http-equiv="Content-Style-Type" content="text/css" />
28 </head>
29 <body>
30     <?php
31         echo $sHtml //affiche le resultat
32     ?>
33 </body>
34 </html>
```

Dans la page membre, on récupère les données de session, redirection vers la page d'authentification si elle n'existe pas

Ajout d'un lien pour ce déconnecter.

```
<?php
//On démarre la session
session_start();

//On détruit toute les variables de notre session
//Rq : cette fonction n'est pas officiellement dépréciée mais
//on lui préférera $_SESSION[]=array();
session_unset();

//On détruit notre session
session_destroy();

//On redirige le visiteur vers la page d'accueil
header('location: login.html');
?>
```

- On récupère la session;
- Et la vide des données qu'elle contient.
- Enfin, on détruit la session et redirige vers la page de connexion.

Quelques notes sur cet exemple !

- **Il ne s'agit que d'un exemple pédagogique !
Je vous déconseille fortement de créer une
page de membre en stockant le login et le mot
de passe dans la session et en clair !!!**



- Oui mais alors ? Comment fait-on ?
 - Tout d'abord, les données des utilisateurs devraient être stockées dans une base de données.
 - Le mot de passe, devrait être crypté par un algorithme de hachage tels que MD5 ou SHA1 (A préférer) et ceux, côté serveur (php) et côté client (javascript). Vous pouvez également utiliser la technique du *grain de sel* (*salt*) pour vous protéger des dictionnaires de mots de passe.
 - Il est également possible de créer une table session dans la base de données pour associer à l'utilisateur son identifiant de session.
 - ...