



Exercice 1

Garder la mémoire

Le protocole HTTP est très déficient au niveau de la mémoire des actions précédentes. L'un des moyens de lui faire conserver des informations en mémoire est de les passer sur l'URL. Pouvez-vous ainsi passer le login d'un utilisateur et son nom sur l'URL d'une page1.php à une page2.php ?

Exercice 2

Champs cachés

Faire de même mais en utilisant un champ caché HTML.

Code 1 – Compteur avec cookies

```
<?php
if (isset($_COOKIE['compteur'])) {
    $message = "Vous êtes déjà venu ". $_COOKIE['compteur']. " fois<BR>\n";
    $valeur = $_COOKIE['compteur'] + 1;}
else { $message = "Je vous met un petit cookie<BR>\n";
    $valeur = 1;}
setCookie("compteur", $valeur);
echo $message;
```

Exercice 3

Cookies

Garder une information sur le login et le nom de l'utilisateur en positionnant une cookie avec le login et une autre avec le nom de l'utilisateur. Ces cookies sont disponibles par le biais du tableau PHP \$_COOKIE[]\$. Récupérez ces infos sur une autre page PHP.

Code 2 – authentification avec session

```
<?php session_start();
if (!isset($_SESSION['login'])) { ?>
<form action='authen.php' method="GET">
<p>Login:<input type="text" name="login">
<p>Pass:<input type="password" name="pass">
<p><input type="submit"></form>
if(isset($_GET['login'])) {
    if ((($_GET['login']=='toto') && ($_GET['pass']=='titi')) {
        $_SESSION['login']="toto";
```

```
header('Location: authen.php');}
else { echo "Autorisation refusee<br>";
      echo "<a href='authen.php'>Recommencez !</a>";} }}
else {
echo "<br>bonjour " . $_SESSION['login'];
echo "<a href='suite.php'>Entrez</a>"; }
```

Exercice 4

Sessions

Les sessions utilisent le tableau associatif \$_SESSION[]. Réalisez une authentification simple avec les sessions puis transformez-la pour vérifier les logins/passwd dans une table de base de données. Pensez à hasher les mots de passe. Quelle fonction PHP permet de hasher un mot de passe ?