

TP PHP

Notions fondamentales de PHP : envoi de fichiers, les cookies et les sessions

Créé par : Neila Ben Lakhal

TP PHP

Notions fondamentales de PHP : envoi de fichiers, les cookies et les sessions

Exercice 1.

- Créer un formulaire de saisie des deux codes couleur préférés du visiteur du site pour la couleur de fond et le texte de la page. Les enregistrer dans deux cookies valables deux mois. À l'ouverture de la page d'accueil, récupérer ces valeurs et créer un style utilisant ces données.
- L'utilisateur a le droit de changer ses préférences à chaque fois qu'il accède à la page.

```
<?php
$expir=time() + 2*30*24*3600;
#first time the page is loaded, no cookies set
if (!isset($_POST['fond']) AND !isset($_POST['texte']) AND !isset($_COOKIE['fond']) AND
!isset($_COOKIE['texte']))
{
echo "veuillez choisir une couleur de texte et de l'arrière plan !!!";
}
#le formulaire est rempli mais pas de cookies existantes
if (isset($_POST['fond']) AND isset($_POST['texte']) AND !isset($_COOKIE['fond']) AND
!isset($_COOKIE['texte']))
{
$fond=$_POST['fond'];
$texte=$_POST['texte'];
setcookie("fond",$fond,$expir);
setcookie("texte",$texte,$expir);
}
#le formulaire est rempli, et les cookies existent, il faut MAJ les cookies
if (isset($_POST['fond']) AND isset($_POST['texte']) AND isset($_COOKIE['fond']) AND
isset($_COOKIE['texte']))
{
$fond=$_POST['fond'];
$texte=$_POST['texte'];
setcookie("fond",$fond,$expir);
setcookie("texte",$texte,$expir);
}
#formulaire vide, mais cookies existent: l'utilisateur a fait des choix déjà
if (!isset($_POST['fond']) AND !isset($_POST['texte']) AND isset($_COOKIE['fond']) AND
isset($_COOKIE['texte']))
{
$fond=$_COOKIE['fond'];
$texte=$_COOKIE['texte'];
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
```



```

$fond=$_COOKIE['couleur'][$'fond'];
$texte=$_COOKIE['couleur'][$'texte'];
}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Couleurs du site</title>
<style type="text/css" >

body{background-color: <?php echo $fond ?> ; color: <?php echo $texte ?> ;}
legend{font-weight:bold;font-family:cursive;}
label{font-weight:bold;font-style:italic;}

</style>
</head>
<body>
<form method="post" action="cookiefavoritecolors.php">
<fieldset>
<legend>Choisissez vos couleurs</legend>
<label>Couleur de fond
<input type="text" name="fond" />
</label><br/><br/>
<label>Couleur de texte
<input type="text" name="texte" />
</label><br />
<input type="submit" value="Envoyer" />&nbsp;&nbsp; 
<input type="reset" value="Effacer" />
</fieldset>
</form>
</body>
</html>

```

Exercice 3.

- Créer un formulaire n'effectuant que le transfert de fichiers ZIP et d'une taille limitée à 1 Mo. Le script affiche le nom du fichier du poste client ainsi que la taille du fichier transféré et la confirmation de réception.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post"
enctype="multipart/form-data">
<fieldset>
<legend><b>Transférez un fichier pdf </b></legend>
<table border="0" >
<tr>
<td>Choisissez un fichier </td>
<td><input type="file" name="fich" /></td>
<td><input type="hidden" name="MAX_FILE_SIZE" accept="application/pdf"
value="1000000"/></td>
</tr>

```

```

<tr>
<td>&nbsp;</td>
<td><input type="submit" value="ENVOI" /></td>
</tr>
</table>
</fieldset>
</form>
</body>
</html>
<?php
if(isset($_FILES['fich']))
{
if($_POST["MAX_FILE_SIZE"] <$_FILES["fich"]["size"])
{
echo "<b>Taille trop grande </b><hr />";
echo "Taille maximale autorisée :",$_POST["MAX_FILE_SIZE"],"
octets<hr />";
echo "Taille du fichier transféré :",$_FILES["fich"]["size"],"
octets<hr />";
}
else
{
//Enregistrement et renommage du fichier
$result=move_uploaded_file($_FILES["fich"]["tmp_name"],$_FILES["fich"]["name"]);
if($result==TRUE)
{
echo "<b>Vous avez bien transféré le fichier</b><hr />";
echo "Le nom du fichier est : ",$_FILES["fich"]["name"],"<hr
/>";
echo "Votre fichier a une taille de
",$_FILES["fich"]["size"],"<hr />";
}
else {echo "<hr /> Erreur de transfert
n°",$_FILES["fich"]["error"];}
}
}
?>

```

Exercice 4.

- Écrivez un script "ajout.php" qui démarre une session et affiche l'identifiant de cette session.
- Compléter le script en permettant à l'utilisateur d'enregistrer plusieurs variables dans la session (le nom de la variable ainsi que sa valeur seront saisies dans un formulaire).
- Affichez aussi les variables (et les valeurs) de la session courante.

Ajout.html

```

<html>
<body>
<p>ajout de variable de session</p>
<form action="ajout.php" method="post">
<table>
<tr>
<td>nom</td><td><input type="text" name="nom"/></td>
</tr>
<tr>
<td>valeur</td><td><input type="text" name="val"/></td>

```

```

</tr>
</table>
<input type="submit" value="ok"/>
</body>
</html>

```

Ajout.php

```

<?php
session_start();
//On récupère les informations de la session :
echo 'nom de la session : ', session_name() , '<br/>' ;
echo 'identifiant utilise : ', session_id() , '<br/>' ;
if (isset($_POST))//une nouvelle variable de session est à ajouter
{
    $nom=$_POST['nom'];
    $val=$_POST['val'];
    $_SESSION[$nom]=$val;
}
echo 'Actuellement les variables de la session sont :<br/>';
foreach($_SESSION as $key => $elem)
echo "$key = $elem<br/>";
?>

```

Exercice 5.

- Afin de tester les variables de session, commencez par créer la page menu.html ci-dessous :

```

<html><body>
<h1>Test de variables de session</h1>
<a href="ouvrir_session.php">Ouvrir une session</a><br/>
<a href="afficher_session.php">Afficher variables de session</a><br/>
<a href="detruire_session.php">Détruire variables de session</a><br/>
</body>
</html>

```

- Le but de cet exercice est de créer les pages **ouvrir_session.php**, **afficher_session.php** et **detruire_session.php**
- Créer le script **ouvrir_session.php** qui :
 - crée une nouvelle session si aucune session n'existe (un SID est engendré et transmis dans un cookie)
 - ou bien restaure la session en cours (connue par son identifiant de session SID)

Pour cela il faudra appeler en début de script la fonction PHP `session_start()`.

Afficher ensuite un formulaire permettant de saisir le nom et le prénom de l'utilisateur. Ces deux informations seront conservées dans des variables de session lorsque le formulaire est envoyé.

```
<?php
session_start();
?>
<html>
<body>
<h1>Ouvrir une session</h1>
<?php
echo "<br/>ID session: " . session_id() ;
echo "<br/>Nom session: " . session_name() ;
?>
<hr>
<form action="creer_session.php" method="get">
Nom: <input type="text" name="nom"/><br/>
Prénom: <input type="text" name="prenom"/><br/>
<input type="submit" value="Envoyer"/>
</form>
</body>
</html>
```

- Créer le script `creer_session.php` qui enregistre les données du formulaire en variables de session : ce script renverra une page html affichant les variables de session, un lien en bas de page permettra de retourner au menu principal.

```
<?php
session_start();
// enregistrer les variables de session
$_SESSION['nom'] = $_REQUEST['nom'] ;
$_SESSION['prenom'] = $_REQUEST['prenom'] ;
?>
<html>
<body>
<h1>Variables de session créées</h1>
<?php
echo "<br/>ID session: " . session_id();
echo "<br/>Nom session: " . session_name();
echo "<br/>Nom: " . $_SESSION['nom'];
echo "<br/>Prénom: " . $_SESSION['prenom'];
?>
<br><a href="menu.html">Retour menu</a>
</body>
</html>
```

- Créer le script `afficher_session.php` affichant les variables de session, vous ajouterez un compteur en variable de session afin d'afficher le nombre de fois que la page a été vue.

```
<?php
session_start();
// compte le nombre de fois que la page a été visitée
if (!isset($_SESSION['compteur'])) {
$_SESSION['compteur'] = 1;
} else {
$_SESSION['compteur']++;
}
?>
<html>
<body>
<h1>Affichage des variables de session</h1>
<?php
echo "Vous avez vu cette page " . $_SESSION['compteur'] . " fois<p>";
echo "<br/>ID session: " . session_id();
echo "<br/>Nom session: " . session_name();
echo "<br/>Nom: " . $_SESSION['nom'];
echo "<br/>Prénom: " . $_SESSION['prenom'];
echo "<hr/>";
print_r($_SESSION);
?>
<br><a href="menu.html">Retour menu</a>
</body>
</html>
```

- Créer le script `detruire_session.php` qui détruit les variables de session, utiliser la fonction PHP `session_destroy()`.

```
<?php
session_start();
$sid = session_id();
session_destroy();
?>
<html>
<body>
<h1>Détruire les variables de session</h1>
<?php
echo "<br/>Variables de la session $sid détruites !\n" ;
?>
<br><a href="menu.html">Retour menu</a>
</body>
```


Exercice 6.

- Reprendre l'exercice 1 en enregistrant les préférences du visiteur dans des variables de session pour afficher toutes les pages du site avec ses couleurs préférées.

```
<?php
session_start();
#first time the page is loaded, no session variables already stored
if (!isset($_POST['fond']) AND !isset($_POST['texte']) AND
    !isset($_SESSION['fond']) AND !isset($_SESSION['texte']))
{
    echo "veuillez choisir une couleur de texte et de l'arrière plan !!!";
}
#le formulaire est rempli mais pas de variables de sessions existantes
if (isset($_POST['fond']) AND isset($_POST['texte']) AND !isset($_SESSION['fond'])
    AND !isset($_SESSION['texte']))
{
    $fond=$_POST['fond'];
    $texte=$_POST['texte'];
    $_SESSION["fond"] = $fond;
    $_SESSION["texte"] = $texte;
}
#le formulaire est rempli, et les variables de sessions existent, il faut MAJ les
variables
if (isset($_POST['fond']) AND isset($_POST['texte']) AND isset($_SESSION['fond'])
    AND isset($_SESSION['texte']))
{
    $fond=$_POST['fond'];
    $texte=$_POST['texte'];
    $_SESSION["fond"] = $fond;
    $_SESSION["texte"] = $texte;
}
#formulaire vide, mais cookies existent: l'utilisateur a fait des choix déjà
if (!isset($_POST['fond']) AND !isset($_POST['texte']) AND
    isset($_SESSION['fond']) AND isset($_SESSION['texte']))
{
    $fond=$_SESSION['fond'];
    $texte=$_SESSION['texte'];
}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Couleurs du site</title>
<style type="text/css" >

body{background-color: <?php echo $fond ?> ; color: <?php echo $texte ?> ;}
legend{font-weight:bold;font-family:cursive;}
label{font-weight:bold;font-style:italic;}

</style>
```

[illegible]

Exercice 7.

CAS D'APPLICATION : AUTHENTIFICATION PAR FORMULAIRE

Un client vous a demandé de faire passer un de ses sites Internet à la vitesse supérieure en mettant en place un extranet pour ses clients. Cette partie implique une forte sécurité, car il souhaite mettre à disposition des informations confidentielles. Vous souhaitez donc pouvoir en restreindre l'accès aux utilisateurs identifiés et appartenant à une liste établie. Le site de votre client est hébergé chez un spécialiste sous la forme d'un hébergement mutualisé.

REALISATION

- Pour créer un espace sécurisé, on fait généralement appel à une des deux méthodes suivantes :
- utilisation des contrôles d'accès intégrés au serveur web,
- utilisation d'une authentification gérée par l'application PHP via des sessions.
- Dans notre cas, nous sommes dans le cadre d'un hébergement mutualisé et nous n'avons aucun moyen de modifier la configuration Apache, que ce soit de manière globale ou de manière locale (via des fichiers `.htaccess`). La seule solution est donc l'utilisation de sessions PHP. Ces sessions vous permettent de garder trace de l'identification donnée par les gens et validée par vos soins. Nous allons donc créer un fichier permettant de s'identifier, un fichier permettant de valider les données soumises et un script de

vérification de session. Ce dernier script sera inclus en haut de chaque page qui aura besoin d'être sécurisée.

TRAVAIL A FAIRE :

- Votre tâche est d'écrire les trois scripts suivants :
 - `auth.php` : le formulaire pour s'identifier : Le formulaire d'authentification comporte deux champs: pseudo et mot de passe et appelle le script `verif.php`
 - `verif.php` : le script vérifiant le couple pseudo/mot de passe et identifiant l'utilisateur (de deux méthodes : pseudo/mot de passe stocké comme constantes puis stockés dans une base de données).
 - `secure.php` : le script qui vérifie si l'utilisateur a accès à la ressource demandée. :
Pour vérifier si l'utilisateur est déjà authentifié, nous allons lire cette variable dans le tableau de session `$_SESSION[]`. Si l'utilisateur est authentifié, alors la variable `$_SESSION['nom']` existe. Dans le cas contraire, c'est que jamais l'utilisateur n'a fourni de mot de passe valide, et il faut le rediriger à la page `auth.php`.
Il vous suffit maintenant d'ajouter `include('secure.php')` dans les scripts réservés pour que seules les personnes authentifiées puissent les utiliser.
Les autres seront redirigées vers le formulaire d'identification.

auth.php :

```
<html>
<head><title>Identification</title></head>
<body>
  <form method="post" action="verif.php">
    <p>
      <label for="nom">Pseudo : </label>
      <input type="text" name="pseudo"/>
    </p>
    <p>
      <label for="motdepasse">Mot de passe : </label>
      <input type="password" name="motdepasse"/>
    </p>
    <p>
      <input type="submit" value="s'identifier"/>
    </p>
  </form>
</body>
</html>
```

verif.php :

```
<?php
// Initialisation de la session
session_start() ;
// Si on a reçu les données d'un formulaire :
if ( isset( $_POST['pseudo'] ) && isset ( $_POST['motdepasse'] ) ) {
    // On les récupère
    $nom = $_POST['pseudo'] ;
    $motdepasse = $_POST['motdepasse'] ;
    // On teste si le mot de passe est valide :
    if ( verification($nom,$motdepasse ) ) {
        // Le mot de passe est valide, l'utilisateur est identifié
        // On sauvegarde donc son nom dans la session
        $_SESSION['nom'] = $nom ;
        $message = 'vous êtes correctement identifié' ;
    } else {
        // sinon on avertit l'utilisateur :
        $message = 'Mauvais mot de passe' ;
        $message .= '<a href="auth.php">retour</a>' ;
    }
} else {
    // Un des champs n'est pas rempli
    $message = 'le login ou le mot de passe est vide' ;
    $message .= '<a href="auth.php">retour</a>' ;
}
function verification( $name, $password)
{
    define("name","admin");
    define("password","adminpass");
    if (($name==name) AND ($password==password))
    {return true;}
    else
    {return false;
    }
}

?>
<html>
<head><title>Identification</title></head>
<body><p>
<?php echo $message ?>
</p></body>
</html>
```

secure.php

```
<?php
session_start() ;
// On vérifie si l'utilisateur est identifié
if ( !isset( $_SESSION['nom'] )) {
    // La variable de session n'existe pas,
    // donc l'utilisateur n'est pas authentifié
    // On redirige sur la page permettant de s'authentifier
    header('Location: auth.php') ;
    // On arrête l'exécution
    exit() ;
}
```