

COOKIES

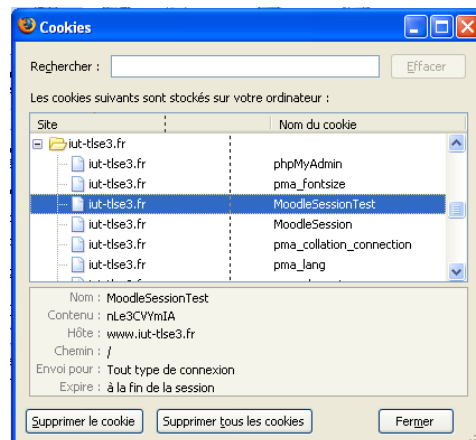
(témoin de connexion)

Cookies

- Mécanisme de stockage d'information sur le poste client (celui qui exécute le navigateur)
- Pour des raisons de sécurité, l'utilisateur peut refuser les cookies (configuration du navigateur)
- Stockés dans un fichier sur le poste client (le lieu dépend du navigateur)
- Contraintes de taille (par exemple minimum 20 cookies sur un même poste et avec une taille de 4 Ko min) pour un site web (un nom de domaine)
- Durée de vie limitée fixée par le concepteur du site

Cookies

- IE enregistre chaque cookie dans un fichier différent.
- Firefox enregistre tous ses cookies dans un seul fichier.



Cookies & PHP

Création d'un cookie

➔ Fonction `setcookie($nom,$valeur,$dateexpiration)`

- nom du cookie (chaîne de caractères) qui sert à l'identifier
- valeur (chaîne de caractères)
- date d'expiration (entier) sous la forme d'un timestamp UNIX

Attention !!!

la fonction `setcookie` doit être la première à écrire dans l'entête, donc pas de HTML, pas d'affichage PHP avant l'appel de cette fonction.

Cookies & PHP

- **Effacer** la valeur d'un cookie

Appel de la fonction `setcookie` avec juste le premier paramètre, le nom du cookie

- **Remplacer** la valeur d'un cookie

Appel de la fonction `setcookie` avec la nouvelle valeur pour le deuxième paramètre

- **Supprimer** un cookie

Appel de la fonction `setcookie` avec comme paramètre une date d'expiration déjà passée

Cookies & PHP

- Lecture d'un cookie en PHP

l'ensemble des cookies stockés sur le poste client (et créés par le site web) sont accessibles dans le tableau `$_COOKIE`

Cookies & PHP

Exemple

// création d'un cookie

```
setcookie("TestCookie", "bonjour", time()+3600);
```

- nom du cookie : TestCookie

- valeur : "bonjour"

- date d'expiration : expire dans une heure

// effacer la valeur du cookie

```
setcookie("TestCookie");
```

// effacer le cookie : date dépassée

```
setcookie("TestCookie", "", time()-3600);
```

Cookies & PHP

// récupération de la valeur du cookie

```
echo $_COOKIE["TestCookie"];
```

→ Affichage de "bonjour"

Cookies

- Les cookies pouvant être interdits par l'utilisateur, ils ne peuvent être utilisés que pour apporter du confort à l'utilisateur pas pour des fonctions importantes du site

Sessions

Sessions

- Le mécanisme des sessions permet de conserver des données lors d'un passage d'une page à une autre page
- Chaque utilisateur accédant à une page web se voit attribuer un identifiant unique appelé « identifiant de session »
- Une session permet d'enregistrer des variables pour pouvoir les réutiliser dans d'autres scripts

Session & PHP

- La gestion des sessions se fait par l'intermédiaire du tableau `$_SESSION`

```
<?php // page1.php
// utilisation des sessions pour cette page
session_start();

// Écriture d'une valeur dans la session
$_SESSION['couleur'] = 'vert';
$_SESSION['animal'] = 'chat';
// lien vers la page 2
echo '<a href="page2.php">page2</a>' ;
?>
```

Session & PHP

```
<?php // page2.php
// utilisation des sessions pour cette page
session_start();

// récupération des valeurs dans la session
echo $_SESSION['couleur'] ;
echo $_SESSION['animal'] ;
?>
```

Session & PHP

Attention !!

pour que les sessions fonctionnent il est nécessaire que le serveur puisse stocker sur le disque du serveur le contenu de `$_SESSION` ;

pour cela le serveur utilise :

- soit le dossier /tmp
- soit un dossier spécial nommé très souvent *sessions* ; en l'absence de l'existence de ce dossier les sessions ne fonctionneront pas.

Session & PHP

Pour pouvoir accéder aux données d'une session, il faut que la page PHP connaisse l'identifiant de la session (le SID)

→ Les pages PHP doivent se transmettre le SID

Deux possibilités :

- utiliser un cookie
 - pb l'utilisateur peut refuser les cookies
- utiliser l'URL
 - le SID est visible

Session & PHP

Attention !!

Comme les sessions peuvent utiliser les cookies, l'appel de la fonction `session_start()` doit se faire avant tout code HTML et tout affichage

Sécurisation

authentification

- Les sessions sont un moyen simple de limiter l'accès à des pages PHP aux seules personnes correctement identifiées
- Utiliser 1 variable de session
Acces : oui/non
- Un formulaire d'identification
Saisie du login et mot de passe
Vérification de la validité
Si ok positionner acces à oui
Sinon positionner acces à non

authentification : accueil.php

accueil.php

veuillez vous identifier

login :

mot de passe :

Menu : [accueil](#) [option1](#) [option2](#)

Tant que l'utilisateur ne s'est pas identifié correctement, on reste sur la page d'accueil

Menu : [accueil](#) [option1](#) [option2](#)

authentification : accueil.php

accueil.php

Une fois identifié correctement, on remplace le formulaire de connexion par celui de déconnexion et toutes les pages sont accessibles

Menu : [accueil](#) [option1](#) [option2](#)

```
<?php
session_start(); // utilisation de session
// si la variable de session n'existe pas, on la crée
if (!isset($_SESSION['acces'])) { $_SESSION['acces']="non"; }
?>
Maenu : <a href="accueil.php">Accueil</a> <a
      href="option1.php">Option1</a> <a href="option2.php">Option2</a>
```

```
<?php
if (isset($_POST["login"])) // click sur le bouton connexion
{ // verif du login et mot de passe
  if ($_POST['nom']=="user" && $_POST['passwd']=="pass")
  { // acces autorisé : variable de session acces = oui
    $_SESSION['acces'] = "oui";
  }
  else
  { // acces non autorisé : variable de session acces = non
    echo "identification incorrecte";
    $_SESSION['acces'] = "non";
  }
}
```

```
// deconnexion : click sur le bouton deconnexion
if (isset($_POST['logout']))
{ $_SESSION['acces'] = "non"; } // acces non autorisé

// si pas identifié correctement, il faut afficher le formulaire d'identification
if ($_SESSION['acces']=="non")
{?>
<form method="post">
<fieldset>
<legend>veuillez vous identifiez</legend>
<label>login :</label> <input type="text" name="nom"/> <br/>
<label>mot de passe :</label><input type="password" name="passwd"/>
<input type="submit" name="login" value="connexion"/>
</fieldset>
</form>
<?php
}
```

```
else {
// si identification correcte : formulaire de deconnexion
?>
<form method="post">
<input type="submit" name="logout" value="deconnexion"/>
</form>
<?php
}
?>
Menu :
<a href="accueil.php">accueil</a>
<a href="option1.php">option1</a>
<a href="option2.php">option2</a>
```

Authentication : option1.php

```
<?php
session_start(); // utilisation des sessions
if ($_SESSION['acces']=="non") // si pas identifié correctement
    // renvoie automatique vers la page d'accueil
    header ("Location:accueil.php");
else {
?>
Menu :
<a href="accueil.php">accueil</a>
<a href="option1.php">option1</a>
<a href="option2.php">option2</a>
page option 1
....

<?php
}
?>
```