Fenêtre de login et d'inscription

Les formulaires XHTML

Dans un premier temps, on crée simplement deux pages : une pour la connexion, un autre pour l'inscription.

- **Formulaire de login :** dans ce formulaire, on demande à l'utilisateur son *login* et son *mot de passe*. Un lien permet de passer au formulaire d'inscription.
- **Formulaire d'inscription :** dans ce formulaire, on demande à l'utilisateur de donner un *login*, et *un mot de passe* (on le demande 2 fois).

Un peut de mise en forme CSS

On réalise une feuille de style commune pour les deux pages (spécifiez les propriétés CSS que vous voulez).

Validation en JavaScript

Dans le formulaire d'inscription, on vérifie lors de la saisie que le login ne contient que des caractères alphanumériques, et que le mot de passe fait au moins 8 caractères. Si le contenu du champ est correct, on l'encadre en vert. Si ce n'est pas le cas :

- on signale le problème de deux manières :
  - en encadrant le champ erroné en rouge ;
  - en affichant un message d'avertissement sous le champ erroné, en rouge ;
- on empêche l'envoi du formulaire.

Le login existe-t-il ? Côté serveur, introduction aux services Web

Sur le serveur, on va créer un service pour vérifier si un login existe déjà. Le service se nomme *loginExists.php*, s'appelle par HTTP (méthode GET ou POST), et prend un paramètre *login* qui a pour valeur le login que l'on veut tester. Il retourne 1 si le login existe déjà, et 0 sinon.

1. Dans un premier temps, on va créer un « service idiot », qui répond toujours la même chose. Dans le fichier *loginExists.php*, on écrit le code suivant :

```
<?php
	print "1";
?>
```

Testez ce script. Attention, vous ne pouvez pas l'afficher directement dans votre navigateur ! Les navigateurs Web ne savent pas exécuter le code PHP. Ce code est traité sur le serveur, vous devez donc l'appeler via votre serveur Apache, qui va l'exécuter et encoyer le résultat au navigateur.

2. Ensuite, on va modifier ce script pour aller un peu plus loin : on retourne 1 si le login est *toto*, et sinon 0.

```
<?php
	$login = $_REQUEST["login"];  // Récupérer le paramètre HTTP "login"
	if($login == "toto")
		print "1";
	else
		print "0";
?>
```

Testez à nouveau ce script, en l'appelant avec votre navigateur, avec un paramètre *login*. Essayez de le modifier, pour qu'il accepte aussi le login *tutu*.

3. Enfin, on va mettre en place un système définitif : les logins seront stockés sous forme de fichiers ; chaque fichier aura pour nom *[lelogin].txt*, et contiendra une seule ligne avec le mot de passe.

▲Pour savoir si un fichier existe en PHP, on peut utiliser la fonction *file_exists("cheminDuFichier").*

Le login existe-t-il ? Côté client, un peu d'Ajax pour l'inscription

Maintenant, on peut ajouter un fonctionnalité au formulaire d'inscription : on vérifie, grâce à un script JavaScript/Ajax (voir le cours) que le login est disponible. Ce script va appeler *loginExists.php*, par la méthode GET, en lui passant un paramètre *login*, et récupérer la réponse. Si la réponse est 1, le login existe déjà ; sinon il n'existe pas encore.

Ainsi, après avoir vérifié que le login ne contient que des caractères alphanumériques (←déjà fait), on vérifie ensuite si il est disponible. Comme précédemment, si c'est le cas, on encadre le champ en vert ; sinon, on l'encadre en rouge et on affiche un message d'erreur sous le champ.

On peut s'inspirer du code ci-dessous :

```
function checkLogin(login) {
  var mygetrequest = new XMLHttpRequest();  // Création de l'objet servant à envoyer une requête HTTP

  mygetrequest.onreadystatechange=function() {  // Création de la fonction qui sera appelée à chaque changement d'état de la requête HTTP
    if(mygetrequest.readyState==4 && mygetrequest.status==200) {  // Tout s'est bien passé !
        if(mygetrequest.responseText==0)  // Si la réponse du service est "0"
          loginFieldExists();  // Appeler la fonction qui traite ce cas
        else  // Sinon
          loginFieldDontExists();  // Appeler la fonction qui traite cet autre cas
    }
  }

  var encodedLogin=encodeURIComponent(login);  // On encode le login au format HTTP
  mygetrequest.open("GET", "loginExists.php?login="+encodedLogin, true);  // On crée la requête HTTP
```

```
  mygetrequest.send(null);  // On envoie la requête HTTP
}


function loginFieldExists() {
  // Lorsque le login existe déjà.
}


function loginFieldDontExists() {
  // Lorsque le login n'existe pas encore.
}
```

Terminer l'inscription côté serveur

Il reste maintenant à ajouter effectivement l'utilisateur lorsqu'il soumet le formulaire d'inscription. On fait en sorte que, lorsqu'il est soumis, le formulaire d'inscription appelle le script *inscription.php*. Cela se fait grâce à l'attribut *action* de la balise *form* : <form action="inscription.php" ...>

On crée ensuite le script *inscription.php*. Ce script prend deux paramètres (*login* et *password*), crée un fichier *[lelogin].txt* qui contient le mot de passe, et affiche le message « Félicitations, vous êtes inscrit. ».

▲Pour écrire dans un fichier en PHP, on peut utiliser la fonction file_put_contents(*"cheminDuFichier", "texte à écrire"*).

Terminer le login côté serveur

De la même manière, on va relier le formulaire de login à un script *login.php*, qui prend deux paramètres (*login* et *password*). En cas de succès, on affiche « Félicitations, vous êtes connecté ». Dans le cas contraire, on affiche à nouveau le formulaire de connexion.

▲Pour lire un fichier en PHP, on peut utiliser la fonction *file_get_contents("cheminDuFichier")*. La fonction retourne le contenu du fichier.