

Cours Système Web PHP

TP4

Intégration Web / PHP

21 novembre 2005

1 Introduction

1.1 Serveur Web

Les scripts PHP doivent être servis par un serveur Web proprement configuré. En conséquence, il faut **impérativement** passer par une URL en `http://` afin de pouvoir accéder au résultat du script PHP (Fichier/Ouvrir, ou une URL en `file://`, ne fera que lire le source du script PHP sans l'interpréter). On se conformera aux instructions données pendant le TP pour savoir où placer les fichiers et comment y accéder par HTTP.

1.2 Squelette PHP

Dans la majorité des cas, le script PHP s'inscrit dans un squelette de document HTML, par exemple :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    <h1>Page de test</h1>
    <?php echo "Hello world!"; ?>
  </body>
</html>
```

2 Formulaire et intégration Web / PHP

1. Créer une page HTML comprenant un formulaire avec les champs suivants :
 - Un champ texte
 - Un champ mot de passe
 - Une liste déroulante à 4 entrées :
 - bleu
 - rouge
 - vert
 - jaune
 - Un groupe de boutons radio Oui/Non
 - Un bouton de réinitialisation du formulaire
 - Un bouton de soumission du formulaire

Remarque : pour avoir du HTML correct, les champs du formulaire (`input`), ainsi que les étiquettes `label` et le reste du contenu ne doivent pas être mis directement dans la balise `form`, mais encapsulés dans des balises `div` (bloc), `p` (paragraphe) ou `fieldset` (ensemble de champs).

On positionnera la méthode du formulaire à `post`. L'attribut `action` du formulaire doit pointer vers un script prenant en charge le traitement des données du formulaires. Indiquer `affiche_parametres.php`; ce script est l'objet de la question suivante. Valider la page HTML obtenue.

2. Les paramètres HTTP passés par la requête sont disponibles dans le tableau associatif PHP `$_REQUEST`. Créer un script `affiche_parametres.php` qui affiche le contenu du tableau associatif `$_REQUEST` sous forme d'un tableau HTML (dont la première colonne contient les noms, et la deuxième colonne les valeurs). Tester avec le formulaire précédent.
3. Ajouter l'attribut `multiple` à la liste d'option du formulaire. Sélectionner plusieurs couleurs (grâce à la touche Ctrl) et regarder le résultat affiché par `affiche_parametres.php`.
Pour pouvoir passer des paramètres avec valeurs multiples (liste avec plusieurs sélections, groupes de case à cocher), PHP requiert que le nom du paramètre se termine par `[]`. Modifier en conséquence le formulaire et adapter le code du script PHP pour tenir compte des paramètres à valeur multiples (la fonction PHP `is_array` pourra être utile).
4. Ajouter au formulaire un champ de soumission de fichier, et adapter le code du script pour afficher, pour les paramètres fichiers, le nom du fichier soumis par le client dans la deuxième colonne.

3 Sessions

Écrire une application PHP comprenant les pages suivantes. Commencer par réaliser un schéma des interactions entre pages. On trouvera sur la page Web du cours un exemple de réalisation de cette application.

1. `login.php` présente un formulaire demandant un nom d'utilisateur et redirigeant vers `bonjour.php`.
2. `bonjour.php` :
 - si on lui passe en paramètre un nom d'utilisateur, ouvre une session, stocke ce nom dans une variable session et affiche un message de bienvenue contenant ce nom.
 - si une session est ouverte, affiche un message de bienvenue contenant le nom d'utilisateur.
 - affiche un message d'erreur et propose de revenir à `login.php` sinonDans les deux premiers cas, `bonjour.php` propose un lien vers `quitter.php`.
3. `quitter.php` affiche un message contenant le nom d'utilisateur et termine la session si la session est ouverte, affiche un message d'erreur sinon. Un lien pour revenir à `login.php` est proposé.

Tester. Ajouter à `bonjour.php` un lien vers `login.php` permettant de modifier le nom d'utilisateur courant. Faire en sorte que `login.php` affiche comme valeur par défaut du nom d'utilisateur le nom d'utilisateur courant.

4 Manipulation de fichiers

Afin d'éviter des confusions, on utilisera ici les termes *download* et *upload* au lieu de l'ambigu *télécharger*.

Réaliser une application PHP permettant de :

- Uploader des fichiers sur le serveur, chaque fichier étant identifié par un titre.
- Afficher la liste des fichiers ayant été uploadé, avec leur titre, leur taille, et un lien pour les downloader.

Bien réfléchir à tout ce qui doit être fait avant de se lancer. En particulier, décrire le schéma des interactions entre page, réfléchir où les fichiers seront stockés sur le serveur, etc.