

Cours Système Web PHP

TP6

Sérialisation, boîtes CSS

12 décembre 2005

1 Sérialisation

La *sérialisation* est la représentation sous forme d'une chaîne de caractères d'un objet ou tableau PHP, de manière à ce que cet objet ou tableau puisse être reconstitué par la suite. Ceci permet de stocker dans un fichier le contenu d'une variable PHP, pour ensuite reconstruire cette variable en relisant le fichier. Les fonctions utiles sont `serialize` et `unserialize`, qui prennent chacun un unique argument et renvoient le résultat de la (dé)sérialisation.

Dans l'exercice 2 du TP4, on affichait les paramètres d'une requête PHP. On veut pouvoir reproduire le même comportement, mais en stockant plusieurs requêtes. Pour ceci, à chaque fois que l'on va soumettre un formulaire, on va stocker dans un fichier d'un répertoire `fichiers` la *sérialisation* du tableau associatif `$_POST`. Au moment d'afficher l'ensemble des requêtes, il suffira de parcourir l'ensemble des fichiers du répertoire et de récupérer leur contenu en le désérialisant. Les fichiers auront comme nom un numéro d'identifiant, qu'on fera croître au fur et à mesure, en stockant le numéro actuel dans un fichier `current_id`.

1. Récupérer la correction du deuxième exercice de l'exercice 2 sur la page Web du cours. Enlever de `formulaire.html` et de `affiche_parametres.php` les parties concernant l'upload de fichiers. Modifier l'action du formulaire pour pointer vers `ajout_enregistrement.php`
2. Dans un fichier `tools.php`, écrire une fonction `get_current_id()` qui :
 - (a) récupère le contenu d'un fichier `current_id` et le stocke dans une variable `$id`
 - (b) écrit dans le fichier `current_id` la valeur de `$id+1`
 - (c) retourne `$id`
3. Écrire `ajout_enregistrement.php`, dont le rôle est de créer un nouveau fichier dans le répertoire `fichiers` (avec pour nom la valeur retournée par `get_current_id()`), contenant la sérialisation du tableau associatif `$_POST`.
4. Modifier `affiche_parametres.php` de manière à parcourir l'ensemble des fichiers du répertoire `fichiers` et à afficher le contenu de chacun de ces fichiers comme une table HTML (première colonne : nom de paramètre, deuxième colonne : valeur(s) du paramètre).

2 Boîtes CSS

En plus des propriétés de *mise en forme* du texte que l'on a vu jusqu'ici (`font-size`, `text-align`, `color`...), CSS propose des propriétés de *mise en page* (`float`, `position`...) permettant de contrôler l'agencement des différents blocs (`p`, `div`, `table`...).

On s'appuie pour ce TP sur des documents réalisés pour un autre cours, à l'Institut de formation d'ingénieurs de Paris-Sud.

1. Lire le cours *CSS — Notions avancées* disponible à l'URL <http://pierre.senellart.com/enseignement/2005-2006/ifips1web/cours3/cours3.pdf>
2. On pourra également utiliser l'aide-mémoire disponible à l'URL <http://pierre.senellart.com/enseignement/2005-2006/ifips1web/cours3/fiche3.pdf>
3. Regarder <http://pierre.senellart.com/enseignement/2005-2006/ifips1web/cours3/modele.html>
4. Sauvegarder cette page HTML en local et écrire une feuille de style qui reproduit la même mise en page.