

TP 7— PHP

Internet et Outils (IO2)

Nous allons nous intéresser dans ce TP à la création de petites pages XHTML générées automatiquement à l'aide du langage php.

Vous trouverez un manuel en ligne (en français) de php sur
`http://fr2.php.net/manual/fr/`

Le langage de programmation php est couramment utilisé pour générer du texte au format (X)HTML. Ce langage présente l'intérêt de pouvoir s'insérer au milieu du (X)HTML, entre la balise ouvrante `<?php` et la balise fermante `?>`.

Les fichiers contenant du code php doivent porter un nom d'extension `.php`. Vous devez créer un répertoire `public_html` dans votre répertoire maison et placer ces fichiers dans l'arborescence de ce répertoire.

Le serveur exécute le code php puis renvoie le résultat (sous forme de texte en principe au format (X)HTML), au navigateur. Ceci représente un gros avantage en matière de diffusion d'un site : on n'a pas besoin de s'assurer que la machine cliente possède une quelconque application (en dehors du navigateur). En contre-partie, en cas de nombreuses requêtes sur le serveur celui-ci risque d'être surchargé.

Exercice 1 — *Premier script php*

Recopiez le texte suivant dans un fichier `~/public_html/essai.php` :

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD HTML 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>mon premier script php</title>
  </head>
  <body>
    <?php phpinfo() ?>
  </body>
</html>
```

Vous pouvez visualiser ce que rend votre script à l'adresse :
`http://pams.script.jussieu.fr:80/~mon_login/essai.php`
où `mon_login` est remplacé par votre login.

... joli, n'est-ce pas ?

Exercice 2 — *Date et Switch*

1. (rappel de cours)
Ecrivez une page qui donne la date en anglais, sous la forme : "Today is : 25/10/2007".

- En bons francophones que nous sommes, nous allons maintenant créer un script `php` qui donne la date et l'heure en français et en en toutes lettres, sous la forme :
 "Aujourd'hui, nous sommes le mardi 25 octobre 2007, il est 14h32".
 Nous allons, pour cela, procéder en deux étapes :
 - ★ Créez des variables dans lesquelles vous allez ranger les valeurs dont vous aurez besoin par la suite.
 - ★ Certaines de ces variables sont directement utilisables par votre script `php`, mais ce n'est pas le cas pour celles que vous allez devoir traduire.
 Utilisez l'instruction `switch` pour "traduire" ces dernières.

L'action du bouton de type `submit` d'un formulaire (voir TP4) est définie par la valeur de l'attribut `action` de l'élément `form`. Dans notre cas, on appellera systématiquement un fichier `php`. On précisera également dans l'élément `form` la méthode d'envoi des données en utilisant `method="get"` (ou `method="post"`).

Le script `php`, ainsi appelé, récupère la valeur d'un `input` de nom `toto` dans la variable `$_REQUEST["toto"]`.

Exercice 3 — *Création d'une page XHTML à partir d'un formulaire*

- Créez une page `form.html` contenant un formulaire demandant à l'utilisateur de fournir un titre, un auteur et un texte de description, et appelant un script `php` noté `form.php`, avec la méthode `get`.
- Créez alors une page `form.php` contenant un script dont le résultat est une page `xhtml` dont le titre (l'entête) est le titre fourni dans le formulaire précédent, contenant, comme sous-titre, l'auteur fourni, et contenant un paragraphe où s'affiche le texte fourni par l'utilisateur.
- (`get` ou `post` ?) Retour au formulaire.
 Si vous avez utilisé l'`input` de type `text` pour le texte demandé, vous remarquerez que cela est peu pratique pour y écrire un texte de plus de vingt caractères... Modifiez donc votre `input` pour que l'utilisateur ait plus de place pour écrire son texte.
 Quel est alors l'inconvénient de la méthode `get` ?
 Une fois que vous avez trouvé la réponse, vous pouvez même deviner la question d'après : utilisez la méthode `post` pour votre formulaire et constatez le progrès...

Exercice 4 — *Convertisseur Euros / Dollars*

- Créez une page `form2.html` contenant un formulaire composé de trois `inputs` : 2 boutons permettant de choisir le sens de conversion que l'on désire (dollars vers euros ou l'inverse), et un champ permettant de saisir la valeur que l'on souhaite convertir.
- Créez alors une page `form2.php` contenant un script qui va convertir la valeur saisie dans le sens précisé. Vous devez pour cela tester quel est le sens qui a été demandé et en déduire quel est le coefficient qu'il faut utiliser (au 14/02/07, les coefficients étaient : 1.313 euros pour 1 dollar, et 0.762 dollar pour 1 euro).
 Le résultat doit être un texte de la forme *"100 dollars valent 76.2 euros"*.

Exercice 5 — *Formulaire et vérification des données fournies*

- Créez une page `form3.html` contenant un formulaire d'inscription, demandant à l'utilisateur de fournir nom, prénom, adresse e-mail, pseudo, mot de passe, confirmation du mot de passe et appelant un script `php` noté `form2.php`. Précisez, sur cette page, que les champs adresse e-mail, pseudo, mot de passe et confirmation du mot de passe sont obligatoires.

2. Créez alors une page `form3.php` contenant un script qui va vérifier si un des champs obligatoires est vide. Si c'est le cas, il doit alors afficher un message précisant quel est le premier champ qui a été oublié. Sinon, il doit afficher un message remerciant l'utilisateur de s'être inscrit, en l'appelant par son pseudo.
3. Enrichissez votre script pour qu'il vérifie que le mot de passe fourni ne fait pas moins de 6 caractères, et, le cas échéant, précisant à l'utilisateur que son mot de passe est trop court.
4. Enfin, enrichissez votre script pour qu'il vérifie que le mot de passe fourni et sa confirmation sont bien les mêmes (et précisant l'erreur, si ce n'est pas le cas).

Exercice 6 — *Carnet d'adresses*

Chargez le fichier `www.pps.jussieu.fr/~tasson/enseignement/I02/tp7/carnet.inc`. Ce fichier contient du code `php` que vous incluez dans votre page grâce à la commande

```
include("carnet.inc");
```

Un tableau `$carnet` y est défini, dont chaque case est elle-même un tableau contenant divers renseignements sur une personne (nom, prénom, téléphone, ville).

1. Ecrivez une fonction `php` qui prend en argument une ligne d'une table `php` et la convertit en une ligne d'un tableau `XHTML`.
2. Ecrivez une fonction `php` qui prend en argument le nom d'une personne et renvoie les renseignements la concernant sous la forme d'une ligne d'une table, à l'aide de la fonction précédente.
3. Ecrivez une fonction qui affiche l'ensemble du carnet d'adresses.
4. Ecrivez une fonction qui affiche les renseignements concernant toutes les personnes habitant une ville donnée.
5. Ecrivez une fonction qui affiche les renseignements concernant toutes les personnes dont on possède le numéro de portable.
6. Ecrivez une page contenant un formulaire dans lequel on donne à l'utilisateur la possibilité d'afficher tout le carnet d'adresses ou de faire une recherche dans le carnet par nom, par ville, ou par disponibilité du numéro de portable.

Exercice 7 — *Compteur de clics*

Ecrivez une page `php` qui s'autoréférence grâce à un formulaire. Faites-en sorte que cette page affiche le message "*Vous n'avez jamais cliqué.*" tant que l'utilisateur n'a effectivement pas encore cliqué sur le bouton `submit` du formulaire, puis le message "*Vous avez déjà cliqué n fois*" où *n* prend la valeur du nombre de clics effectués.

Ajoutez le message "*C'est fini, oui ?*" quand l'utilisateur a déjà cliqué dix fois ou plus.

Indications :

- Il faut "retenir" le nombre de clics déjà effectués dans une variable `php`.
- Afin d'incrémenter à chaque clic cette variable, vous pouvez utiliser l'`input` de type `hidden`, dans votre formulaire.
- N'oubliez pas d'initialiser cette variable à 0 avant le premier clic.