

TP 01 — Prise en main de PHP

fabien.givors@unice.fr

2014-2015

1 Mise en place

Le serveur de page personnelles de l'Université interprète les pages PHP, c'est lui que nous allons utiliser pour les TP. Vous allez donc créer dans votre répertoire personnel une place de dépôt dans laquelle les pages PHP des TP seront entreposées.

1.1 Préparation de dossiers

Le dossier doit être créé à un endroit particulier, en respectant la casse¹ et les espaces éventuels. Ces dossiers doivent également être accessibles en lecture et en exécution au serveur web.

Le plus simple reste d'ouvrir un terminal et de taper les commandes suivantes²

```
mkdir -p ~/www/php
chmod -R o=rx ~/www
chmod o+x ~/
```

1.2 Test du bon fonctionnement et visualisation des pages

Vous pouvez créer une page de test (qui demeurera éditable par la suite), via la commande³:

```
echo '<?php echo "Hello World!";' > ~/www/php/index.php
```

Consultez ensuite avec votre navigateur la page:

<http://www-mips.unice.fr/~votre-login/php/index.php>

1.3 Lors du début d'un TP

Pensez à créer un dossier dédié à chacun des TP et à lui affecter les bons droits, par exemple pour le TP 1, via les commandes:

```
mkdir ~/www/php/tp01/
chmod -R o+rx ~/www/php/tp01
```

2 Première page: échauffement

Exercice 1. Dans une page `tp01/sommes.php`,

1. écrire un bloc de code PHP qui affiche la somme des n premiers entiers pour chaque n inférieur à 200;
2. modifier le bloc précédent pour qu'il affiche la somme en rouge si elle est un nombre pair⁴.

¹casse: la distinction entre les majuscules et les minuscules.

²Détail des commandes:

- la première ligne crée, s'il n'existe pas, le dossier `www` dans votre répertoire personnel, et crée, s'il n'existe pas, un sous-dossier `php` dans celui-ci;
- la seconde ligne donne accès en lecture et exécution au serveur web à tous les fichiers et dossiers du répertoire `www`;
- la troisième ligne donne au serveur web le droit d'accès à votre répertoire personnel.

³Détail de la commande: cette ligne crée (ou écrase) un fichier `~/www/php/index.php` contenant le code `<?php echo "Hello World!";`.

⁴On utilisera l'opérateur modulo `%` qui permet de connaître le reste de la division d'un entier par un autre (un nombre étant pair si le reste de sa division par deux est nul.)

Astuce 1 Astuce: N'oubliez pas la structure de base du document HTML à l'intérieur du fichier PHP. Vous pouvez partir de modèle suivant:

Listing 1: sommes-exemple.php

```
<!doctype html>
<html>

<head>
  <meta name="author" content="Votre Nom"/>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Somme des premiers entiers</title>
</head>
<body>
<?php
  // calculs et affichage ici
?>
</body>
</html>
```

Astuce 2 La syntaxe HTML pour mettre du texte en couleur (par exemple en rouge) est explicite dans cet exemple:

Listing 2: Mot en couleur

```
| Une phrase avec un <span style="color: red;">mot</span> en rouge.
```

3 Lister les fichiers contenus dans un sous-dossier

L'archive suivante : <http://chezlefab.net/ens/slzio012/tp-01-news.tar.gz> contient les fichiers suivant:

```
news
  monde
    01.news
  politique
    1.news
    2.news
    3.news
    4.news
    5.news
  sports
    01.news
    02.news
    03.news
    04.news
```

Commencez par l'extraire dans votre dossier tp01, par exemple via les commandes⁵:

```
wget -O ~/tp-01-news.tar.gz "http://chezlefab.net/ens/slzio012/tp-01-news.tar.gz"
tar xf ~/tp-01-news.tar.gz -C ~/www/php/tp01/
```

Ces fichiers constituent le fil rouge de vos tps : vous allez appliquer les différents concepts à ce cadre (navigation, session, manipulation des fichiers, formulaires, etc).

Créez un nouveau fichier `listarticle.php`.

Exercice 2. Commençons par lister le contenu du dossier « sports »

Dans ce nouveau fichier `listarticle.php`, avec `php`, générez la liste html (voir Astuce 1) des fichiers « .news » contenus dans le dossier « ./news/politique ».

Chaque item de cette liste doit être un lien (voir Astuce 2) vers le fichier `.news` correspondant.

⁵Détail des commandes:

- la première commande télécharge l'archive depuis le site et l'enregistre dans le répertoire personnel;
- la deuxième commande extrait l'archive téléchargée dans le dossier `tp01`.

Pour cela, utilisez la fonction `glob` (voir annexe 1) parcourez le résultat avec une boucle `foreach`. Construisez la liste dans cette boucle. Ne l'affichez (`echo`) que dans le corps de la page.

Essayez votre page en ajoutant / retirant des fichiers (`.news`) dans le dossier « `./news/politique` » et en rafraîchissant votre page dans votre navigateur. Essayez les liens que vous avez générés.

Astuce 1 Syntaxe pour la création d'une liste ordonnée:

Listing 3: Liste ordonnée

```
<ol>
  <li>Premier élément de la liste</li>
  <li>Deuxième ...élément</li>
  <li>...</li>
</ol>
```

Astuce 2 Syntaxe pour la création d'un lien

Listing 4: Lien

```
<a href="page2.html">Lien vers la page 2</a>
```

Exercice 3. Nous allons lister le contenu de tous les dossiers contenus dans le dossier « `news` »

Modifiez votre code, en utilisant une nouvelle fois `glob` (avec l'option `GLOB_ONLYDIR`) dont le résultat sera parcouru par une autre boucle `foreach` englobant la première. Utilisez les valeurs du nouveau `glob` pour générer le chemin (nom des sous-dossiers) pour le second.

Essayez votre page en ajoutant / retirant des fichiers (`.news`) et en rafraîchissant votre page dans votre navigateur. Essayez les liens que vous avez générés.

Essayez votre page en ajoutant / retirant des fichiers (`.news`) dans les sous-dossiers de « `./news` » et en rafraîchissant votre page dans votre navigateur. Essayez les liens que vous avez générés.

4 Annexe 1: La fonction `glob()`

La fonction `glob()` permet de lister tous les fichiers et dossiers dont le chemin satisfait un certain motif donné en paramètre.

Le prototype de la fonction `glob()` est le suivant:

```
array glob ( string $pattern [, int $flags = 0 ] )
```

Cela signifie que la fonction prend deux arguments. Le premier de type chaîne de caractère qui correspond à un *motif*, et le second de type entier qui correspond à un ensemble d'options. Enfin, la fonction renvoie un tableau de tous les fichiers et dossiers concernés.

La documentation des options⁶ nous indique:

| Option | Description |
|----------------------------|---|
| <code>GLOB_MARK</code> | Adds a slash to each directory returned |
| <code>GLOB_NOSORT</code> | Return files as they appear in the directory (no sorting) |
| <code>GLOB_NOCHECK</code> | Return the search pattern if no files matching it were found |
| <code>GLOB_NOESCAPE</code> | Backslashes do not quote metacharacters |
| <code>GLOB_BRACE</code> | Expands a,b,c to match 'a', 'b', or 'c' |
| <code>GLOB_ONLYDIR</code> | Return only directory entries which match the pattern |
| <code>GLOB_ERR</code> | Stop on read errors (like unreadable directories), by default errors are ignored. |

Exemple d'utilisation: pour lister tous les dossiers dans le répertoire `data`:

```
1 $dossiers = glob ("./data/*", GLOB_ONLYDIR);
2
3 foreach($dossiers as $dossier)
4 {
5     echo $dossier;
6 }
```

⁶<http://php.net/>