TP PHP n°1

I. Truck, Université Paris 8,
Master 1 MIASHS, 2019–2020

Rappels

— En cas de doute sur un concept, une fonction, etc., consulter la documentation `http://www.w3schools.com/php`. C'est probablement la plus complète actuellement.
— En PHP, les déclarations de variables sont *implicites*. Expliquer et donner les 5 principaux types de variables, sans oublier les **variables prédéfinies**.
— On concatène du texte avec des variables avec l'opérateur '.' . Expliquer. Pour afficher les chaînes de caractères, que préfèrera-t-on ? Les *quotes* ou les guillemets ? Pourquoi ?
— Conditions : quelles différences y a-t-il entre `if-else`, `elseif`, `switch` et l'opérateur ternaire ' ? :' ?
— Boucles : quelle différence y a-t-il entre `while` et `for` ?
— Quelle différence y a-t-il entre `print` et `echo` ?
— Comment insérer du code PHP dans une page HTML ? Et comment déclare-t-on une fonction PHP ?
— Il faut toujours donner des noms de variables **clairs** et **explicites**. Il faut penser à bien **indenter** son code. Il faut créer **un nouveau répertoire par fiche TP**. Faites-le.
— Le site officiel PHP contenant la référence des fonctions se trouve à l'adresse : `http://fr.php.net/manual/fr/funcref.php`
— Que font les fonctions `strlen(), str_replace(), str_shuffle(), strtolower()` ?
— Il existe deux types de tableaux : les tableaux numérotés et les tableaux associatifs. Donner un exemple en PHP, avec les deux types. Quelle est la différence entre les 2 ? Comment les parcourir ? (*cf.* `for, foreach, print_r`). Comment rechercher une valeur ? (*cf.* `array_key_exists, in_array, array_search`)

1 Premier programme en PHP

— Créer, en utilisant l'éditeur `emacs` (`Ctrl+X+S` permet de sauvegarder ; `Ctrl+X+C` permet de quitter), une page nommée `index.php` constituée d'une entête, d'un corps et d'un pied de page (grâce à des balises HTML `<div>`).
Les parties entête, corps et pied de page devront être des appels à des pages PHP (appel à la fonction PHP `include()`). On obtient donc en tout 4 pages : `index.php, entete.php, corps.php, piedPage.php`. Le fichier `corps.php` contiendra simplement le numéro du TP et de l'exercice.
— Dans le fichier `entete.php`, faire appel à la fonction `date()` ainsi qu'aux **tableaux** (associatifs ou numérotés ?) pour afficher la date et l'heure *en français* (exemple : "Mardi 2 mars 2020, 18h30"). Afficher ensuite une deuxième fois la date et l'heure en français en utilisant la fonction `setlocale()`.
— Tester le tout en copiant les fichiers sur le serveur `handiman (193.54.166.23)` sous le répertoire `public_html/TP1`. Se reporter aux instructions données par M. Archambault en début d'année.

2 Transmettre des données d'une page à l'autre

— Dans le fichier `corps.php`, ajouter un lien vers un fichier nommé `form.php`. Insérer un formulaire à l'aide de la balise `<form>` dans `form.php`. On utilisera la méthode GET et l'on modifiera le fichier `corps.php` pour qu'il affiche le nom et le prénom de l'internaute. L'action de `form` sera donc `corps.php`, les `input` du formulaire seront donc de deux types : `text` et `submit` et ils seront nommés `nom, prenom, OK`. Aide : on affichera les valeurs grâce à `$_GET['nom']`.
— Dans le même formulaire, ajouter un champ de type `password`, nommé `motPasse`.
— Que se passe-t-il au niveau de l'URL, lors du passage des paramètres nom, prénom et mot de passe ?
— Contrôler les paramètres transmis en vérifiant, avant de les afficher, s'ils sont bien positionnés (avec `isset()` et `empty()`). S'ils ne sont pas bien positionnés, afficher un message d'erreur.

— Dans le même formulaire, ajouter deux champs de type `text` pouvant contenir respectivement un nombre et une phrase (comme une punition qu'on écrirait plusieurs fois). Nommer ces champs `punition` et `nbFois` et mettre 10 comme valeur par défaut au champ `nbFois`. Modifier le fichier `menu.php` de sorte qu'il affiche la punition `nbFois` fois. Vérifier que le champ `nbFois` est bien un nombre entier en le *castant* (de *cast* : conversion) en `int`. Au niveau de l'URL, quels dangers le passage de ce paramètre peut-il impliquer ? Ajouter une vérification sur la taille du nombre (par exemple, si supérieur à 100, alors forcer la valeur à 100).
— Dupliquer le fichier `form.php` en `form.1.php`. Dans `form.php`, changer la méthode GET par la méthode POST. Qu'est-ce qui change ?
— **Ne pas sous-estimer la vulnérabilité** du fichier `corps.php` (contenant le traitement des données issues du formulaire) : il peut être appelé par n'importe quelle page (pas seulement `form.php`). Quelles conséquences ? Par exemple, mettre du code HTML dans le champ `nom` (balise `h1`, par exemple). Puis, mettre du code Javascript pour générer une alerte (aide : balises `script`, type `text/javascript` et fonction `alert()`, *cf.* la fameuse faille XSS).
— Comment résoudre le problème précédent ?
Il faut *échapper* le code HTML transmis avec la fonction `htmlspecialchars()`. Tester cette fonction. On peut également utiliser la fonction `strip_tags()` qui supprime les balises HTML du champ. Tester également cette fonction.

3 Exercice : protéger une page par mot de passe

— Créer une page `formulaire.php` contenant un simple titre et un formulaire avec un champ `Mot de passe` et un champ `Accéder` qui appellera la page `pageProtegee.php` avec la méthode `POST`.
— Créer la page `pageProtegee.php` qui affiche un texte (par exemple, "Voici le plus grand secret de tous les temps : le cheval d'Henri IV était blanc.") seulement si le mot de passe est correct. Sinon, un autre texte est affiché (par exemple, "Vous n'êtes pas autorisé à afficher cette page : mot de passe incorrect." ou bien "Vous n'êtes pas autorisé à afficher cette page : merci de renseigner le mot de passe.").
— Reprendre la page du formulaire en *intégrant* le contenu de `pageProtegee.php` dans `formulaire.php`. Le formulaire pointe donc sur lui-même.

4 Variables superglobales : les sessions

— Qu'est-ce qu'une variable superglobale ?
— Les principales sont : `$_SERVER, $_ENV, $_SESSION, $_COOKIE, $_GET, $_POST, $_FILES`. En dire deux mots.
A quoi sert la valeur contenue dans `$_SERVER['REMOTE_ADDR']` ?
— Pour conserver des variables sur toutes les pages du site et pendant toute la durée de présence de l'internaute, on utilise les sessions (plus facile qu'avec `GET` ou `POST`). On démarre une session (**sur toutes les pages du site**) avec la fonction `session_start()` (**avant d'écrire tout autre code**) et on la termine avec `session_destroy()`. `session_start()` démarre une session ou **reprend la session déjà ouverte**. `session_destroy()` est appelée automatiquement en cas d'inactivité prolongée, mais on peut aussi l'invoquer en créant un bouton (ou un lien) `Déconnexion`.
— Créer le fichier `login.php` en ajoutant un lien à partir de `corps.php`. Dans `login.php`, créer 3 variables de session : `$_SESSION['nom']`, `$_SESSION['prenom']` et `$_SESSION['motPasse']`.
— Reprendre le fichier `index.php` qui doit maintenant faire appel à `entete.php`, `login.php`, `piedPage.php`, comme le montre la figure 1. Ne pas oublier d'appeler `session_start()` partout où cela est nécessaire.
— Reprendre la page `login.php` et mettre comme `action` un appel à elle-même (comme dans l'exercice précédent). Ajouter un lien dans `login.php` vers une autre page `infos.php`, qu'il faut donc créer. Vérifier, en les affichant, que `infos.php` voit bien les 3 variables de session.
— Dans `infos.php`, créer un lien "Déconnexion" qui ramène l'internaute à la page `index.php` et qui supprime la session. Pour ce faire, il faut créer une page de déconnexion intermédiaire (`logout.php`) qui va d'abord appeler la fonction `unset($_SESSION)` avant `session_destroy()`. L'appel à `index.php` se fait grâce à `header('Location: index.php')` qui permet de rediriger le navigateur (si `index.php` est dans le même répertoire que le fichier appelant, bien sûr). Lire le manuel PHP au sujet de `header()` si besoin.

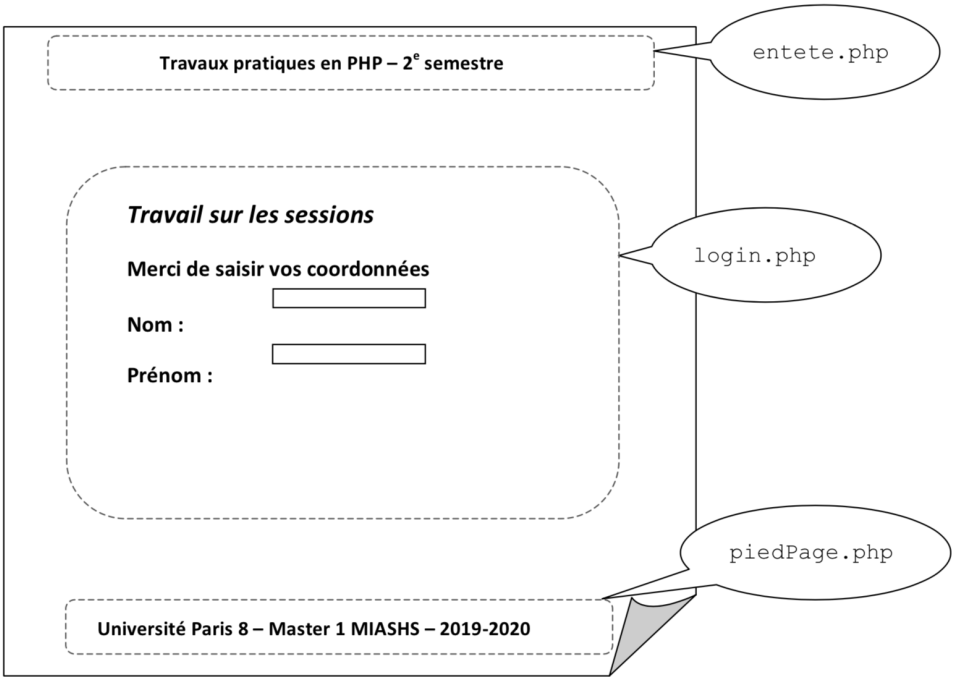

FIGURE 1 – Contenu du fichier `index.php`.

— Copier le fichier `infos.php` en le nommant `infos2.php`. Ajouter un autre lien dans `login.php` vers la page `infos2.php`. `infos2.php` va déconnecter l'utilisateur au bout de 30 secondes et afficher "Vous avez été déconnecté". Ensuite, si la session existe encore (donc avant l'écoulement des 30 secondes), il faut afficher les informations de session (nom, prénom). Pour tester, vous pouvez rafraichir la page avant 30 secondes, puis attendre 30 secondes et rafraichir de nouveau. Vous devriez voir apparaître le message "Vous avez été déconnecté" la deuxième fois.
Nota bene : pour faire cette déconnexion automatique, servez vous de cette page :
`https://stackoverflow.com/questions/520237/how-do-i-expire-a-php-session-after-30-minutes`
avec le code :

```
if (isset($_SESSION['LAST_ACTIVITY']) && (time() - $_SESSION['LAST_ACTIVITY'] > 30)) {
    // la derniere activité a eu lieu il y a plus de 30 secondes
    session_unset();     // unset $_SESSION
    session_destroy();   // detruit les données de session de la mémoire
}
$_SESSION['LAST_ACTIVITY'] = time(); // met à jour le timeStamp de la dernière activité
```