

# TP 06 (noté) — Exercices récapitulatifs

fabien.givors@unice.fr

2014-2015

- Le TP est à rendre par mail à l'adresse ci-dessus avec pour sujet « [PHP-TP6] votre nom ».
- Le TP peut être réalisé seul ou en binôme. Alors, un seul mail sera envoyé comportant les deux noms.
- Le sujet est découpé en trois parties.  
Vous devez rendre la première partie au plus tard le mercredi 12/11/2014 23h59 UTC+0100.  
Vous devez rendre les deuxième et troisième parties au plus tard le mercredi 19/11/2014 23h59 UTC+0100.  
Les parties sont indépendantes, mais les exercices au sein d'une partie sont liés et à faire dans l'ordre.
- La présence en TP reste obligatoire aux horaires habituels (sous réserve des places disponibles...)

## 1 Mini-calculatrice

### Exercice 1. Tables d'opération

1. Créer un fichier vide `ex1-fonctions.inc`. Ce fichier ne générera pas une page HTML mais comportera simplement des fonctions et variables globales.
2. Définir dans ce fichier un tableau associatif `$operations` tel que `$operations["plus"]` soit égal à "+", et ainsi de suite avec "moins", "fois" et "par".
3. Définir à la suite une fonction `calcule($n1, $n2, $oper)` qui, suivant que la variable `$oper` soit égale à la chaîne de caractère "plus", "moins", "fois" ou "par" retourne (sans afficher) respectivement la valeur de  $\$n1+\$n2$ ,  $\$n1-\$n2$ ,  $\$n1*\$n2$  ou  $\$n1/\$n2$ .
4. Créer un fichier `ex1-test.php` affichant dans une page HTML les tables d'additions, soustraction, multiplication et division pour les entiers de 1 à 9. On pourra utiliser le tableau `$operations` et la fonction `calcule()` précédents en incluant le fichier précédemment créé.

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

```
<?php
    // calculs et inclusions ici
?><!doctype html>
<html>

<head>
    <meta name="author" content="Votre Nom"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Titre de la page</title>
</head>
<body>
<?php
    // affichage là
?>
</body>
</html>
```

### Exercice 2. Formulaire

1. Créer un fichier `ex2-calc.php` qui, s'il a reçu par POST un paramètre "calc", un paramètre "n1", un paramètre "n2" et un paramètre "oper" applique la fonction "calcule" à ces paramètres après avoir effectué les vérifications de sécurité d'usage, puis affiche le résultat.

- Créer un fichier `ex1-formulaire.php` dont le rôle sera d'afficher un formulaire HTML contenant deux champs texte (destinés à envoyer les nombres `n1` et `n2`) et une liste déroulante "oper" permettant de choisir l'une des quatre opérations évoquées ci-dessus.
- Modifier le fichier `ex1-calc.php` pour que, s'il n'a pas reçu de paramètres valides, il redirige automatiquement vers la page `ex1-formulaire.php`. Rajouter également un lien vers cette page en dessous de l'affichage de la valeur calculée.

## 2 Connexion

### Exercice 3. Authentification

- Créer un fichier `ex3-bdd-users.inc`, dans lequel il faudra initialiser le tableau `$users` à deux dimension suivant :

id	username	hash	realname
0	admin	21232f297a57a5a743894a0e4a801fc3	Admin
1	toto	f71dbe52628a3f83a77ab494817525c6	T. Toto
2	foo	e10adc3949ba59abbe56e057f20f883e	Foo Bar

Tel que `$users[1]["username"]` vaille « toto », `$users[0]["realname"]` vaille « Admin », etc.

- Créer une fonction `verifie_utilisateur($user, $pass)` qui vérifie si le tableau comporte bien une ligne dans laquelle l'utilisateur est `$user` et le hash est le md5 de `$pass` et renvoie l'id de l'utilisateur s'il est dans la liste, ou -1 sinon.
- Créer un fichier `ex3-connexion.php` qui, s'il reçoit les données d'un formulaire de connexion, vérifie si ces données sont correctes et affiche le résultat, et s'il ne reçoit pas les données d'un formulaire de connexion, affiche un formulaire de connexion avec deux champs pour le nom d'utilisateur et le mot de passe.
- On veut maintenant que le formulaire puisse être utilisé depuis n'importe quelle page. Regrouper le code dans des fonctions dans un fichier `ex3-fonctions.inc` de manière à rendre cela possible en limitant la duplication de code.

### Exercice 4. Session

L'objectif est maintenant de faire en sorte que l'authentification soit mémorisée dans la session.

- Modifier les fichiers de l'exercice précédent de manière à ce qu'ils utilisent le mécanisme de session pour se souvenir si l'utilisateur s'est connecté ou non.
- Créer une page `ex4-etat.php` affichant une page HTML indiquant si l'utilisateur est connecté ou non.
- Créer une page `ex4-deconnexion.php` fermant la session.

## 3 Upload de fichiers

### Exercice 5. Gestion d'un avatar

Nous allons maintenant utiliser les fonctionnalités d'upload de fichiers en les appliquant à la gestion d'une image d'avatar.

- Dans un fichier `ex5-avatar.php`, tester si le fichier `images/avatar.png` existe. Si c'est le cas, afficher une page HTML affichant l'image en question. Sinon, afficher une page HTML indiquant qu'il n'y a pas de tel fichier.
- Modifier le fichier précédent pour qu'il affiche en plus un formulaire permettant de choisir une nouvelle image à uploader.
- Modifier enfin le fichier pour que, s'il reçoit les données du formulaire, teste si la nouvelle image est au bon format, puis l'enregistre dans `images/avatar.png`.
- Rajouter sur la page, uniquement dans le cas où il y a déjà un avatar, un formulaire permettant de supprimer l'avatar. Et gérer sa suppression en cas de réception des données de ce formulaire.