

## TP PHP 3 - Sessions, fichiers, XML

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter. Pensez à consulter le site [PHP](#).

### Exercice 1. (Cookies, sessions)

1. Le formulaire **cookies.php** permet au visiteur de soumettre les couleurs de fond et de texte qu'il souhaite pour la page. Les **codes CSS** peuvent être donnés sous forme symbolique, hexadécimale ou RGB. Le script doit appliquer le style demandé à l'ensemble de la page et enregistrer les valeurs fournies dans deux cookies valables 10 secondes.

Le visiteur pourra ensuite :

- resoumettre de nouveaux codes auquel cas le style de la page et la valeur des cookies seront réactualisés,
- ou resoumettre le formulaire sans spécifier de codes. S'il soumet avant expiration des cookies, le script régénérera le formulaire avec la même mise en forme en se servant des cookies. Sinon, la page reprendra ses couleurs par défaut (noir sur blanc).

Complétez **cookies.php** pour obtenir le comportement attendu. Notez que Firefox permet d'inspecter et de supprimer individuellement les cookies comme illustré en Figure 1.

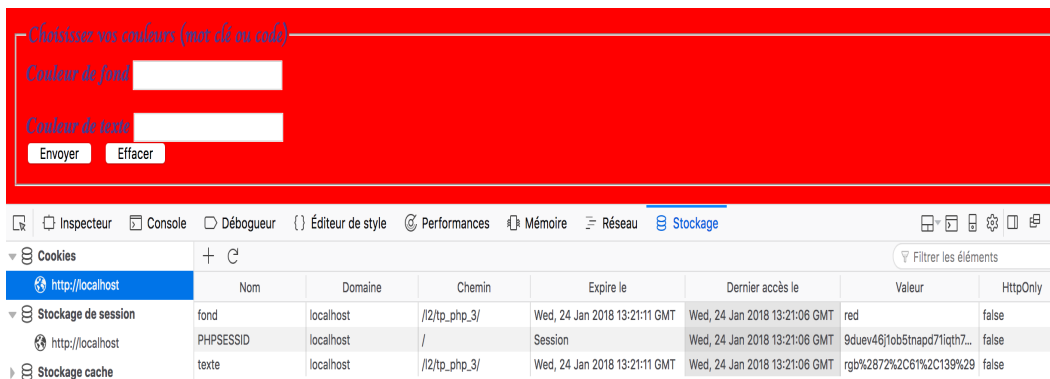


FIGURE 1 – Page customisée par cookies

2. Complétez le script **sessions.php** pour enregistrer les préférences du visiteur dans des données de session et non plus dans des cookies. Les données de session resteront valides jusqu'à soumission de nouveaux codes par le visiteur - aucun délai d'expiration n'est à programmer. Complétez ensuite le fichier **sessions-B.php** (accessible à partir de la page principale) pour que lui soit appliquée la mise en forme demandée tel qu'illustré en Figure 2.

Choisissez vos couleurs (mot clé ou code)

Couleur de fond

Couleur de texte

Contenu de la page principale

[Lien vers la page B qui aura ces couleurs](#)

FIGURE 2 – Page annexe customisée par données de session

Exercice 2. (Fichiers CSV/XML, sessions)

L'objectif est de développer pas à pas différents scripts permettant d'inscrire des employés et d'en afficher la liste. Cette liste sera d'abord stockée sous forme de fichier au format CSV. Vous développerez ensuite un fichier au format XML.

1. Complétez le fichier `accueil.php` de sorte que l'hyperlien redirige vers le fichier `employee_form.php` tel qu'illustré en Figure 3.



FIGURE 3 – Lien de redirection

2. Le fichier `employee_form.php` est un formulaire permettant d'inscrire un nouvel employé (Figure 4) ou de revenir à la page d'accueil. Chaque inscription met immédiatement la page à jour en affichant la liste des employés sous forme de tableau HTML (Figure 5).

Inscrire un employé

Nom :

Salaire :

Age :

[Retour à l'accueil](#)

FIGURE 4 – Formulaire d'inscription

Inscrire un employé

Nom :

Salaire :

Age :

[Retour à l'accueil](#)

ID	NOM	SALAIRE	AGE
1	superman	20000	80

FIGURE 5 – Inscription d'un employé

Toute inscription autorisée est ajoutée dans le fichier `employees.csv` dont le format est le suivant (les sauts de ligne correspondent au caractère `\n`) :

```
1;superman;20000;80
2;batman;10000;70
```

Afin de créer et mettre à jour ce fichier, utilisez les fonctions :

- `fopen` (en mode lecture ou ajout), `fclose` et `flock` pour verrouiller/déverrouiller l'accès au fichier.

- **fgetcsv** pour lire et décomposer chaque ligne du fichier.

Vous prendrez garde à l'inscription des employés dénommés :

```
<h1>My HTML</h1>
<script>alert('Gotcha!');window.history.back();</script>
```

3. On souhaite rejeter les doubles inscriptions en cours de “session” où deux individus seront considérés identiques s'ils ont le même nom et le même âge. Un exemple de rejet est illustré en Figure 6.

FIGURE 6 – Rejet d'inscription

Modifiez les fichiers en utilisant le mécanisme des sessions afin de mémoriser chaque inscription effectuée et de rejeter toute nouvelle tentative (un véritable test consisterait à analyser le fichier à chaque tentative). Notez que Firefox permet de supprimer le cookie de session PHPSESSID généré par PHP.

4. Créez le script **employees2xml.php** qui génère un fichier XML d'employés à partir du fichier CSV **employees.csv**. Le fichier XML devra être conforme à la DTD **employee.dtd**. Exemple de fichier valide :

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE employees SYSTEM 'employees.dtd'>
<employees>
  <employee>
    <id>1</id>
    <name>superman</name>
    <salary>20000</salary>
    <age>80</age>
  </employee>
  <employee>
    <id>2</id>
    <name>batman</name>
    <salary>10000</salary>
    <age>70</age>
  </employee>
</employees>
```

5. Modifiez ensuite la page d'accueil de sorte que :

- le tableau des noms d'employés s'affiche s'il en existe (voir Figure 7),
- chaque nom soit un hyperlien qui, par invocation du script **get\_employee.php**, permet de récupérer toutes les informations sur cet employé (voir Figure 8).

Le script **get\_employee.php** doit lire et extraire du fichier XML les informations relatives à l'employé demandé. Utilisez l'API **SimpleXML** ou l'API **DOM** avec requête **XPath** pour rechercher les bons éléments.



NOM
superman
batman

FIGURE 7 – Listing des employés

employee: id=2 name=batman salary=10000 age=70

FIGURE 8 – Informations sur un employé