

# Cookies ; connexion à une base de données

Licence Informatique 3ème année

Alexandre Niveau — Jean-Marc Lecarpentier

## Cookies ; connexion à une base de données

### Notes de cours

- [Cookies HTTP](#)
- [Interaction PHP-SGBD avec PDO](#)

## Travail personnel

### Objectifs

Le premier exercice vise à vous faire comprendre comment fonctionnent les cookies, et à les appliquer sur un exemple simple. Le deuxième vous fait construire un site simple utilisant une base de données. Le troisième, destiné aux plus rapides, montre un cas d'application classique des cookies, l'enregistrement de préférences de l'internaute.

---

### Exercice 1 — Cookies

#

#### 🐼 Manipulation basique

1. Créer un script PHP `affichage_cookies.php` sur votre serveur, et y mettre le contenu suivant :

```
<?php  
var_export($_COOKIE);
```

2. Aller à l'URL correspondante avec Firefox : normalement, le tableau est vide (pas forcément, mais ce n'est pas grave).
3. Créer un autre script PHP `creation_cookie.php` sur votre serveur. Dedans, utilisez la fonction `header()` pour ajouter l'en-tête HTTP Set-Cookie : `prenom=VOTREPRENOM` (avec votre prénom à la place de `VOTREPRENOM`).
4. Accéder à l'URL correspondant à votre script `creation_cookie.php`, puis à celle de `affichage_cookies.php`. Constaté que votre cookie apparaît maintenant dans le tableau. **Grâce au cookie**, on a, en quelque sorte, **transmis une information d'une page à une autre** (sur le même site).
5. Rien n'empêche d'utiliser `header()` pour positionner des cookies, comme on l'a fait, mais la fonction PHP `setcookie()` est plus pratique, notamment pour préciser des options. Remplacer l'appel à `header` par un appel à `setcookie` (voir la doc ou le cours) avec comme troisième paramètre le code `time()+10`. Accéder à `creation_cookie.php`, puis à `affichage_cookies.php` ; le cookie devrait toujours être là. Attendre 10 secondes et actualiser `affichage_cookies.php` : le cookie devrait avoir disparu. Si vous ne comprenez pas pourquoi, demander à votre chargé-e de TP.

## Observation

1. Ouvrir le *moniteur réseau* de Firefox (clic droit sur la page > Inspecter > onglet « Réseau » ; raccourci : control-shift-E).
2. Accéder à `creation_cookie.php`, et observer les en-têtes HTTP de la réponse : vous devriez trouver le Set-Cookie ajouté par PHP, avec sa date d'expiration.
3. Actualiser puis accéder rapidement (avant 10 secondes) à `affichage_cookies`, et observer les en-têtes HTTP de la requête (pas de la réponse !) : vous devriez voir le cookie transmis.
4. Ouvrir l'*inspecteur de stockage* de Firefox (clic droit sur la page > Inspecter > onglet « Stockage » ; raccourci : shift-F9). Vous pouvez voir les cookies enregistrés par le navigateur pour les domaines utilisés par la page, même ceux qui sont expirés. Essayer de modifier la date d'expiration et la valeur du cookie, et actualiser la page.
5. Conclure sur la confidentialité et la sécurité de l'utilisation des cookies.

## Utilisation

1. Écrire un script `date.php` qui crée une page HTML simple (mais valide !) affichant la date et l'heure courantes.
2. **En utilisant un cookie**, faire en sorte que *si l'internaute a déjà visité la page*, alors la date et l'heure de sa dernière visite sont également affichées.

# Cookie

Nous sommes le dimanche 01 octobre 2023 à 22:11:32.

Votre dernière visite remonte au mardi 20 mars 1984 à 05:43:17.

Exemple de résultat

3. Modifier le code précédent pour que le site ne se souvienne de l'internaute que pendant un an. Vérifiez la date d'expiration du cookie avec le moniteur réseau de Firefox (ctrl-shift-E, voir plus haut).
4. Ouvrir l'inspecteur de stockage de Firefox (shift-F9, voir plus haut). Comment se comporte la page si vous supprimez le cookie ? Et si vous mettez du HTML dedans ?

## Exercice 2 — Galerie d'images

#

Cet exercice vise à découvrir le fonctionnement de PDO en affichant une galerie d'images cliquables.

### Description de la base et mise en place

Les informations sur les images à afficher sont dans une table `images` en base de données. Les images elles-mêmes **ne sont pas** dans la base de données : elles sont hébergées par [Flickr](#). La description de la table est la suivante :

```
mysql> desc images;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
title	varchar(255)	YES		NULL	
creation_date	datetime	YES		NULL	
thumb_url	varchar(255)	YES		NULL	
image_page_url	varchar(255)	YES		NULL	
author_name	varchar(255)	YES		NULL	
author_url	varchar(255)	YES		NULL	

7 rows in set (0.00 sec)

Il y a trois URL :

- `thumb_url` est celle de la miniature (*thumbnail*) de l'image,

- `image_page_url` est celle de la page Flickr de l'image,
- `author_url` est celle de la page Flickr de l'auteur.

**NB : il n'y a pas d'URL de l'image en grand format**, on n'utilise que la miniature dans cet exercice.

Le fichier [images.sql](#) contient le code SQL permettant de créer et remplir cette table.

## 🐼 Galerie

1. Utiliser le fichier fourni pour créer la table `images` dans une de vos bases de données MySQL. Voir [cette page de la FAQ du département](#) pour savoir comment y accéder. À noter en particulier, le mot de passe initial de votre BD est dans le répertoire Protected de votre \$HOME. Si vous n'avez pas l'habitude, **le plus simple** pour ajouter la table à votre BD est d'utiliser la fonction d'import sur l'application phpMyAdmin, accessible à `https://dev-LOGIN.users.info.unicaen.fr/phpmyadmin`. Attention, il faudra peut-être créer une base, et tous les noms ne sont pas autorisés !
2. Dans un script `galerie.php`, créer une instance de PDO qui se connecte à la base, et la configurer pour qu'elle renvoie des exceptions en cas de problème lors d'une requête.
3. Avec la méthode `query()` de votre instance de PDO, exécuter la requête suivante : `test de requête!`. C'est bien sûr une requête incorrecte, donc **vous devez absolument obtenir une exception** : si rien ne s'affiche, c'est que vous n'avez pas configuré correctement votre instance de PDO, relisez la question précédente !
4. Afficher une liste des titres de toutes les images de la base. Attention, les identifiants ne sont pas contigus (il y a des trous). Ça ne doit poser aucun problème si votre code est correct.
5. Modifier le code pour qu'il affiche la miniature de l'image à côté (ou à la place) de chaque titre.
6. **(Optionnel)** Ajouter un peu de CSS pour que la liste ressemble davantage à une galerie. (Ne pas y passer plus de cinq minutes, ce n'est pas l'objet de l'exercice !)

## 🐼 Pages individuelles des images

On va maintenant faire un autre script permettant d'avoir une page pour chaque image, avec un peu plus d'infos.

1. Avant de commencer le script, modifier `galerie.php` pour ajouter à chaque image un lien vers l'URL `image.php?id=34` (avec à la place de 34 l'identifiant de l'image).

2. Créer le script `image.php`, qui affiche toutes les infos sur l'image dont l'identifiant a été passé en paramètre : titre, miniature, lien vers la page Flickr de l'image, date, et nom de l'auteur avec un lien vers sa galerie Flickr. **NB** : on n'affichera pas l'image en taille réelle (son URL n'est pas stockée dans notre base), c'est un peu bizarre de n'avoir que la miniature mais c'est comme ça.
3. Y a-t-il des redondances de code entre les deux scripts `galerie.php` et `image.php` ? Si oui, essayez de mutualiser les parties communes dans un troisième fichier, qui sera inclus depuis les deux scripts.
4. Que se passe-t-il si vous allez sur la page d'URL « `image.php?id=1; DROP TABLE images; --` » ? Si ce n'est déjà fait, protégez votre BD contre les injections SQL en utilisant uniquement des *requêtes préparées*.

### Optionnel, pour les rapides

1. Faire en sorte qu'il soit possible de contrôler le nombre d'images affichées dans la galerie avec un paramètre `size` dans l'URL.
2. Faire en sorte qu'il soit possible de contrôler quelles images sont affichées dans la galerie avec un paramètre `ordre`, qui modifie l'ordre d'affichage (chronologique, chronologique inverse, ou alphabétique) et des paramètres de filtrage comme avant et après (une certaine date) ou auteur, par exemple.

---

## Exercice 3 (optionnel) — Stockage de préférences avec un cookie #

1. Écrire un script PHP qui crée une page HTML simple avec un titre et du texte.
2. Créer deux fichiers CSS (très simples !) pour cette page, `clair.css` et `sombre.css`, qui donnent des rendus très différents, le premier étant un thème sur fond clair et le second sur fond sombre.
3. Ajouter un formulaire sur la page permettant à l'internaute de choisir le thème qu'il ou elle préfère, et enregistrer cette préférence dans un cookie, afin que la page s'affiche toujours avec le thème choisi. NB : pour vos tests et debugging, n'oubliez pas l'outil « inspecteur de stockage » de Firefox, qui permet de supprimer facilement un cookie.