

LES COOKIES

L'appel d'une page à une autre passe par le protocole HTTP. C'est un protocole non connecté (*protocole sans états*). Cela signifie que chaque requête est traitée indépendamment des autres et qu'aucun historique des différentes requêtes n'est conservé. Ainsi le serveur web ne peut pas se "souvenir" de la requête précédente. La nouvelle page ne peut donc pas « récupérer » des informations de pages précédentes.

Une façon de pallier à ce problème est l'utilisation des **Cookies**.

Donc un cookie est un moyen d'enregistrer des informations sur le poste du client.

L'utilisation du protocole HTTP est obligatoire pour pouvoir tester les exemples suivants. (Consulter le guide d'installation de XAMP dans la séance PHP pour installer et configurer un serveur http sur votre poste).

FONCTIONNEMENT DU COOKIE

Un *cookie* peut être généré soit par une application côté serveur (PHP par exemple), soit côté client avec du JavaScript.

Un *cookie* contient 5 données :

- **Une clé**
- **Une valeur**
- Une date d'expiration (optionnel)
- Un domaine (optionnel)
- Une arborescence. (optionnel)

La paire *clé=valeur* permet de stocker l'information souhaitée.

La *date d'expiration* donne la durée de vie de cette information. (1 jour par défaut)

Le *domaine* est le nom du serveur par défaut. Autrement, il faut spécifier le nom de l'hôte (exemple : *domaine.fr*).

L'arborescence spécifie à partir duquel le cookie est valide.

Dès qu'une page Web est appelée, tout cookie enregistré qui fait parti du même domaine et de la même arborescence est ajouté à l'entête http.

LIMITES DES COOKIES

Un cookie ne peut pas excéder 4 Ko.

Une application ne peut dépasser 20 cookies.

Tout navigateur est limité à 300 cookies au maximum (ou moins).

UTILISATION DES COOKIES

L'objet *document* fournit l'objet *cookie*.

Il permet d'écrire et lire les cookies.

Exemple de syntaxe d'écriture d'un cookie :

```
document.cookie = "nom=toto";expires=200000 ;path=/
```

nom représente la clé, toto représente la valeur.

expires donne la date d'expiration, 2000000 une date en milliseconde

path référence l'arborescence

Méthode qui permet de créer ou modifier un cookie

```
function createCookie(name,value,days) {  
    // permet de créer un cookie  
    if (days) {  
        // si le nombre de jour est renseigné, on le transforme en millieme de  
        seconde  
        var date = new Date();  
        date.setTime(date.getTime()+(days*24*60*60*1000));  
        var expires = "expires="+date.toGMTString();  
    }  
    else var expires = "";  
    //le cookie doit contenir clé=valeur;expires=temps;path=nomDomaine  
    document.cookie = name+"="+value+"; " + expires+"; path="/;  
}
```

Pour lire les cookies, il faut utiliser la syntaxe suivante :

```
var listeCookies = document.cookie;
```

La variable *listeCookies* récupère l'ensemble des paires clé=valeur, séparées par des ';'.
ATTENTION:

En écriture, document.cookie permet de créer un cookie

En lecture, document.cookie renvoie la liste de tous les cookies

Méthode qui permet de lire la valeur d'un cookie

```
function readCookie(name) {  
    // on récupère tous les cookies du site en une fois, séparé par ;  
    // on range dans un tableau chaque cookie  
    var listeCookies = document.cookie.split(';');  
    for(let i=0;i < listeCookies.length;i++) {  
        // pour chaque cookie, on sépare en tableau la clé et la valeur  
        var unCookie = listeCookies[i].split("=");  
        // si la clé correspond au cookie cherché, on renvoi la valeur  
        if (unCookie[0]==name) return unCookie[1];  
    }  
    return null;  
}
```

SUPPRIMER UN COOKIE

Pour supprimer un cookie, il faut mettre la valeur à *null* ou vide ; et modifier la date d'expiration à une date antérieure à la date du jour.

```
function eraseCookie(name) {  
    // pour supprimer un cookie, on le périmé  
    createCookie(name,"",-1);  
}
```

VOIR UN COOKIE

Les cookies sont stockés dans le profil.

Dans Firefox, On y accède par

C:\Users\VotreMatricule\AppData\Roaming\Mozilla\Firefox\Profiles\NumeroAleatoire.default

Le fichier de stockage est cookies.sqlite. Il peut être lu avec 'DB Browser for SQLite' par exemple

Dans Chrome, on y accède par

C:\Users\1701871\AppData\Local\Google\Chrome\User Data\Default

Le fichier de stockage est cookies. Il peut être lu avec 'DB Browser for SQLite' par exemple

Exercices

Créer une page Html qui affiche le nombre de fois qu'elle a été visitée par le même utilisateur.

Utiliser pour se faire un cookie qui conservera cette information.

Ajouter également un bouton qui permet de réinitialiser le nombre de visites à 1.

Résultat à obtenir :

