

## Écriture d'un cookie

La création d'un cookie se fera par l'intermédiaire d'une fonction JavaScript que nous allons créer (peu importe son nom) comme suit :

```
/* Définition de la fonction JavaScript de création d'un cookie */
function creationCookie(nomCookie, valeurCookie, expirationCookie,
cheminCookie, domaineCookie, securiteCookie)
{
    document.cookie = nomCookie + ' = ' + escape(valeurCookie) + ' ' +
    ((expirationCookie == undefined) ? '' : ('; expires=' +
    expirationCookie.toGMTString())) + ((cheminCookie == undefined) ? '' :
    ('; path=' + cheminCookie)) + ((domaineCookie == undefined) ? '' :
    ('; domain=' + domaineCookie)) + ((securiteCookie == true) ? ';'
    securiteCookie' : '');
}


// La fonction escape réalise un encodage des caractères spéciaux en les
// remplaçant par une séquence %remplacement où remplacement est un nombre
// sous forme hexadécimale

/* Création d'un cookie par appel de la fonction creationCookie */
var dateExpiration = new Date();
dateExpiration.setTime(dateExpiration.getTime() + 300 * 1000);
// NB : - dateExpiration est un objet de type date (par défaut initialisé
//        à la date système)
//        - La date de fin de validité du cookie (il sera supprimé) est
//        calculée à partir de la date système (5 minutes plus tard ici,
//        c'est-à-dire 300 secondes)
creationCookie("monCookie", "Editions ENI ", dateExpiration, "/");
// Nom du cookie : monCookie
// Valeur du cookie : Editions ENI
// Expiration du cookie : Dans 5 minutes (300 secondes)
// Accessibilité : Accessible à tout le site

/* Message de contrôle */
document.write("Le cookie monCookie (contenant Editions ENI) a normalement
été écrit sur le disque dur");
```

Pour la fonction `creationCookie` deux paramètres obligatoires sont requis, le nom et la valeur du cookie.

La valeur du cookie peut contenir des caractères spéciaux. La fonction JavaScript `escape()` permet de remplacer ces caractères spéciaux (accentuations...) par des codes hexadécimaux. Bien évidemment, à la relecture du cookie un codage inverse devra être réalisé par la fonction `unescape()`.

 Selon la source [developer.mozilla.org](https://developer.mozilla.org), les fonctions `escape` et `unescape` permettent de protéger des chaînes de caractères en encodant et décodant les caractères spéciaux. Avec `escape`, chaque caractère qui peut être interprété comme un caractère spécial est remplacé par la forme `%NN` où `NN` représente la valeur en hexadécimal du caractère dans le jeu de caractères ISO Latin 1 (ISO-8859-1).

Le codage par la fonction `escape()` n'a pas été fait ici car le nom prévu pour le cookie (`monCookie`) ne le requiert pas.

dateExpiration est un objet de type date qui n'est pas obligatoire. Nous y stockons la date système plus un délai de 5 minutes comme suit :

```
var dateExpiration = new Date();  
dateExpiration.setTime(dateExpiration.getTime() + 300 * 1000);
```

En cas d'absence de ce paramètre facultatif, le cookie sera automatiquement supprimé à la fin de la session (séquence de code, en général du PHP, pendant laquelle la conservation des cookies est prévue).

Le paramètre cheminCookie précise le chemin sur le serveur. Avec la valeur /, le cookie sera utilisable sur tout le site. Si le paramètre n'est pas précisé ou s'il vaut null, la valeur par défaut sera l'URL de la page Web.

Le paramètre domaineCookie joue sensiblement le même rôle que cheminCookie mais cette fois-ci pour le domaine signalé.

Enfin, il est possible de fournir un dernier paramètre, lié à la sécurité. Le paramètre securiteCookie devra être égal à true pour une utilisation en connexion sécurisée https.

À l'exécution du script, nous obtenons :

