

# PHP - Les Cookies

Les **cookies** constituent un mécanisme fondamental du web permettant de stocker des informations côté client, dans le navigateur, afin de conserver un état ou des préférences entre plusieurs requêtes HTTP. En PHP, la gestion des **cookies** est simple mais requiert une compréhension précise de leur fonctionnement, de leur portée et de leurs limites.

## 1 - Définition et Utilité des Cookies

- Un **cookie** est un petit fichier texte enregistré par le navigateur à la demande du serveur.
- Il permet de *mémoriser des informations* sur l'utilisateur entre différentes visites ou pages, facilitant ainsi la personnalisation de l'expérience ou la gestion de sessions légères.

Exemple :

```
<?php
// Création d'un cookie valable 1 heure
setcookie('nom_utilisateur', 'Jean', time() + 3600);
?>
```

php

### 1.1 - Portée et Durée de Vie

Un **cookie** possède une **portée** (domaine, chemin) et une **durée de vie** (date d'expiration).  
Par défaut, un cookie expire à la fermeture du navigateur, sauf si une date d'expiration est spécifiée.

#### 1.1.1 - Paramètres de `setcookie`

La fonction `setcookie()` accepte plusieurs paramètres :

- nom : le nom du cookie,
- valeur : la valeur du cookie,
- expiration : timestamp Unix indiquant quand le cookie expire,
- chemin : le chemin sur le serveur où le cookie est accessible,
- domaine : le domaine pour lequel le cookie est valide (si "/" est spécifié, le cookie est accessible sur tout le domaine),
- secure : si `true`, le cookie n'est envoyé que sur des connexions HTTPS,
- httponly : si `true`, le cookie n'est pas accessible via JavaScript.

Exemple :

```
<?php
// Cookie accessible uniquement via HTTPS et non accessible en JavaScript
setcookie('token', 'abc123', time() + 3600, '/', 'monsite.com', true, true);
?>
```

php

## Lecture et Utilisation des Cookies

Les **cookies** envoyés par le navigateur sont accessibles via le tableau associatif `$_COOKIE`.

Il est possible de lire leur valeur à chaque requête, tant que le cookie n'a pas expiré.

Exemple :

```
<?php
if (isset($_COOKIE['nom_utilisateur'])) {
    echo "Bonjour, " . $_COOKIE['nom_utilisateur'];
}
?>
```

php

## 2.1 - Modification et Suppression

Pour **modifier** un cookie, il suffit de réutiliser `setcookie()` avec le même nom et une nouvelle valeur.

Pour **supprimer** un cookie, on lui attribue une date d'expiration passée.

Exemple :

```
<?php
// Suppression du cookie
setcookie('nom_utilisateur', '', time() - 3600);
?>
```

php

## Sécurité et Bonnes Pratiques

Les **cookies** peuvent contenir des informations sensibles. Il est donc essentiel d'appliquer des mesures de sécurité pour éviter les risques de vol ou de falsification.

### 3.1 - Sécurisation des Cookies

- Utiliser le paramètre `secure` pour restreindre l'envoi du cookie aux connexions HTTPS.
- Employer le paramètre `httponly` pour empêcher l'accès au cookie via JavaScript ( `XSS` ).
- Éviter de stocker des données sensibles en clair dans les cookies.

### 3.2 - Limites des Cookies

- Taille limitée (environ 4 Ko par cookie).
- Nombre limité par domaine.
- Dépendance à l'acceptation des cookies par l'utilisateur.

## À RETENIR

Points essentiels :

- Les **cookies** permettent de stocker des informations côté client pour persister des données entre requêtes.
- La fonction `setcookie()` sert à créer, modifier ou supprimer un cookie.
- Les cookies sont accessibles via le tableau `$_COOKIE` .
- Il est crucial de sécuriser les cookies avec les options `secure` et `httponly` .
- Les cookies ne doivent pas contenir d'informations sensibles non chiffrées.

Concept	Description
<code>cookie</code>	Donnée stockée côté client pour conserver un état ou une préférence.
<code>setcookie()</code>	Fonction PHP pour créer, modifier ou supprimer un cookie.
<code>\$_COOKIE</code>	Tableau associatif contenant les cookies reçus par le serveur.
<code>secure</code>	Option pour restreindre l'envoi du cookie aux connexions HTTPS.
<code>httponly</code>	Option pour interdire l'accès au cookie via JavaScript.
<code>expiration</code>	Date à laquelle le cookie devient invalide.