

COOKIES ET SESSIONS

1. LES COOKIES

La navigation sur un site web peut gagner en efficacité par la mémorisation de données personnelles liées à l'utilisateur, par exemple : l'historique de navigation, les pages préférées, une configuration d'affichage, un identifiant de connexion, le contenu d'un panier ...

Un cookie est un petit fichier déposé par un site WEB sur l'appareil de l'internaute, sous réserve que le navigateur de celui-ci l'accepte. Un cookie contient des informations textuelles utilisables par le site WEB lors d'un retour ultérieur de cet internaute.

Un cookie peut être écrit et lu par un script PHP.

Les cookies font l'objet de limites d'emploi :

- Un cookie n'est accessible que par le site qui l'a déposé.
- Les informations stockées dans un cookie sont disponibles pendant une durée limitée (1 heure, 1 an, 30 secondes, jusqu'à la fin de la connexion, ...).
- Le nombre de cookies qu'un site peut déposer est limité : selon le navigateur, la limitation s'applique à partir de 20 cookies.
- Le contenu des informations stockées dans un cookie ne peut pas dépasser 4096 octets.

Remarque : Les cookies peuvent être supprimés à partir des paramètres du navigateur.

- Pour FF : Vie privée et sécurité → Cookies et données de sites → Effacer les données ...
- Pour Chrome : Confidentialité et sécurité → Effacer les données de navigation.

1.1. Principe

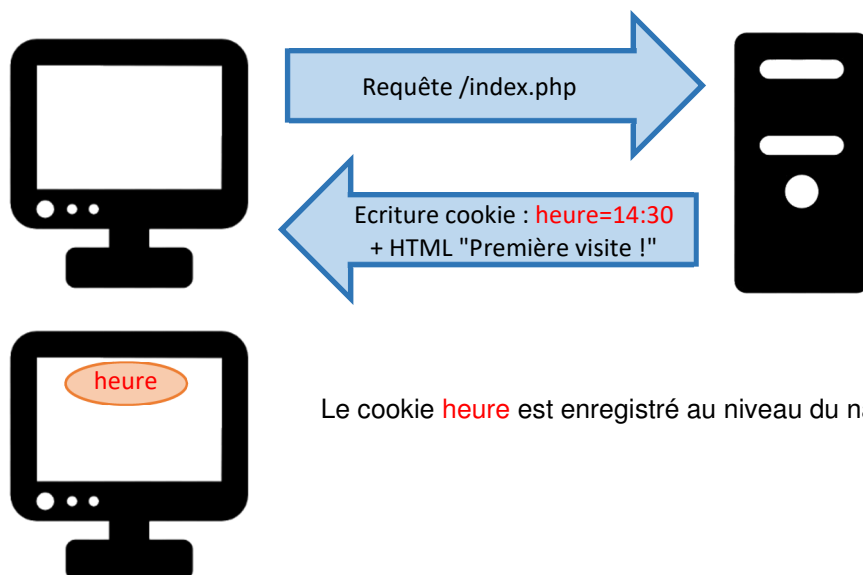
L'utilisation de cookies obéit aux règles suivantes :

1. L'écriture des cookies se fait par le serveur avec de la fonction PHP `setcookie()`. Un cookie est envoyé vers le client tout au début de la connexion, avant l'envoi de la première balise HTML.
2. Les cookies déjà stockés sur un client sont renvoyés vers le serveur qui les a déposés. Cet envoi se fait automatiquement à chaque fois que le client émet une requête vers ce serveur.
3. Les cookies récupérés par le serveur sont accessibles dans le tableau `$_COOKIE`. Ce tableau est utilisable dans le script PHP exécuté par la page demandée par le client.

Exemple : Enregistrement de l'heure de connexion dans un cookie.

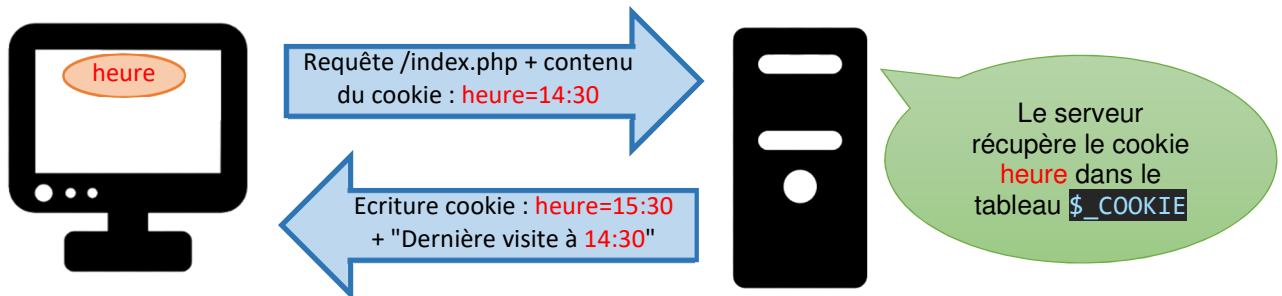
Un client se connecte pour la première fois sur le site `www.truc.fr` à 14h30 :

- Le serveur écrit le cookie **heure** avec la valeur **14:30** sur le client.
- Le serveur envoie le contenu HTML de la page `index.php` : "C'est votre première visite !".



Le client se connecte à nouveau sur le serveur `www.truc.fr` à 15h30 :

- Le client envoie automatique le cookie **heure** avec la requête
- Le serveur récupère l'heure de la précédente visite dans `$_COOKIE["heure"]`.
- Le serveur écrit une nouvelle valeur (**15:30**) dans le cookie **heure** sur le client.
- Le serveur envoie le contenu HTML de la page `index.php` : "Votre dernière visite était à 14h30 !".



1.2. Ecriture de cookies

Important : Pour écrire un cookie sur le poste client, il est impératif qu'aucun contenu HTML n'ait été envoyé au client avant l'envoi du cookie. L'écriture d'un cookie devra par conséquent être effectuée au début du script, avant l'envoi d'une quelconque balise HTML, ou d'un simple texte.

La fonction `setcookie()` permet de déposer un cookie sur le poste de l'internaute.

Prototype : `setcookie(string name[, string value, int expire, ...]) : bool`

Avec :

- `name` : nom du cookie ;
- `value` : valeur (texte) stockée dans le cookie ;
- `expire` : date d'expiration du cookie (instant UNIX) ;

La fonction `setcookie()` retourne `TRUE` si l'instruction a pu être exécutée et `FALSE` dans le cas contraire.

Suite à l'appel de cette fonction, un cookie portant le nom `name` et contenant la valeur `value` est envoyé vers le navigateur; s'il existe déjà un cookie portant le même nom, ce dernier est mis à jour avec la nouvelle valeur. Le paramètre `expire` permet de déterminer la date d'expiration du cookie ; si ce paramètre n'est pas spécifié (ou égal à 0) le cookie expire à la fin de la session (c'est-à-dire, lorsque le navigateur est refermé).

Exemple : `setcookie('prenom', 'Rémi', time()+3600);`

Exemple sécurisé : L'option `httpOnly` permet de rendre un cookie inaccessible en JavaScript.

`setcookie('prenom', 'Rémi', time()+3600, null, null, false, true);`

Le dernier paramètre à `TRUE` active l'option `httpOnly`.

Exemple : script d'écriture de deux cookies (fichier `cookie1.php`)

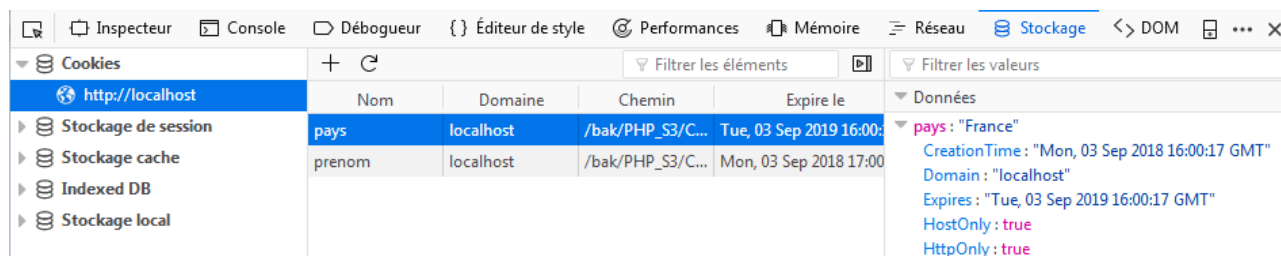
```
<?php
    setcookie('prenom', 'Rémi', time()+3600, null, null, false, true);
    setcookie('pays', 'France', time()+365*24*3600, null, null, false, true);
?>
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Cookie</title>
</head>
<body>
    <p>
```

```

    Bonjour
    <a href="cookie2.php">lien vers la page suivante</a>
  </p>
</body>
</html>

```

Sur FF, les outils de développement web (touche F12), onglet Stockage, permettent de gérer les cookies :



1.3. Lecture de cookies

Lors du chargement d'une page, PHP récupère automatiquement les informations des cookies préalablement déposés sur le poste client par le même site. Ces informations sont placées dans le tableau `$_COOKIE`.

Exemple : script de lecture des deux cookies (fichier `cookie2.php`).

```

<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Cookie</title>
</head>
<body>
  <p>
    Bonjour, votre prénom est : <?=$_COOKIE['prenom']?><br>
    Pays : <?=$_COOKIE['pays']?><br>
  </p>
</body>
</html>

```

Bonjour, votre prénom est : Rémi
Pays : France

Remarques :

- Le cookie est déposé sur le poste de l'internaute par la fonction `setcookie()` du fichier `cookie1.php` puis est renvoyé ultérieurement lors de la visite de n'importe quelle autre page du même site.
- Le cookie n'est donc pas disponible immédiatement pour la page qui le dépose : il faut recharger une page du site pour que le cookie soit récupéré dans le tableau `$_COOKIE`.

1.4. Modification d'un cookie

Pour modifier un cookie existant il suffit de refaire appel à la fonction `setcookie()` en conservant le même nom du cookie et en échangeant sa valeur.

Exemple : `setcookie('pays', 'Canada', time()+365*24*3600, null, null, false, true);`

1.5. Suppression d'un cookie

Pour supprimer un cookie, il faut définir une valeur vide et/ou une date de validité antérieure à la date actuelle, tout en conservant le nom utilisé lors de sa définition :

- Le cookie sur le navigateur est rendu inaccessible
- Sa valeur restera disponible dans `$_COOKIE` jusqu'au rechargement de la page qui contient le code de suppression.

Exemple : `setcookie('prenom', '', 1);`

1.6. Exercice

Créez un compteur de visite dans une page à l'aide d'un cookie.

2. LES SESSIONS

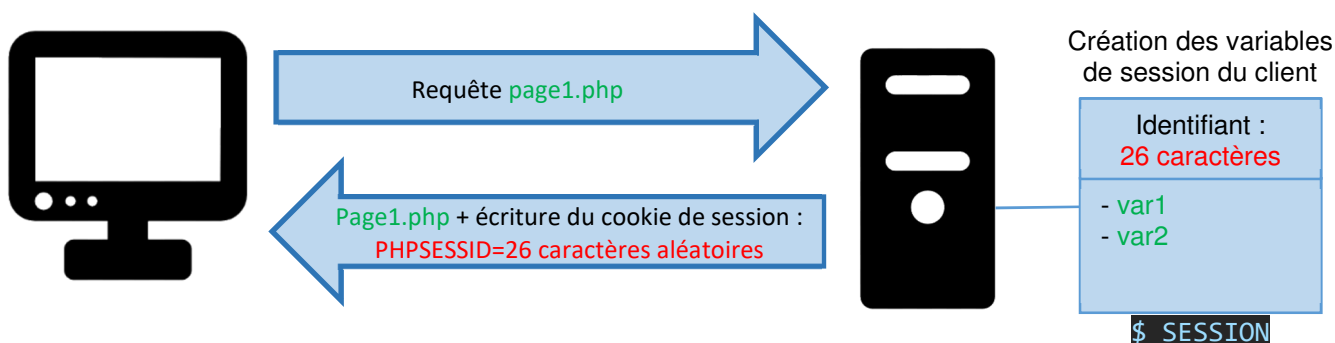
Les variables de sessions permettent de transmettre des données entre toutes les pages d'un site pendant la visite d'un internaute. Les variables sont stockées au niveau du serveur dans le tableau `$_SESSION`.

2.1. Principe

Une session fournit un mécanisme sécurisé d'échange de données entre les pages web d'un même site. Les scripts de chaque page du site peuvent accéder au tableau `$_SESSION` pour y lire et écrire des informations. Une session est définie par un nom (par défaut `PHPSESSID`) et un identifiant unique de 26 caractères aléatoires.

Exemple : Navigation entre deux pages d'un même site, utilisant les variables de session.

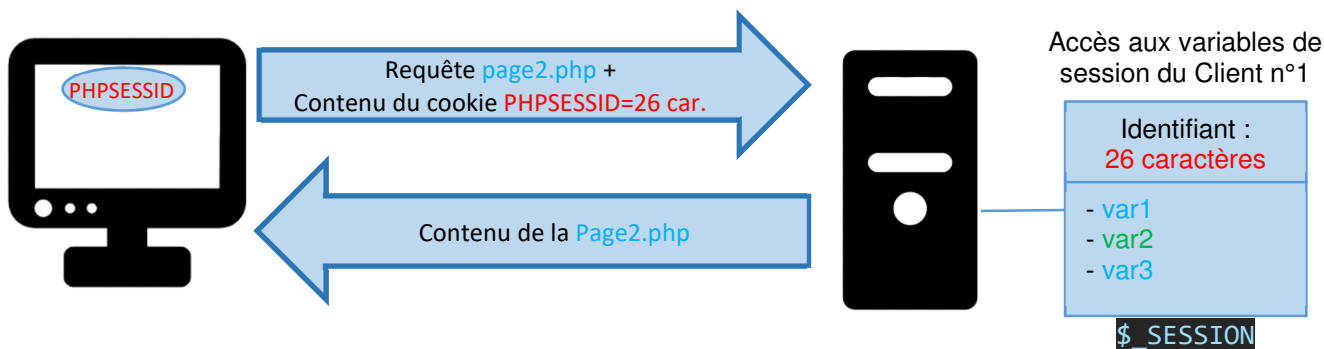
Première connexion : création des variables de session



Le script `page1.php` :

- Initialise une nouvelle session (avec un **identifiant de 26 caractères aléatoires**).
- Envoie **automatiquement** vers le client un cookie de session nommé **PHPSESSID**, contenant l'identifiant aléatoire.
- Enregistre des informations dans le tableau `$_SESSION`, par exemple : `var1` et `var2`.
- Envoie vers le client le contenu HTML généré par `page1.php`

Connexion suivante : Accès aux variables de session



Le script `page2.php` :

- Récupère automatiquement l'identifiant de session à partir du cookie **PHPSESSID**.
- Accède à la session par l'**identifiant de 26 caractères** qui a été récupéré du cookie.
- A accès aux variables de session `var1` et `var2`, au travers du tableau `$_SESSION`.
- Enregistre des informations dans `$_SESSION`, par exemple : modification de `var1` et création de `var3`.
- Envoie vers le client le contenu HTML généré par `page2.php`

→ Tous les scripts des pages visitées par le client peuvent accéder aux variables de la session identifiée par le cookie **PHPSESSID**.

→ Si un autre client se connecte au serveur, une nouvelle session est créée pour lui, avec un autre identifiant.

2.2. Mise en œuvre

Les fonctions ci-dessous permettent de gérer les sessions dans un script PHP :

Nom	Rôle
<code>session_start()</code>	Initialise une nouvelle session, ou accède à une session existante. → Cette fonction doit être utilisée en tout début de chaque page.
<code>session_id() : string</code>	Lit et/ou modifie l'identifiant courant de session
<code>session_name() : string</code>	Lit et/ou modifie le nom de la session (par défaut PHPSESSID)
<code>session_unset()</code>	Supprime le contenu du tableau <code>\$_SESSION</code> .
<code>session_destroy()</code>	Détruit la session en cours. Cette fonction peut être appelée dans une page de déconnexion. Attention : Cette fonction ne supprime pas le cookie de la session.

2.3. Exemple

Première connexion : création des variables de session

La page `session1.php` initialise une session avec `session_start()`, et y enregistre la variable `nom` (cf. `$_SESSION['nom']`).

(Un lien permet de naviguer vers la page nommée `session2.php`).

```
<?php
session_start();
echo 'session_id : '.session_id().'\n';
$_SESSION['nom']='Dupond';
echo $_SESSION['nom'].'\n';
?>
<a href='session2.php'>Vers la page session2.php </a>
```

```
session_id : gpou2kimjkkgt3g5va3jn591m7
Dupond
Vers la page session2.php
```

→ Remarque : A ce stade, un cookie de nom `PHPSESSID` ayant pour valeur l'identifiant de session est automatiquement enregistré sur le poste client par la fonction `session_start()` du script `session1.php`.

Connexion suivante : Accès aux variables de session

Dans la page `session2.php`, la fonction `session_start()` accède à la session déjà présente, dont l'identifiant a été récupéré automatiquement par le cookie `PHPSESSID`.

```
<?php
session_start();
echo 'session_id : '.session_id().'\n';
echo 'Bonjour '.$_SESSION['nom'].'\n';
echo '<a href="session1.php">Vers la page session1.php </a>';
?>
```

```
session_id : gpou2kimjkkgt3g5va3jn591m7
Bonjour Dupond
Vers la page session1.php
```

→ Remarque : Un fichier nommé `sess_gpou2kimjkkgt3g5va3jn591m7` ayant pour contenu `nom|s:6:"Dupond";` est créé dans un dossier du serveur par le script `session1.php` (pour laragon, le dossier est `:/laragon/tmp`)

2.4. Exercice

Tester les deux scripts ci-dessous :

Script : page1.php

```
<?php
    session_start();
    $_SESSION['prenom']='Luc';
    $_SESSION['nom']='Dupond';
    $_SESSION['age']=25;
?>

<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Page 1</title>
</head>
<body>
    <?= 'Identifiant de session : '.session_id() ?>
    <p>
        Bonjour
    </p>
    <a href="page2.php" title="Lien vers page2.php">Lien vers page2</a>
</body>
</html>
```

Script : page2.php

```
<?php
    session_start();
?>

<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Page 2</title>
</head>
<body>
    <?= 'Identifiant de session : '.session_id() ?>
    <p>
        Re-bonjour <?= $_SESSION['prenom'].' '.$_SESSION['nom'] ?><br>
        Vous avez : <?= $_SESSION['age'] ?> ans.
    </p>
</body>
</html>
```

- Observez la création du cookie de session avec les outils de développement web du navigateur.
- Créez la variable de session `pays` dans `page2.php` et affichez-la dans `page1.php` (il faudra ajouter un lien vers `page1.php` dans `page2.php`).

- Créez le script `page3.php` qui affiche toutes les variables de session à l'aide de la fonction `var_dump()` (ajoutez un lien vers `page3.php` dans les scripts `page1.php` et `page2.php` ainsi que les liens vers `page1.php` et `page2.php` dans le script `page3.php`).
- Créez le script `page4.php` dans lequel la session est détruite.
Supprimez également le cookie de session (voir remarque ci-dessous).
Testez les scripts et vérifiez le contenu du cookie de session. (Ajoutez dans chaque script les liens permettant d'atteindre les autres scripts).
Affichez les variables de session juste après la destruction de la session. Que constatez-vous ?
Rechargez la page (touche F5). Que constatez-vous ? Explication ?
Comment faire pour supprimer les variables de session dès le 1^{er} chargement de `page4.php` ?

Remarque : Pour supprimer le cookie de session, il faut utiliser l'instruction :

```
setcookie(session_name(), '', 1, '/');
```

A noter : Le 4^{ème} paramètre de la fonction `setcookie()` indique l'emplacement du cookie. Les cookies de session sont placés sur la racine ('/').

2.5. Session sans cookie

Si le navigateur n'accepte pas les cookies, il faut transmettre l'identifiant de session d'une page du site à l'autre page par un autre moyen.

Une solution est de passer l'identifiant de session dans l'URL (méthode GET).