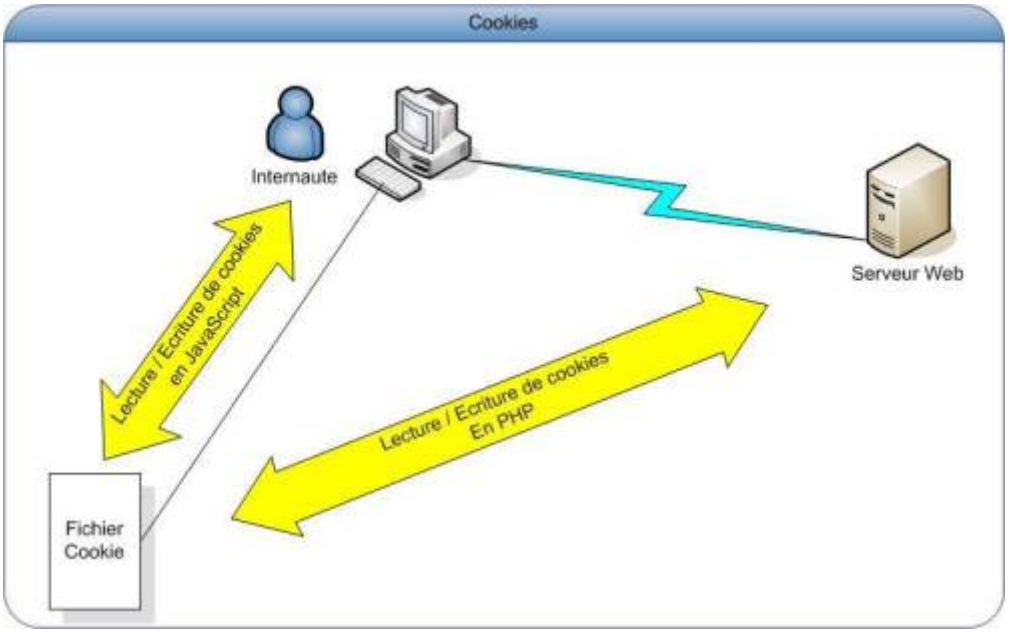
Les sessions

En programmation Web, les échanges entre le client et le serveur sont dits “**Stateless**”. C'est à dire qu'il n'y a aucune donnée mémorisée entre deux requêtes distinctes d'un internaute.
Les sessions sont un moyen technique qui vient combler cette insuffisance en permettant de stocker durablement des données individuelles pour chaque utilisateur. Elles peuvent être utilisées pour faire persister des informations entre plusieurs pages.

Les cookies

Les cookies sont l'outil sur lequel reposent les sessions.
Les cookies sont des fichiers stockés **côté client** qui permettent de créer une persistance d’informations au fil des pages visitées par l’internaute :

- Ce sont des ensembles de données stockées par le poste client sur son disque local sous la forme de fichiers « texte » ;
- Les données sont enregistrées dans un cookie comme des paires « clé / valeur » ;
- Chaque navigateur Web stocke les cookies dans un espace qui lui est propre ;
- L’utilisateur peut interdire l’emploi des cookies sur son poste, mais certains sites ne fonctionneront plus normalement dans ce cas ;
- S’ils sont autorisés, les cookies sont accessibles au serveur par le fait que le protocole HTTP prévoit leur transport systématique et bidirectionnel dans les entêtes HTTP à chaque échange (REQUEST/RESPONSE) entre client et serveur ;

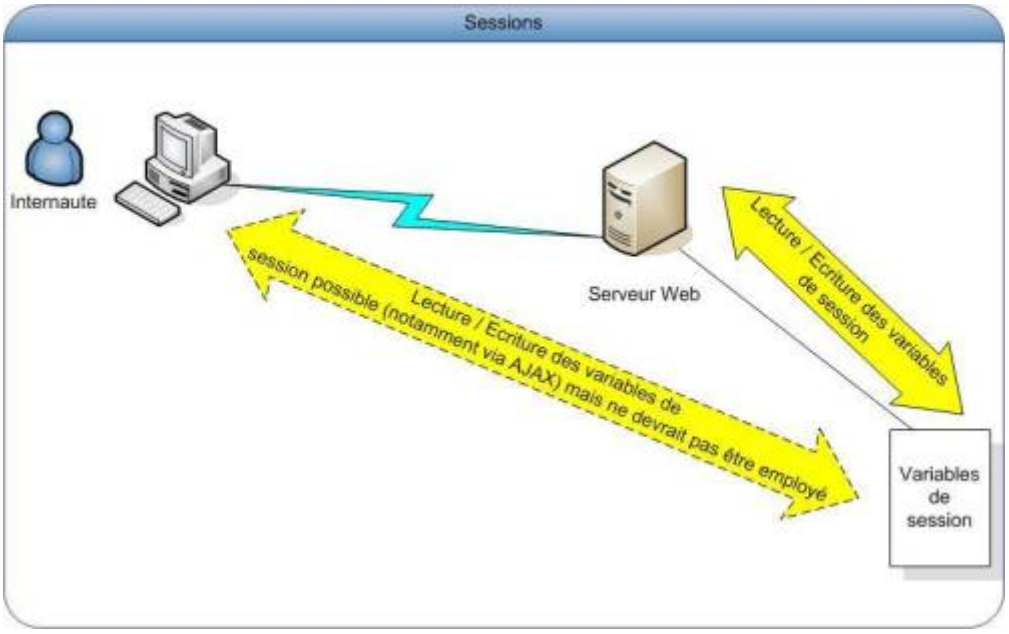


Les sessions

Les sessions sont une technique mise en œuvre **côté serveur** pour assurer la persistance

d'informations au fil des pages visitées par un internaute :

- Ouvrir une session consiste à demander au serveur de fabriquer un numéro qui identifiera l'internaute de manière unique (id. de session) et à mettre à disposition une zone mémoire dédiée pour y stocker des variables de session ;
- Les variables de session restent disponibles aussi longtemps que la session n'est pas, soit fermée explicitement, soit fermée implicitement parce que sa durée de vie est atteinte ;
- La durée de vie d'une session est paramétrable sur le serveur ;
- Pour fiabiliser le mécanisme, l'id. de session est échangé entre client et serveur sous la forme d'un cookie ;
- Même si c'est techniquement possible, les variables de session n'ont pas vocation à être exploitées par le client;



Mettre en œuvre les sessions en PHP

Démarrer une session

Il est possible de conserver des valeurs d'une page à l'autre en utilisant le mécanisme des sessions. Une session est un ensemble d'informations relatives à un utilisateur connecté sur le site. Ces informations sont enregistrées sur le serveur PHP et existent donc quelle que soit la page PHP courante. Toute page souhaitant accéder à ces informations de session doit commencer par :

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

Attention, ces lignes doivent impérativement être les premières du fichier. Une ligne vide avant, par exemple, provoquera une anomalie de fonctionnement !

Chaque session possède un identifiant qui conserve la même valeur au fil des pages PHP. Cette valeur

de type chaîne de caractères peut être utilisée de la manière suivante :

```
<?php
   echo session_id();
?>
```

Utiliser les variables de session

Utilisation courante

Lorsqu'une session est créée, il est possible de déclarer des variables de session et de leur donner une valeur. Voici l'exemple d'une page "**sessions2.php**" qui définit deux variables de session et comporte un lien vers une page "**sessions3.php**" :

sessions2.php

```
<?php
   session_start();
?>
<html>
<body>
<?php
    $_SESSION["nom"]='Dupont';
    $_SESSION["prenom"]='Paul';

    // session_register("prenom") permettra de créer une variable de
session
    // nommée prenom vide de contenu
?>
   <a href="sessions3.php">sessions3.php</a>
</body>
</html>
```

La page "**sessions3.php**" peut ensuite utiliser les deux variables définies :

sessions3.php

```
<?php
   session_start();
?>
<html>
<body>
<?php
    echo '<p>Nom : '.$_SESSION["nom"].'</p>';
    echo '<p>Prénom : '.$_SESSION["prenom"].'</p>';
?>
</body>
```

```
</html>
```

Ce sont pourtant 2 pages distinctes qui n'échangent formellement aucune donnée ...

Etat d'une variable de session

Comme pour toute variable en PHP, les fonctions **isset** et **empty** permettront de tester respectivement l'existence et la vacuité (le fait de contenir du vide) d'une variable.

```
if (!isset($_SESSION["nom"])
{
    $_SESSION["nom"]='Dupont';
}
```

```
if (empty($_SESSION["format"])
{
    $_SESSION["format"]='XML';
}
```

Suppression d'une variable de session

Pour supprimer une variable de la session courante :

```
session_unset($_SESSION["nom"]);
```

Pour supprimer toutes les variables de session en une seule ligne, au choix :

```
session_unset();
```

```
$_SESSION = array() ;
```

Clôture d'une session

```
session_destroy();
```

Attention, **session_destroy** ne détruit pas les variables associées à la session (cf **session_unset**, plus haut), ni le cookie sous-jacent, s'il existe. Il reste donc des traces de la session, notamment sur le client, ce qui peut avoir des effets indésirables.

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc3:sessions&rev=1681207303

Last update: **2023/04/11 12:01**