IIT
INSTITUT INTERNATIONAL TECHNOLOGIE
Université Nord Américaine Privée

Programmation web II

Mohamed Karim Abdmouleh
karim.abdelmoula@iit.ens.tn

1ère année Génie Informatique

(Version 0)

Chapitre 4 : Cookies et Sessions

Cookies

- Le langage php utilisé sur le web est **sans état**; chaque page est **indépendante**; ce qui signifie que chaque requête HTTP est traitée de manière indépendante, **sans avoir de connaissance de requêtes précédentes ou ultérieures**. Cela signifie que **PHP ne stocke pas automatiquement** les données de session entre les requêtes.
- Les cookies et les sessions de PHP permettent de s'affranchir cette contrainte.
- Les cookies sont des petites informations déposées dans le navigateur de la **machine cliente** pour pouvoir se **souvenir** d'elle. Un cookie est comme un **badge** : **vous dites qui vous êtes au serveur**.
- Chaîne de caractère **générée par le serveur** et stockée **sur le disque du client**
- Utilisés pour maintenir des informations de navigations
- Voyagent dans les headers http

Cookies

Il peut y avoir **plusieurs cookies pour un site** pour différentes raisons, notamment :

- Pour stocker des informations de **session**, telles que le fait que l'utilisateur est connecté ou non, les **préférences de langue**, etc.
- Pour stocker des informations de **suivi** de l'utilisateur, telles que les **pages visitées**, les **produits consultés**, etc. Ces informations peuvent être utilisées pour **personnaliser les publicités** ou les offres qui sont présentées à l'utilisateur.
- Pour stocker des informations de **sécurité**, telles que des jetons d'authentification, des codes de vérification, etc.
- Pour stocker des informations de fonctionnalité, telles que les **choix de l'utilisateur**, les **préférences** de lecture, etc.

Chacun de ces types de cookies peut avoir un nom et une valeur différents et peut être stocké pour une **durée variable en fonction des besoins du site.**

Comment peut on consulter les cookies d'un site?

- Tout d'abord, ouvrez le site web pour lequel vous souhaitez consulter les cookies.

1. Cliquer sur Inspecter/Application/cookies (on trouve la liste de tous les cookies associés à ce site web, avec leur nom, leur valeur et leur date d'expiration.)
2. Autre méthode: Vous pouvez également utiliser des **extensions** de **navigateur** comme "**EditThisCookie**" pour afficher et modifier les cookies stockés sur votre navigateur.

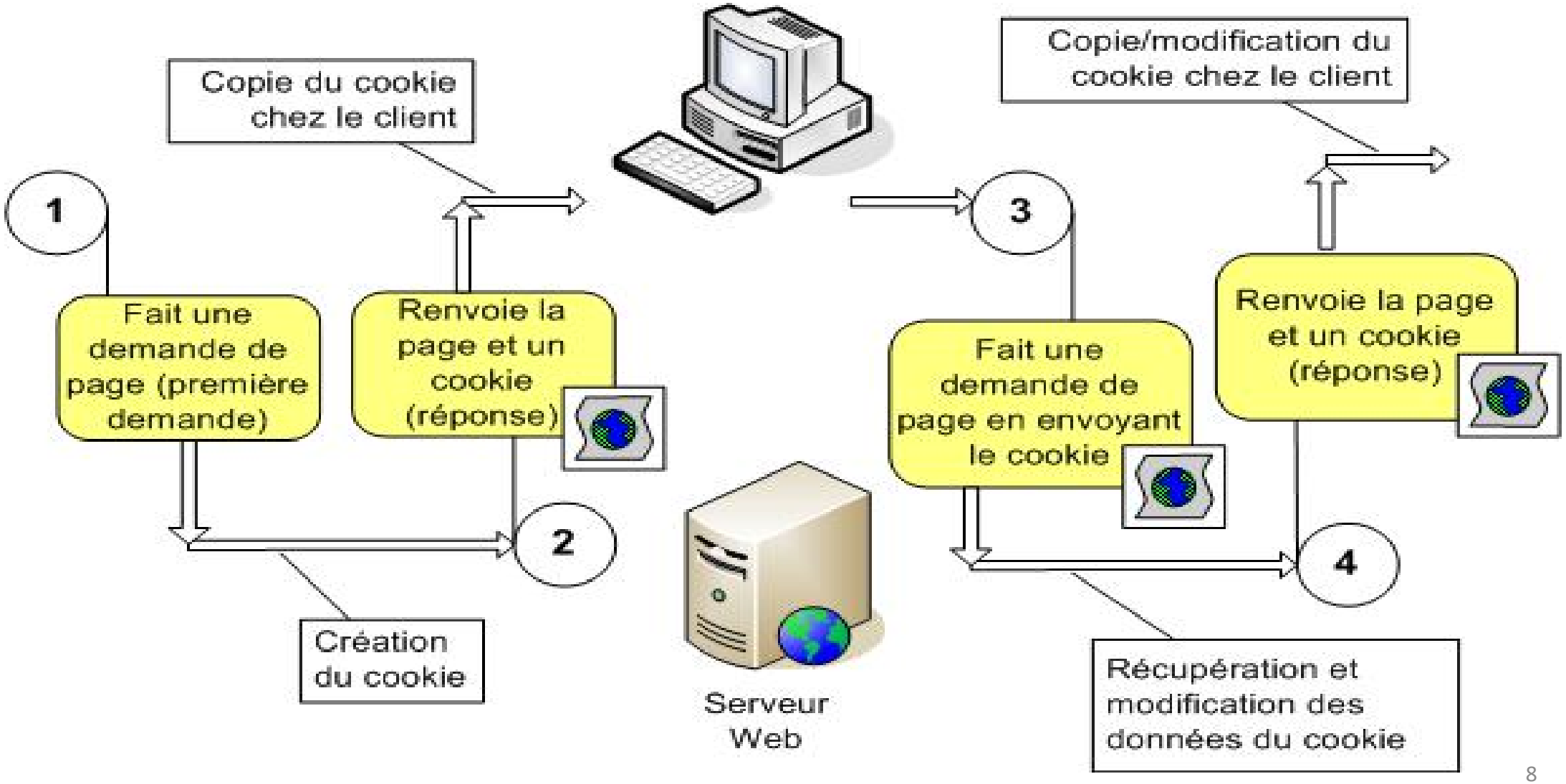
Cookies

- Structure:
  - Nom
  - Valeur
  - Date d’expiration
  - Chemin de validité
  - Domaine de validité
  - Attribut de sécurité

Cookies

- Paire de nom=valeur passé du serveur au navigateur dans l'entête de la réponse.
- Lorsque le navigateur envoie une requête, il inclut tous les cookies non expirés reçus du même serveur à l'intérieur de la requête.
- Problèmes :
  - Les cookies ne sont pas supportés par l'ensemble des navigateurs pour des raisons de confidentialité.
  - Le client peut désactiver les cookies.
- Avantages :
  - Placer des données (non sensibles) **côté client**
  - Cibler les attentes de l'internaute

Cookies


Copie du cookie chez le client
Copie/modification du cookie chez le client
1
Fait une demande de page (première demande)
Renvoie la page et un cookie (réponse)
2
Création du cookie
3
Fait une demande de page en envoyant le cookie
Renvoie la page et un cookie (réponse)
4
Récupération et modification des données du cookie
Serveur Web

Cookies

- Ecriture d'un cookie : définir un cookie qui sera envoyé avec le reste des en-têtes de la communication HTTP.

setcookie (nom, valeur, date_d_expiration, chemin, domaine, secure)

- Paramètres :
  - **Nom** : seul argument obligatoire
  - **date d'expiration** du cookie en secondes
    - une date du passé permet d'effacer un cookie
    - sans, le cookie est détruit à la fermeture du navigateur : c'est alors un cookie de session.
    - time()+60*60*24 expire dans 24 heures
    - time()- 10000 permet d'"expirer" le cookie
    - le cookie n'étant accessible qu'au chargement de la prochaine page (puisqu'il faut envoyer par HTTP le cookie vers le client puis que le client envoie sa valeur du cookie), un test évite le message d'erreur.

Cookies

- Le **chemin** est celui de la page qui a inscrit le cookie
  - ceci détermine la page qui pourra relire le cookie.
  - sinon, c'est le chemin du fichier courant qui est retenu
  - "/chemin" détermine que les pages de répertoires ou sous-répertoires de chemin peuvent lire le cookie
  - "/" implique que toutes les pages du domaine peuvent lire le cookie
  - Le cookie n'est valide que pour le chemin et sa sous-arborescence
- Le **nom de domaine** permet d'identifier le cookie parmi tous ceux stockés sur la machine
  - détermine la page qui pourra relire le cookie.
  - sinon, c'est le domaine courant
  - " @URL" indique que toutes les pages des sous-domaines de " @URL" peuvent lire le cookie
- **secure** est une valeur booléenne : indique si l'accès au cookie est protégé, donc transmissible uniquement par HTTPS.

Exemple

```
setcookie (nom, valeur, date_d_expiration, chemin, domaine, secure);

setcookie("couleur", "jaune", time()+3600*24*30,    "/",      " ",    0);
setcookie("langue", "français", time()+3600*24*365  );
```

- time()+3600*24*30 La durée de ce cookie est 1 mois (1h*24*30j)
- Le chemin "/" indique le chemin sur le serveur pour lequel le cookie est disponible.
- " " détermine les domaines qui peuvent relire le cookie, e.g. example.com
- 0 indique que l'accès au cookie est **non** protégé, donc transmissible par HTTP.

Cookies

- **Ex: setcookie**("nom_cookie",$valeur,time()+$second,"/","",0);

- Il faut appeler la fonction **Setcookie** avant toute balise <HTML> ou <HEAD> car les cookies doivent "passer" avant les autres en-têtes.

- Le cookie ne sera accessible qu'au chargement de la prochaine page, ou au rechargement de la page courante.

- Lorsque le navigateur appelle une page via HTTP, il regarde s'il a des cookies valides pour cette page et auquel cas, les envoie dans l'entête de requête HTTP.

Cookies

- **$_COOKIE**
  superglobale des cookies : contient les cookies transmis dans l'entête de la requête HTTP.
- $_COOKIE["nom_cookie"] = $valeur;

  => Doit être avant le premier caractère HTML (y compris espace)
- Pour consulter une variable de cookie :

  $_COOKIE['nom_cookie']
- Lecture d'un cookie :

```
if (isset($_COOKIE["nom_cookie"])) {
$valeur = $_COOKIE["nom_cookie"];  }
```


Cookies

- Une fois le cookie créé, il est possible de le récupérer dans les pages suivantes via le tableau $_COOKIE['nom_cookie'].
- *setcookie("TestCookie", $value, time()+3600);*

// expire dans une heure

- La suppression d'un cookie se fait en le recréant avec une date d'expiration passée.
- *setcookie ("TestCookie", "", time() - 3600);*

// a expiré il y'a une heure

- Ou aussi, faire un setcookie ("nom_cookie") en ne spécifiant aucune valeur.

Exemple

```
<?php
$Last = false;
if (isset ($_COOKIE['date_connexion'])){
   $Last = $_COOKIE['date_connexion'];
}

setcookie ('date_connexion', date ("d-m-Y H:m:s"));
if ($Last)
   echo "Votre dernière connexion: ".$Last;
?>
```

Application

Créer **deux** pages, une première qui **créé** un cookie dans lequel on placera des informations sur le navigateur de l'utilisateur, avec un **lien vers une seconde page** dans laquelle **s'affichera** cette information.

On en rajoutera une **troisième pour tuer le cookie.**

Correction

ex01.php :

```
<?php
$var = $_SERVER['HTTP_USER_AGENT'];
setcookie("monCookie", $var , time()+3600);
?>
<HTML>
<HEAD><TITLE>Création  d'un cookie</TITLE></HEAD>
<BODY>
Bonjour à tous ! <BR>
<A href="ex02.php">Allez  à la page suivante.</A>
</BODY>
</HTML>
```

Correction

ex02.php :

```
<HTML>
<HEAD><TITLE>Utilisation d'un cookie</TITLE></HEAD>
<BODY>
Rebonjour!<BR>
<?php
echo "Sur votre précédente feuille, vous utilisiez un navigateur :
                    $_COOKIE["monCookie"] " ;
?>
<BR>
<A href="ex03.php">Voir la page où on tue les cookies.</A>
</BODY>
</HTML>
```

Correction

ex03.php :

```
<?php
setcookie("monCookie","",time()-60);
?>
<HTML>
<HEAD><TITLE>Suppression d'un cookie</TITLE></HEAD>
<BODY>
Rebonjour encore une fois !<BR>
<?php
if (isset($_COOKIE["monCookie"]))
 echo " Le Cookie n’est pas mort !";
else
 echo " Le Cookie est mort !";
?>
</BODY>
</HTML>
```

Application 2

Créer un cookie qui permet de calculer le nombre de visite à une page web.

```php
<?php
if (isset ($_COOKIE['nb_visites']))
{
    $n=$_COOKIE['nb_visites'];
    $n++;
}

else
    {$n=1;}
    setcookie('nb_visites', $n, time()+3600*24*365);
    echo  "vous avez effectue $n visite(s)";
    ?>
```

Third-party cookies

- sont souvent utilisés pour le suivi en ligne, la publicité ciblée et l'analyse du comportement des utilisateurs à travers différents sites web.

Sessions

- Les sessions permettent de **conserver des informations** relatives à un utilisateur lors de son **parcours** sur un site web.

- **But : conserver des valeurs d'une page à l'autre.**

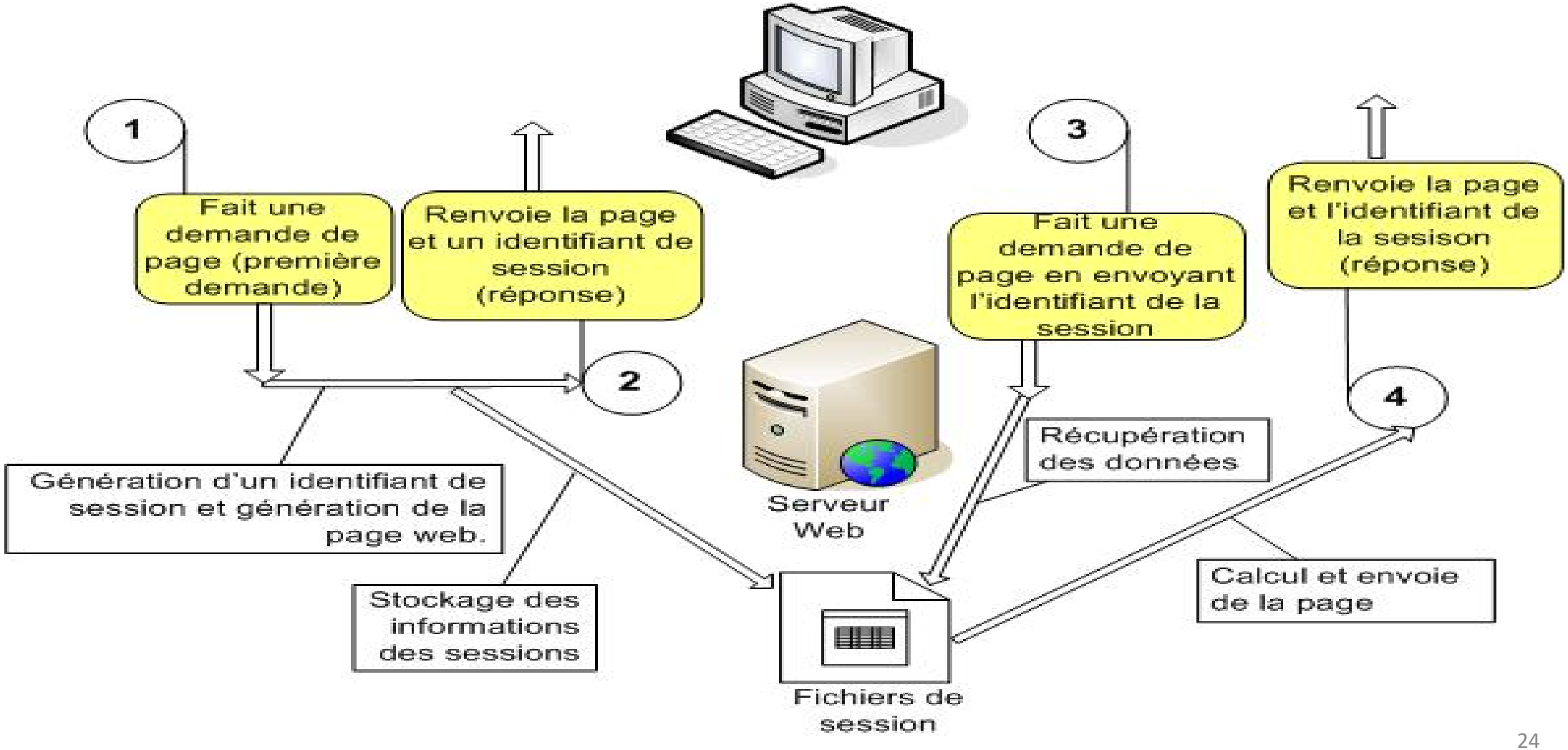
- Contrairement aux cookies, les sessions sont basées sur des fichiers créés sur le **serveur**.

- Avantages :
  - Conserver des données d'une page à l'autre.
  - Très simple utilisation.
  - Utile notamment dans les formulaires de commande en ligne.

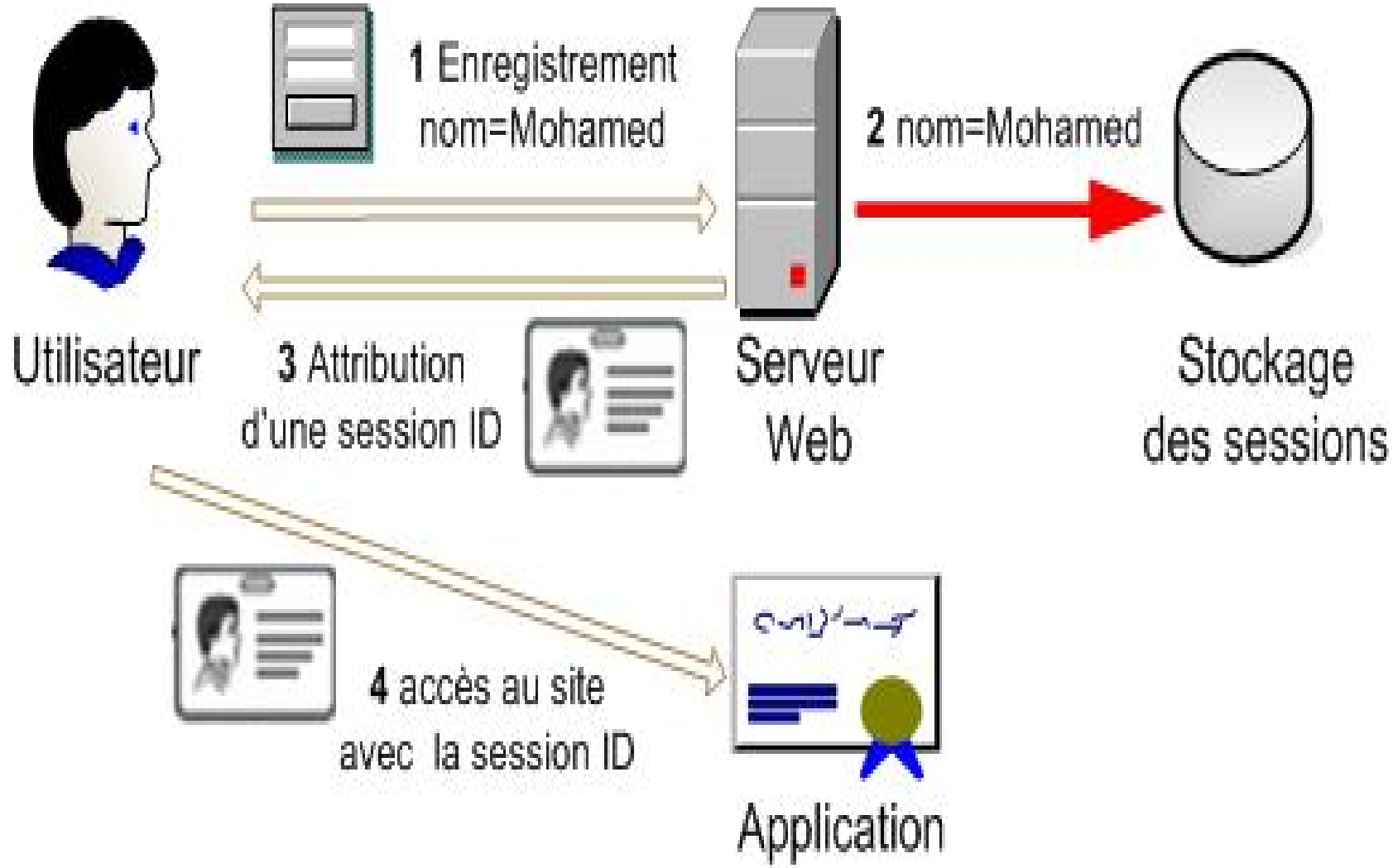
Sessions

- Les opérations effectuées dans différents pages web et réalisées par le même individu nécessite une session
- Exemple transaction avec sa banque
- Transaction commerciale avec un site marchand

Sessions


1
Fait une demande de page (première demande)
Renvoie la page et un identifiant de session (réponse)
2
3
Fait une demande de page en envoyant l'identifiant de la session
Renvoie la page et l'identifiant de la sesison (réponse)
4
Serveur Web
Récupération des données
Calcul et envoie de la page
Génération d'un identifiant de session et génération de la page web.
Stockage des informations des sessions
Fichiers de session

Sessions


1 Enregistrement
nom=Mohamed
2 nom=Mohamed
Utilisateur
3 Attribution
d'une session ID
Serveur
Web
Stockage
des sessions
4 accès au site
avec la session ID
Application

Sessions

- Chaque visiteur en se connectant à un site **reçoit un numéro d'identification** dénommé identifiant de session (SID)

- La fonction **session_start()** se charge de **générer automatiquement cet identifiant** unique de session et de créer un répertoire. Elle doit être placée au début de chaque page afin de **démarrer ou de continuer** une session.

```
<?php
 session_start();
 $Session_ID = session_id();
 // $Session_ID = 7edf48ca359ee24dbc5b3f6ed2557e90     ?>
```

- Les modifications aux variables de session sont enregistrées automatiquement.
- L'utilisation des sessions par une page php n'est pas automatique (sauf si on modifie la valeur de session.autostart dans php.ini).
- Par défaut, la durée de vie de session est ***24 minutes.***

Sessions

- Pour utiliser les sessions, Il faut écrire : **session_start();**

  Dans le cas ou true est retournée, cette fonction permet soit de créer une session (si pas encore définie) soit d'utiliser les variables de la session en cours.

- Si aucune session n'existe, php créé automatiquement un identifiant de session qu'on peut visualiser avec la fonction : **session_id() ;**

- Pour retourner le nom de la session en cours : **session_Name();**

- L'emplacement du fichier créé par php est indiqué par :

  session_save_path() ;

Sessions

- On peut stocker et interroger les variables d'une session de la manière suivante :

  $_SESSION['nomVariable'];

  L'affectation du tabeau $_SESSION permet de créer des variables de session (*il faut que session_start() soit en entête de page*).

- Autre méthode : pour enregistrer une variable en session et indiquer initialement qu'elle sera accessible d'une page à l'autre : **session_register ();**

- Si on veut supprimer une session, on utilise la fonction : **session_destroy();**

- Tant que la page n'est pas rechargée, les variables de session restent accessibles. Pour les décharger de la mémoire : **session_unset();**

Remarque

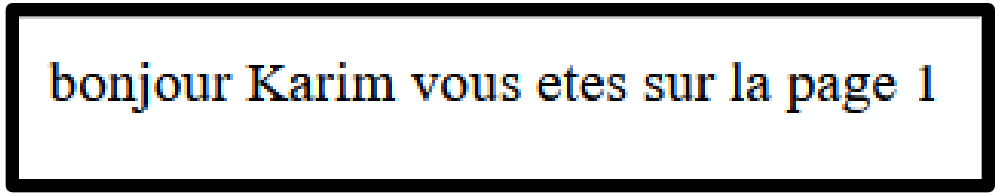
- Il est nécessaire de toujours **réécrire** **session_start()** au début de chaque script PHP qui utilise les sessions, car cette fonction **démarre** une session **ou restaure une session** existante pour l'utilisateur en cours. **Sans l'appel à session_start(), les variables de session ne pourront pas être utilisées.**

- De plus, il est important de noter que session_start() doit être appelé avant tout autre contenu envoyé au navigateur, y compris les espaces blancs ou les lignes vides.

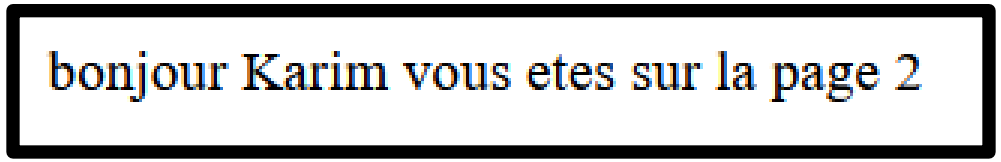
Exercice

Formulaire.html

Reponse.php



page1.php



page2.php

Correction Formulaire.html


```
<!DOCTYPE html>
<html lang="en">
<body>
    <form action="reponse.php" method="GET">
        nom :  <input type="text" name="nom">

        <input type="submit" value="envoyer">
    </form>
</body>
</html>
```

Reponse.php

```php
<?php
if (!empty($_GET["nom"]))
echo "bonjour  ". $_GET["nom"];
else
echo "erreur";
?>

<html>
<form method="post">
<input type="submit" value = "page 1" name="p1">
<input type="submit" value = "page 2" name="p2">
</form>
</html>
```


```php
<?php
session_start();

$_SESSION["user"]=$_GET["nom"];

if (isset($_POST["p1"]) )
header("location: page1.php");
if (isset($_POST["p2"]) )
header("location: page2.php");
?>
```

page1.php

```php
<?php
session_start();
echo "bonjour    " .  $_SESSION["user"]. "  vous etes sur la page 1";
?>
```

bonjour Karim vous etes sur la page 1

page2.php

```php
<?php
session_start();
echo "bonjour    " .  $_SESSION["user"]. "  vous etes sur la page 2";
?>
```

bonjour Karim vous etes sur la page 2

Sessions

Application :

Créer une page qui créé une session. Faire afficher l'identifiant et l'emplacement du fichier de session.

Entrer deux valeurs dans deux variables de session et créer un lien vers une seconde page.

Cette seconde page se chargera d'afficher le contenu de ces deux variables ainsi que la manière dont ces deux variables sont stockées dans le fichier de session.

Sessions

Page01.php

```
<?php
session_start();
?>
<HTML>
<BODY>
<?php
echo "Bienvenue, ton ID de session, <B>".session_id()."</B><BR>";
echo "Ton fichier se trouve: <B>".session_save_path()."</B><BR>";
$_SESSION['nom'] = "Mohamed Karim Abdmouleh";
$_SESSION['mdp'] = "12345";
?>
<A href="page02.php">Page suivante</A>
</BODY>
</HTML>
```

Sessions

Page02.php

```
<?php
session_start();
?>
<HTML>
<BODY>
<?php
echo "Les variables ont été enregistrées  ";
echo "Nom : ".$_SESSION['nom'];
echo " Mdp : ".$_SESSION['mdp'];
?>
</BODY>
</HTML>
```