

TP Sessions PHP

TOUS LES FICHIERS CREEES ICI SERONT SUR VOTRE ESPACE PERSONNEL SUR LE SITE WEB

Apport de connaissances : Les variables de session

Apport de connaissances :

Vous connaissez à présent les variables classiques (\$var1, \$var2, ...) et les variables spéciales associées aux échanges d'information par formulaire html (tableaux \$_POST[] et \$_GET[]).

Il existe également des variables « de session » dont la portée s'étend à TOUTES les pages d'un même site.

Une variable de SESSION est conservée sur le serveur PHP et reste donc confidentielle.

On comprend l'intérêt de ces variables « super globales » quand une information doit suivre l'utilisateur dans toutes les pages du site, comme par exemple son nom d'utilisateur.

Quand une session est démarrée, un numéro de session est créé sur le serveur et une copie de ce numéro est envoyée sous forme de cookie chez le client (le navigateur doit donc disposer de cette fonction). Ce numéro sera valable pendant toute la durée de la connexion. Quand on ferme l'onglet du navigateur, la connexion est rompue et la session (avec ses variables) est supprimée. On a donc un système relativement sécurisé.

Pour démarrer une SESSION PHP, il faut utiliser la fonction :

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

NB : Certains serveurs Web démarrent automatiquement une session quand une page PHP est appelée.

Une fois la session démarrée, on utilise le tableau associatif \$_SESSION[].

Ex : \$_SESSION['nom'] = "Dupont" ;

TRAVAIL 1 :

Dans la suite de ce TP, nous allons créer des pages privées accessibles par un nom / mot de passe. C'est une variable de session qui servira à vérifier que l'authentification a eu lieu.

A faire :

A partir des fichiers fournis :

1. Modifier la page **index.php** comme suit :
 - Le formulaire nom / mot de passe doit fonctionner selon la méthode POST.
 - Le *submit* doit appeler la page nommée **accueil_prive.php** : notez le code php qui a été ajouté au début :

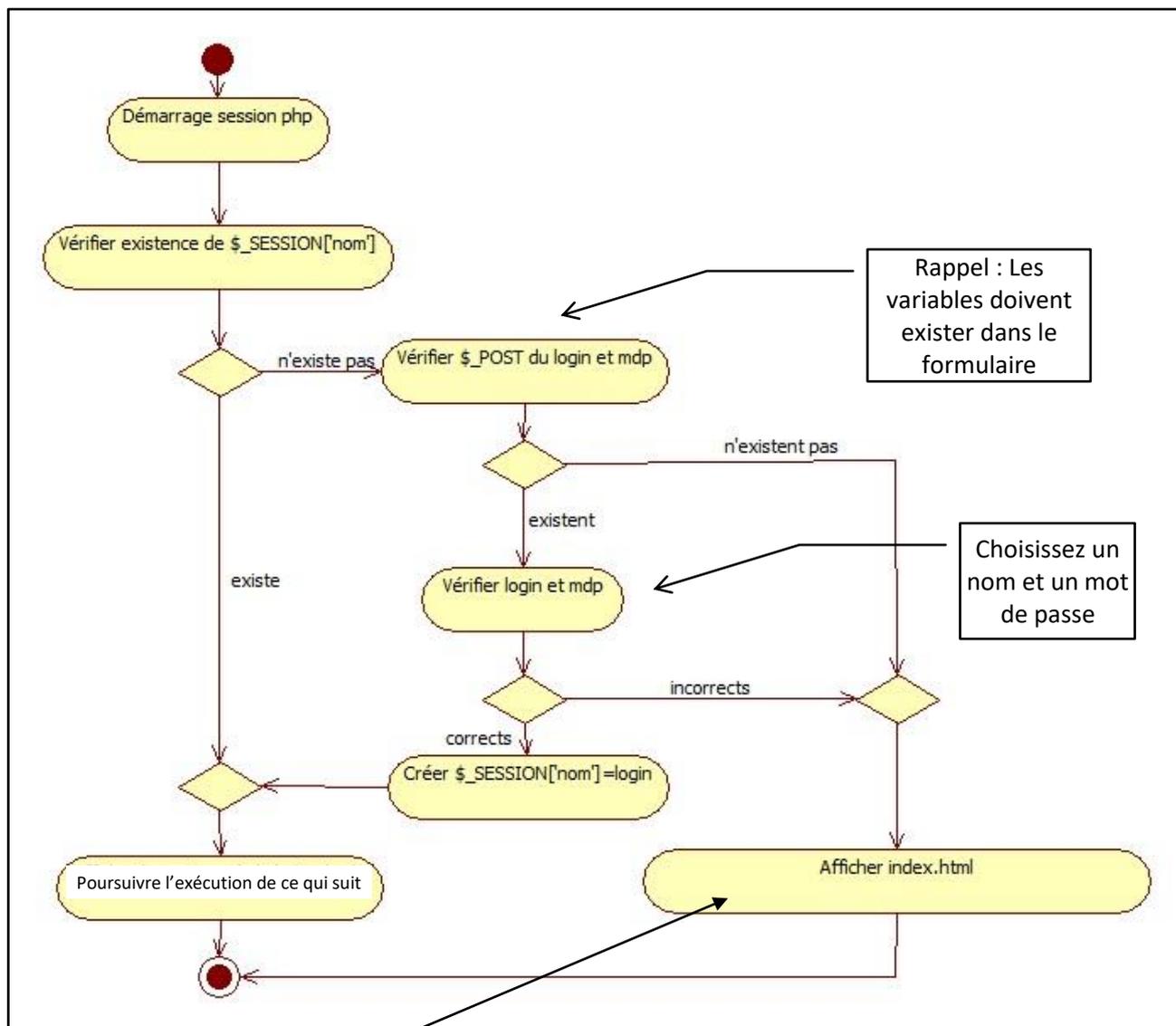
```
<?php require("connexion.php"); ?>
```

ce code appelle la page connexion.php avant de continuer son exécution.

Si la page n'existe pas ou a un problème, l'exécution s'arrête. C'est une sécurité.

2. **Il faut créer cette page** : Codez la page **connexion.php** selon l'algorithme suivant :

Fonctionnement du script de la page connexion.php



Note : Pour l'action « Afficher index.html », utiliser pour cela la fonction Php suivante :

```
header("Location:index.html"); die();
```

Validation : Si votre travail est correct, il sera impossible d'accéder à accueil_prive.php directement (par son adresse url), sans passer par l'authentification de la page d'accueil.

TRAVAIL 2 : Réutiliser son travail

3. Ajouter ce code au début des autres pages du site (**prive1.php**, **prive2.php**) :

```
<?php require("connexion.php"); ?>
```
4. **Validation** : Si votre travail est correct, il ne devrait pas être possible d'accéder aux pages du site directement (par leur adresse url), sans passer par l'authentification de la page d'accueil.

ATTENTION : Les navigateurs conservent dans leur cache les pages consultées. Il faut donc quitter le navigateur pour être certain que la session est détruite. Nous réglerons cet inconvénient dans la partie suivante.

TRAVAIL 3 : Détruire une session

Ajouter aux pages du site un lien ou un bouton de déconnexion qui appellera une nouvelle page « deconnexion.php » que vous devrez créer. Cette page exécutera la fonction PHP :

```
session_destroy();
```

+ et qui reviendra à la page d'accueil (avec un header).

Pour réaliser ce travail, vous utiliserez

soit un lien : `Quitter`

Soit un bouton : `<button onclick="location.href='deconnexion.php' ">Quitter</button>`

TRAVAIL 4 : Ajouter un COOKIES pour forcer une déconnexion.

En vous servant de votre document de cours, créez un cookie valable 2 min dès que la session est validée. Ensuite, si la page est rechargée et que le cookie n'est plus valable, on force la déconnexion de la session.