



Avant de commencer j'aimerais tout d'abord vous dire qu'il est possible d'arrêter une session. En arrêtant la session tout ce qui était sauvegardé sera bien entendu perdu. C'est un peu comme quand on se déconnecte d'un site, il nous redemandera de nous reloguer.

Il suffit de faire un « `session_unset()` » ou « `unset($_SESSION['nomDeLaVariable'])` » qu'on verra un peu plus tard.

#### (Exos)

- 0) Améliorez le code du jeu du hasard. Je veux qu'à chaque fois que vous introduisez un numéro il continue de vous dire si c'est plus grand ou plus petit etc. Mais je veux que ce numéro aléatoire soit gardé en mémoire tant que nous ne l'avons pas trouvé. Si on le trouve, il donnera bien entendu le message comme quoi on l'a trouvé mais il faudra aussi qu'il le libère de la mémoire. Que quand on relance jeu du hasard il recrée un nouveau nombre et ainsi de suite.

Nous allons maintenant créer un dossier « **functions** » là où se trouve nos fichiers. Dans ce dossier on va créer un fichier qui s'appelle « **authentification.php** ». Dans ce fichier PHP on va créer une fonction qui va s'appeler « **is\_connected** », elle ne recevra rien en paramètre et retournera un booléen.

```
🐘 authentification.php U X
functions > 🐘 authentification.php > ...
1  <?php
2      function is_connected() : bool {
3          return true;
4      }
5  ?>
```

On va **modifier** le contenu de la fonction de manière à vérifier que l'utilisateur est connecté, on pourra se baser sur le principe des sessions que nous venons de voir.

```
return !empty($_SESSION['connected']);
```

Si ce n'est pas vide, ça va me renvoyer true, si c'est vide ça va me renvoyer false.

Maintenant on peut utiliser cette fonction la dans un fichier qu'on va appeler « **dashboard.php** »(pas dans functions). Dashboard en anglais veut dire tableau de bord. En gros c'est une représentation visuelle des informations importantes. On aurait pu l'appeler autrement si on voulait.

On va créer un onglet dashboard qui exécutera le fichier.

### (Exos)

#### 1) Créez l'onglet dashboard (basez-vous sur les autres onglets).

Dans ce fichier on va tester d'appeler la fonction « **is\_connected** » et vérifié grâce aux sessions si ça fonctionne bien.

```
dashboard.php > ...
1  <?php
2      $nav = "DashBoard";
3      $title = "Dashboard";
4      session_start();
5      require "functions/authentication.php";
6      var_dump(is_connected());
```

Ici on voit bien que j'importe bien le fichier **authentication.php** qui se trouve dans le dossier **functions**. On n'oublie pas de lancer la session avant. Puis on fait un **var\_dump** pour vérifier que vaut l'appelle de la fonction **is\_connected()**.

Si on clique sur l'onglet dashboard de notre site. On verra bien que le booléen est à false. Ça veut dire qu'on n'est pas connecté.

Si durant la session on donne la valeur de 1 à notre variable globale. Rajoutez cette ligne juste au-dessus du require.

```
$_SESSION['connected']=1;
```

Il retournera désormais true sur notre site. Vous auriez pu aussi lui donner la valeur true directement.

Voilà une manière de savoir comment on pourra définir si un utilisateur est connecté.

Vous remarquez qu'à chaque fois, je démarre la session. Il y a moyen de vérifier si une session est démarrée ou pas. C'est en utilisant la fonction « **session\_status** », elle se trouve bien entendu dans la documentation officielle sur php.net. Elle permet de renvoyer le statut courant de la session. Vous avez 3 constantes qui représente les différentes valeurs :

#### Valeurs de retour

- **PHP\_SESSION\_DISABLED** si les sessions sont désactivées.
- **PHP\_SESSION\_NONE** si les sessions sont activées, mais qu'aucune n'existe.
- **PHP\_SESSION\_ACTIVE** si les sessions sont activées, et qu'une existe.

On peut donc faire quelques choses comme cela dans le fichier **authentification.php** :

```
function is_connected() : bool {  
    if (session_status() === PHP_SESSION_NONE){  
        session_start();  
    }  
    return !empty($_SESSION['connected']);  
}
```

En gros, si la session n'est pas lancée, il fera un `session_start()`.

On peut maintenant supprimer la ligne où on lançait la session dans notre fichier **dashboard.php**.

Si votre session est toujours active il suffit de faire un `session_unset()` ou un `unset($_SESSION['connected'])` après le `var_dump()`.

Maintenant il faut empêcher l'utilisateur d'avoir accès à cette page s'il n'est pas connecté, et le rediriger vers une page qu'on créera qui s'appellera « **login.php** ».

Pour rediriger un utilisateur vers une autre page il suffit de mettre une entête `location`, elle permet d'indiquer vers quelle page doit être rediriger l'utilisateur.

Il suffit de faire un `header('Location : cheminDeRedirection')`, dans notre cas ici on va le rediriger vers la page de login.

Dans notre fichier « **dashboard.php** » on va mettre ceci :

```
dashboard.php > ...  
1  <?php  
2      $nav = "DashBoard";  
3      $title = "Dashboard";  
4      require "header.php";  
5      require "functions/authentification.php";  
6      var_dump(is_connected());  
7      header('Location: coursphp/login.php');  
8  
9      require "footer.php";
```

Maintenant il nous reste à créer notre fichier **login.php**.

(Exos)

## 2) Créer la page login, un formulaire avec 2 variables « pseudo » et « password » :

Mon Site Accueil Contact A propos Jeu du Hasard Dashboard Login

**Login**

Votre login

Votre mot de passe

Se connecter

Accueil Contact A propos

© 2022 Cfitech, Inc

Maintenant qu'on a notre page login, si on clique sur l'onglet dashboard, il nous redirigera sur la page login pour devoir se connecter.

Dans le code on va essayer de récupérer ce qu'on écrit dans notre formulaire de connexion. On va créer un compte avec comme pseudo « **votre prénom** » et comme mot de passe « **12345** ». On devra vérifier qu'on a bien introduit ces deux valeurs pour pouvoir être connecté.

Faites ces modifications dans votre fichier login.php :

```
<?php
$nav = "login";
$title = "Login";
$erreur = null;
if (!empty($_POST['pseudo']) || !empty($_POST['password'])) {
    if ($_POST['pseudo'] === "" && $_POST['password'] === "12345") {

    } else {
        $erreur = "Identifiants incorrects !";
    }
}
require "header.php";
if ($erreur) : ?>
    <div class="alert alert-danger">
        <?php echo $erreur ?>
    </div>
<?php endif;
?>
```

Ici je n'ai pas encore défini ce qu'on doit faire quand on se connectera correctement.

Je crée une variable \$erreur qui va me servir de retourner une erreur si on n'a pas introduit les bons identifiants. Je vérifie ensuite si dans notre formulaire on a bien écrit quelques choses dans pseudo et dans password. Si oui je vérifie ensuite que ce qu'il y a dedans correspond à la condition. Sinon je donne à notre variable erreur un message.

Ensuite je vérifie si la variable erreur est vide, si elle est toujours à null, c'est qu'on n'aura pas eu d'erreur donc il n'entrera pas dans le IF. Par contre si on a eu une erreur de pseudo ou de password, il aura une valeur donc il entrera dans le IF et affichera l'erreur sous forme d'alerte grâce à la class alert alert-danger.

Allez maintenant sur votre site et introduisez un mauvais mot de passe ou un mauvais pseudo et vous devriez avoir normalement cette alerte avec ce message ci :

[Mon Site](#) [Accueil](#) [Contact](#) [A propos](#) [Jeu du Hasard](#) [Dashboard](#) [Login](#)

Identifiants incorrects !

## Login