

## Authentification PHP sur site web :

Pour y mettre en place une authentification PHP, il vous faudra en premier lieu un serveur web opérationnel avec des pages web avec l'extension PHP.

Sur nos pages, qu'on veut protéger, on va ajouter un bout de code au début de notre fichier pour renvoyer l'utilisateur vers la page de connexion s'il ne s'est pas connecté auparavant :

```
<?php
// On prolonge la session
session_start();
// On teste si la variable de session existe et contient une valeur
if(empty($_SESSION['login']))
{
    // Si inexistante ou nulle, on redirige vers le formulaire de login
    header('Location: connexion.php');
    exit();
}
?>
```

Une fois en place, on va configurer notre fichier connexion.php

Ci-dessous, la partie PHP :

```
<?php
// Definition des constantes et variables
define('LOGIN','test');
define('PASSWORD','test');
$errorMessage = '';

// Test de l'envoi du formulaire
if(!empty($_POST))
{
    // Les identifiants sont transmis ?
    if(!empty($_POST['login']) && !empty($_POST['password']))
    {
        // Sont-ils les mêmes que les constantes ?
        if($_POST['login'] !== LOGIN)
        {
            $errorMessage = 'Identifiant incorrect !';
        }
        elseif($_POST['password'] !== PASSWORD)
        {
            $errorMessage = 'Mot de passe incorrect !';
        }
        else
        {
            // On ouvre la session
            session_start();
            // On enregistre le login en session
            $_SESSION['login'] = LOGIN;
            // On redirige vers le fichier admin.php
            header('Location: accueil.php');
            exit();
        }
    }
    else
    {
        $errorMessage = 'Veuillez entrer vos identifiants !';
    }
}
?>
```

Le principe est simple, si l'utilisateur ne tape pas le bon login et le bon mot de passe qui est défini dans les variables, il ne pourra jamais passer.

Cela n'est pas très propre du fait que les identifiants se trouvent directement dans le fichier et non pas dans une base de données externe. Cependant, par le fait que le PHP est un langage backend, les utilisateurs du site ne pourront jamais voir cette partie. Pensez quand même à mettre votre site en HTTPS pour éviter que vos mots de passe tapés soient en clair.

Pour la partie HTML, un formulaire basique fera l'affaire :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <body style=background-color:#737373;>
      <font color="white">
        <title>Formulaire d'authentification</title>
      </head>
    <body>
      <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="post">
        <fieldset>
          <legend>Guarda</legend>
          <?php
            // Rencontre-t-on une erreur ?
            if(!empty($errorMessage))
            {
              echo '<p>', htmlspecialchars($errorMessage) , '</p>';
            }
          ?>
          <p>
            <label for="login">Identifiant :</label>
            <input type="text" name="login" id="login" value="" />
          </p>
          <p>
            <label for="password">Mot de passe :</label>
            <input type="password" name="password" id="password" value="" />
            <br>
            <br>
            <input type="submit" name="submit" value="Se connecter" style="background-color: #737373
          </p>
        </fieldset>
      </form>
    </font>
  </body>
</html>
```

Pour aller plus loin, vous pouvez créer un bouton qui renvoie vers une page de déconnexion, pensez juste à placer ces 2 lignes PHP sur la page de déconnexion pour détruire la session :

```
<?php
// On prolonge la session
session_start();
// On teste si la variable de session existe et contient une valeur
session_destroy();
?>
```