

CR TDM07 PHP 2

SCRIPTS & commentaires

1 Compteur PHP version 2

Enoncé

Reprenez le compteur PHP développé la semaine dernière et modifiez-le afin de lui ajouter un second compteur de visites qui s'incrémente uniquement pour chaque première visite au cours d'une même session pour un utilisateur. Aucune saisie de login/mdp n'est nécessaire, il s'agit uniquement de s'appuyer sur les sessions.

Quelques méthodes utiles et remarques:

- Mettre le *compteur2.php* dans */var/www*
- **fread(file, length)** : les 2 arguments sont requis, *length* : pour spécifier le nb max des bytes à lire

Script

compteur.inc.php

```
<?php
function compteur($nomdufichier,$flagincrement=TRUE) {
    if (file_exists($nomdufichier)) {
        $fichier=fopen($nomdufichier,'r+');
        flock($fichier,LOCK_EX);
        $compteur=fgets($fichier,100);
        if (empty($compteur)) {
            $compteur=0;
        }
        else {
            settype($compteur,"integer");
        }
        if ($flagincrement) {
            $compteur++;
        }
    }
    else {
        $fichier=fopen($nomdufichier,'c');
        flock($fichier,LOCK_EX);
        $compteur=1;
    }
    if ($flagincrement) {
        fseek($fichier,0);
        fputs($fichier,$compteur);
    }
}
```

```

    }
    flock($fichier, LOCK_UN);
    fclose($fichier);
    return $compteur;
}
?>

```

pagecompteurs.php

```

<?php session_start();
require('compteur.inc.php');?>
<html>
  <head>
    <title>Compteurs</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  </head>
  <body>
    <p>Compteur de hits : <?php echo compteur('compteur.txt'); ?></p>
    <p>Compteur de visites :
    <?php echo compteur('compteurvisite.txt',
    !isset($_SESSION['dejapasse'])); ?></p>
    <?php
      if (!isset($_SESSION['dejapasse'])) {
        $_SESSION['dejapasse']=TRUE;
      }
    ?>
    <p><a href="logout.php">Réinitialisation de la connexion.</a></p>
  </body>
</html>

```

logout.php

```

<?php
  session_start();
  session_destroy();
  header("Location:../pagecompteurs.php");
?>

```

2 Serveur bibliographique (2) : PHP et SimpleXML

Enoncé

Reprenez les fichiers développés au cours du TDM sur XML et créez une page PHP affichant, à l'aide de SimpleXML, le résultat de la recherche d'une publication saisie à partir d'un champs texte (par exemple une recherche sur l'année). Seul le premier résultat sera affiché. NB : l'affichage étant fait ici par lecture du contenu d'un fichier XML à l'aide de SimpleXML, pour appliquer la CSS il est

nécessaire de réinsérer des balises HTML pour lesquelles il existe un traitement CSS

Quelques méthodes utiles et remarques:

Script

serveurbiblio.php

```
<!DOCTYPE html>
<?php
    setlocale(LC_ALL,"fr_FR");
?>
<html>
    <head>
        <title>Biblio</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
        <link href="data/styleBibliographie.css" rel="stylesheet" type="text/css"/>
    </head>
    <body>
        <form action="serveurbiblio.php" method="post">
            <p><label>Annee de la publication : <input type="text" name="annee"/></label></p>
            <p><input type="submit" value="Afficher"/></p>
        </form>
        <?php
            $publications=simplexml_load_file("data/liste-publications.xml");
            foreach ($publications->publication as $publication) {
                if ($_POST['annee']==$publication->annee) {
                    echo "<p>Publication : ";

                    // Les auteurs
                    foreach ($publication->listeauteurs->auteur as $auteur) {
                        echo "<auteur>".$auteur."</auteur>";
                    }

                    // Le titre
                    echo "<titre>".$publication->titre."</titre>";

                    // Le type
                    $type=$publication->attributes();
                    if ($type=='article') {
                        echo "Journal : <journal>".$publication->journal."</journal>";
                    }
                    elseif ($type=='journal') {
                        echo "Conférence : <conference>".$publication->conference."</conference>";
                    }
                    // Pages et année
                    echo "<pages>".$publication->pages."</pages><annee>".$publication->annee."</annee>";
                }
            }
        </?php>
    </body>
</html>
```

```

    }
  }
  ?>
</body>
</html>

```

3 Forum ASI: PHP v2

Enoncé

Reprenez la page permettant le dépôt de messages en mode non connecté réalisée lors du TDM précédent et modifiez-la afin de gérer la persistance des messages à l'aide de SQLite à la place d'un fichier simple. Vous encapsulerez maintenant la gestion des messages dans une classe.

Quelques méthodes utiles et remarques:

Script

createDB.php

```

<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Création de la BD pour le Forum ASI</title>
  </head>
  <body>
    <?php
      error_reporting(E_ALL);
      try {
        $requete="CREATE TABLE IF NOT EXISTS messages (
          id INTEGER PRIMARY KEY,
          date INTEGER NOT NULL,
          email TEXT NOT NULL,
          texte TEXT NOT NULL
        )";

        /* creation de la BD */
        $db=new PDO("sqlite:./DATABASE/messages.sqlite");

        /* errors->exceptions */
        $db->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

        /* requete de creation */
        $db->query($requete);
        unset($db);
        echo 'Table created';
      }
    </?php
  </body>
</html>

```

```

        catch(PDOException $e) {
            echo $e->getMessage();
        }
        ?>
    </body>
</html>

```

forum.inc.php

```

<?php
class Message {
    private $timestamp;
    private $email;
    private $texte;

    public function __construct($email,$texte,$date=null) {
        if ($date==null) {
            $this->timestamp=time();
        }
        else {
            $this->timestamp=$date;
        }
        $this->email=htmlentities($email);
        $this->texte=htmlentities($texte);
    }

    public function getTimeStamp() {
        return $this->timestamp;
    }

    public function getDate() {
        return strftime("%A %d %B %Y %T",(int)$this->timestamp);
    }

    public function getEmail() {
        return html_entity_decode($this->email);
    }

    public function getTexte() {
        return html_entity_decode($this->texte);
    }
}

class Database {
    private $db;

    public function __construct() {

```

```

        error_reporting(E_ALL);
        try {
            $this->db=new PDO("sqlite:./DATABASE/messages.sqlite");
            $this->db->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
        } catch(Exception $e) {
            echo $e->getMessage();
        }
    }

    public function __destruct() {
        try {
            unset($this->db);
        } catch(Exception $e) {
            echo $e->getMessage();
        }
    }

    public function ecrireMessage($message) {
        try {
            $this->db->query("INSERT INTO messages VALUES(NULL,'" . $message->getTimestamp()."', '
            ". $message->getEmail()."', '" . $message->getTexte()."'");
        } catch(Exception $e) {
            echo $e->getMessage();
        }
    }

    public function lireMessages() {
        $messages=array();
        $liste_messages=$this->db->query("SELECT * FROM messages");
        foreach ($liste_messages as $row) {
            $message=new Message($row['email'],$row['texte'],$row['date']);
            $messages[]=$message;
        }
        return $messages;
    }
}
?>

```

forum.php

```

<?php
header('Content-Type: text/html; charset=UTF-8');
setlocale(LC_TIME,"fr_FR.utf8");
require('forum.inc.php');
$db=new Database();
?>

```

```

<!DOCTYPE html>
<html>
  <head>
    <title>ForumASI</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <link href="Ressources/style.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <header id="top">
      
      <h1>ForumASI</h1>
    </header>
    <p id="probleme"></p>
    <form name="formulaire" action="forum.php" method="post">
      <fieldset>
        <legend>Coordonnées:</legend>
        <label for="name">Nom:</label><input placeholder="Saisissez votre nom"
          id="name" name="name" type="text" size="30"/>
        <label for="email">E-mail:</label>
        <input placeholder="Saisissez votre email" id="email" name="email" type="text"
          size="30"/>
      </fieldset>
      <fieldset>
        <legend>Message:</legend><textarea rows="4" cols="50" id="message"
          name="message"></textarea>
      </fieldset>
      <input type="button" value="Poster le message"
        onclick="Javascript:verification();" />
      <input type="button" value="Effacer" onclick="Javascript:nettoyage();" />
    </form>
    <?php //enregistrement du message si submit
      if (isset($_POST) && !empty($_POST['name'])) {
        $message=new Message($_POST['name'].("".$_POST['email']."),$_POST['message']);
        $db->ecrireMessage($message);
      }
    ?>
    <h1>Liste des messages postés</h1>
    <?php // affichage du livre d'or
    foreach ($db->lireMessages() as $message):
    ?>
    <table width="80%" border="1">
      <tr>
        <td class="date"><?php echo $message->getDate()?></td>
        <td class="email"><?php echo $message->getEmail()?></td>
      </tr>

```

```

<tr>
<td colspan="2"><pre><?php echo $message->getTexte()?></pre></td>
</tr>
</table>
<br/>
<?php
endforeach;
?>
<footer>
<style type="text/css" scoped>
    object{ overflow: auto; }
</style>
<style type="text/css" scoped>
    p#probleme{ color: Red; }
</style>
<script type="text/javascript">
    function verification() {
        var emailBox=document.getElementById("email");
        var probleme=document.getElementById("probleme");
        if (emailBox.value=="") {
            if (!probleme.hasChildNodes()) {
                emailBox.style.outline="solid Red";
                probleme.appendChild(document.createTextNode("Champ obligatoire."));
            }
        }
        else {
            document.forms['formulaire'].submit();
        }
    };

    function nettoyage() {
        if (confirm('Etes vous sur ?')) {
            document.getElementById("email").style.outline="initial";
            var probleme=document.getElementById("probleme");
            if (probleme.hasChildNodes()) {
                probleme.removeChild(probleme.childNodes.item(0));
            }
            document.forms['formulaire'].reset();
        }
    };
</script>
</footer>
</body>
</html>

```


Points importants CM

Tableaux superglobaux

- `$_GLOBAL` contient des références sur les variables de l'environnement d'exécution global (clefs = noms des variables globales)
- `$_SERVER` variables fournies par le serveur web
- `$_GET` et `$_POST` variables transmises par les méthodes GET et POST du protocole HTTP
- `$_COOKIE`, `$_REQUEST`, `$_SESSION`, `$_FILES`, `$_ENV`

Constantes

- *Syntaxe*
`define("NOM_DE_LA_CONSTANTE", valeur)`
- ne commencent pas par `$`
- sont définies et accessibles globalement dans tout le code
- ne peuvent pas être redéfinies
- ne peuvent contenir que des booléens, des entiers, des flottants et des chaînes de caractères

Tableaux

- *Exemple fruits.php*

```
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Page PHP</title>
</head>
<body>
  <?php
    $tab=array("fruit"=>"pomme",42,"légume"=>"salade",1.5e3);
    foreach($tab as $cle => $valeur) {
echo "<p>".$cle."=>".$valeur."</p>";
    }
    echo "<p>tab[1]=".$tab[1]."</p>";
    $tab[]="peu importe";
    echo "<p>tab[2]=".$tab[2]."</p>";
    echo "<p>tab['fruit']=".$tab["fruit"]."</p>";
  ?>
</body>
```

- `count($array)` : nombre d'éléments
- `sort($array)` : trie le tableau
- `array_pop($array)` : récupère et supprime le dernier élément d'une liste

- (i.e. fonctionne comme une pile)
- `array_push($array, $elem1, ...)` : ajoute des éléments à la fin d'une
- liste (i.e. fonctionne comme une pile)
- `array_shift($array)` : récupère et supprime le premier élément d'une liste
- `array_unshift($array, $elem1, ...)` : ajout d'éléments en début de liste
- `array_merge($array1, $array2, ...)` : fusionne plusieurs tableaux
- `in_array($elem, $array)` : recherche d'un élément dans un tableau
- `array_key_exists($key, $array)` : recherche une clef dans un tableau
- `array_flip($array)` : inverse les clef et les valeurs d'un tableau

Type d'une variable

```
string gettype($var);
```

Test de type

```
is_integer($var); is_double($var); is_array($var);
```

Conversion dynamique

```
$result = (type-désiré)$var;
```

Vérification d'existence (formulaires)

- `isset($var)`; retourne FALSE si \$var n'est pas initialisée ou a la valeur NULL, TRUE sinon
- `empty($var)` retourne TRUE si \$var n'est pas initialisée, a la valeur 0 , "0" , ou NULL, FALSE sinon
- `is_null($var)`; retourne TRUE si \$var n'est pas initialisée ou vaut NULL, FALSE sinon (inverse de `isset`).