Programmation Web Serveur

Bilan intermédiaire 2 - Formulaires

D'après les cours de Philippe Renevier

Fabien Givors

Université de Nice Sophia Antipolis

Département Informatique

fabien.givors@unice.fr

FONCTIONS

- Mot clef *function*
- Nombre de paramètre quelconque
- Valeur par défaut des paramètre
- Variables internes locales à la fonction
- Résultat unique retourné avec return

```
function maFonction($parametre1 ; $parametre2 =
   "valeur par defaut")
   {
   // ….
      return $unResultat;
   }

$resultat = maFonction();
```

STRUCTURATION DU CODE

- `require()` et `include()` incluent et exécutent un fichier PHP.
  - La commande `require()` se remplace elle-même par le contenu du fichier spécifié
  - `require()` et `include()` sont identiques, sauf dans leur façon de gérer les erreurs. `include()` produit une Alerte (warning) tandis que `require()` génère une erreur fatale. Notamment lorsque le fichier manque.
- `require_once()` et `include_once()`
  - La principale différence est qu'avec `require_once()`, vous êtes assurés que ce code ne sera ajouté qu'une seule fois, évitant de ce fait les redéfinitions de variables ou de fonctions, génératrices d'alertes.
- Structuration du code
- Partage de code

REDIRECTION

- Fonction `header("Location: $url");`
- $url : localisation de la page de redirection
  - Chemin local
  - Chemin absolu (http://...)
- Utilise un champ d'une réponse http
- **DONC A FAIRE AVANT D'EMETTRE UNE REPONSE**
  - i.e. avant d'être dans le document html
  - Le moindre espace compte

NOTION DE SESSION

- Partage de données entre des pages
  - (ou de l'exécution d'une même page)
- Création d'un fichier sur le serveur
  - Unique par navigateur connecté au site
  - Identification par une clef (l'id de session)
  - Stockage dans un cookie (par défaut)
  - Chargement des valeurs automatiquement (sur demande)
- En php

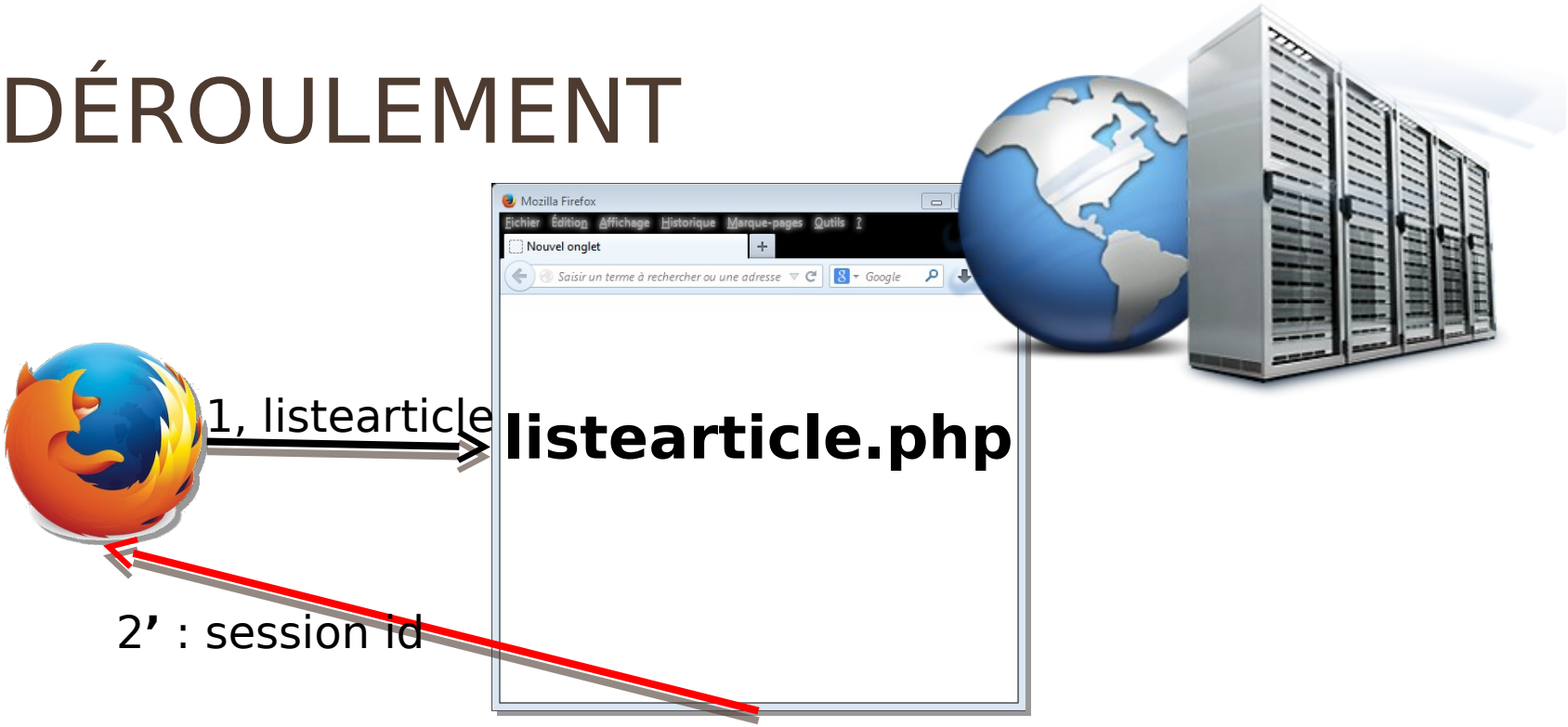
```
// en cas de partage de serveur web
session_name("votre login");
// à faire avant tout retour , echo , caractère « invisible », etc.
session_start();
```

- Au début de la session : création et remplissage de $_SESSION
  - tableau contenant toutes les variables de session
  - Affection = création ou mise à jour
  - `$_SESSION["style"]="blue.css"` : crée une variable de session « style » qui vaut « blue.css »
  - Test d'existence : `isset : if ( isset($_SESSION["style"]) )` …
  - Effacecement : `unset($_SESSION["style"])`
- Destruction de session par un code particulier (c.f. cours 04)

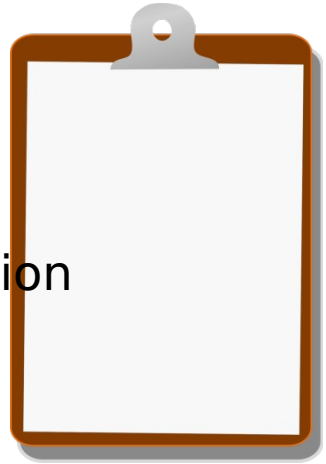
SESSION : COMMUNICATION ENTRE PAGES

- `listearticle.php` :
  - Si nécessaire, initialise la session pour tous les articles
  - Affiche vu / pas vu (nouveau)
- `tp2.php`
  - Si nécessaire, initialise la session pour l'article en cours
    - Attention, on peut arriver directement sur cette page !
  - Indique dans la session que l'article en cours est désormais vu

DÉROULEMENT



2, initialisation… et génération



session

```
session_start();

if (! isset($_SESSION["pagesvues"])) $_SESSION[
"pagesvues"] = array();
```

INITIALISATION POUR CHAQUE PAGE

```php
/** seconde partie : liste de toutes les dossiers **/
$dossiers = glob("news/*", GLOB_ONLYDIR);
// [...]
// parcours du tableau / liste
foreach($dossiers as $indice_dossier => $d)
    {
    // [...]
    // creation si necessaire de la variable de session
    if (! isset($_SESSION["pagesvues"][$indice_dossier]))
        {
        $_SESSION["pagesvues"][$indice_dossier] = array();
        }
    // pour chaque dossier, on recupere la liste des fichiers
    // $d contient le chemin
    $fichiers = glob("$d/*.*");
    // [...]
    foreach($fichiers as $indice_nouvelle => $f)
        {
        // [...]
        // variable de session
        if (! isset($_SESSION["pagesvues"][$indice_dossier][$indice_nouvelle]))
            {
            $_SESSION["pagesvues"][$indice_dossier][$indice_nouvelle] = false;
            }
        // [...]
        }
    // [...]
    }
```

DÉROULEMENT

RETOUR SUR LISTEARTICLE

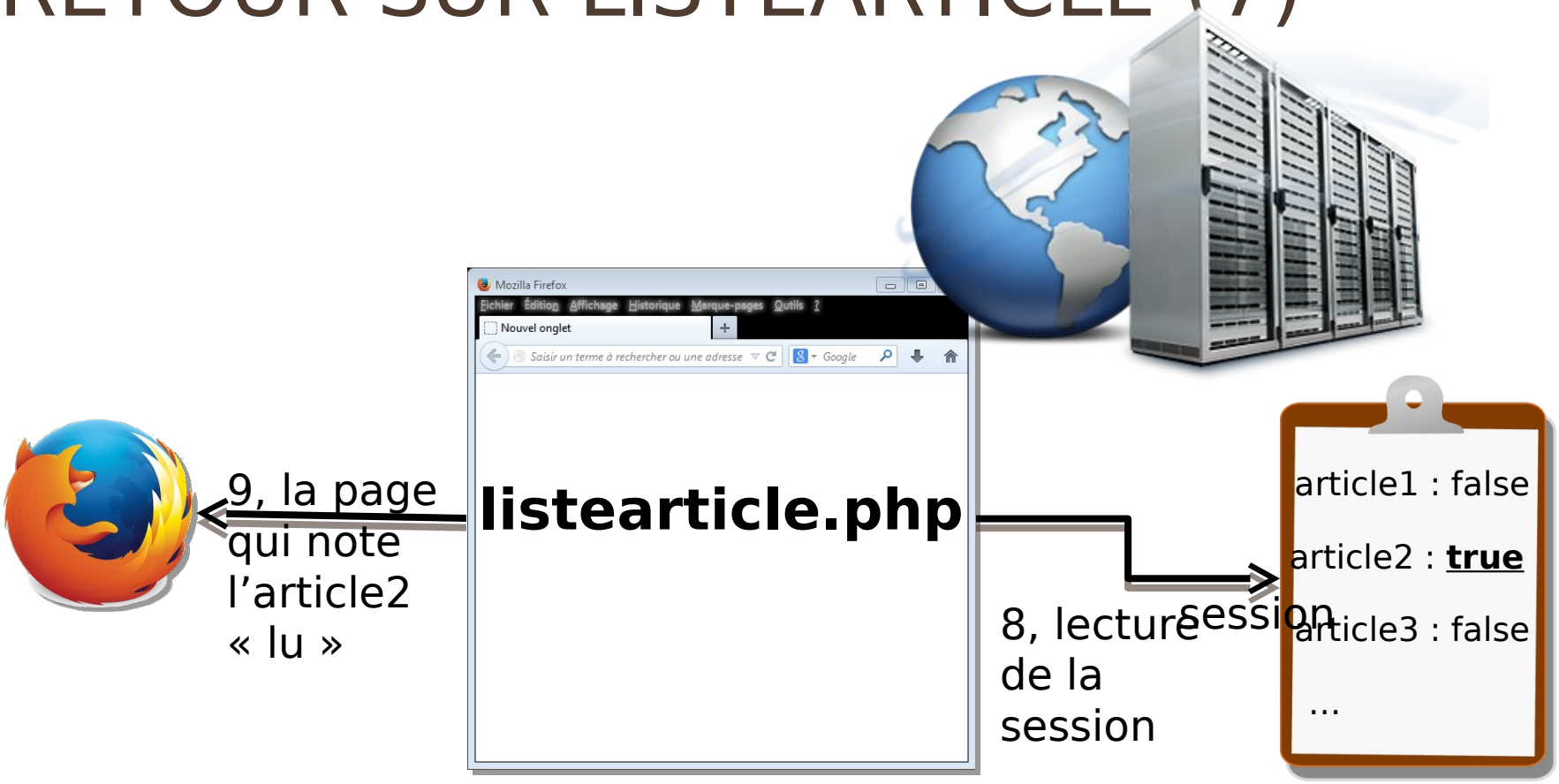
TP2.PHP ET LA SESSION (1/2)

```php
<?php
session_start();
// variable de session :
if (! isset($_SESSION["pagesvues"])) $_SESSION["pagesvues"] = array();


//initialisation des variables d'affichages
// [...]
// dossier courant
$dossiers = glob("news/*", GLOB_ONLYDIR);
$indice_dossier = 2;
$nbDossiers = count($dossiers);
// recuperation s'il y a du dossier en cours

if (isset($_GET["dossier"])) {
    $indice_dossier = intval($_GET["dossier"]);
}

if (($indice_dossier >= 0) && ($indice_dossier < $nbDossiers))
{
    // creation si necessaire de la variable de session
    if (! isset($_SESSION["pagesvues"][$indice_dossier]))
        {
        $_SESSION["pagesvues"][$indice_dossier] = array();
        }
    // [...]
```

TP2.PHP ET LA SESSION (2/2)

```php
    // recuperation s'il y a de l'article en cours
    if (isset($_GET["indice"])) {
        $indice = intval($_GET["indice"]);
    }
    // [...]
    if (($indice >= 0) && ($indice < $nbFichiers))
        {
        //le ficher est lu, il faut le marquer comme tel
        if (! isset($_SESSION["pagesvues"][$indice_dossier][$indice]))
            {
            // creation de la variable de session correspondante si necessaire
            $_SESSION["pagesvues"][$indice_dossier][$indice] = true;
            }
        // sinon mise à jour de la variable de session
        else $_SESSION["pagesvues"][$indice_dossier][$indice] = true;
        // [...]
        }
    // [...]

}
else
{
    // [...]
}
```

RETOUR SUR LISTEARTICLE (7)



LISTEARTICLE ET LA LECTURE DE LA SESSION

```php
foreach($fichiers as $indice_nouvelle => $f)
    {
    // texte gras ou pas
    $texte = $f;
    // fin de phrase, vide par defaut
    $fin = "";

    // variable de session
    if (! isset($_SESSION["pagesvues"][$indice_dossier][$indice_nouvelle
    ]))
        {
        $_SESSION["pagesvues"][$indice_dossier][$indice_nouvelle] = false;
        }

    if (! $_SESSION["pagesvues"][$indice_dossier][$indice_nouvelle])
        {
        // si ce n'est pas lue
        $fin = " (nouvelle non lue)";
        $texte = "<strong>$texte</strong>";
        }

    // on cree une liste item (li) par fichier
    $tousDossiers = $tousDossiers . "<li><a href='tp02.php?indice=
    $indice_nouvelle&dossier=$indice_dossier'>$texte</a>$fin</li>";
    }
```

Pour passer des caractères spéciaux (« / » ; « : » ; etc.) en valeur

ENCODAGE URL

CODAGE D’UNE URL

Tabulation	%09
Espace	%20
"	%22
#	%23
%	%25
&	%26
(	%28
)	%29
+	%2B
,	%2C
.	%2E
/	%2F
:	%3A
;	%3B

<	%3C
=	%3D
>	%3E
?	%3F
@	%40
[	%5B
\	%5C
]	%5D
^	%5E
'	%60
{	%7B
|	%7C
}	%7D
~	%7E

ENCODAGE / DÉCODAGE URL ET PHP

- `urlencode()`
  - Encode une chaîne en URL
  - Cette fonction est utile lors de l'encodage d'une chaîne de caractères à utiliser dans la partie d'une URL, comme façon simple de passer des variables vers la page suivante.
- `urldecode()`
  - Décode une chaîne encodée URL
  - Décode toutes les séquences `%##` et les remplace par leur valeur. Les caractères '+' sont décodés en un caractère d'espacement.
  - Pas besoin avec `$_GET` (c'est fait automatiquement)

Entrer des données depuis un navigateur

FORMULAIRE

PRINCIPES

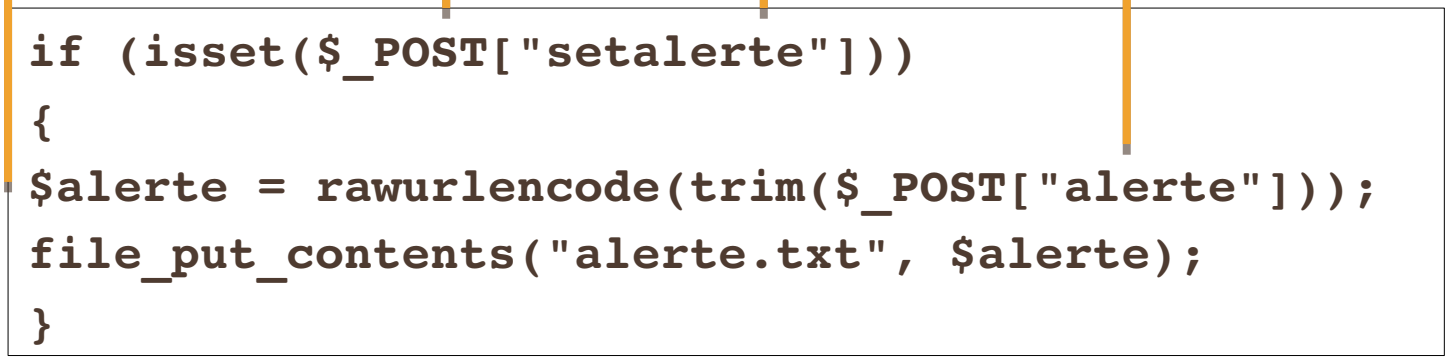
- Envoi de données
  - depuis le navigateur
  - Pour traitement dans le serveur
- Deux façons de faire
  - Passage des informations dans l’url (méthode GET, déjà utilisée)
  - Passage des informations dans la requête (méthode POST)
    - Attention : ce n’est pas sécurisé

EXEMPLE: CÔTÉ NAVIGATEUR / HTML

```
<form action="page.php" method="post">
<fieldset>
    <legend>Placez une alerte</legend>
    <p>Message de l'alerte : <input type="text" name="alerte" /></p>
    <input type="submit" name="setalerte" value="declenchez l'alerte" />
</fieldset>
</form>
```

CÔTÉ SERVEUR / PHP

```
<form action="page.php" method="post">
<fieldset>
    <legend>Placez une alerte</legend>
    <p>Message de l'alerte : <input type="text" name="alerte" /></p>
    <input type="submit" name="setalerte" value="declenchez l'alerte" />
</fieldset>
</form>
```



```
if (isset($_POST["setalerte"]))
{
$alerte = rawurlencode(trim($_POST["alerte"]));
file_put_contents("alerte.txt", $alerte);
}
```

BALISES DE FORMULAIRE : FORM

- Contient des éléments de contrôle de formulaire (bouton, champs, etc.)
  - « block » (sauf form) ou script
  - Attributs
    - action (uri)
    - method ("get" ou "post")
      - get : envoi dans l'url des paires key/value : ? toto=val&titi=val2&...
      - post : envoi
    - enctype (pour une méthode "post")
      - Par défaut : **application/x-www-form-urlencoded -** encodage : espace devient + et les autres non alphanumériques %HH et les retours à la ligne : "CR LF" (i.e., ` %0D%0A')
      - **multipart/form-data - ** envoi en différentes parties (types à préciser à la source)
    - accept-charset (liste - , - d'encodage possible pour les caractères acceptés par le server)
    - Accept (liste - , - de types de contenu acceptés par le server)
    - events : onsubmit et onreset

BALISES DE FORMULAIRE : TYPE DE INPUT

- text : champs d’entrée de texte.
- password : l’écho sont des ‘*’. sécutité pauvre.
- checkbox
- radio (radiobutton)
- submit : un bouton pour envoyer
- image : un bouton submit graphique. Attribut src donne l’URI de l’image. Utiliser l’attribut alt. Les coordonnées du clic sont passés au server sous la forme name.x et name.y
  - problème d’accéssibilité : navigateur non graphique, clic difficile, etc.
  - à remplacer par plusieurs boutons submit ou par des scripts côtés client.
- reset (bouton).
- button : bouton sans comportement prédéfini (script)
- hidden : champs caché (parfois utile pour passer une valeur masquée)
- file : sélection d’un fichier
- + type html 5 : date, email, etc.

BALISES DE FORMULAIRE : INPUT

- Balise vide
- Attributs
  - type
  - name : nom de contrôle (très important)
  - value (valeur initiale ou libellé) : optionnel sauf pour radio et checkbox
  - size (en pixel sauf pour text et password où c’est un nombre de caractère)
  - maxlength : pour text ou password : nombre de caractères maximum
  - checked : pour radio et checkbox
  - src : pour image : la source (ne pas oublier alt)

BALISES DE FORMULAIRE : SELECT

- select : menu
  - (optgroup | option)+
  - attributs
    - name : nom de contrôle
    - size (nombre) : nombre d'éléments visibles pour une scroll list
    - multiple (pas de valeur) : permet la sélection multiple
- option
  - #pcdata (texte)
  - attributs
    - selected : pour présélectionner l'élément
    - value (texte) : pour donner une valeur autre que le texte (#pcdata)
    - label (texte) : pour faire apparaître un autre nom (plus court) à la charge du navigateur !! (pas sûr que cela fonctionne !!)
- optgroup
  - regrouper les options : (option)+
  - attribut : label (texte) : libellé

BALISES DE FORMULAIRE : TEXTAREA

- Champs d’entrée sur plusieurs lignes
- textarea
  - #PCDATA : texte initiale
  - Attributs
    - name : nom de contrôle
    - cols : nombre de colonne
    - rows : nombre de ligne

BALISES DE FORMULAIRE : LABEL

- Permet d’associer un texte à un élément de formulaire sans texte
  - inline
  - *Start tag:* ***required****, End tag:* ***required***
- Attaché par l’attribut for
  - Valeur = id d’un champ de contrôle

EXEMPLE DE FORMULAIRE

```
<form action="" method="post" id="ajouterarticle" style="float: left;margin-right:
2em;">
<fieldset>
  <legend>Ajouter une news</legend>

    <select name="categorie">
    <option value='france'  >france</option>
    <option value='monde'  >monde</option>
    <option value='politique'  >politique</option>
    <option value='sports'  >sports</option>
  </select><br />

  <label for="titre">titre : </label><input type="text" name="titre" value='' /><br
/>
  <label for="auteur">auteur : </label><input type="text" name="auteur" value=''
/><br />
  <label for="texte">texte : </label><textarea  name="texte"></textarea ><br />
  <input type="submit" /></fieldset>
</form>
```

FORMULAIRES : RÉCEPTION EN PHP

- Page qui reçoit le formulaire : attribut « action » du form
- Valeur(s) accessible(s) par :
  - tableau associatif : index est l'attribut « name » de l'input
    - `$_POST` ou `$_GET` sont des « superglobales »
- Un peu de sécurité
  - `trim(htmlspecialchars(addslashes( ) ) )`
  - `str_replace` pour remplacer des caractères « spéciaux »
  - Tests complémentaires…

- Les valeurs peuvent être des tableaux (si le name de l'input est du style nom[]

EXEMPLE DE RÉCEPTION DE FORMULAIRE

```php
// si toutes les valeurs du formulaires sont bien présentes
if (isset($_POST["titre"],$_POST["auteur"],$_POST["texte"],$_POST["categorie"]))
{
  // et si elles sont toutes remplies avec quelques choses
  if ($_POST["titre"] && $_POST["auteur"]
                      && $_POST["texte"] && $_POST["categorie"])
  {
  // on prépare le contenu de la nouvelle / article
  $article = trim($_POST["titre"]) . "\n";
  $article .= trim($_POST["auteur"]) . "\n";
  $article .= trim($_POST["texte"]) ;
  // il faut déterminer un nom de fichier
  $nomFichier = "news/".$_POST["categorie"]."/".$_POST["titre"].".news";
  // si le fichier n'existe pas...
  if (! file_exists($nomFichier) )
    {
     // ... il est créé
    file_put_contents($nomFichier, $article);
    }
  }
}
```