

TP05: Session et fichiers

Ce tp est la suite du TP04. Il faut donc finir le TP04 avant de faire le TP05. Le but de cette suite est de poursuivre la construction du site commencé, un site d'affichage de nouvelles. En plus de la notion de session introduite au TP04, ce TP introduit les notions de fichier :

1. Nous formaterons (afficherons mieux) l'affichage des nouvelles (php et css).
2. Nous ferons la distinction entre les nouvelles lues, lues mais modifié et non lues
3. Nous introduirons une fonction et l'inclusion de fichier pour partager cette fonction

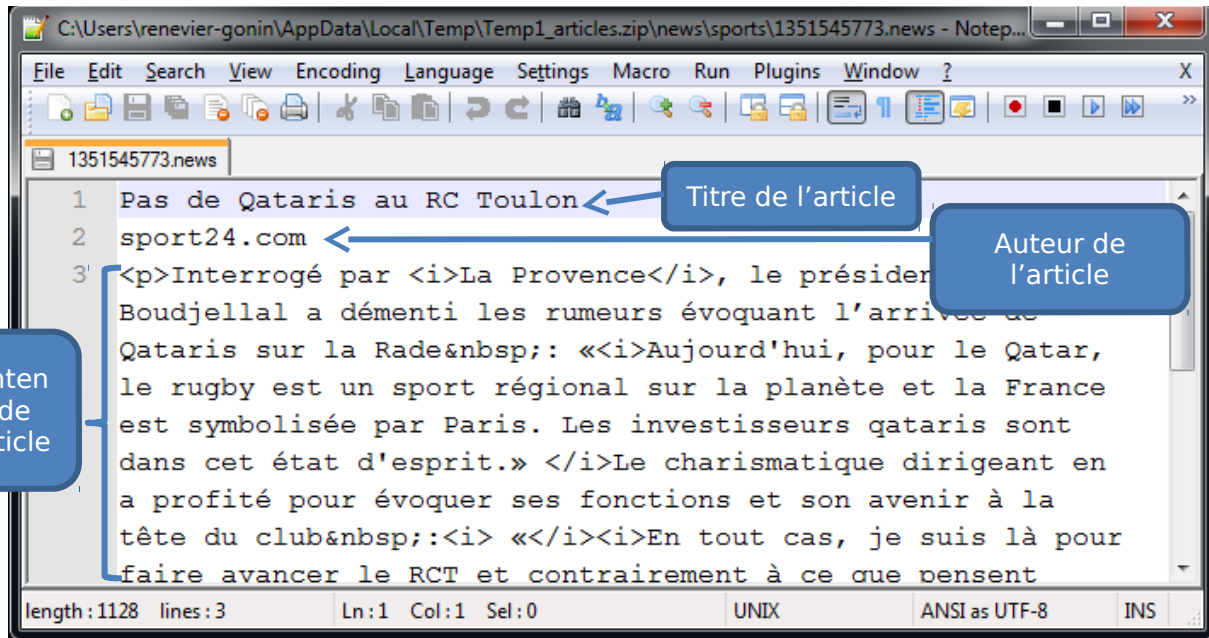
Pour essayer vos pages, vous aurez à modifier des nouvelles dans les catégories avec un éditeur de texte. Comme ce TP est la continuation du TP précédent, vous pouvez travailler dans le même dossier.

1. Affichage (mieux) des nouvelles

Un fichier « .news » est un fichier qui contient un article. Son contenu est le suivant :

1. Sur la première ligne, il y a le titre de l'article
2. Sur la seconde ligne, il y a l'auteur
3. Puis le reste du fichier est le contenu de l'article

Il faut donc modifier le code de tp02.php. Il faut remplacer la fonction « `file_get_contents` » par du code plus évolué. Avec la fonction **file**, vous récupérer le contenu d'un fichier sous forme d'un tableau de chaîne de caractère. Chaque élément du tableau est une ligne du fichier. Dans le cas d'un fichier « .news », le premier élément du tableau retourné par la fonction **file** sera le titre de l'article, le deuxième sera l'auteur, le reste sera le contenu de l'article (c.f. l'illustration ci-dessous)



A partir de ce tableau, il faudra générer un code html semblable à celui ci-dessous, à condition qu'il y ait au moins 2 lignes dans le fichier « .news » (la première pour le titre, la seconde pour l'auteur). Sinon, utilisez file_get_contents (comme avant).

```
<div class="article">
```

```
    <h2>le titre</h2>
```

```
    <p>l'auteur</p>
```

Le reste de l'article, c'est-à-dire la fin du tableau (à partir de l'indice 2, jusqu'au bout)

```
</div>
```

Avec du css associé à la page « article.php » (à mettre dans TP02-style.css) :

- Centrez le titre de l'article, changez sa couleur de fond pour un gris clair opaque à 0.5 (50%), placez un trait en bas (avec border-bottom) [sélecteur div.article > h2 { } : un h2 contenu directement dans une div de classe « article »]
- Le nom de l'auteur sera aligner à droite, écrit en italique [sélecteur div.article > h2 + p { } : un paragraphe qui suit immédiatement un h2 contenu directement dans une div de classe « article »]

Puisque vous êtes à modifier le css, ajoutez à toutes les div une marge en bas de 0.5em.

2. Nouvelles non lues, lues mais modifiées, lues

Pour faire la différence entre les nouvelles lues et les nouvelles non lues, nous avons mémorisé, pour chaque nouvelle, dans `$_SESSION` un booléen : faux si la nouvelle n'était pas lue, vraie si elle l'était.

Entre deux lectures, un article peut avoir été modifié, et de ce fait le nouveau contenu n'est pas connu (cela dépend de la modification). Il serait utile à l'internaute d'avoir un message après le lien :

- « » (rien) si la nouvelle a été lue et n'a pas été modifiée
- « (modifiée) » si la nouvelle a été modifiée depuis qu'elle a été lue
- « (non lue) » si la nouvelle n'a pas été lue

Or, il est possible, avec la fonction **filemtime**(\$fichier) de connaître la date (en temps qu'un nombre de secondes écoulées depuis 1970) de modification du fichier \$fichier, où \$fichier est un chemin vers le fichier, typiquement une des valeurs retournées par la fonction **glob**.

Donc, si nous connaissons la date de lecture du fichier, alors il est possible de la comparer avec la date de modification. Voici comment vous allez faire :

- Dans `tp02.php`, lorsqu'une nouvelle est lue, au lieu de créer/changer sa variable de session à `true`, mémoriser plutôt la date actuelle dans cette même variable de session (à la place d'un booléen):
`$_SESSION["pagesvues"][$indice_dossier][$indice] = strtotime("now") ;`
- Dans `listarticle.php`, la condition pour décorer le lien (gras ou pas, message « non lue » et maintenant « modifiée ») vers la nouvelle se fera selon la valeur de `$_SESSION["pagesvues"][$indice_dossier][$indice]` :
 - o Si `$_SESSION["pagesvues"][$indice_dossier][$indice]` vaut faux (par exemple 0) alors la nouvelle n'a pas été lue
 - o Sinon si `$_SESSION["pagesvues"][$indice_dossier][$indice]` est une date plus petite que la date de modification (fonction **filemtime**) alors la nouvelle a été modifiée depuis la dernière lecture
 - o Sinon la nouvelle est lue et n'a pas été modifiée (pas de message).

Essayez cette nouvelle version de `listarticle.php`, en modifiant des fichiers avec un éditeur de texte, en allant le lire, etc. et en revenant sur `listarticle.php`.

3. Une fonction, qui plus est, partagée

Vous remarquerez que pour trouver le nom d'une nouvelle (ou d'une catégorie), vous utilisez le même code recopié plusieurs fois, un code qui ressemble à :

```
$nomarticle = explode("/", $fichiers[$indice]);  
$nomarticle = $nomarticle[count($nomarticle)-1];
```

Nous allons partager ce code entre les pages tp02.php (là où il est utilisé) et listearticle.php (là où vous vous en servirez pour n'affichez que le nom du dossier, par exemple « sports », à la place de « news/sports »). Créez un dossier « inclusion ». Dans ce dossier, créez un fichier « nom.inc ». Dans ce fichier, créez en php (entre <?php et ?>) une fonction nom, qui prend un paramètre un chemin (vers un fichier) et qui retourne le nom de ce fichier (ou dossier).

Importez ce fichier « inclusion/nom.inc » avec la fonction **include** dans les pages « listearticle.php » et « tp02.php ». Utilisez ensuite cette fonction « nom » aux endroits adéquants (en remplaçant du code dans tp02.php et pour générer les h2 dans listearticle.php).

Pourquoi des fichiers .inc

L'objectif de ne pas utiliser l'extension .php est de faire la différence entre les pages du site (.php) et les fichiers qui structurent la programmation du site (.inc). Par ailleurs, avec une bonne configuration du serveur web apache, il est possible de rendre inaccessible directement certains fichiers :

```
<Files *.inc>
  Order allow,deny
  Deny from all
</Files>
```

Ce principe s'applique aussi extension de fichiers (csv, txt, etc.)

4. Des urls plus stables

Plutôt que d'avoir une indexation pour le dossier et l'article dans les URL et dans \$_SESSION par des numéros, essayez de remplacer ces indices par les noms des dossiers et des fichiers. Ceci permettra de conserver toujours le même lien sur un article (il ne dépendra plus de ses indices).

Pour le passage en paramètre dans l'url, il faudra utiliser la fonction [urlencode](#). Cette fonction transformera les caractères spéciaux (espace, accents, etc.) pour que cela soit compatible avec les caractères possibles dans une URL.

Pour retrouver le fichier ou le dossier à partir de son nom dans la liste retournée par la fonction glob, il faudra utiliser : [array_search](#).

Cette fonction permet de retrouver l'indice d'une valeur dans un tableau. Le premier paramètre est la valeur recherchée, le second le tableau dans lequel on recherche.

Voici un exemple d'utilisation de cette fonction :

```
// dossier courant
$dossiers = glob("news/*", GLOB_ONLYDIR);
// TP 05 : une valeur par défaut... a condition qu'il y ait au moins un dossier...
// l'indice servira pour faire les liens suivant / precedent...
$indice_dossier = false;
// valeur initiale
```

```

$nomDossier= "";
if ($nbDossiers > 0)
{
    if (isset($_GET["dossier"])) {
        // $_GET decode automatiquement
        $nomDossier_temp = $_GET["dossier"];

        // TP 05 et la verification se fera avec "is_dir"
        if (is_dir("news/$nomDossier_temp"))
        // voir avec array_search pour chercher l'indice...
        {
            $nomDossier = $nomDossier_temp;
            $indice_dossier = array_search("news/$nomDossier",
$dossiers);
        }
    }
    else {
        // valeur par default
        $indice_dossier = 0;
        $nomDossier = nom($dossiers[$indice_dossier]);
    }
}

```