

Les Médecins Maîtres-Toile

[Accueil](#) ▶ [Boîte à Outils](#) ▶ [Syndication de contenu](#) ▶ [Afficher un fil RSS/Atom sur votre site 2/3](#)

Publié le : 9 juillet 2005

 [Imprimer cet article](#)

Auteur :
Alain Sichel †



Afficher un fil RSS/Atom sur votre site 2/3

2) Adaptation du script

Après avoir vu dans [l'article précédent](#) l'utilisation de **Fil_LE** version 2.1 de Dominique WOJYLAC dans les situations courantes, nous allons voir comment améliorer la présentation ou rattraper les erreurs de certains fichiers de syndication [1].

Sommaire

- [Le principe](#)
- [Exemple avec les fils des MMT](#)
- [Autres suggestions de nettoyage](#)
- [En résumé](#)

Le principe

La classe `rss_read` version 2.1 `rss_read.inc.php` [2] ne doit pas être modifiée.

L'affichage se fait dans un fichier dont [l'article précédent](#) vous donnait 3 exemples :

- `exemple-1.php`,
- `exemple_a.php`,
- `exemple_b.php`.

C'est dans ce fichier que l'on va traiter les données pour améliorer l'affichage. Dominique WOJYLAC propose déjà dans `exemple-1.php` (lignes 29-30) l'option supprimer les images au sein de la fonction `clean_text()`.

Je préfère utiliser une nouvelle fonction car elle ne servira que pour l'élément *description*, contrairement à `clean_text()` qui a une utilisation plus large. Cette fonction est écrite après `clean_text()`, voici son principe :

```
function nettoyage($description) {
    $avant = array ("1er élément cherché'si", "2e élément cherché'si", "3e
élément cherché'si");
    $apres = array ("élément de remplacement 1", "élément de remplacement
2", "élément de remplacement 3");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Cette fonction est appelée plus loin par :

```
if (trim($items[$i]['description'])!='') {
    echo nettoyage(clean_text($items[$i]['description'], $encode));
}
```

qui remplace :

```
if (trim($items[$i]['description'])!='') {
    echo clean_text($items[$i]['description'], $encode);
}
```

Exemple avec les fils des MMT

J'ai repris *exemple_b.php* en mettant les informations sur les fichiers de syndication des MMT, ce qui donne ceci : *exemple_c.php*. Ce résultat n'est pas terrible :

1. un peu trop long avec tous les paragraphes,
2. les images alignées à gauche perturbent la mise en page,
3. des images sont manquantes,
4. les lignes horizontales ne sont pas utiles...

et encore, j'ai corrigé les squelettes pour que le logo apparaisse !

A chaque problème sa solution :

1. Pour supprimer les paragraphes, je remplace `<p>` par " " et `</p>` par " " ou par `
` si je préfère des retours à la ligne simples.
 2. Je remplacer `align="left"` par `align="right"`, ainsi les images ne gêneront pas la mise en page.
-

- Je constate dans le code du fil des brèves qu'il manque le nom de domaine dans l'URL des images, je remplace donc `<img src='IMG/'` par `http://www.mmt-fr.org/IMG/` et les images seront visibles [3].
- Je remplace `<hr>` par " " et les lignes horizontales disparaissent.

L'écriture sous la forme `"<p>'si"` n'est pas sensible à la casse, c'est-à-dire que cela fonctionne aussi bien avec `<p>` qu'avec `<P>`. Sous la forme `"<p.*?>'si"` on élargit aux éventuels caractères supplémentaires comme par exemple `<p class="spip">`

La fonction nettoyage devient alors :

```
function nettoyage($description) {
    $avant = array ("<p.*?>'si", "</p>'si", "<br.*?>'si",
    "'align=\"left\"'si", "'<img src='IMG/'si", "'<hr.*?>'si");
    $apres = array (" ", " ", " ", "align=\"right\"", "<img src='http://www.mmt-
fr.org/IMG/', " ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Pour voir le résultat, regardez [exemple-2.php](#) pour le fil articles seul et [exemple_d.php](#). Tout n'est pas parfait, mais c'est présentable [4].

Autres suggestions de nettoyage

Suppression des images :

Si les images posent trop de problèmes, on peut les supprimer complètement avec :

```
function nettoyage($description) {
    $avant = array ("<img.*?>'si");
    $apres = array (" ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Suppression des liens :

Si on veut supprimer les liens, il suffit de faire :

```
function nettoyage($description) {
    $avant = array ("<a.*?>'si", "</a>'si");
    $apres = array (" ", " ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

```
}
```

Suppression des caractéristiques du texte :

Si on vous êtes gêné par des balises précisant la couleur, la taille du texte... vous pouvez les supprimer avec :

```
function nettoyage($description) {
    $avant = array ("<font.*?>'si", "</font>'si");
    $apres = array (" ", " ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Suppression du style du texte :

Si par contre c'est le texte en gras, en italique ou souligné qui vous chagrine, voilà comment faire :

```
function nettoyage($description) {
    $avant = array ("<i.*?>'si", "<em.*?>'si", "<b.*?>'si", "<strong.*?>'si",
    "<u.*?>'si", "</i>'si", "</em>'si", "</b>'si",
    "</strong>'si", "</u>'si");
    $apres = array (" ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Suppression des listes :

Voilà la solution si vous ne voulez plus des listes à puces ou numérotées :

```
function nettoyage($description) {
    $avant = array ("<ul.*?>'si", "<ol.*?>'si", "<li.*?>'si",
    "</ul>'si", "</ol>'si", "</li>'si");
    $apres = array (" ", " ", " ", " ", " ", " ", " ");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

Suppression des tableaux :

Voilà comment éliminer les tableaux :

```
function nettoyage($description) {
```

```
$avant = array ("<table.*?>'si", "<tr.*?>'si", "<td.*?>'si",
"</table>'si", "</tr>'si", "</td>'si");
$apres = array (" ", " ", " ", " ", " ", " ", " ", " ");
$description = preg_replace($avant, $apres, $description);
return $description;
}
```

En résumé

On dispose avec classe `rss_read` de [fil_LE](#) version 2.1 d'un lecteur facile à mettre en œuvre, qui récupère bien mieux et plus vite les éléments d'un fil de syndication que les lecteurs dont nous disposions avant. Il gère aussi bien mieux les caractères spéciaux.

Bref, il ne lui manque qu'un cache pour éviter de trop en demander au serveur pour un fil que l'on souhaite présenter régulièrement sur son site... ce sera l'objet de notre [troisième article](#).

[1] Pour bien comprendre cet article, il faut avoir lu [l'article précédent](#).

Volontairement, pour ne pas faire trop long, je ne vais pas expliquer tout le code, je crois que pour cela les explications de la documentation [doc_rss_read.html](#) sont suffisantes, je me limite donc à quelques exemples d'adaptation du script.

[2] Pour télécharger `fil_LE` [cliquez ici](#).

[3] Il n'y a pas que le site des MMT qui soit dans ce cas, j'ai vu cela notamment sur de nombreux fils générés par SPIP (en particulier pour les brèves), revoyez vos squelettes pour que les liens soient absolus !

[4] C'est d'ailleurs [présenté ici](#).

Document joint à l'article



Exemples_2.zip - Ce fichier contient `exemple_c.php` et `exemple_d.php` décrits dans cet article.



 [Imprimer cet article](#)

