http://dutailly.net/la-dist-de-spip-sur-mesure-linguistique

La dist de SPIP : sur-mesure linguistique

- Spip et al -

Date de mise en ligne : lundi 15 décembre 2014

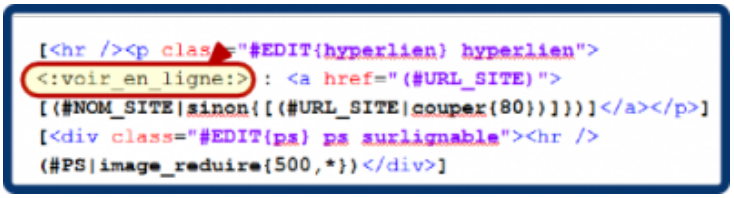
Date de parution : 15 décembre 2014

Tutoriels et logiciels libres Licence Creative Commons Attribution - Partage dans les Mêmes Conditions 4.0 International

Dans la série sur la personnalisation de la dist de SPIP j'avais signalé le fait que le multilinguisme fait partie de l'ADN de SPIP, mais sans aller plus loin. Ce tutoriel a pour propos de préciser comment personnaliser et adapter la dist au niveau linguistique et de laisser, le cas échéant, la porte ouverte à une internationalisation du site s'il devait, dans un premier temps n'être que dans une seule langue.

Des expressions bizarroïdes

En travaillant sur le fichier sommaire.html, on a pu voir des mentions un peu bizarres [1] du type :



Écriture de la commande La ligne de commande dans un éditeur de texte.

Soit :

```
<:voir_en_ligne:> :
```

Et on a pu être tenté de modifier directement la mention de cette façon :

```
 Sur d'autres sites :
```

voire, pour reprendre l'exemple du site Numéricoach [http://numericoach.net] :

```
Les derniers articles de Tutoriels et logiciels libres
```

Qui affichera très exactement ceci :

Derniers articles Ligne Â« dernier articles Â» du squelette numericoach

Rien à redire sur cette façon de procéder sauf que... si on veut modifier l'intitulé il faudra revenir modifier le ou les fichiers qui comportent la mention. Et, bien évidemment, si on veut, plus tard, que notre site soit dans d'autres langues, ça ne fonctionne plus puisque SPIP affichera le texte saisi ainsi « en dur ».

Cette syntaxe qui encadre l'élément textuel entre deux-points et symboles <> indique à SPIP d'aller chercher l'information dans un fichier de langue annexe. Ce qui facilite le multilinguisme en particulier et les modifications en général puisqu'on n'a pas à aller à la pêche aux éléments qui comportent l'expression quand on veut la corriger ou la changer. À la place, on utilisera des fichiers de langues.

Le principe

En fait SPIP, comme d'ailleurs les autres CMS et la plupart des logiciels, fonctionne sur la base d'un système de référence qui « nomme » les différentes mentions et indique ce qui doit être affiché à raison d'un fichier par langue. Sous SPIP, ces fichiers sont des fichiers php [2] dont la deuxième partie du nom comporte **obligatoirement** l'étiquette de la langue concernée séparée du nom du fichier par un trait de soulignement (undescore), touche « 8_ \ » du pavé alphabétique.

Il doit, tout aussi obligatoirement, être rangé dans le dossier « lang » de « squelettes », à créer si ce n'est déjà fait.

Exemple concret : pour le site Numéricoach le fichier pourrait s'appeler « numericoach_en » pour la version anglaise et la version espagnole serait « numericoach_es ».

Le nom du fichier, avant l'étiquette de langue, n'est pas impératif sauf dans certains cas. Par exemple, le squelette Escal est livré avec un dossier « lang » et des fichiers personnalisés (et surtout à personnaliser) appelés « local ».

Il n'y a aucune hiérarchie entre les divers fichiers de langue et il n'y a pas à déclarer ces fichiers comme on le fait pour les feuilles de style.

Chaque fichier de langue doit commencer par ceci :

```
 <?php
if (!defined('_ECRIRE_INC_VERSION')) return;
$GLOBALS[$GLOBALS['idx_lang']] = array(
```

et terminer par cela :

```
 );
?>
```

Tous les éléments figurant entre les parenthèses forment le « dictionnaire de références » de notre site.

La syntaxe

Comme d'habitude en informatique, il faut respecter très exactement la syntaxe sinon on n'obtient pas le résultat attendu (voire on n'obtient rien du tout d'ailleurs).

Comme il n'y a aucune hiérarchie entre les langues, prendre le fichier de celle dans laquelle on est le plus familier, ici le français, et saisir les noms des expressions (ou chaine de caractère) également dans notre langue favorite, donc en français dans ce cas de figure, mais ça pourrait aussi bien être du letton, du russe, du coréen, l'anglais...

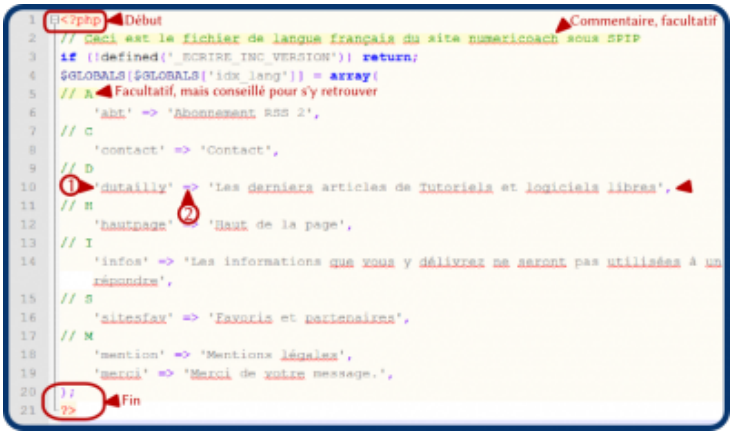
Une référence (ou chaine de caractères) s'indique de cette façon :

```
'merci' => 'Merci de votre message.',
```

où

1. le nom de la mention figure entre apostrophe : « ' », il vaut mieux qu'il soit facilement mémorisable et court et sans caractères dits spéciaux ;
2. la flèche se fait par le clavier, touche égal, =, et supérieur > ;
3. l'expression à afficher est également entre apostrophes et suivie impérativement par une virgule.

Sinon... une fois que votre fichier est sur le serveur ainsi que le ou les fichiers html qui comportent la mention, votre site ne sera pas accessible et vous aurez un message d'erreur qui vous indique quelle ligne du fichier fautif il faut corriger.



Capture d'écran du fichier de langue en php Le glossaire personnalisé décrypté.

Comme on peut le voir dans l'exemple un mot peut équivaloir à une expression nettement plus longue.

Pour les autres fichiers, on remplace l'expression à afficher par la version dans la langue concernée en gardant le nom **(1)** et évidemment la flèche **(2)**.

Les trois versions pour une mention Accès direct et Accueil en trois versions !

L'indication sur les fichiers html

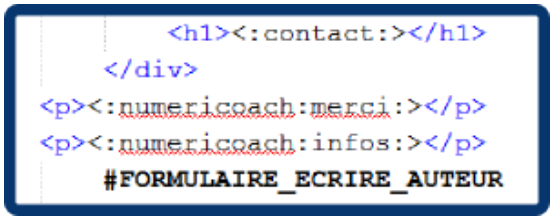
Là encore il faut être extrêmement respectueux de la syntaxe, sinon pas d'affichage du terme voulu [3].

On écrira l'indication de la façon suivante :

```
<:local:terme:>
```

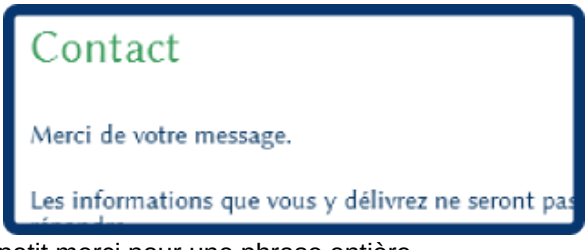
en faisant extrêmement attention au nombre de deux points ! Si on oublie par exemple celui de fin, la mention n'est simplement pas prise en compte.

Exemple concret :



Indication d'une mention sur un fichier html SPIP. Attention aux deux points qui permettent que ce contenu soit dynamique.

Un simple « merci » qui donne à l'affichage :



Page contact numéricoach Un petit merci pour une phrase entière.

Pour aller plus loin

Sait-on jamais, si après tout cela vous vous sentez des ailes et une envie dévorante de traduire SPIP dans une langue encore inédite, ou de finir la traduction de l'interface :

- Comment traduire SPIP ? [http://www.spip.net/fr_article1.html]

---

[1] Pour le débutant tout au moins, l'initié, lui, sait.

[2] Rassurez-vous, vous n'aurez pas besoin d'apprendre ce langage pour les créer.

[3] J'ai, hum, testé...