

Memento Smarty

Catégorie : Fiches techniques

Publié par [DuGris](#) le 16/04/2005

INTRODUCTION Ce document a pour but d'expliquer le rôle joué par Smarty au sein de Xoops, et de décrire l'essentiel à connaître de Smarty dans le cadre de son utilisation avec Xoops :

- les méthodes pour les scripts,
- les fonctions et variables pour les templates.

Pour tout savoir sur Smarty, une documentation très complète et en français, est disponible sur le site officiel de [Smarty](#) **Sommaire** 1) [Généralités sur Smarty](#)

- Les templates
- Le rôle de Smarty
- Le cache 2) [Notions de base](#) 3) [Les méthodes pour les scripts](#) 4) [Les variables Smarty](#) 5)

Les fonctions Smarty

1) Généralités sur Smarty Les templates

Les templates ont été créés pour permettre la séparation entre le traitement de l'information et son affichage. On a ainsi :
le Php pour le traitement des données: interface avec la base de données, calculs, gestion des cookies et sessions, traitement des chaînes de caractères, etc.
le Html, associé aux feuilles de style, pour la présentation des données dans des tableaux, avec couleurs et images. Les templates sont donc des fichiers Html particuliers, dans lesquels on trouvera :

- des **variables** (,) qui seront remplacées, au moment de l'appel de la page, par le contenu que leur aura assigné le script Php
- des **fonctions** (,) permettant de conditionner l'affichage de certaines parties au résultat d'un test, d'inclure un autre template, ou encore de balayer un tableau de données transmis par le script Php. Les templates permettent donc de modifier facilement l'apparence d'une page avec un éditeur Html, sans avoir à intervenir dans le code Php.

Le rôle de Smarty

Smarty est une application indépendante de Xoops que ce dernier utilise pour l'affichage par templates, ainsi que leur compilation et la gestion du cache.

C'est ce que l'on appelle un moteur de template dont le rôle est d'assurer l'interfacage entre le script Php et le template.

La configuration générale de Smarty est réalisée par Xoops, qui se charge également de l'interface utilisateur pour le paramétrage du cache, qui, nous le verrons ensuite, est une fonctionnalité de Smarty (et non de Xoops).

L'une des particularités de Smarty est d'intégrer un compilateur de templates, lesquels seront enregistrés dans la base de données. C'est la raison pour laquelle, après toute modification de template, il faut exécuter une mise à jour du module ou du système pour que ces modifications soient prises en compte.

Le cache

L'un des inconvénients de l'utilisation d'un moteur de templates est qu'il a besoin lui aussi de ressources serveur pour fonctionner. Le temps de traitement d'une page utilisant les templates sera donc plus long qu'un affichage direct par le script Php.

Pour compenser cet inconvénient, les développeurs de Smarty ont donc intégré un système de mise en cache : certaines pages, ou portions de pages, ne sont pas générées à chaque appel du navigateur, mais stockées directement en Html, prêt à l'emploi, dans un répertoire spécifique ([templates_c](#)).

En outre, même si le paramétrage du cache et l'interfaçage avec l'utilisateur pour les modules ou blocs à mettre en cache est réalisé par Xoops, ce système de cache est totalement géré par Smarty.

[Retour au sommaire](#)

2) Notions de base [Méthodes](#)

Les méthodes sont utilisées dans les scripts **Php**.

Elles permettent d'assigner des valeurs aux variables smarty par valeur ou référence, sous forme de valeurs simples ou de tableaux.

La classe XoopsTpl étant une classe dérivée de la classe Smarty, toutes les méthodes de Smarty peuvent être utilisées avec

`$xoopsTpl->methode_Smarty` (ex. `xoopsTpl->append('stories', $story)`)

[Méthodes Smarty](#)

[Délimiteurs](#)

Les constantes, variables et fonctions, utilisées dans les **templates**, doivent être délimitées.

Smarty autorisant la personnalisation des délimiteurs, les symboles **et }> sont ceux défini par Xoops.** [Commentaires](#)

Il est possible d'insérer des commentaires dans les templates sous la forme :

Attention, ils n'apparaîtront pas dans le code source de la page, contrairement aux commentaires Html. [Fonctions](#)

Les fonctions de Smarty sont les fonctions utilisables dans les templates:

Elles permettent notamment :

- l'inclusion d'autres templates
- des instructions conditionnelles
- le parcours de tableaux pour en afficher les données

Ne seront détaillées que les fonctions utilisées couramment avec Xoops (include, section, if, etc.)

[Fonctions Smarty](#)

[Variables](#)

Les variables sont utilisées dans les templates :

- soit pour être affichées directement
- soit pour être utilisées comme arguments de fonctions ou au sein d'instructions conditionnelles.

On distingue 3 types de variables :

- Les variables réservées Smarty
- Les variables chargées depuis des fichiers de configuration (non développées

ici)

- Les variables assignées depuis Php; nous listerons les variables assignées par Xoops, comme ou

[Variables Smarty](#) [Modificateurs de variables](#)

Les modificateurs de variables peuvent être appliqués aux variables, fonctions et chaînes de caractères.

Ils sont utilisés pour des modifications du texte : mise en majuscules, minuscules, troncature, suppression d'espace, etc.

Indiqués pour mémoire, n'étant en principe pas utilisés avec Xoops, les traitements de chaînes de caractères étant réalisés dans le Php.

Ex : troncature du texte 'sujet' à 40 caractères, suivi de ...

[Retour au sommaire](#)

[Smarty : méthodes](#) La classe `xoopsTpl` (fichier [class/template.php](#)) est une classe dérivée de la classe `Smarty`.

Toutes les méthodes de la classe `Smarty` peuvent donc être utilisées.

Sauf exception (voir [display](#)), cette classe n'a pas besoin d'être instanciée, puisque la création de l'objet `template` est fait par `xoops` dans `header.php` :

```
require_once XOOPS_ROOT_PATH.'/class/template.php';
xoopsTpl = new XoopsTpl();
```

[assign](#) `Smarty` assignation d'une valeur

[assign_by_ref](#) `Smarty` assignation d'une valeur par référence

[append](#) `Smarty` ajout d'une valeur

[append_by_ref](#) `Smarty` ajout d'une valeur par référence

[display](#) `Smarty` affichage du template

[xoops template clear module cache](#) `xoopsTpl` Effacement du cache du module

NB: de nombreuses méthodes (de la classe `Smarty`, ou de la classe dérivée) ne sont pas écrites, étant utilisées principalement par le core de Xoops.

[assign](#)

Permet d'assigner une valeur à un template.

La valeur est soit une chaîne, soit un tableau associatif. [assign \(string | array \\$tpl_var, mixed \\$value = null\)](#)

`$tpl_var` string ou array nom de la variable de template à créer

`$value` mixed valeur à lui transmettre

[Exemples](#)

// Passage d'une paire nom/valeur

```
$xoopsTpl->assign('displaynav', false);
```

// Passage d'un tableau associatif

```
$xoopsTpl->assign(array(
    "lang_partner" => _MD_PARTNER,
    "lang_desc" => _MD_DESCRIPTION,
    "lang_hits" => _MD_HITS, ));
```

[assign_by_ref](#)

Identique à assign, sauf que la donnée est passée par référence (et non par valeur).

`assign_by_ref (string | array $tpl_var, mixed $value = null)`

`$tpl_var` string ou array nom de la variable de template à créer

`$value` mixed Référence de la valeur à lui transmettre

append

Permet d'ajouter une valeur à une variable tableau d'un template.

Si le tableau n'existe pas, il est d'abord créé.

Dans le template, le tableau peut être parcouru avec la fonction [foreach](#) ou [section](#)

`append (string | array $tpl_var, mixed $value = null)` [Exemples](#) (avec fonction section dans le template)

// Dans le code PHP

```
for ( $i = 0; $i < count($partner); $i++) {
    $partner[$i]['hits'] = $array_partners[$i]['hits'];
    $partner[$i]['title'] = $array_partners[$i]['title'];
    $partner[$i]['descr'] = $array_partners[$i]['description'];
    $xoopsTpl->append("partners", $partner[$i]);
}
```

append_by_ref

Identique à append, sauf que la donnée est passée par référence (et non par valeur) [Exemple](#) (avec boucle foreach dans le template)

// Dans le code PHP

```
while (list($id, $name) = $xoopsDB->fetchRow($result)) {
    $category = array();
    $category['name'] = $name;
    $category['id'] = $id;
    $sql = 'SELECT faq_id, faq_title FROM '.$xoopsDB->prefix('xoops_faq').'
    WHERE cat_id='.$id;
    $result = $xoopsDB->query($sql);
    while ($myrow = $xoopsDB->fetchArray($result)) {
        $category['questions'][] = array('id' => $myrow['faq_id'], 'title' =>
        $myrow['faq_title']);
    }
    $xoopsTpl->append_by_ref('categories', $category);
    unset($category);
}
```

display

Affichage du template.

Pas ncessaire lorsque l'on utilise l'affichage par le template principal (template_main) puisque le footer.php se charge d'effectuer cette action.

A n'utiliser que dans les cas particuliers d'affichage sans le theme (exemples : site closed, redirect header, image_manager,...) `$xoopsTpl->display (string $template)`

Exemple

```
include_once XOOPS_ROOT_PATH.'/class/template.php';
$xoopsTpl = new XoopsTpl();
$xoopsTpl->assign(array('sitename' => $xoopsConfig['sitename'],
'xoops_themecss' => ... ));
$xoopsTpl->xoops_setCaching(1);
$xoopsTpl->display('db:system_siteclosed.html');
exit();
```

xoops_template_clear_module_cache

Permet de vider le cache du module. `xoops_template_clear_module_cache (int $mid)`

\$mid int identifiant du module

Exemple (aprAs une m Ã j dans l'admin d'un module)

```
include_once XOOPS_ROOT_PATH.'/class/template.php';
xoops_template_clear_module_cache($xoopsModule->getVar('mid'));
```

[Retour au sommaire Smarty](#)

Version Xoops de rÃfÃrence: 2.06

C. Felix AKA theCat le 22/06/04

[Smarty : variables](#) Ces variables smarty, qu'elles soient assignÃes par Xoops ou rÃservÃes pour Smarty sont utilisables dans tous les templates. [Variables assignÃes](#) - Variables assignÃes par Xoops dans le fichier `header.php`, Ã partir des donnÃes de la Bdd.

Ces valeurs sont celles enregistrÃes dans la page Admin systÃme >> PrÃfÃrences >> MÃta balises et pied de page

Balise mÃta: mots clefs

Balise mÃta: description

Balise mÃta: auteur

Balise mÃta: copyright

Balise mÃta: robots

Pied de page

- Variables assignÃes par xoops dans le fichier `header.php`

Nom du site *

Titre de la page *

Nom du theme *

Url de la feuille de style du theme *

Inclusion de javascript du fichier include/xoops.js *

Affichage bannière *

Affichage du contenu de la page *

0 ou 1 selon que le bloc centre doit être affiché *

0 ou 1 selon que le bloc gauche doit être affiché *

0 ou 1 selon que le bloc droit doit être affiché *

Url demandée, ayant conduit sur cette page

Slogan du site

Url des images du thème *

1 si administrateur

1 si user

id de l'utilisateur, 0 si anonyme

pseudo de l'utilisateur, sinon terme choisi pour anonyme

* Variables nécessaires au template du thème: theme.html

- Variables assignées par xoops dans le fichier **template.php**

Ex. ISO-8859-1

Code de la langue 2 caractères: en, fr, etc.

Url du site

Chemin physique du répertoire de xoops

Version de Xoops (ex. Xoops 2.0.5.1)

Url du répertoire d'upload

Variables réservées >

Permet d'accéder directement aux constantes Php depuis le template.

Utilisable en particulier pour afficher les 'defines' de langage, sans passer par un xoopsTpl->assign dans le Php.

Ex :

Permet d'afficher le timestamp courant

Ex :

Variables de requêtes

Pour mémoire.

Permettent d'accéder à différentes variables de GET, POST, COOKIES, SERVER, ENVIRONNEMENT et SESSION

[Retour au sommaire](#)

C. Felix AKA theCat le 15/06/04

Smarty : fonctions

Smarty dispose en standard de plusieurs fonctions utilisables dans les templates.

On distingue les fonctions natives (directement intégrées au langage) et les fonctions utilisateurs (répertoire class/smarty/plugins).

Ne seront décrites que les fonctions utilisées habituellement avec Xoops. De même tous les attributs des fonctions ne sont pas forcément décrits.

Consultez le [site Smarty](#) pour une information plus complète.

include	native	inclusion d'un template dans un autre
if, elseif, else	native	instruction conditionnelle
foreach, foreachelse	native	parcours d'un tableau associatif
simple		
section, sectionelse	native	parcours d'un tableau associatif
complexe		
cycle	utilisateur	détermination cyclique de valeurs

[include](#)

Permet d'inclure un template à l'intérieur d'un autre template.

Il est possible de transmettre des variables au template inclus, sous forme d'attributs de la fonction.

Xoops stockant les templates en Bdd, le nom du template inclus sera précédé de **db :**

Nom attribut	Requis	Type	Description
file	Oui	string	nom du template à inclure
[var...]	Non	mixed	variable(s) à passer au template

Include simple

Include avec passage de variable (story)

[if, elseif, else](#)

Instruction conditionnelle permettant l'exécution de tests.

Doit obligatoirement se terminer par la balise .

Permet d'utiliser les opérateurs logiques (forme littérale ou symbolique).

Attention, les opérateurs doivent être entourés d'espaces.

Opérateurs

==

!=

>

>= !
|| &&
 neq eq gt
gte lte lt not
or
 and
is even
 is odd is not even
is not odd is odd by
 is even by div by
 mod

Exemple

```

if $gender == 'male'>
  Bonjour Monsieur
elseif $gender == 'female'>
  Bonjour Madame
else}>
  Bonjour
  
```

Exemple avec opérateurs

```

if ($price >= 100 and $nbre > 9) or $code == 'promo' >
  Remise de 10%
  
```

Exemple pour affichage sur 3 colonnes

Dans une table, passage à la ligne suivante lorsque la valeur du compteur est un multiple de 3.

```

if $category.count is div by 3}>
  
```

/if}>

foreach, foreachelse

Permet de parcourir un tableau associatif simple.

Doit obligatoirement se terminer par la balise

Les boucles peuvent être imbriquées (pas de limitation de nombre).

foreachelse sera exécuté si aucune valeur n'est trouvée dans la variable de

	Nom attribut	Requis	Type	Description
l'attribut	from	Oui	string	nom du tableau à parcourir
accéder à l'élément	item	Oui	string	nom de la variable pour
accéder à ses propriétés	name	Non	string	nom de la boucle foreach pour

Exemple

_____ Parcours du tableau associatif \$topics contenant les paires clefs/valeurs :
post_image, post_title, poster_uname.
foreach item = sujet from = \$topics)>

/foreach}>

section, **sectionelse**
Permet de parcourir un tableau. Utilis e pour les tableaux complexes(multidimensionnels p.ex.).
Doit obligatoirement se terminer par la balise
Les sections peuvent  tre imbriqu es (pas de limitation de nombre).
sectionelse sera execut e si aucune valeur n'est trouv e.
Nom attribut Requis Type Descrip