MÃ©mento Smarty

CatÃ©gorie : Fiches techniques

PubliÃ© par DuGris le 16/04/2005

INTRODUCTION Ce document a pour but d'expliquer le rÃ´le jouÃ© par Smarty au sein de Xoops, et de dÃ©crire l'essentiel Ã¡ connaitre de Smarty dans le cadre de son utilisation avec Xoops :

- les mÃ©thodes pour les scripts,
- les fonctions et variables pour les templates.

Pour tout savoir sur Smarty, une documentation trÃ¨s complÃ¨te, et en franÃ§ais, est disponible sur le site offficiel de Smarty **Sommaire**

1) GÃ©nÃ©ralitÃ©s sur Smarty Les templates

Les templates ont Ã©tÃ© crÃ©Ã©s pour permettre la sÃ©paration entre le traitement de l'information et son affichage. On a ainsi :

- le Php pour le traitement des donnÃ©es: interfaÃ§age avec la base de donnÃ©es, calculs, gestion des cookies et sessions, traitement des chaines de caractÃ¨res, etc.
- le Html, associÃ© aux feuilles de style, pour la prÃ©sentation des donnÃ©es dans des tableaux, avec couleurs et images. Les templates sont donc des fichiers Html particuliers, dans lesquels on trouvera :

- des **variables** (, .) qui seront remplacÃ©es, au moment de l'appel de la page, par le contenu que leur aura assignÃ© le script Php
- des **fonctions** (,) permettant de conditionner l'affichage de certaines parties au rÃ©sultat d'un test, d'inclure un autre template, ou encore de balayer un tableau de donnÃ©es transmis par le script Php. Les templates permettent donc de modifier facilement l'apparence d'une page avec un Ã©diteur Html, sans avoir Ã¡ intervenir dans le code Php.

Le rÃ´le de Smarty

Smarty est une application indÃ©pendante 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 rÃ´le est d'assurer l'interfacage entre le script Php et le template.

La configuration gÃ©nÃ©rale de Smarty est rÃ©alisÃ©e par Xoops, qui se charge Ã©galement de l'interface utilisateur pour le paramÃ©trage du cache, qui, nous le verrons ensuite, est une fonctionnalitÃ© de Smarty (et non de Xoops).

L'une des particularitÃ© de Smarty est d'intÃ©grer un compilateur de templates, lesquels seront enregistrÃ©s dans la base de donnÃ©es. C'est la raison pour laquelle, aprÃ¨s toute modification de template, il faut exÃ©cuter une mise Ã¡ jour du module ou du systÃ¨me pour que ces modifications soient prises en compte.

Le cache

L'un des inconvÃ©nients 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 inconvÃ©nient, les dÃ©veloppeurs de Smarty ont donc intÃ©grÃ© un systÃªme de mise en cache : certaines pages, ou portions de pages, ne sont pas gÃ©nÃ©rÃ©es Ã¡ chaque appel du navigateur, mais stockÃ©es directement en Html, prÃªtes Ã¡ l'emploi, dans un rÃ©pertoire spÃ©cifique (templates_c).

LÃ¡ encore, mÃªme si le paramÃªtrage du cache et l'interfacage avec l'utilisateur pour les modules ou blocs Ã¡ mettre en cache est rÃ©alisÃ© par Xoops, ce systÃªme de cache est totalement gÃ©rÃ© par Smarty.

Retour au sommaire

2) Notions de base MÃ©thodes

Les mÃ©thodes sont utilisÃ©es dans les scripts **Php.**

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

La classe XoopsTpl Ã©tant une classe dÃ©rivÃ©e de la classe Smarty, toutes les mÃ©thodes de Smarty peuvent Ãªtre utilisÃ©es avec

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

MÃ©thodes Smarty

DÃ©limiteurs

Les constantes, variables et fonctions, utilisÃ©es dans les **templates**, doivent Ãªtre dÃ©limitÃ©es.

Smarty autorisant la personnalisation des dÃ©limiteurs, les symboles **et }> sont ceux dÃ©fini par Xoops.**

Commentaires

Il est possible d'insÃ©rer des commentaires dans les templates sous la forme :

Attention, ils n'apparaitront 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 donnÃ©es

Ne seront dÃ©taillÃ©es que les fonctions utilisÃ©es couramment avec Xoops (include, section, if, etc.)

Fonctions Smarty

Variables

Les variables sont utilisÃ©es dans les templates :

- soit pour Ãªtre affichÃ©es directement

- soit pour Ãªtre utilisÃ©es comme arguments de fonctions ou au sein d'instructions conditionnelles.

On distingue 3 types de variables :

- Les variables rÃ©servÃ©es Smarty

- Les variables chargÃ©es depuis des fichiers de configuration (non dÃ©veloppÃ© ici)

- Les variables assignÃ©es depuis Php; nous listerons les variables assignÃ©es par Xoops, comme ou

Variables Smarty Modificateurs de variables

Les modificateurs de variables peuvent Ãªtre appliquÃ©s aux variables, fonctions et chaines de caractÃªres.

Ils sont utilisÃ©s pour des modification du texte : mise en majuscules, minuscules, troncature, suppression d'espace, etc.

IndiquÃ©s pour mÃ©moire, n'Ã©tant en principe pas utilisÃ©s avec Xoops, les traitements de chaines de caractÃªres Ã©tant rÃ©alisÃ©s dans le Php.

Ex : troncature du texte 'sujet' Ã¡ 40 caractÃªres, suivi de ...

Retour au sommaire

Smarty : mÃ©thodes La classe xoopsTpl (fichier class/template.php) est une classe dÃ©rivÃ©e de la classe Smarty.

Toutes les mÃ©thodes de la classe Smarty peuvent donc Ãªtre utilisÃ©es.

Sauf exception (voir display), cette classe n'a pas besoin d'Ãªtre instanciÃ©e, puisque la crÃ©ation 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 rÃ©fÃ©rence
append	Smarty	ajout d'une valeur
append_by_ref	Smarty	ajout d'une valeur par rÃ©fÃ©rence
display	Smarty	affichage du template
xoops_template_clear_module_cache	xoopsTpl	Effacement du cache du module

NB: de nombreuses mÃ©thodes (de la classe Smarty, ou de la classe dÃ©rivÃ©e) ne sont pas dÃ©crites, Ã©tant utilisÃ©es principalement par le core de Xoops.

---

assign

Permet d'assigner une valeur Ã Â un template.

La valeur est soit une chaine, soit un tableau associatif. assign (string | array $tpl_var, mixed $value = null)

$tpl_var	string ou array	nom de la variable de template Ã Â crÃ©er
$value	mixed	valeur Ã Â 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 Ã Â assign, sauf que la donnÃ©e est passÃ©e par rÃ©fÃ©rence (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 Ã Â crÃ©er

$value mixed RÃ©fÃ©rence de la valeur Ã Â lui transmettre

---

append

Permet d'ajouter une valeur Ã Â une variable tableau d'un template.
Si le tableau n'existe pas, il est d'abord crÃ©Ã©.
Dans le template, le tableau peut Ãªtre parcouru avec la fonction foreach ou section .
append (string | array $tpl_var, mixed $value = null)
Exemples (avec fonction section dans le template)

```
// Dans le codeÂ¨Php
for ( $i = 0; $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 Ã Â append, sauf que la donnÃ©e est passÃ©e par rÃ©fÃ©rence (et non par valeur)
Exemple (avec boucle foreach dans le template)

```
// Dans le codeÂ¨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 nÃ©cessaire lorsque l'on utilise l'affichage par le template prinipal (template_main) puisque le footer.php se charge d'exÃ©cuter cette action.

A n'utiliser que dans les cas particuliers d'affichage sans le thÃªme (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 (aprÃªs une m Â Â 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 rÃ©fÃ©rence: 2.06

C. Felix AKA theCat le 22/06/04

Smarty : variables Ces variables smarty, qu'elles soient assignÃ©es par Xoops ou rÃ©servÃ©es pour Smarty sont utilisables dans tous les templates. Variables assignÃ©es - Variables assignÃ©es par Xoops dans le fichier header.php, Ã¡ partir des donnÃ©es de la Bdd.

Ces valeurs sont celles enregistrÃ©es dans la page Admin systÃªme >> PrÃ©fÃ©rences >> MÃ©ta balises et pied de page

Balise mÃ©ta: mots clefs

Balise mÃ©ta: description

Balise mÃ©ta: auteur

Balise mÃ©ta: copyright

Balise mÃ©ta: robots

Pied de page

- Variables assignÃ©es par xoops dans le fichier header.php

Nom du site *

Titre de la page *

Nom du thÃªme *

Url de la feuille de style du thÃªme *

Inclusion de javascript du fichier include/xoops.js *

Affichage banniÃªre *
Affichage du contenu de la page *
0 ou 1 selon que le bloc centre doit Ãªtre affichÃ© *

0 ou 1 selon que le bloc gauche doit Ãªtre affichÃ© *

0 ou 1 selon que le bloc droit doit Ãªtre affichÃ© *

Url demandÃ©e, ayant conduit sur cette page

Slogan du site
Url des images du theme *
1 si administrateur
1 si user
id de l'user, 0 si anonyme
pseudo de l'user, sinon terme choisi pour anonyme

** Variables nÃ©cessaires au template du thÃªme: theme.html*

- Variables assignÃ©es par xoops dans le fichier template.php

Ex. ISO-8859-1
Code de la langue 2 caratÃªres: en, fr, etc.

Url du site
Chemin physique du rÃ©pertoire de xoops
Version de Xoops (ex. Xoops 2.0.5.1)
Url du rÃ©pertoire d'upload

Variables rÃ©servÃ©es >

Permet d'accÃ©der 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 requÃªtes
Pour mÃ©moire.
Permettent d'accÃ©der Ã¡ diffÃ©rentes 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 intÃ©grÃ©es au langage) et les fonctions utilisateurs (rÃ©pertoire class/smarty/plugins).

Ne seront dÃ©crites que les fonctions utilisÃ©es habituellement avec Xoops. De mÃªme tous les attributs des fonctions ne sont pas forcÃ©ment dÃ©crits.

Consultez le site Smarty pour une information plus complÃ¨te.

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 dÃ©termination cyclique de valeurs

---

include

Permet d'inclure un template Ã Â l'intÃ©rieur 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 prÃ©cÃ©dÃ© de db :

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

Include simple

Include avec passage de variable (story)

---

if, elseif, else

Instruction conditionnelle permettant l'exÃ©cution de tests.

Doit obligatoirement se terminer par la balise .

Permet d'utiliser les opÃ©rateurs logiques (forme littÃ©rale ou symbolique).

Attention, les opÃ©rateurs doivent Ãªtre entourÃ©s d'espaces.

OpÃ©rateurs

= =

!= >

>=

!

||

&&

eq

neq

gt

gte

lt

lte

not

or

and

is even

is odd

is not even

is not odd

is even by

is odd by

div by

mod

Exemple

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

Exemple avec opÃ©rateurs

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

Exemple pour affichage sur 3 colonnes

Dans une table, passage Ã Â 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 Ãªtre imbriquÃ©es (pas de limitation de nombre).

foreachelse sera executÃ© si aucune valeur n'est trouvÃ©e dans la variable de l'attribut from

Nom attribut	Requis	Type	Description
from	Oui	string	nom du tableau Ã Â parcourir
item	Oui	string	nom de la variable pour accÃ©der Ã Â l'Ã©lÃ©ment en cours
name	Non	string	nom de la boucle foreach pour accÃ©der Ã Â ses propriÃ©tÃ©s

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. UtilisÃ©e pour les tableaux complexes(multidimensionnels p.ex.).

Doit obligatoirement se terminer par la balise

Les sections peuvent Ãªtre imbriquÃ©es (pas de limitation de nombre).

sectionelse sera executÃ© si aucune valeur n'est trouvÃ©e.

Nom attribut Requis Type Descrip