

TP de Smarty

LAFON Sylvain

12 novembre 2011

Table des matières

I	Introduction	2
II	Une documentation	3
III	Exercices	4
1	Variables	4
1.1	Utilisation d'une variable simple	4
1.2	Utilisation d'un tableau	4
1.3	Utilisation de \$smarty	4
2	Les fonctions	4
2.1	Les fonctions simples	4
2.2	{include}, {extends} et {block}	4
2.2.1	{include}	4
2.2.2	{block}	5
2.2.3	{extends}	5
2.3	Foreach	5
IV	Conclusion	6

Première partie

Introduction

Smarty est un moteur de template, c'est un système permettant de séparer la mise en forme d'une page de ses données. En soi : on va traiter les données dans le php (*.php) et s'occuper de les afficher dans des templates (*.tpl)

Commencez par installer Smarty dans un dossier. Prendre un fichier html basique (appelez le maPage.tpl) :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma page (html 5)</title>
  </head>
  <body>
    Ceci est une page (en html 5, ce n'est pas obligatoire mais bon..) <br/>
    Oh, et voici une variable : {$maVariable}
  </body>
</html>
```

Creez un index.php contenant ceci :

```
<?php
    require("dossier/Smarty.class.php");
    $smarty = new Smarty();

    $smarty->assign("uneVariable", "uneValeur");
    $smarty->display("maPage.tpl");
?>
```

Résultat :

- Si vous ouvrez maPage.tpl, vous obtiendrez une page html dans laquelle est notemment écrit : "Oh, et voici une variable : {\$maVariable}"
- Si vous ouvrez index.php, vous obtiendrez une page html dans laquelle est notemment écrit : "Oh, et voici une variable : uneValeur"

Sachez que pour assign, vous pouvez également passer en paramètre un tableau associatif.

```
// Plus propre que plusieurs appels d'assign
$smarty->assign(Array(
    "unNom" => "uneValeur",
    "unAutreNom" => "uneAutreValeur"
));
```

Et on peut aussi passer un objet en valeur !

Sachant déjà tout cela, vous pouvez commencer par faire de bonnes pages

Deuxième partie

Une documentation

Smarty contient une très bonne documentation en ligne : <http://www.smarty.net/docsv2/fr/>.

Vous pouvez aussi regarder ma présentation (sautez les parties sur la fabrication de fonctions (qui ne doit pas être à jour :/))

Vous devrez, pour réaliser ces exercices, avoir la documentation sous la main.

Vous pouvez dès à présent feuilleter la doc, notamment pour voir :

- Les variables (et les tableaux !)
- La variable `$smarty`
- Les modificateurs (je n'en utilise pas si je me souviens)
- Les fonctions natives
- La boucle `foreach`
- La fonction `{assign}`
- La fonction `{extends}` et `{block}` (voir sur ma présentation)

Troisième partie

Exercices

1 Variables et Modificateurs

1.1 Utilisation d'une variable simple

Exercice 1 : Votre but est d'afficher la somme de deux variables passées par la barre d'adresse (faire la somme dans le php)

Exercice 2 : Votre but est d'afficher en majuscule une chaîne passée par la barre d'adresse

1.2 Utilisation d'un tableau

Exercice 3 : Votre but est d'afficher l'ensemble des jours de la semaine en passant par un tableau indexé (utilisation de foreach non demandée pour le moment, tapez les 7 cases du tableau à la main)

Exercice 4 : Votre but est d'afficher la traduction des 12 mois de l'année en passant par un tableau associatif (utilisation de foreach toujours pas demandée, tapez les 12 cases à la main)

1.3 Utilisation de \$smarty

Exercice 5 : Votre but est d'afficher la date d'aujourd'hui de manière 'lisible' (on se fiche de la langue)

Exercice 6 : Refaire l'exercice 2 sans utiliser Smarty::assign() dans le php

2 Les fonctions

2.1 Les fonctions simples

Exercice 7 : Refaire l'exercice 6 en vérifiant l'existence de la variable (utiliser isset() dans votre test du côté template, bien sûr)

2.2 {include}, {extends} et {block}

Exercice 8 : Nous allons créer un site ayant une sorte de 'modèle'.

2.2.1 {include}

Pour commencer, créer une page index.php et main.tpl : index.php va pour le moment ne faire qu'appeler main.tpl.

Dans main.tpl, commencer l'architecture de votre site en plusieurs templates : utilisez {include} pour séparer votre main.tpl en plusieurs fichiers, par exemple pour le menu

2.2.2 {block}

Ensuite, mettez à l'endroit où devra se trouver le contenu un `{block name=contenu}` Ceci sera le contenu de la page `{/block}`

Vous pouvez en ajouter plusieurs : ce seront les endroits de la page qui pourront être modifiés

2.2.3 {extends}

Créez maintenant un template `page1.tpl` (que votre `index.php` va appeler). La première ligne devra contenir `{extends file="main.tpl"}` .

Ajoutez ensuite : `{block name=contenu}`Hey ! C'est énoorme!!!`{/block}` Vous pouvez observer que l'on retrouve la page "main.tpl" avec seul le contenu qui a changé.

2.3 Foreach

Exercice 9 : Reprendre l'exercice 3 et 4 en utilisant `foreach`

Exercice 10 : Reprendre l'exercice 9 en ajoutant le numéro du jour et le nom en anglais du mois

Exercice 11 : Reprendre l'exercice 10 en remplaçant l'indice du premier jour par : "Premier jour" et le dernier par : "Dernier jour"

Exercice 12 : Utiliser `{assign name="monTableau" value=Array("January", "February", "March", "April", "May", "June", "Juillet", "August", "September", "October", "November", "December")}`

Puis utiliser `foreach` pour les afficher

Quatrième partie

Conclusion

Maintenant que vous avez un peu mieux compris (je l'espere) comment fonctionne Smarty, voici comment je l'ai utilisé dans le projet Bob.

index.php est notre fichier principal, il appelle la classe Bob, qui appelle à son tour d'autres classes telles que Membre, Admin, Categories, etc.. ensuite, index.php va traiter la requete : que doit-on faire ? Selon la requete il va choisir quelles opérations faire et aussi quel template appeller.

Chaque template qu'index.php appelle étant une extension du meme template /templates/modele/main.tpl : cela nous permet d'avoir une meme page type, et de n'avoir que son contenu qui change.