# Smarty

- Un moteur de template
- Séparer la logique métier de la logique de présentation
- Un autre langage
- Gestion d'un cache

http://www.smarty.net/docsv2/fr/

# Caracteristiques

- Il est très rapide.
- Il est efficace, le parser PHP s'occupe du sale travail.
- Pas d'analyse de template coûteuse, une seule compilation.
- Il sait ne recompiler que les fichiers de templates qui ont été modifiés.
- Vous pouvez créer des fonctions utilisateurs et des modificateurs de variables personnalisés, le langage de template est donc extrémement extensible.
- Syntaxe des templates configurable, vous pouvez utiliser {}, {{}}, <!--{}-->, etc. comme délimiteurs tag.
- Les instructions if/elseif/else/endif sont passées au parser PHP, la syntaxe de l'expression {if...} peut être aussi simple ou aussi complexe que vous le désirez.
- Il est possible d'inclure du code PHP directement dans vos templates, bien que celà ne soit pas obligatoire (ni conseillé), vû que le moteur est extensible.
- Support de cache intégré.
- Fonctions de gestion de cache personnalisables.
- Architecture de plugins

# Installation

```
<?php

// charge la bibliothèque Smarty
define('SMARTY_DIR', '…/smarty/libs/');
require_once(SMARTY_DIR . 'Smarty.class.php');

$smarty = new Smarty();

$smarty->template_dir = '/web/www.example.com/smarty/livredor/templates/';
$smarty->compile_dir = '/web/www.example.com/smarty/livredor/templates_c/';
$smarty->config_dir = '/web/www.example.com/smarty/livredor/configs/';
$smarty->cache_dir = '/web/www.example.com/smarty/livredor/cache/';

$smarty->assign('name','Ned');

$smarty->display('index.tpl');
?>
```

# Bases

Note
Toutes les balises Smarty sont entourées de délimiteurs. Par défaut, ils sont { et }, mais ils peuvent être modifiés.

## Commentaires

```
{* ceci est un comentaire *}
```

## Variables

Les variables de template commencent par un signe dollar ($).

```
{$foo}        <-- affiche une variable simple (qui n'estpas un tableau ou un objet)
{$foo[4]}     <-- affiche le 5ème élément d'un tableau indexé
{$foo.bar}    <-- affiche la clé "bar" d'un tableau, identique à $foo['bar'] en PHP
```

## Variables insérées dans des chaînes de caractères

```
{func var="test $foo test"}       <-- comprends $foo
```

## Variables assignées depuis PHP

```
$smarty->assign('foo', $bar);
$smarty->assign('foo', array('contacts' => array('Doug', 'Lucie')));
```

# Variable réservée {$smarty}

### Variables de requête

Les variables de requête comme $_GET, $_POST, $_COOKIE, $_SERVER, $_ENV et $_SESSION peuvent être utilisées : déconseillé.

{$smarty.now} : avoir le timestamp courant

### Modificateurs de variables (mise en forme)

capitalize : première lettre de chaque mot d'une variable en majuscule
cat : concatener des valeurs
count_characters : nombre de caractères dans une variable
count_words : nombre de mots dans une variable
date_format : formatage des dates à partir d'un timestamp ($smarty.now)
default : définir une variable par défaut à une variable
lower : mettre en minuscule la valeur d'une variable
nl2br : changer les \n en <br />
replace : remplacer une portion de la valeur d'une variable
strip_tags : supprime toutes les balises html de la valeur d'une variable
truncate : Tronque une variable à une certaine longueur
upper : mettre en majuscule la valeur d'une variable

```
{$titreArticle|lower|truncate:30}
{$titreArticle|replace:'Garden':'Vineyard'}
```

## Opérations mathématiques

Les opérations mathématiques peuvent être directement appliquées aux variables.

{assign var="foo" value="`$foo+$bar`"}

## Désactiver l'analyse de Smarty

Utiliser les balises {literal}

# Fonctions

Les balises Smarty affichent une variable ou invoquent une fonction.
Elles sont appelées lorsqu'elles sont entourées, ainsi que leurs paramètres, des délimiteurs Smarty.

```
{config_load file='colors.conf'}
{include file='header.tpl'}
{insert file='banner_ads.tpl' title='Smarty est cool !'}
```

**Fonctions natives**

{capture}
{foreach},{foreachelse}
    .index
    .iteration
    .first
    .last
    .show
    .total

```
 {* L'en-tête du block est affiché toutes les 5 lignes *}
<table>
 {foreach from=$items key=myId item=i name=foo}
 {if $smarty.foreach.foo.index % 5 == 0}
 <tr><th>Title</th></tr>
 {/if}
 <tr><td>{$i.label}</td></tr>
 {/foreach}
</table>
```

{if},{elseif},{else}
{include}
{insert}

Mêmes fonctions, mais insert ne met jamais en cache

{literal}
{section},{sectionelse}
    .index
    .index_prev
    .index_next
    .iteration
    .first
    .last
    .rownum
    .loop
    .show
    .total

```
{* $rows[row.index] et $rows[row] sont identiques *}
<table>
  <tr>
    <th>index</th><th>id</th>
    <th>index_prev</th><th>prev_id</th>
    <th>index_next</th><th>next_id</th>
  </tr>
{section name=row loop=$rows}
  <tr>
```

```
   <td>{$smarty.section.row.index}</td><td>{$rows[row]}</td>
   <td>{$smarty.section.row.index_prev}</td><td>{$rows[row.index_prev]}</td>
   <td>{$smarty.section.row.index_next}</td><td>{$rows[row.index_next]}</td>
  </tr>
{/section}
</table>
```

# Fonctions développeurs

### Affichage du template

display() : Affiche le template
is_cached() : indique si un template est déjà en cache

```
<?php
include(SMARTY_DIR.'Smarty.class.php');
$smarty = new Smarty();
$smarty->caching = true;

// ne fait un appel à la base de données que si le fichier
// de cache n'existe pas
if(!$smarty->is_cached('index.tpl')) {

    // quelques données
    $address = '245 N 50th';
    $db_data = array(
                    'Ville' => 'Lincoln',
                    'Pays' => 'Nebraska',
                    'Code postal' = > '68502'
     );

    $smarty->assign('Nom','Fred');
    $smarty->assign('Adresse',$address);
    $smarty->assign($db_data);

}

// affichage
$smarty->display('index.tpl');
?>
```

### Définition de fonctions utilisateur

register_function()

```
<?php
$smarty->register_function('date_now', 'print_current_date');

function print_current_date ($params) {
    extract($params);
    if(empty($format))
        $format="%b %e, %Y";
    echo strftime($format,time());
}

?>
```

register_block()

```
<?php
// Déclaration de la fonction
```

```
function do_translation ($params, $content, &$smarty, &$repeat) {
    if ($content) {
        $lang = $params['lang'];
        // fait de la traduction avec la variable $content
        echo $translation;
    }
}

// Enregistrement avec Smarty
$smarty->register_block('translate', 'do_translation');
?>


{translate lang='br'}Bonjour le monde !{/translate}
```

clear_all_assign()
clear_all_cache()
clear_assign()
clear_cache()

# Cache

Activer le cache

```
<?php
require('Smarty.class.php');
$smarty = new Smarty;

$smarty->caching = 1;

$smarty->display('index.tpl');
?>
```

Réglage individuel de $cache_lifetime

```
<?php
require('Smarty.class.php');
$smarty = new Smarty;

$smarty->caching = 2; // régler la durée de vie individuellement

// règle la durée de vie du cache a 15 minutes pour index.tpl
$smarty->cache_lifetime = 300;
$smarty->display('index.tpl');

// règle la durée de vie du cache à 1 heure pour home.tpl
$smarty->cache_lifetime = 3600;
$smarty->display('home.tpl');

// NOTE : le réglage suivant ne fonctionne pas quand $caching = 2. La durée de vie
// du fichier de cache de home.tpl a déja été réglée a 1 heure et ne respectera
// plus la valeur de $cache_lifetime. Le cache de home.tpl expirera toujours
// dans 1 heure.
$smarty->cache_lifetime = 30; // 30 secondes
$smarty->display('home.tpl');
?>
```

Groupes de fichiers de cache

```
<?php
require('Smarty.class.php');
$smarty = new Smarty;

$smarty->caching = true;

// efface tous les fichiers de cache avec "sports|basketball" comme premiers
// groupes d'identifiants de cache
$smarty->clear_cache(null,'sports|basketball');

// efface tous les fichiers de cache "sports" comme premier groupe d'identifiants.
// Inclue donc "sports|basketball" ou "sports|nimportequoi|nimportequoi|..."
$smarty->clear_cache(null,'sports');

$smarty->display('index.tpl','sports|basketball');
?>
```

# Bonnes pratiques

- Tous les contenus au format texte uniquement (pas d'objets php)
- Balises {literal} pour le javascript
- Ne pas utiliser de code php dans les templates
- Utiliser les sous dossiers pour les dossiers de template et de cache : $use_sub_dirs=true