

Tutoriel templates Joomla! 1.6

Copyright : Cédric KEIFLIN

Licence : GNU/GPL

<http://www.ck-web-creation-alsace.com>

<http://www.joomlack.fr>

SOMMAIRE

Table des matières

1	Cahier des charges	4
2	La création graphique	5
3	Création de la structure de fichiers	7
	index.php	7
	templateDetails.xml	8
	index.html	8
	template_thumbnail.png	8
	template_preview.png	8
	favicon.ico	8
	template.css	8
	image.jpg	8
	Suppléments de fichiers	8
4	Créer le dossier	10
5	Créer un fichier 'index.php'	10
	Début de la page	10
	Entête	10
	Corps du document	11
	Bloc principal de la page	12
	Bloc avec les 3 modules	16
	Bloc de pied de page	17
	Largeur de colonne centrale adaptable	18
	Emplacement de slideshow	19
	3 modules à largeur adaptable	20
6	Création du fichier XML	21
7	Création des styles CSS	23
8	Création des images d'aperçu et de l'icône	30
9	Override des images système	31
10	Packaging et installation	32
11	Annexes	33
	Les commentaires conditionnels	33
	Positions du template 1870	34
	Positions usuelles	35
	Migrer son template Joomla! 1.5 en 1.6	36
	Librairie mootools	36
	Fichier d'installation	36
	Fichiers de langue	36
	Paramètres	38
	Liens vers d'autres tutoriels Joomla! 1.6	39

C'est parti, lançons nous dans la création de template Joomla!. Tout d'abord on va mettre les choses à plat en répondant à quelques questions basiques qu'on peut se poser.

Qu'est-ce qu'un template Joomla! ?

C'est un ensemble de fichiers HTML et CSS qui permettent de mettre en forme le site. C'est le template qui permet d'afficher le graphisme et de gérer la structure du site. En clair, c'est un gabarit.

Est-ce compliqué de faire un template ?

Rien n'est jamais difficile pour celui qui sait. Ce n'est en tout cas pas plus compliqué que faire un site basique, ce n'est qu'un fichier HTML associé à une feuille de style CSS.

Bien sur si vous n'avez aucune notion de ces deux langages, je vous invite à visiter ces sites :

<http://www.siteduzero.com/>

<http://www.w3schools.com/>

Est-ce qu'un template Joomla! 1.5 est compatible avec Joomla! 1.6 ?

Non, mais la mise à jour n'est pas très compliquée. Il s'agit surtout de mettre à jour le fichier XML que nous verrons plus tard.

1 Cahier des charges

Bon, qu'est ce qu'on va bien pouvoir faire ? Je me dis que j'aimerais bien un template avec un style simple et fort, qui reprendrait les éléments suivants :

- un logo
- une bannière
- un top menu
- un emplacement de slideshow ou photo
- un fil d'ariane
- 3 colonnes, gauche, centrale, droite
- 3 modules en bas de page
- un pied de page

Ca reste assez simple comme objectif, l'idée ici étant d'apprendre tranquillement comment faire un template, et pas de faire une usine à gaz. Ce n'est pas pour autant que ce sera simple !

Au contraire allons un peu plus loin :

- la colonne centrale doit s'adapter en largeur si la colonne de droite n'existe pas (aucun module publié)
- les 3 modules du bas doivent s'adapter en largeur en fonction du nombre de modules publiés

Hé hé, vous voyez qu'on peut vite corser les choses !

Avec tout ça on aura une bonne base de travail et un template facilement adaptable à tous les besoins.

Attaquons nous maintenant à la première étape, la création graphique.

2 La création graphique

On a dit tout à l'heure qu'on veut un design simple et fort, c'est à dire avec un contraste élevé. Mais j'aime bien aussi mettre un peu de relief par ci par là, histoire d'apprendre à mettre tout ça en place.

Définissons d'abord les éléments principaux :

- largeur du site : 1000px
- hauteur de la bannière : 100px
- hauteur du topmenu : 50px
- hauteur du slideshow : 220px
- un petit fil d'ariane
- largeur des colonnes gauche et droite : 200px

Et voici le résultat :

TEMPLATE 1870

CRÉATION DU TUTORIEL DE CÉDRIC KEIFLIN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros.

ACCUEIL

JOOMLA OVERVIEW

CONTACT



Accueil

Menu principal

Lien de menu
Description du lien

Titre de la page

Titre de l'article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.

Titre de l'article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.

Login

Identifiant

Mot de passe

Connexion

Titre module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum.

Titre du module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti.

Titre du module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti.

Titre du module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti.

3 Création de la structure de fichiers

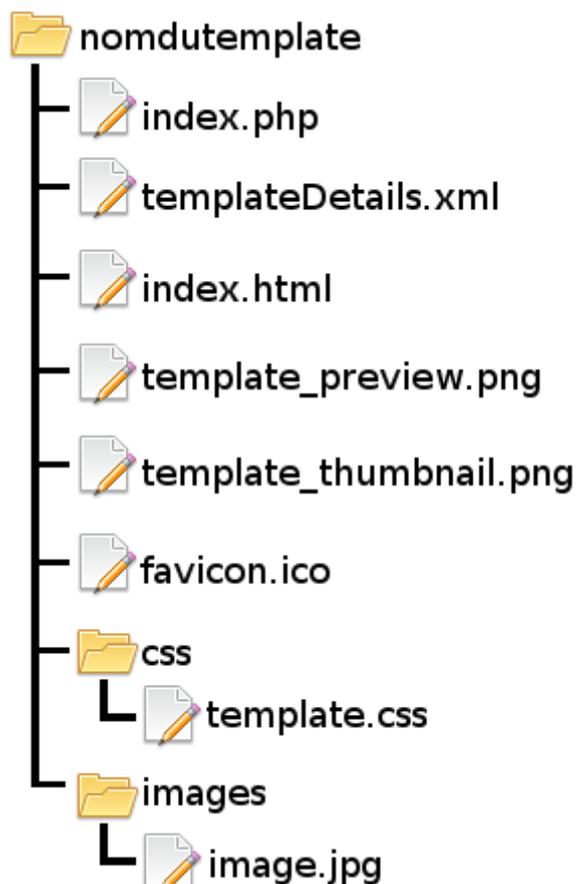
Pour fonctionner contrairement à ce qu'on peut croire le template n'a besoin que de 2 fichiers :

- index.php
- templateDetails.xml

Le fichier XML permet l'installation et l'assignation du template, le fichier PHP renvoie la structure HTML du site.

Pourtant notre template ne va pas se limiter à ces deux fichiers, on va ajouter au moins une feuille de style CSS pour donner de l'allure à la page, des images également, une petite image d'aperçu utilisée dans l'administration, etc... Voyons tout cela en détail :

Liste des fichiers et dossiers d'un template basique :



index.php

Fichier principal qui contient le code HTML de la structure du site, les appels CSS et Javascript, les balises meta, etc...

templateDetails.xml

Fichier d'installation du template, c'est aussi lui qui permet de lister le template dans l'administration de Joomla!, lister les positions disponibles dans les listes déroulantes, et c'est ici que sont définis les paramètres d'administration du template.

index.html

Simple fichier vide qui évite que quelqu'un ne voit le contenu du dossier.

template_thumbnail.png

Petite image de 200x150px pour avoir l'aperçu du template dans l'administration

template_preview.png

Cette image est utilisée dans l'onglet 'templates' de l'administration des templates. Lorsque l'on clique sur la miniature pour l'agrandir, c'est cette image qui s'affiche alors dans une lightbox.

favicon.ico

Icone affichée dans la barre d'adresse du navigateur

template.css

Fichier contenant les styles CSS du template, ce fichier est appelé à partir du fichier 'index.php'

image.jpg

Exemple d'image, elles sont utilisées pour ajouter une touche de design. Elles sont toutes placées dans le dossier 'images' et généralement appelées à partir du fichier CSS.

Suppléments de fichiers

component.php

Ce fichier est utilisé pour le rendu des pages lorsque seul le composant est appelé, par exemple pour l'impression des pages, lorsque l'url comporte '&tmpl=component'.

error.php

Ce fichier est aussi présent dans le template 'system', si on le met à la racine de notre template on

peut l'override, c'est à dire le personnaliser. Il sert essentiellement à afficher certaines pages d'erreur.

dossier language

Ce dossier est intégré au template dans le package d'installation pour ajouter des termes traduits dans plusieurs langues.

Dans le dossier 'images' on peut ajouter un dossier 'system', il contiendra les images système qu'on veut override (remplacer pas les nôtres). Par exemple 'arrow.png' pour le fil d'ariane. Les images d'origine se trouvent dans 'media/system/images'.

4 Créer le dossier

Créez un répertoire nommé '1870' qui est le nom que l'on donne à notre super nouveau template. C'est à l'intérieur de ce répertoire que l'on va stocker tous nos fichiers et sous-dossiers.

5 Créer un fichier 'index.php'

Début de la page

On commence par mettre l'entête de sécurité :

```
<?php  
defined('_JEXEC') or die;  
?>
```

Ceci vérifie que l'accès au fichier provient de Joomla! Et pas d'une personne mal intentionnée.

On ajoute ensuite le Doctype pour définir le type de document html :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

On ouvre le document html en définissant la langue du visiteur et la direction de lecture (ltr ou rtl)

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;  
?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction ?>">
```

Entête

On ouvre l'entete du fichier html

```
<head>
```

Ensuite on appelle la fonction joomla! Qui charge les balises meta, title, les flux rss, la favicon et les inclusions de script et css (si définis dans une extension).

```
<jdoc:include type="head" />
```

On appelle les CSS basiques de joomla! pour appliquer les styles par défaut

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?  
>/templates/system/css/system.css" type="text/css" />  
<link rel="stylesheet" href="<?php echo $this->baseurl ?  
>/templates/system/css/general.css" type="text/css" />
```

On appelle ensuite le fichier CSS du template :

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo  
$this->template ?>/css/template.css" type="text/css" />
```

`$this->baseurl` : renvoie l'adresse de base du site

`$this->template` : renvoie le nom du template

Et on ferme l'entête de la page

```
</head>
```

Le reste des éléments seront placés dans le corps de la page entre les balises <body> et </body>.

Petit récapitulatif des éléments HTML

bloc	<div></div>
paragraphe	<p></p>
lien	
balise en ligne	
image	

Corps du document

On ouvre le corps :

```
<body>
```

Bon c'est maintenant que les choses sérieuses commencent ! Tout va se jouer ici, on va donner un peu de vie à notre document et on va enfin pouvoir meubler tout ça.

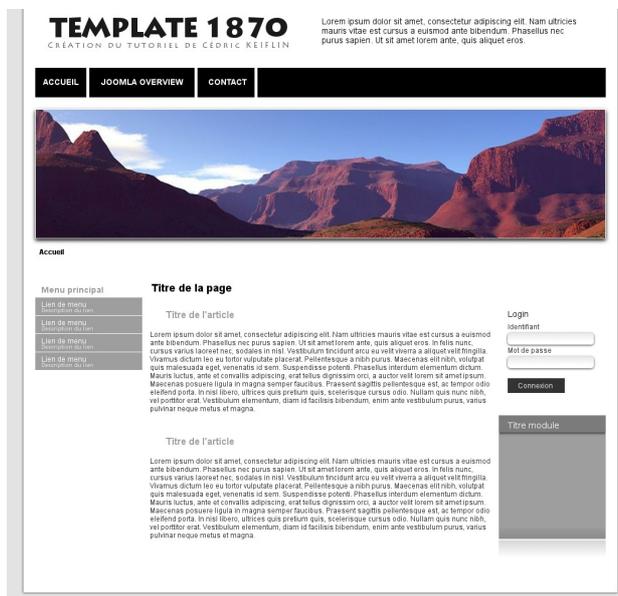
Commençons par réfléchir, que nous faut-il ?

- Un bloc qui contient toute la page avec
 - un bloc principal qui contient la bannière, le topmenu, le slideshow, et les 3 colonnes de contenu
 - un bloc qui contient les 3 modules
 - un bloc qui contient le pied de page

Le bloc qui contient toute la page, appelons-le 'wrapper' et ouvrons-le tout de suite.

```
<div id="wrapper">
```

Bloc principal de la page



Ce bloc commence tout en haut de la page, il a des ombres sur le côté et des ombres en bas. On va créer

- un bloc qui contient les ombres de côté
- un bloc à l'intérieur du premier qui contient le contenu
- un bloc en bas qui contient les ombres du bas

Voici notre structure :

```
<div class="ombre_cote">
  <div id="conteneur">

  </div>
</div>
<div class="ombre_bas"></div>
```

Je comprends pas, on utilise des classes pour les ombres et un ID pour l'autre bloc. Pourquoi on fait ça ?

Alors l'explication est très simple, regardons le graphisme on s'aperçoit que les blocs des 3 modules et du pied de page ont la même configuration d'ombre sur le côté et en bas. Donc pourquoi s'embêter à créer des styles pour des tas de blocs avec ID alors qu'ils sont identiques ? On leur applique une classe et hop !

Par contre pour identifier le bloc conteneur de manière précise dans les CSS, lui on le nomme avec un ID.

Dans le conteneur on va ensuite créer tous les autres blocs et insérer le contenu. Dans le div 'conteneur' on ajoute :

- un bloc de 480px pour le logo (on y insère un lien vers la page d'accueil)
- un bloc de 520px pour les news à droite

- un bloc de topmenu
- un bloc de slideshow
- un bloc de fil d'ariane (breadcrumbs)
- un bloc de colonne gauche de 200px
- un bloc de contenu de 600px
- un bloc de colonne droite de 200px

On crée d'abord la structure HTML et on définira les dimensions dans les styles CSS plus tard.

Voici la structure HTML à insérer dans le `<div id="conteneur">` :

```
<div id="logo"><a href="index.php"></a></div>
<div id="news">
    <jdoc:include type="modules" name="news" />
</div>
<div class="clr"></div>
<div id="topmenu">
    <jdoc:include type="modules" name="position-1" />
</div>
<div class="clr"></div>
<div id="slideshow">
    <jdoc:include type="modules" name="position-12" />
</div>
<div id="breadcrumbs">
    <jdoc:include type="modules" name="position-2" />
</div>
<div id="gauche">
    <jdoc:include type="modules" name="position-7" style="xhtml" />
</div>
<div id="centre">
    <div class="interne">
        <jdoc:include type="message" />
        <jdoc:include type="component" />
    </div>
</div>
<div id="droite">
    <jdoc:include type="modules" name="position-6" style="xhtml" />
</div>
<div class="clr"></div>
```

C'est très simple comme structure. Mais c'est quoi ces trucs 'jdoc:include' ?? C'est pas du HTML ça ?! Ben non, ce sont les appels Joomla! qui permettent de remplir avec le contenu des modules et des articles. Voyons ça plus en détail :

- `<jdoc:include />`** : appelle la fonction Joomla! qui charge les données
- `type="modules" name="news"`** : charge les modules qui sont publiés dans la position 'news'
- `type="component"`** : charge le composant de la page (article, catégorie, etc, tel que défini dans l'administration du site)
- `type="message"`** : permet d'afficher les messages système
- `style="xhtml"`** : utilise le style 'xhtml' pour créer la structure du module

Les styles par défaut sont créés dans le fichier 'templates/system/html/modules.php'
Voici un récapitulatif des types et de leur structure HTML (simplifiée, regardez le fichier pour avoir toutes les infos) :

style="table"

```
<table>
  <tr>
    <th>
      titre du module
    </th>
  </tr>
  <tr>
    <td>
      contenu du module
    </td>
  </tr>
</table>
```

style="horz"

```
<table>
  <tr>
    <td>
      (appel du style "table", ce qui crée 2 tables imbriquées)
    </td>
  </tr>
</table>
```

style="xhtml"

```
<div>
  <h3>titre du module</h3>
  contenu du module
</div>
```

style="rounded"

```
<div>
  <div>
    <div>
      <div>
        <h3>titre du module</h3>
        contenu du module
      </div>
    </div>
  </div>
</div>
```

Petite question subsidiaire : on a donné des noms aléatoires aux positions de modules... mais y a pas des noms à respecter dans Joomla! ? Avec Joomla! 1.5 on avait des trucs comme 'left', 'right', 'user1', 'user2' et là on fait comme on veut ?

Oui et non, on peut faire comme on veut en donnant les noms désirés et en publiant les modules dans ces positions. Pour coller aux données d'exemple fournies avec Joomla! 1.6 j'ai décidé

d'utiliser la convention de nommage (les positions usuelles) décrite en Annexe. Pour les autres positions comme 'news', il faut publier un module pour que le contenu s'affiche. Dans la position 'position-12' (du slideshow), vous devez également publier un module de galerie ou de diaporama, l'image des montagnes dans la maquette ne sert que d'illustration et ne fait pas partie du template.

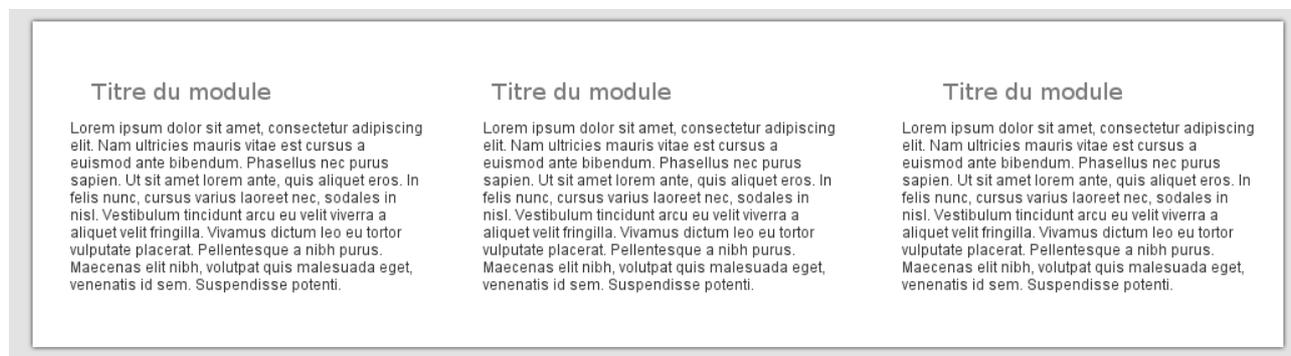
Il semblerait que dans Joomla! 1.6 il n'y ait plus vraiment de standard établi pour les positions comme c'était le cas dans Joomla! 1.5, mais j'ai listé en annexe une corrélation de positions usuelles en fonction des données d'exemple fournies.

L'essentiel est que toutes ces positions soient ensuite listées dans le fichier XML que nous créerons plus tard.

C'est quoi le `<div class="clr"></div>` ? C'est tout simplement un bloc qui va stopper la flotaison (alignement en `float:left;`) des blocs qui sont juste au dessus. C'est une technique très utilisée en CSS.

Pour qu'une position de module soit fonctionnelle, elle doit être créée dans le fichier 'index.php' avec l'appel Jdoc, puis créée dans le fichier XML, et enfin les styles CSS permettront de donner les dimensions et l'aspect esthétique du module. Pour ajouter un module vous pouvez suivre le tutoriel '[Ajouter une position de module](#)' sur Joomlack.

Bloc avec les 3 modules



On retrouve une partie de la structure du bloc principal, sauf qu'il y a une bordure ombrée en haut en plus. A l'intérieur on va ensuite créer 3 blocs côte à côte.

Structure HTML :

```
<div class="ombre_haut"></div>
<div class="ombre_cote">
  <div class="modules_bas">
    <div class="interne">
      <jdoc:include type="modules" name="module-1" style="xhtml" />
    </div>
  </div>
  <div class="modules_bas">
    <div class="interne">
      <jdoc:include type="modules" name="module-2" style="xhtml" />
    </div>
  </div>
  <div class="modules_bas">
    <div class="interne">
      <jdoc:include type="modules" name="module-3" style="xhtml" />
    </div>
  </div>
  <div class="clr"></div>
</div>
<div class="ombre_bas"></div>
```

Tiens v'là qu'on crée des DIV avec une classe 'modules_bas' ? En fait les trois modules auront le même style.

Les div avec la classe 'interne' permettent de donner des marges pour placer le contenu. Pourquoi ne pas les donner aux div avec la classe 'modules_bas' ? Tout simplement parce qu'il vaut mieux éviter d'appliquer une largeur et des marges sur un même élément, ça engendre des problèmes de compatibilité entre navigateurs à cause d'une différence d'interprétation. Cette technique est appelée la techniques des div imbriquées.

Bloc de pied de page

Copyright - Création de Cédric KEIFLIN - 2011 - Template pour Joomla!

Alors là c'est facile ! Il nous suffit de reprendre la même structure qu'avant et de charger un module tout simplement. Allons-y !

Structure HTML :

```
<div class="ombre_haut"></div>
<div class="ombre_cote">
  <div id="footer">
    <jdoc:include type="modules" name="position-14" style="xhtml" />
  </div>
</div>
<div class="ombre_bas"></div>
```

Et surtout il ne faut pas oublier de fermer la page, et le wrapper !

```
</div>
  <jdoc:include type="modules" name="debug" />
</body>
</html>
```

La position 'debug' est utilisée pour le débogage du système, on la met généralement à la fin de la page.

Voilà pour notre structure HTML, ensuite on va pouvoir passer à la construction des styles CSS pour mettre tout ça en forme, mais avant tout on va créer le fichier XML pour pouvoir installer le template.

... j'ai raté quelque chose ? Ah oui, le cahier des charges nous imposait d'avoir des colonnes et des modules adaptables en fonction des modules publiés dans les positions. Bon ben hop, on fait ça avant d'attaquer le XML.

Largeur de colonne centrale adaptable

Pour l'instant on a juste créé 3 colonnes : gauche, centre et droite. Pour ajouter de la flexibilité à notre template on va faire en sorte que si aucun module n'est publié à droite, la colonne centrale s'élargit pour prendre le reste de la place.

Si on a un module au moins publié à droite, la colonne centrale fait 590px.

Si aucun module n'est publié à droite, la colonne centrale fait 790px.

Ce que l'on va devoir faire, c'est tester le chargement de module dans 'droite' (position-6) avec
`$this->countModules('position-6')`

Cette fonction renvoie le nombre de modules publiés dans la position.

On l'utilise pour l'intégrer dans une condition PHP, et assigner une classe CSS qui nous permettra de fixer les tailles :

```
<?php
$centreclass = 'large';
if ($this->countModules('position-6')) :
    $centreclass = 'normal';
endif;
?>
```

On insère le code juste avant la balise `</head>`.

Ensuite on utilise la variable `$centreclass` dans la classe du `<div>` concerné :

```
<div id="centre" class="<?php echo $centreclass; ?>">
    <div class="interne">
        <jdoc:include type="message" />
        <jdoc:include type="component" />
    </div>
</div>
```

Dans les CSS on pourra identifier le bloc de la manière suivante :

```
#centre.normal {
    float: left;
    width: 590px;
}

#centre.large {
    float: left;
    width: 790px;
}
```

Et on ajoute aussi la condition pour l'insertion de la colonne de droite, sinon ça risque de nous poser des soucis d'affichage même si elle est vide.

```
<?php if ($this->countModules('position-6')) : ?>
<div id="droite">
    <jdoc:include type="modules" name="position-6" style="xhtml" />
</div>
<?php endif; ?>
```

Emplacement de slideshow

On va utiliser exactement la même technique pour afficher la position du slideshow uniquement si un module y est publié. Sinon on se retrouve avec un gros bloc vide. Voilà le code qu'on obtient :

```
<?php if ($this->countModules('position-12')) : ?>
    <div id="slideshow">
        <jdoc:include type="modules" name="position-12" />
    </div>
<?php endif; ?>
```

3 modules à largeur adaptable

On va utiliser la même méthode que précédemment pour la colonne centrale. Sauf que maintenant on va tester 3 positions de modules et en fonction du résultat on définit une variable de classe.

```
<?php
$nbmodules = 0;
if ($this->countModules('module-1')) $nbmodules++;
if ($this->countModules('module-2')) $nbmodules++;
if ($this->countModules('module-3')) $nbmodules++;
if ($nbmodules == 1) $modulesclasse = 'entier';
if ($nbmodules == 2) $modulesclasse = 'moitie';
if ($nbmodules == 3) $modulesclasse = 'tiers';
?>
```

On ajoute le code ci-dessus juste avant la balise </head>.

En fonction de la présence de modules dans les positions on incrémente la variable \$nbmodules, et en fonction de sa valeur on définit la classe qui sera appliquée aux blocs de modules.

On insère donc cette classe et on profite aussi de la variable \$nbmodules pour savoir si on doit afficher le bloc complet :

```
<?php if ($nbmodules) : ?>
  <div class="ombre_haut"></div>
  <div class="ombre_cote">
    <?php if ($this->countModules('module-1')) : ?>
      <div class="modules_bas <?php echo $modulesclasse; ?>">
        <jdoc:include type="modules" name="module-1" style="xhtml" />
      </div>
    <?php endif; ?>
    <?php if ($this->countModules('module-2')) : ?>
      <div class="modules_bas <?php echo $modulesclasse; ?>">
        <jdoc:include type="modules" name="module-2" style="xhtml" />
      </div>
    <?php endif; ?>
    <?php if ($this->countModules('module-3')) : ?>
      <div class="modules_bas <?php echo $modulesclasse; ?>">
        <jdoc:include type="modules" name="module-3" style="xhtml" />
      </div>
    <?php endif; ?>
  <div class="clr"></div>
</div>
<div class="ombre_bas"></div>
<?php endif; ?>
```

Et dans les CSS on agira comme cela pour définir les tailles :

```
div.entier {
    width: 100%;
}

div.moitie {
    width: 50%;
}

div.tiers {
    width: 33%;
}
```

6 Création du fichier XML

Le fichier XML a plusieurs rôles :

- installer le template et le lister dans l'administration du site
- créer la liste des positions de modules disponibles pour le template
- lister les informations de l'extension (auteur, licence, version, description, etc...)

Il faut créer un fichier nommé 'templateDetails.xml' dans le répertoire du template. Pensez à l'encoder en utf-8.

Débutons par les premières lignes. On définit le type de fichier (xml) et ensuite on dit à Joomla! qu'on veut installer une extension de type template en version 1.6.

```
<?xml version="1.0" encoding="utf-8"?>
<extension version="1.6" type="template" client="site">
```

Les lignes suivantes n'ont rien d'extraordinaire, ce sont juste les informations concernant l'extension. A vous de juger :

```
<name>1870</name>
<creationDate>Janvier 2011</creationDate>
<author>Cédric KEIFLIN</author>
<authorEmail>ced1870@gmail.com</authorEmail>
<authorUrl>http://www.joomlack.fr</authorUrl>
<copyright>Copyright (C) 2011 Cédric KEIFLIN</copyright>
<license>GNU General Public License version 2 ou supérieure</license>
<version>1.0.0</version>
<description>Template 1870 du tutoriel de Cédric Keiflin sur
http://www.joomlack.fr</description>
```

Ce qu'il faut noter c'est que la balise <name> doit absolument coller avec le nom du répertoire du template.

On continue en listant les fichiers et dossiers à copier pendant l'installation.

```
<files>
  <folder>css</folder>
  <folder>images</folder>
  <filename>index.php</filename>
  <filename>index.html</filename>
  <filename>favicon.ico</filename>
  <filename>templateDetails.xml</filename>
  <filename>template_preview.png</filename>
  <filename>template_thumbnail.png</filename>
</files>
```

<folder> : permet d'installer le dossier et tous les fichiers qu'il contient

<filename> : installe le fichier spécifié

On verra plus tard l'utilité de tous les fichiers que l'on vient de lister.

La prochaine étape est de lister les positions du template, celles qu'on a créées dans le fichier 'index.php'.

```
<positions>
  <position>news</position>
  <position>position-1</position>
  <position>position-12</position>
  <position>position-2</position>
  <position>position-7</position>
  <position>position-6</position>
  <position>module-1</position>
  <position>module-2</position>
  <position>module-3</position>
  <position>position-14</position>
  <position>debug</position>
</positions>
```

On retrouve toutes les positions que l'on a défini au début, elles seront disponibles dans l'administration des modules pour la sélection de la position de publication.

Et on termine le document par une balise fermante :

```
</extension>
```

Voilà, rien de bien compliqué là dedans. Maintenant on pourrait zipper notre template et l'installer dans Joomla!. Le seul problème c'est qu'on vient de lister des fichiers qu'on n'a pas encore créés, du coup l'installateur nous renverra une erreur. Alors continuons.

7 Création des styles CSS

Pour créer nos styles on va devoir mettre en place un dossier nommé 'css' à l'intérieur du template. On y placera toutes les feuilles de style CSS qu'on pourra créer.

Petit rappel pour ceux qui l'ont déjà oublié, dans le fichier 'index.php' on a déjà inséré un appel à une feuille de style 'css/template.css' :

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo $this->template ?>/css/template.css" type="text/css" />
```

Créons donc maintenant le fichier 'template.css' dans ce dossier 'css'. Prenez tout de suite soin d'encoder le fichier en utf-8 pour éviter tout souci, d'ailleurs tous les fichiers doivent être encodés en utf-8.

Allez, on attaque ! C'est parti !

Tout d'abord il faut donner les styles standards pour commencer à mettre à plat et formater la page.

```
html {
    height: 101%;
}

body {
    color : #333;
    background : #e3e3e3;
    font-family: Arial, sans-serif;
    font-size : 12px;
    padding : 0;
    margin : 0;
}

img {
    border : none;
}

a {
    color : #9e9e9e;
}

a:hover {
    text-decoration : none;
}

a:focus {
    outline: none;
}

.clr {
    clear : both;
```

```
}  
li {  
    list-style : none;  
}  
p {  
    padding : 4px 0 4px 0;  
}  
* {  
    margin : 0;  
    padding : 0;  
}  
h1, h2, h3, h4, h5 {  
    font-size : 14px;  
    font-weight : normal;  
}
```

Ca serait trop long de tout détailler, mais pour faire simple :

* : permet d'appliquer à tous les styles (ici marges 0 ce qui initialise tous les navigateurs de la même manière)

body : on définit les styles par défaut dans la page (couleur, police, taille et couleur de fond)

a : on définit la couleur des liens et leur comportement (souligné par défaut et non souligné lors du survol)

Reprenons ensuite notre structure HTML, on a un wrapper qui englobe toute la page, c'est à lui qu'on va donner la largeur fixe et qui va nous permettre de tout centrer. La largeur est de 1000px plus les ombres de 5px de chaque côté, soit 1010px au total.

```
#wrapper {  
    width: 1010px;  
    margin: 0 auto;  
}
```

Ensuite on va donner les styles pour les blocs avec les ombres. Tous les blocs utilisent les mêmes classes, ça va donc être facile.

```
div.ombre_cote {  
    background: url(../images/ombre_cote.png) top center repeat-y;  
    width: 1010px;  
    margin: 0;  
    padding: 0;  
}  
  
div.ombre_bas {  
    background: url(../images/ombre_bas.png) top center no-repeat;  
    height: 10px;  
    width: 1010px;  
    margin: 0;  
    padding: 0;  
}
```

```
div.ombre_haut {
    background: url(../images/ombre_haut.png) bottom center no-repeat;
    height: 10px;
    width: 1010px;
    margin: 0;
    margin-top: 20px;
    padding: 0;
}
```

On s'occupe ensuite des différents conteneurs de la page, ceux qu'on a définis au fil de la conception du fichier 'index.php'. Ici rien de compliqué, on définit les hauteurs et largeurs essentiellement.

```
#logo {
    height: 100px;
    width: 490px;
    float: left;
    background: url(../images/logo.png) top left no-repeat;
}

#logo a {
    width: 450px;
    height: 90px;
    display: block;
}

#news {
    height: 100px;
    width: 500px;
    float: left;
}

#topmenu {
    height: 50px;
}

#conteneur {
    padding : 10px;
}

#slideshow {
    height: 235px;
    clear: both;
    margin: 10px 0 10px 0;
}

#breadcrumbs {
    height: 20px;
}

#footer {
    padding: 10px;
    text-align: center;
}
```

Il nous reste à définir les colonnes principales gauche, centre et droite. Là ça se corse, il faut les faire flotter les unes à côté des autres puis jouer sur les classes du centre pour lui affecter la bonne largeur en fonction de la publication des modules en droite.

```
#gauche {
    width: 200px;
    float: left;
}

#centre.normal{
    float: left;
    width: 590px;
}

#centre.large {
    float: left;
    width: 790px;
}

#centre div.interne {
    padding: 10px;
}

#droite {
    width: 200px;
    float: left;
}
```

Il faut bien veiller à donner le padding sur le div.interne et pas sur celui qui contient l'information 'width'. Cette technique est appelée 'div imbriquée' et permet d'assurer une compatibilité multi-navigateurs. On en a déjà vaguement parlé un peu plus tôt.

Dans le même principe on joue sur les classes spécifiques attribuées dans 'index.php' pour définir les tailles des modules du bas. Rappelons qu'on avait défini 3 différentes classes en fonction de la publication des modules : 'entier', 'moitié', 'tiers'.

```
/** modules bas */
div.entier {
    width: 100%;
}

div.moitié {
    width: 50%;
}

div.tiers {
    width: 33%;
}

div.modules_bas {
    float: left;
}

div.modules_bas div.interne {
    padding: 10px;
}

div.modules_bas ul li {
```

```
        line-height: 20px;
        margin-left: 10px;
    }

div.modules_bas h3 {
    line-height: 35px;
}
```

Dans l'ensemble on fait flotter les 3 modules, on leur applique une largeur, un padding au div.interne et un peu de style pour le titre en h3.

Allez, on continue ! Qu'est ce qu'on n'a pas fait encore ? ... le topmenu ! Alors que nous faut-il, aligner les items horizontalement avec un float, un fond noir et couleur blanche pour les textes, un peu de gris pour le survol. Et hop !

```
/** topmenu **/
#topmenu li {
    background: #000;
    height: 50px;
    padding: 0 10px;
    float: left;
    margin: 0 2px;
    width: auto;
    line-height: 45px;
    text-transform: uppercase;
    font-size: 14px;
    font-weight: bold;
}

#topmenu li a {
    color: #fff;
    padding: 0;
    margin: 0;
    line-height: 45px;
}

#topmenu li a:hover, #topmenu li.current {
    color: #9e9e9e;
}
```

Après le topmenu c'est le tour du menu dans les colonnes. Un petit peu de couleur et de taille essentiellement, rien à signaler de particulier ici.

```
/** menu colonne **/
h3 {
    color: #9e9e9e;
    font-size: 14px;
    font-weight: bold;
    padding-left: 20px;
}

ul.menu li {
    background: #9e9e9e;
    border-bottom: 1px solid #fff;
    line-height: 30px;
}
```

```
margin: 0 5px;
}

ul.menu li ul li{
margin-left: 10px;
background: #666;
}

ul.menu li a {
color: #fff;
text-decoration: none;
line-height: 25px;
padding-left: 10px;
}

ul.menu li a:hover {
color: #000;
}
```

Sur notre maquette on avait aussi imaginé un module de couleur gris foncé avec un petit relief en bas. Allons-y pour celui-ci. Pour l'ombre du bas, il suffit de regarder l'image que j'appelle en background, j'ajoute un padding bas pour assurer la marge avec le texte.

```
/** module gris */
div.moduletable_gris h3 {
height: 36px;
background: url(../images/titre_modulegris.png) left top repeat-x;
line-height: 30px;
color: #fff;
font-weight: normal;
}

div.moduletable_gris {
background: #9e9e9e url(../images/bas_modulegris.png) left bottom repeat-x;
padding-bottom: 35px;
color: #fff;
margin-top: 10px;
}

div.moduletable_gris a {
color: #fff;
padding-left : 10px;
line-height: 20px;
}
```

On style le titre h3 avec la hauteur et l'image de fond répétée sur la largeur. On ajoute l'image ombrée sur le div principal et de la couleur.

'_gris' est le suffixe qui est défini dans l'administration du module dans le paramètre 'Suffixe de classe de module'.

Pour finir on met en forme le module de login avec les images pour les champs input et de la couleur pour le bouton de validation.

```
/** formulaires */
input#modlgn-username, input#modlgn-passwd {
    width: 153px;
    height: 28px;
    line-height: 25px;
    background: url(../images/login_form.png) left top no-repeat;
    border: none;
    margin: 0;
    padding: 0px 0 5px 10px;
}

#login-form fieldset{
    border: none;
}

#login-form li{
    line-height: 20px;
}

input.button {
    background: #333;
    color: #fff;
    border: none;
    padding: 7px;
    cursor: pointer;
}
```

Voilà nos CSS s'arrêtent ici, je ne vais pas plus loin dans la personnalisation, je colle à la maquette. Si vous voulez personnaliser le template plus en profondeur, vous pouvez ajouter vos propres styles dans le fichier 'template.css'. Aidez-vous des outils comme Webdeveloper et Firebug dans Firefox pour identifier les éléments HTML et identifiants CSS sur les pages du site.

Pendant tout le long de la mise en place des styles il faut bien sur veiller à enregistrer les images utilisées dans le répertoire 'images' du template.

8 Création des images d'aperçu et de l'icône

Pour améliorer l'expérience utilisateur / administrateur, Joomla! prévoit des images d'aperçu du template :

- **template_thumbnail.png** : 200x150 px environ c'est la miniature qui apparait dans l'administration des templates
- **template_preview.png** : 640x400 px environ, c'est un aperçu de grande taille que l'on peut voir dans l'administration en cliquant sur la miniature

On va donc faire une capture de notre template et l'enregistrer pour créer ces deux images que l'on stocke directement dans le répertoire du template

Ensuite une chose très simple, on va créer une petite icône qui s'affichera à côté de l'url dans la barre du navigateur. Elle peut faire 16x16 ou 32x32 px, pour convertir votre image ou logo en icône il existe de très bon outils gratuits en ligne comme Converticon : <http://converticon.com/>.

On met l'icône à la racine du template et on la nomme 'favicon.ico'.

9 Override des images système

Un petit truc qui me chagrine dans le template, c'est la couleur orange de la flèche de navigation dans le fil d'ariane. Je n'ai jamais demandé à ce qu'elle soit orange et je ne sais même pas d'où elle vient.

Cette image est ce qu'on appelle une image système au même titre que les icônes d'impression, d'email à côté des articles. Elles se trouvent dans le dossier 'media/system/images'. La bonne nouvelle c'est qu'on peut créer la nôtre dans le template et l'utiliser à la place sans avoir à modifier les images au coeur de Joomla!.

Pour cela il faut créer un dossier 'system' dans le dossier 'images' du template. Ensuite on peut copier l'image original 'arrow.png' et la coller dans notre nouveau dossier. Puis on l'édite, on met un beau gris et on enregistre. Regardons l'effet sur notre page, c'est bien notre image qui a été prise en compte, et notre flèche est maintenant grise ! Chouette cette technique d'override des images.

10 Packaging et installation

On n'en a pas trop parlé pendant la conception du template, mais n'oubliez pas de mettre une page 'index.html' à la racine du template et dans tous les dossiers qu'il contient. Ce fichier permet d'éviter qu'une personne mal intentionnée ne puisse voir le contenu des dossiers et accéder aux fichiers directement.

Voilà, maintenant que notre template est construit il ne nous reste plus qu'à le zipper. Prenez le dossier '1870' et compressez le. Vous pouvez le mettre en format 'zip', ou 'tar.gz'.
Ce package est maintenant un template installable dans Joomla!. Bravo ! Vous êtes venus à bout du tutoriel.

11 Annexes

Les commentaires conditionnels

Dans certains cas on n'arrive pas à faire en sorte qu'Internet Explorer affiche correctement les styles, et cela malgré tous nos efforts. Rassurez-vous il existe un moyen pour contourner ce problème : l'inclusion de fichiers CSS via des commentaires conditionnels.

On va déterminer si la page est visualisée avec IE, on peut choisir d'appeler des feuilles de styles CSS spécifiques.

Exemple :

```
<!--[if IE]>
...je charge mes styles...
<![endif]-->
```

Ce code est à insérer dans l'entête du fichier 'index.php'. C'est un code générique qui cible toutes les versions d'IE.

On peut cibler plus précisément grâce à des tests plus poussés. Par exemple pour cibler uniquement IE6 :

```
<!--[if IE 6]>
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
$this->template ?>/css/ie6.css" type="text/css" />
<![endif]-->
```

Dans l'exemple ci-dessus, on charge une feuille de style 'ie6.css' spécifiquement pour IE6. Voilà et on peut continuer comme cela pour de nombreuses situations.

IE 5.0 pour IE5

IE 5.5 pour IE5.5

IE 6 pour IE6

IE 7 pour IE7

En poussant plus loin on peut cibler une catégorie jusqu'à une certaine version :

```
<!--[if lte IE 6]>
...je charge mes styles...
<![endif]-->
```

lt : inférieure à

lte : inférieure ou égale à

gt : supérieure à

gte : supérieure ou égale à

Cette technique est compatible W3C et permet d'appliquer des styles à n'importe quelle version d'IE.

Il faut veiller à appeler les feuilles de styles pour IE après l'appel à la feuille 'template.css', car en CSS c'est la dernière information lue qui est appliquée.

Positions du template 1870

The image shows a Joomla! 1.6 Template 1870 layout with several content positions marked. The main header features the template name and a navigation menu. Below the header is a large banner image with a central position. The main content area is divided into three columns, each with a title and a text block. At the bottom, there are three module positions, each with a title and a text block. A footer position is also present.

TEMPLATE 1870
CRÉATION DU TUTORIEL DE CÉDRIC KEIFLIN

NEWS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros.

ACCUEIL JOOMLA OVERVIEW CONTACT **POSITION-1**

POSITION-12

Accueil **POSITION-2**

Menu principal
Lien de menu
Lien de menu
Lien de menu
POSITION-7

Titre de la page
Titre de l'article
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.
Titre de l'article
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.

Login
Identifiant
POSITION-6
Connexion
Titre module

Titre du module
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.
MODULE 1

Titre du module
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.
MODULE 2

Titre du module
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ultricies mauris vitae est cursus a euismod ante bibendum. Phasellus nec purus sapien. Ut sit amet lorem ante, quis aliquet eros. In felis nunc, cursus varius laoreet nec, sodales in nisi. Vestibulum tincidunt arcu eu velit viverra a aliquet velit fringilla. Vivamus dictum leo eu tortor vulputate placerat. Pellentesque a nibh purus. Maecenas elit nibh, volutpat quis malesuada eget, venenatis id sem. Suspendisse potenti. Phasellus interdum elementum dictum. Mauris luctus, ante et convallis adipiscing, erat tellus dignissim orci, a auctor velit lorem sit amet ipsum. Maecenas posuere ligula in magna semper faucibus. Praesent sagittis pellentesque est, ac tempor odio eleifend porta. In nisi libero, ultrices quis pretium quis, scelerisque cursus odio. Nullam quis nunc nibh, vel porttitor erat. Vestibulum elementum, diam id facilisis bibendum, enim ante vestibulum purus, varius pulvinar neque metus et magna.
MODULE 3

Copyright - Création **POSITION-14** pour Joomla!

Positions usuelles

Cette liste est basée sur les données d'exemple de Joomla! 1.6, elle n'est pas exhaustive.

Joomla! 1.5	Joomla! 1.6	Position visuelle
left	position-7	Colonne de gauche
droite	position-6	Colonne de droite
user4	position-0	Bouton de recherche
breadcrumbs	position-2	Fil d'ariane
footer	position-14	Pied de page
top	position-12	Au dessus du contenu

Migrer son template Joomla! 1.5 en 1.6

Librairie mootools

Tout au début du fichier 'index.php' on peut trouver l'appel à la librairie javascript Mootools. Dans ce cas il faut le mettre à jour.

Joomla! 1.5

```
JHTML::_('behavior.mootools');
```

Joomla! 1.6

```
JHTML::_('behavior.framework', true);
```

Fichier d'installation

Le plus gros des modifications se fait dans le fichier 'templateDetails.xml'. Il faut remplacer les balises d'installation, les paramètres et les appels aux fichier de langue.

Joomla! 1.5

Balises d'installation

```
<install version="1.5" type="template">  
... contenu du fichier ...  
</install>
```

DOCTYPE

```
<!DOCTYPE install PUBLIC "-//Joomla! 1.5//DTD template 1.0//EN"  
"http://www.joomla.org/xml/dtd/1.5/template-install.dtd">
```

Joomla! 1.6

Balises d'installation

```
<extension version="1.6" type="template" client="site">  
... contenu du fichier ...  
</extension>
```

DOCTYPE

```
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN"  
"http://www.joomla.org/xml/dtd/1.6/template-install.dtd">
```

Fichiers de langue

Les fichiers de langue permettent de traduire des termes insérés dans votre template dans plusieurs langues. Avec Joomla! 1,6 il y a quelques modifications, il faut créer une architecture de dossiers et fichiers spécifiques, ajouter un fichier .sys.ini qui sert à traduire les termes lors de la phase d'installation, et modifier les fichiers .ini en ajoutant des guillemets doubles aux termes traduits.

Dans le fichier 'templateDetails.xml' il faut modifier l'appel aux fichiers.

Joomla! 1.5

```
<languages>
  <language tag="en-GB">en-GB.tpl_MONTEMPLETE.ini</language>
</languages>
```

Joomla! 1.6

```
<languages folder="language">
  <language tag="en-GB">en-GB/en-GB.tpl_MONTEMPLETE.ini</language>
  <language tag="en-GB">en-GB/en-GB.tpl_MONTEMPLETE.sys.ini</language>
</languages>
```

Au niveau de la structure de fichiers, dans Joomla! 1.6 il faut créer des répertoires spécifiques pour les fichiers de langue, par exemple pour l'anglais avec le tag 'en-GB' :

[nomdutable]

|- language

 |- en-GB

 |- en-GB.tpl_NOMDUTEMPLATE.ini

 |- en-GB.tpl_NOMDUTEMPLATE.sys.ini

Dans les fichiers .ini il faut également modifier la manière dont on écrit les traductions, mais aussi l'entête du fichier. Il faut remplacer les /* par des ;

Joomla! 1.5

```
/* *
 * @copyright      Copyright (C) 2011 Cédric KEIFLIN alias ced1870
 * http://www.ck-web-creation-alsace.com
 * http://www.joomlack.fr
 * @license        GNU/GPL
 * */
```

Joomla! 1.6

```
; @copyright      Copyright (C) 2011 Cédric KEIFLIN alias ced1870
; http://www.ck-web-creation-alsace.com
; http://www.joomlack.fr
; @license        GNU/GPL
; Double quotes in the values have to be formatted as "_QQ_"
```

Ensuite pour traduire les termes il faut les mettre entre guillemets double ".

Si on veut traduire des termes en utilisant des guillemets double, par exemple pour insérer du code html, il faut remplacer les " par "_QQ_".

Joomla! 1.5

```
TERME_A_TRADUIRE = <p style ="background:#efefef;">Mon texte traduit ici</p>
```

Joomla! 1.6

```
TERME_A_TRADUIRE = "<p style ="_QQ_"background:#efefef;"_QQ_">Mon texte traduit  
ici</p>"
```

Paramètres

Un autre changement qui intervient dans le fichier XML se trouve au niveau des paramètres. Les balises changent et certains nouveaux paramètres voient le jour.

Retrouvez la liste des paramètres sur le wiki :

http://docs.joomla.org/Template_parameters#Standard_parameter_types_in_detail

Joomla! 1.5

```
<params>
  <param name=... paramètres spécifiques voir liste... />
  ...
</params>
```

Joomla! 1.6

```
<config>
  <fields name="params">
    <fieldset name="advanced">
      <field name=... paramètres spécifiques voir liste... />
    ...
  </fieldset>
</fields>
</config>
```

Attention il y a un paramètre qui a changé, c'est celui qui permet de générer la liste des menus :

Joomla! 1.5

```
<param name="menutype" type="mos_menu" default="mainmenu" label=""
description="" />
```

Joomla! 1.6

```
<field name="menutype" type="menu" default="mainmenu" label="" description="" />
```

Si votre template contient un dossier nommé 'html' c'est qu'il utilise des techniques de template override. Dans ce cas je vous conseille vivement de supprimer ce dossier pour votre première mise en ligne et de mettre à jour les fichiers qu'il contient en accord avec la nouvelle version de Joomla! 1.6. Il faut récupérer les fichiers 1.6 de tous les composants qui sont overrideés dans ce dossier et les modifier ensuite.

Liens vers d'autres tutoriels Joomla! 1.6

Tutoriel pour la création de catégories dans Joomla!1.6

<http://tutoriels-joomla16.joomlack.fr...joomla-16.html>

Tutoriel pour la création de sous-menu

<http://tutoriels-joomla16.joomlack.fr...ous-menus.html>

Voir les positions de son template

<http://tutoriels-joomla16.joomlack.fr...-template.html>

Convertir un template 1.5 en 1.6

<http://tutoriels-joomla16.joomlack.fr...a15-en-16.html>

Différence entre les templates 1.5 et 1.6 (anglais)

<http://www.compassdesigns.net/joomla...d-16-templates>

Créer une nouvelle position de module dans son template

<http://tutoriels-joomla16.joomlack.fr...de-module.html>

Nouvelle technique de Layout override dans Joomla! 1.6 (anglais)

http://docs.joomla.org/Layout_Overrides_in_Joomla_1.6

Nouvelle classe JFactory pour dialoguer avec la base de données (anglais)

<http://www.theartofjoomla.com/home/9...joomla-16.html>

Migrer son plugin vers Joomla! 1.6 (anglais)

<http://www.theartofjoomla.com/home/4...joomla-16.html>

Migrer son extension vers Joomla!1.6 - Tableau comparatif pour les tables et événements (anglais)

<http://www.theartofjoomla.com/conver...xtensions.html>