

Création de templates

Catégorie : Fiches techniques

Publié par [Fooups](#) le 29/09/2005

Remarque

Il faut bien faire la distinction entre :

- les scripts php qui traitent les données (saisie, insertion/extraction dans la bdd, calculs, etc)
- les templates qui permettent d'afficher ces données (ordre, position, affichage conditionnel, etc)
- les thèmes qui permettent de modifier la présentation (couleurs de police, de fond, etc)

L'utilisation des templates permet de séparer le traitement des données de leur affichage ; on peut ainsi modifier la présentation des données, sans toucher au code, en modifiant simplement le template.

Bien qu'il soit souvent plus facile de créer une variable unique en php (par ex. nom du poster + date + heure), il est préférable de transmettre les variables séparément et de les mettre en forme dans le template : cela permettra une plus grande souplesse pour des modifications ultérieures, par vous-même ou vos futurs utilisateurs.

Coté php

Pour afficher les données d'un script via un template, 2 opérations sont nécessaires dans le code:

- indiquer au script le nom du template à utiliser .
- transmettre au template les données à afficher .

Structure du fichier :

```
include("header.php");
```

```
// si vous avez des fonctions :
```

```
include("include/functions.php");
```

```
$xoopsOption['template_main'] = "mymodule_template_1.html"
```

```
// nom du template
```

```

include XOOPS_ROOT_PATH.'/header.php'

// si vous utilisez la navigation par page :

include(XOOPS_ROOT_PATH."/class/pagenav.php");
// votre code à€

// exemple assignation de valeurs uniques

xoopsTpl->assign('lang_total_post', _MD_TOTAL_POST);

xoopsTpl->assign("counter", $total);
// votre code à€

// exemple d'assignation de tableau

$array_msg = array();

while ($sqlfetch=xoopsDB->fetchArray($result)) {

    $ array _msg['title'] = $myts->sanitizeForDisplay($sqlfetch["title"],
    0, 0, 0);

    $ array _msg['msg'] = $myts->sanitizeForDisplay($sqlfetch["message"],0,1,1);

    xoopsTpl->append('posts', $array_msg);

}
// sans oublier de finir par :

include(XOOPS_ROOT_PATH."/footer.php");

```

CotÃ© html

* Affichage de valeurs uniques

:

* Affichage des valeurs du tableau par une boucle

```
//  
facultatif: pour alterner la couleur de fond des lignes
```

```
">
```

```
">
```

Idem, mais avec inclusion d'un second template

Dans ce second template, les valeurs seront affichées par

* Autre méthode d'affichage des valeurs d'un tableau
(pour tableau multidimensionnel notamment)

```
.....
```

Remarques

Dans la boucle qui génère le tableau, on peut avoir des instructions

conditionnelles genre

```
If ($variable1>variable2) {
```

```
$ array _msg['title'] = $montexte
```

```
}
```

Si la condition n'est pas remplie, l'Ã©lÃ©ment

`$ array _msg['title']` n'existera pas

dans le tableau transmis au template et gÃ©nÃ©rera (en mode debug php) un :

Notice [PHP]: Undefined index: title in file

Bien que cela n'empÃªche pas votre module de fonctionner, il est prÃ©fÃ©rable de dÃ©clarer dans ce cas l'Ã©lÃ©ment du tableau

- soit systÃ©matiquement avant le test avec une chaine nulle

- soit d'utiliser une boucle `if â€¦ else`

```
if ($variable1>variable2)
```

```
{
```

```
$ array _msg['title'] = $montexte;
```

```
} else {
```

```
$ array _msg['title'] = â€œ";
```

```
}
```

// ou sa version condensÃ©e

```
($variable1>variable2) ? $ array _msg['title'] = $montexte : $ array  
_msg['title'] = â€œâ€œ;
```

Convention

Pour toutes les variables contenant un libellÃ© (par opposition aux variables dynamiques) utilisez le prÃ©fixe **\$lang_**

(ex. \$lang_hits, \$lang_title, etc.)