Les pseudos−frames
Cet tutorial a pour but de répondre aux questions fréquentes posées sur le forum à savoir : "Comment faire un site avec des url du type domain.com?page=news". Ce type d'architecture s'appelle les pseudo−frames, car elles remplacent plus qu'avantageusement l'archaïque système de frame HTML. A savoir que les url de ce type ne servent pas qu'à la création de pseudo−frames, mais c'est le rôle qu'elles auront ici.

L'inclusion de fichier

Quelques explications

Voyons tout d'abord ce que sont les pseudo−frames. Il s'agit en fait de découper votre site en plusieurs parties, chacune dans un fichier différent, puis les regrouper dans une page. Pourquoi donc ? Et bien afin de faciliter les mises à jour, tout comme c'était le cas avec les frames originales. En effet imaginons que vous avez un menu sur votre site. Ce menu est affiché sur toutes les pages de votre site. Si vous deviez le recopier dans chaque page, cela voudrait dire qu'à chaque mise à jour du menu, il faudrait aller le corriger dans chaque page.

Avec les pseudo−frames, ce problème ne se pose plus.Vous créez un fichier menu.htm, et vous appeler ce fichier dans chaque page. Lorsque vous mettez à jour le fichier menu.htm, votre menu est mis à jour sur chaque page automatiquement. Cela n'explique pas encore la raison du "page=news" dans l'url, mais cela va nous servir de point de départ.

Un peu de pratique

Nous allons donc créer notre menu dans le fichier menu.htm. Il s'agit ici simplement d'un peu de HTML, notez qu'il n'y a pas de balise ***<head>, <html>*** ou ***<body>***, car comme nous l'avons vu, cette page sera insérée dans une autre qui, elle, contiendra ces balises. On n'y met donc que le strict nécessaire. Notez également que ce code sera modifé par la suite, pour le moment on se contente d'un menu simple, sans pseudo−frames.

Le code de la page menu.htm

```
<ul>
  <li>
    <a href="accueil.php"
title="L'accueil">Accueil</a>
  </li>
  <li>
    <a href="news.php" title="Les News
">News</a>
  </li>
</ul>
```

Puis nous allons créer deux autres pages pour notre site, la page accueil.php et la page news.php (qui sont les pages sur lesquelles pointe notre menu).

Le code de la page accueil.php

```
<div class="colonneGauche">
<?php
  include('menu.htm');  // Nous
appelons notre menu
?>
</div>
<div class="colonneDroite">
  <p>Ici nous aurions le contenu de
```

```
notre page d'accueil.</p>
</div>
```

Le code de la page news.php

```
<div class="colonneGauche" >
<?php
  include('menu.htm');  // Nous
appelons notre menu
?>
</div>
<div class="colonneDroite">
  <p>Ici nous aurions les news de notre
site.</p>
</div>
```

Explication

Comme vous pouvez le voir, le processus est le même sur ces deux pages. Nous utilisons la structure de langage ***include()*** qui permet d'inclure un ficher et d'exécuter son contenu. Là en l'occurence il s'agit simplement de HTML, donc il n'a rien à exécuter mais cela aurait très bien pu être du PHP comme nous allons le voir plus loin. Je vous invite à vous documenter sur cette structure de langage dans la doc PHP afin d'en connaître le fonctionnement.

Ce que nous avons fait aura pour effet d'afficher le menu sur ces deux pages. Les avantages sont que le rendu final se fera sur une page unique, et, comme nous l'avons vu plus haut, en cas de modification du menu, vous ne modifiez que menu.htm. Comme c'est le même fichier qui est inclus à chaque fois, les changements se répercuteront automatiquement sur tout le site (du moins partout où vous avez inclus ce fichier).

Tout ceci est bien beau mais cela ne nous explique pas la raison du paramètre d'url "page=news", vu plus haut...

Les pseudos−frames

Encore quelques explications

Nous avons donc vu qu'il était possible et très facile d'inclure un fichier dans un autre fichier en PHP. Cela va nous ouvrir la voie des pseudos−frames. En effet imaginons que nous soyons capable de prévoir quelle page doit être incluse et quand, nous pourrions découper tout notre site en plusieurs morceaux tout gardant une page centrale qui se chargerait d'appeler le bon morceau au bon moment.

C'est ici que va nous servir le paramètre d'url. En effet c'est lui va nous permettre de savoir que quand l'utilisateur clique sur tel lien, c'est tel page qu'il faut appeler et afficher. Comment ? C'est ce que nous allons voir...

La preuve par l'exemple

Veuillez noter que le code qui va suivre contient un énorme trou de sécurité. Il n'est qu'à titre d'exemple et d'apprentissage par pallier. Le code sécurisé suit, merci donc de lire le tutorial jusqu'au bout.

Nous allons commencer par créer les morceaux de pages. Ici nous avons besoin de l'accueil et des news. Le menu lui aussi va être légèrement modifié, vous comprendrez plus tard pourquoi.

Le code de la page accueil.php

```
<p>Mettez ici simplement le contenu que vous voulez voir afficher en accueil.</p>
```

Le code de la page news.php

```
<p>Mettez ici simplement le contenu que vous voulez voir afficher dans les news. Il y a de forte chance que cela soit du code PHP qui vous génère ces news. Cela ne pose aucun problème, procédez comme à votre habitude: requête, boucle et echo.</p>
```

Le code de la page menu.htm

```
<ul>
  <li>
    <a href="index.php?page=accueil" title="L'accueil">Accueil</a>
  </li>
  <li>
    <a href="index.php?page=news" title="Les News ">News</a>
  </li>
</ul>
```

Nous allons maintenant créer une page index.php qui va être le noyau de notre site. Elle va se charger d'inclure tout les éléments découpés afin de construire une page complète.

Le code de la page index.php

```
<div class="entete" >
<?php
  include('entete.htm');  // Nous appelons l'entete du site
?>
</div>
<div class="colonneGauche" >
<?php
  include('menu.htm');   // Nous appelons notre menu
?>
</div>
<div class="colonneDroite">
<?php
  include($_GET['page'].'.php');   // Nous appelons le contenu central de la page
?>
</div>
```

```
<div class="pied">
<?php
  include('pied.htm');   // Nous
appelons le pied de page
?>
</div>
```

Comme vous l'avez sûrement compris, tout le problème consiste à savoir quand il faut inclure les news ou l'accueil. Notez les modifications de la page menu : les liens pointent dorénavant tous sur la page index.php, mais il y a plus. Ces derniers ont été complétés avec un paramètre ***page*** auquel nous affectons une valeur en fonction de ce que nous voulons afficher.

Sur la page ***index.php*** nous récupérons ce paramètre ainsi que la valeur qu'il contient à l'aide de $_GET['page']. Cette dernière est ce l'on appelle une variable globale. Tout comme $_POST permet de récupérer des valeurs passées par formulaires (voir le tutoriel 5), $_GET permet de récupérer des valeurs passées dans l'URL.

Conclusion

Donc lorsque le visiteur clique sur le lien news, le paramètre ***page*** avec la valeur news est transmis à la page ***index.php***. Sur cette dernière on récupère cette valeur. Et donc l'include qui dans le code est `include($_GET['page'].'.php')` va devenir dynamiquement `include('news.php')` et c'est donc bien la page contenant le news qui va être incluse dans la page noyau. De même pour l'accueil. Cette fois le paramètre ***page*** sera égal à accueil et donc l'include sera modifié en fonction.

Problème de sécurité

Le problème

Si vous avez bien lu, vous savez que le paramètre page est affiché en clair dans la barre d'adresse du navigateur. Il est donc facilement modifiable par l'utilisateur. Ce dernier peut dès lors inclure n'importe quoi dans votre page index.php ! Il convient donc de tester ce que vaut ce paramètre avant d'inclure la page demandée et surtout vérifier que cela correspond bien à ce que vous permettez.

Nous allons pour cela créer un tableau contenant un listing de toutes les pages que vous autorisez dans cet include.

Le code

Le code de la page index.php

```
<div class="entete" >
<?php
  include('entete.htm');  // Nous appelons l'entete du site
?>
</div>
<div class="colonneGauche" >
<?php
  include('menu.htm');   // Nous appelons notre menu
?>
</div>
<div class="colonneDroite">
<?php

  // On définit le tableau contenant les pages autorisées
```

```php
  // -------------------------------------------------------
  $pageOK = array('news' => 'news.php',
                  'accueil' => 'accueil.php');

  // On teste que le paramètre d'url existe et qu'il est bien
autorisé
  //
-------------------------------------------------------------------
  if ( (isset($_GET['page'])) && (isset($pageOK[$_GET['page']])) )
{
    include($pageOK[$_GET['page']]);   // Nous appelons le contenu
central de la page
  } else {
    include('accueil.php');   // Page par défaut quant elle
n'existe pas dans le tableau
  }

?>
</div> <div class="pied">
<?php
  include('pied.htm');   // Nous appelons le pied de page
?>
</div>
```

Explication

Comme vous le voyez nous avons créé un tableau associatif. Ce tableau est construit ainsi : son index représente les paramètres d'inclusion autorisés dans l'url, les valeurs représentent le chemin réel du fichier. En l'occurence les fichiers sont dans le même répertoire que la page ***index.php***, mais vous auriez pu vouloir inclure un fichier dans le répertoire ***colGauche*** contenu dans le répertoire courant. Dans ce cas vous auriez dû rajouter une ligne au tableau de cette forme : `'page' => 'colGauche/fichier.php'`.

Ensuite nous testons l'existence de cette ligne du tableau avec comme index du tableau, le paramètre passé en url. Si la ligne existe c'est que le paramètre est autorisé, on inclu donc le fichier en utilisant la valeur de la ligne concernée du tableau.

Note : vous l'avez sûrement remarqué, j'ai spécifié des classes dans mes divs. Elles sont là à titre purement indicatif, vous pouvez les modifier à loisir.

Conclusion

Voilà c'est terminé. Si vous avez bien suivi vous pouvez maintenant créer des architectures de sites plus ou moins complexes qui vous faciliteront les mises à jour et le graphisme tout en restant parfaitement accessibles (si vous respectez les standards au niveau du code HTML bien sûr). N'oubliez jamais le problème de sécurité et le moyen d'y remédier sans quoi vous pourriez rencontrer quelques surprises.

Sachez également que vous pouvez utiliser plus d'un paramètre dans vos url, il suffit pour cela de les séparer par des "&" (domain.com?paramOne=1&paramTwo=2).

Voilà cette fois c'est tout, bonne chance.