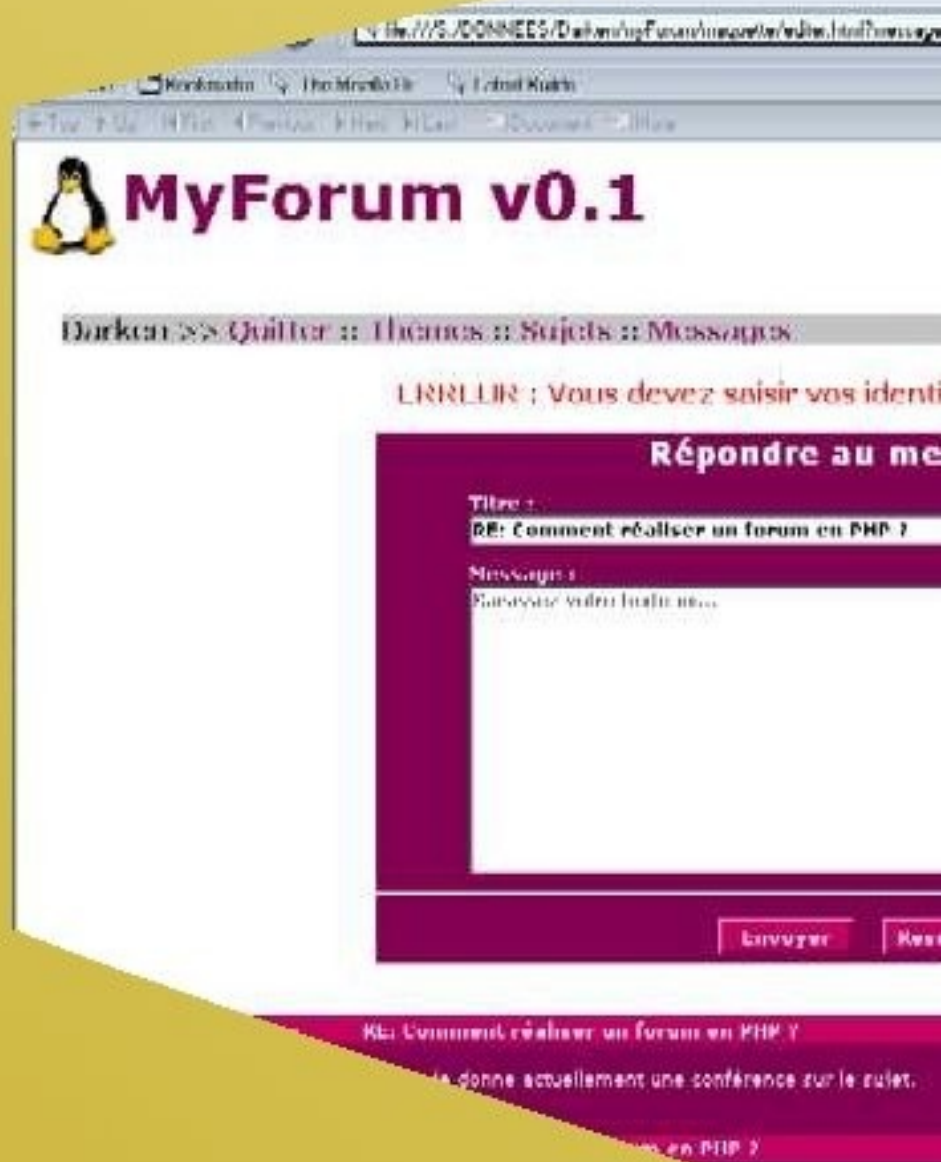


# Créer une Application WEB avec PHP / MySQL

par Philippe Bousquet <Darken33@free.fr>



# **Créer une application WEB avec PHP / MySQL**

*Par Philippe Bousquet <Darken33@free.fr>*

Ce livre électronique est une réédition d'un article de préparation d'une conférence que j'avais effectué au lycée Casler à Talence en 2003/2004 dans le cadre des « conférences de l'ABUL ».

*Ce Livre est distribué selon les termes de la [GNU Free Documentation License](#).  
Copyright (c) 2003-2008 Philippe BOUSQUET*

## Table des matières

I. Introduction	5
II. Le modèle de données	6
III. Création de la base de données	11
IV. Maquettage de l'application	21
V. Le XHTML	23
VI. Les pages du forum en XHTML	36
VII. La mise en forme par CSS	41
VIII. Programmation de l'application	61
IX. Le langage PHP	62
X. L'application my Forum	81
XI. Références	104

# I.Introduction

Au travers de ce livre, nous allons voir les différentes étapes pour créer une application WEB, pour cela nous nous proposons de créer un forum de discussion basé sur les technologies suivantes :

- **MySQL** - Pour le stockage et la gestion des données
- **XHTML** et **CSS** - Pour le rendu de nos pages HTML
- **PHP** - Pour la programmation des parties dynamiques de notre application

## II. Le modèle de données

Avant de se lancer dans la programmation, il convient de bien déterminer l'ensemble des données ainsi que les liaisons qu'elles ont entre elles. En effet la programmation ne représente qu'à peu près 20% dans la création d'une application, le reste étant partagé entre l'Etude de ce que l'on veut réaliser et les tests finaux.

### Définition des fonctionnalités de l'application

Nous devons dans un premier temps donc bien définir ce que notre application est censée faire. Un forum de discussion est en fait un mécanisme permettant à des utilisateurs de discuter sur des sujets divers à travers des messages écrits.

Les utilisateurs doivent pouvoir créer de nouveaux sujets de discussions, ils doivent pouvoir lire les sujets des autres utilisateurs et bien entendu pouvoir y répondre.

### Organisation générale des données

Une fois les fonctionnalités figées, nous devons organiser nos données afin de pouvoir constituer notre base de données. C'est celle ci qui sera le cœur de notre application, donc nous devons y consacrer une grande attention pour que celle ci ne soit pas mal conçue, ce qui rendrait notre application moins fonctionnelle.

Dans l'étape précédente nous avons vu apparaître trois notions distinctes :

- **L'utilisateur** - c'est une personne qui émettra et lira des messages.
- **Le sujet** - il s'agit du point de départ de la discussion, il regroupe les messages le concernant.
- **Le message** - c'est l'objet qui permet aux utilisateurs de s'exprimer sur un sujet donné.

Donc nous voyons ici que notre application devra au moins gérer ces trois entités. Donc nous devons créer trois tables pour constituer notre base de données.

Cependant pour que notre application soit moins rébarbative, dans le sens ou au bout d'un moment les utilisateurs risquent de se perdre dans la jungle des sujets, nous allons intégrer une nouvelle notion : La catégorie.

En effet nous allons décider de regrouper nos sujets par catégories, ce aura pour effet que nos utilisateur, au lieu de lancer une discussion dans le vide, nous allons leur proposer des catégories dans lesquelles ils pourront créer leurs sujets de discussions.

Par exemple :

**Utilisateur** : darken

**Catégorie** : technique

**Sujet** : envoyer des mails avec OO

**Message** : "Il y a bien dans outils / options /programmes auxiliaires la possibilité de choisir un client de messagerie par défaut. Mais je n'arrive pas à savoir comment envoyer des mails....."

Nous aurons donc quatre Tables dans notre base de données :

- **utilisateurs** : qui rassemblera les infos sur les utilisateurs
- **categorie** : qui liste l'ensemble des catégories du forum
- **sujets** : contiendra les sujets des différentes discussions
- **messages** : contiendra tous les messages émis par les utilisateurs

## **Organisation générale des données**

Maintenant que nous avons déterminé quelles tables constitueront notre base de données, il nous faut définir les informations (champs) que contiendront chacune des tables.

### ***Table Utilisateurs***

Cette table contient les utilisateurs ayant accès au forum, chaque utilisateur devra dans un premier temps, passer par une procédure d'inscription afin de pouvoir utiliser le forum. Nous aurons donc besoins des informations suivantes :

- **code utilisateur** : il s'agit de l'identifiant de l'utilisateur, celui ci doit être unique
- **mot de passe** : nécessaire au login de l'utilisateur
- **email** : cette information est utile pour renvoyer le mot de passe à l'utilisateur qui l'aurait oublié
- **nom** et **prenom** : ces deux informations ne sont pas vraiment utiles mais il est toujours intéressant de les posséder

## ***Table Categories***

Cette table contient les thèmes de discussions du forum, chaque sujet de discussion sera englobé dans l'un des thèmes présent en table. Cette table ne sera disponible qu'en lecture seule, c'est à dire que c'est le DBA qui ajoutera les thèmes éventuels dans la table, il n'y aura pas de mise à jour par programme.

Peu d'informations sont nécessaires pour identifier un thème :

- **un numero identifiant** : il s'agit d'un numéro auto-incrémenté qui permet d'identifier un thème, ce numéro est unique
- **un nom** : Il s'agit du nom du thème, par exemple : Devel (pour un thème sur le développement)
- **une description** : cette information permet d'expliquer un tant soit peu le thème, par exemple : "Discussions sur le développement de logiciel libres"

## ***Table Sujets***

Cette table contient l'ensemble des sujets du forum, Elle doit posséder les informations suivantes :

- **un numero d'identifiant** : il s'agit d'un numéro auto-incrémenté qui permet d'identifier un sujet, ce numéro est unique
- **un titre** : titre qui décrit le sujet
- **date creation** : la date de création du sujet
- **heure de creation** : l'heure de création du sujet

Cependant un sujet est créé par un utilisateur, il est intéressant de connaître cette information, donc nous ajoutons :

- **code utilisateur** : c'est le code de l'utilisateur ayant créé le sujet, il doit exister dans la table users.

De plus nous avons précisé qu'un sujet était en rapport avec une catégorie donc cette information doit être également dans la table :

- **identifiant catégorie** : s'est l'identifiant de la catégorie à laquelle le sujet est rattaché, la catégorie doit exister dans la table thèmes.



## **Table Messages**

Cette table contient l'ensemble des messages du forum, Elle doit posséder les informations suivantes :

- **un numero d'identifiant** : il s'agit d'un numéro auto-incrémenté qui permet d'identifier un message, ce numéro est unique
- **un titre** : titre qui décrit le message
- **un texte** : il s'agit du corps du message
- **date creation** : la date de création du message
- **heure de creation** : l'heure de création du message

Cependant un message est créé par un utilisateur, il est nécessaire de connaître cette information, donc nous ajoutons :

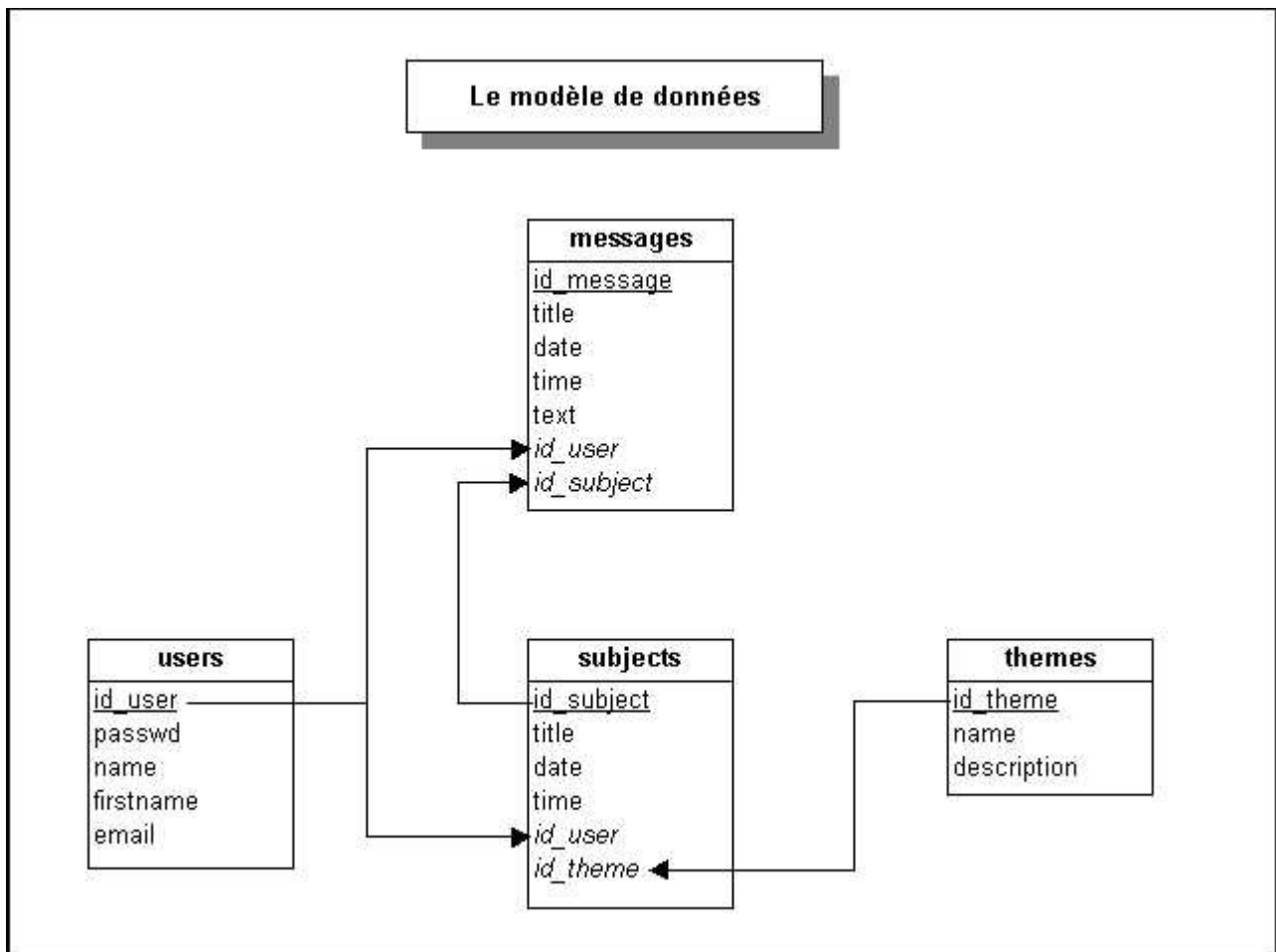
- **code utilisateur** : c'est le code de l'utilisateur ayant créé le message, il doit exister dans la table users.

De plus nous avons précisé qu'un message était en rapport avec un sujet donc cette information doit être également dans la table :

- **identifiant sujet** : s'est l'identifiant du sujet auquel le message est rattaché, le sujet doit exister dans la table sujets.

## La structure complète de la base de données

Nous avons donc défini, de façon théorique la structure de notre base de données, voici le schéma général :



## III. Création de la base de données

Maintenant que nous avons définis les différentes données que notre application devra gérer, nous allons créer physiquement notre base de données.

Pour cela nous allons utiliser le SGBD MySQL, qui est l'un des plus connus dans le monde du WEB et du logiciel libre.

Afin de faciliter, la manipulation de MySQL, nous allons utiliser PhpMyAdmin.

Cet outil, écrit en PHP, permet de gérer ses bases de données sans avoir besoin d'une grande connaissance de MySQL. De plus du fait qu'il est écrit en PHP (donc sur votre serveur HTTP) vous pouvez surtout gérer vos bases à distance.

### **Préparation et configuration**

Il n'est pas recommandé d'utiliser le "compte" administrateur MySQL (root) pour manipuler une base de données donc nous allons dans le client MySQL, créer un utilisateur qui permettra de gérer notre base de données (création de table, ajout d'enregistrements, ...).

Premièrement, il faut se connecter à MySQL en tant que root via le client MySQL, dans le but de créer notre base et d'ajouter l'utilisateur (qui permettra de gérer la base sous PhpMyAdmin) :

```
[darken@localhost darken]$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.0.15

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Il faut maintenant créer notre base de données :

```
mysql> create database myforum;
Query OK, 1 row affected (0.00 sec)
```

Nous allons maintenant créer l'utilisateur qui sera le DBA de notre base de données, nous allons donc nous positionner sur la base myforum, puis donner à l'utilisateur wwwadmin toutes les autorisations sur cette base :

```
mysql> grant all privileges on myforum.* to wwwadmin@localhost;
Query OK, 0 rows affected (0.82 sec)

mysql> grant all privileges on myforum.* to wwwadmin;
Query OK, 0 rows affected (0.82 sec)
```

Nous pouvons donc maintenant quitter le client MySQL :

```
mysql> quit
Bye
[darken@localhost darken]$
```

Donnons maintenant un mot de passe pour l'utilisateur wwwadmin :

```
[darken@localhost darken]$ mysqladmin -p -uwwwadmin password 'azerty'
Enter password:
[darken@localhost darken]$
```

Vérifions maintenant que la configuration de PhpMyAdmin, afin que lorsque l'on se connecte à MySQL via PhpMyAdmin il demande un nom d'utilisateur et un mot de passe. dans le fichier /var/www/html/phpMyAdmin/config.inc.php

Remplacer la ligne :

```
$cfg['Servers'][$i]['auth_type'] = 'config';
```

Par :

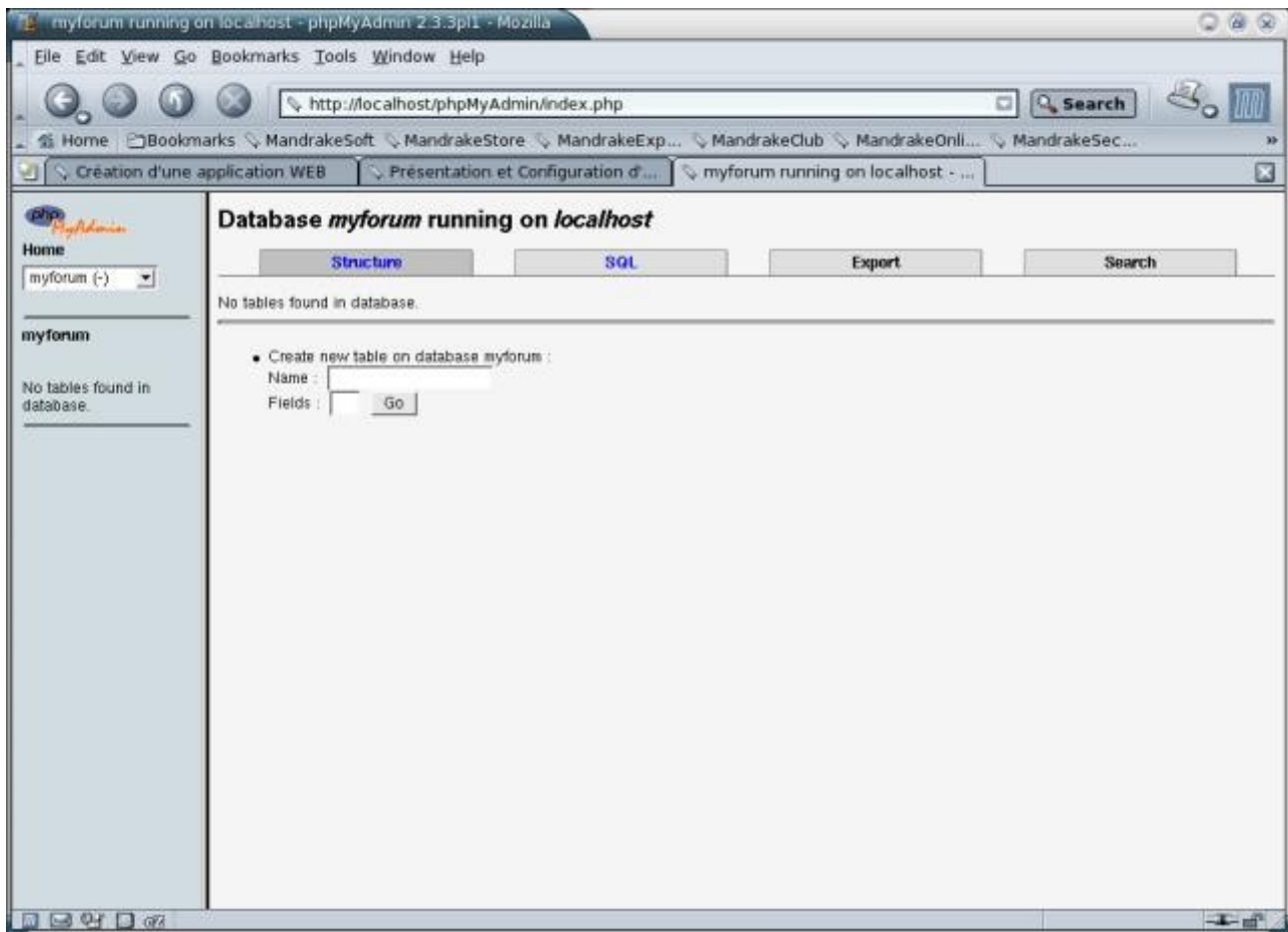
```
$cfg['Servers'][$i]['auth_type'] = 'http';
```

On peut maintenant se connecter à PhpMyAdmin via votre navigateur préféré : <http://localhost/phpMyAdmin> :



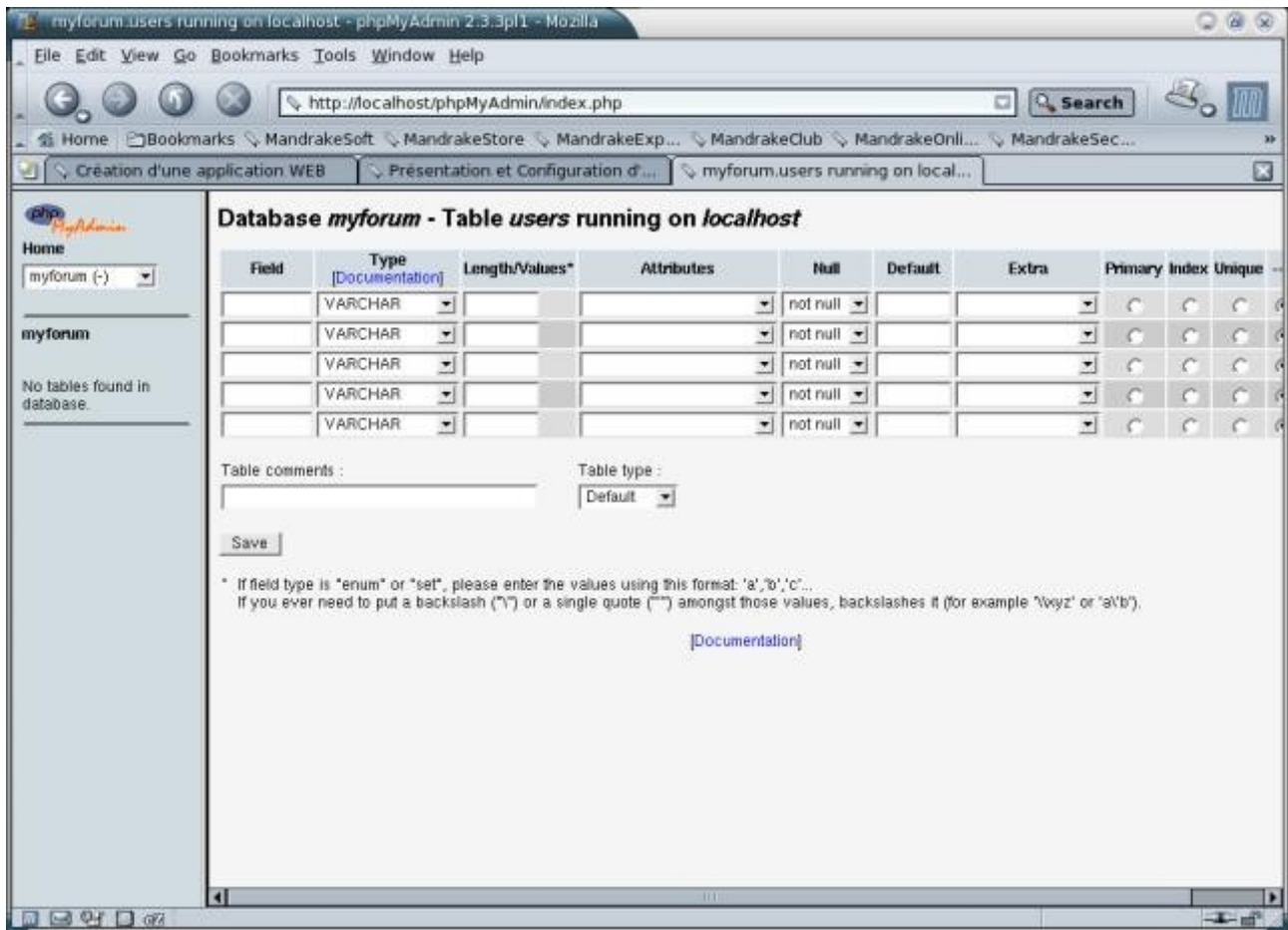
## Création de nos tables

Maintenant que l'on est connecté à MySQL sous PhpMyAdmin, nous allons donc créer nos tables dans notre base MyForum : pour cela sélectionnons la base "myforum".



Nous arrivons sur un écran qui nous propose de créer une nouvelle table, en effet pour l'instant c'est la seule action que nous pouvons effectuer car notre base est vide.

Nous allons donc créer la table "users" qui contiendra les informations sur les utilisateurs. Indiquons le nom de la table ("users") et le nombre de champs qu'elle contiendra (5).



Nous voyons donc 5 lignes qui correspondent à nos cinq champs que nous voulons décrire, pour cela il faut fournir plusieurs informations :

- **Field** : il s'agit du nom des champs
- **Type** : il s'agit du type de données que se champ contiendra (caractères, nombres, dates...)
- **La longueur du champ** : Par exemple un code postal contient 5 caractères.
- **Des attributs particuliers** (non signés, binaire, ...) je conseille de ne les utiliser que si vous en avez vraiment besoin.
- **Null** : permet de définir si le champ peu ne pas être renseigné.
- **Default** : qui permet de définir la valeur par défaut du champ.
- **Extra** : permet de définir des nombres auto incrémentés.
- **Primary** : détermine si le champ est la clé de la table.

- **Index** : permet de déterminer si le champ fait parti de l'index secondaire.
- **Unique** : Permet d'indiquer que la valeur du champ doit être unique dans la table.

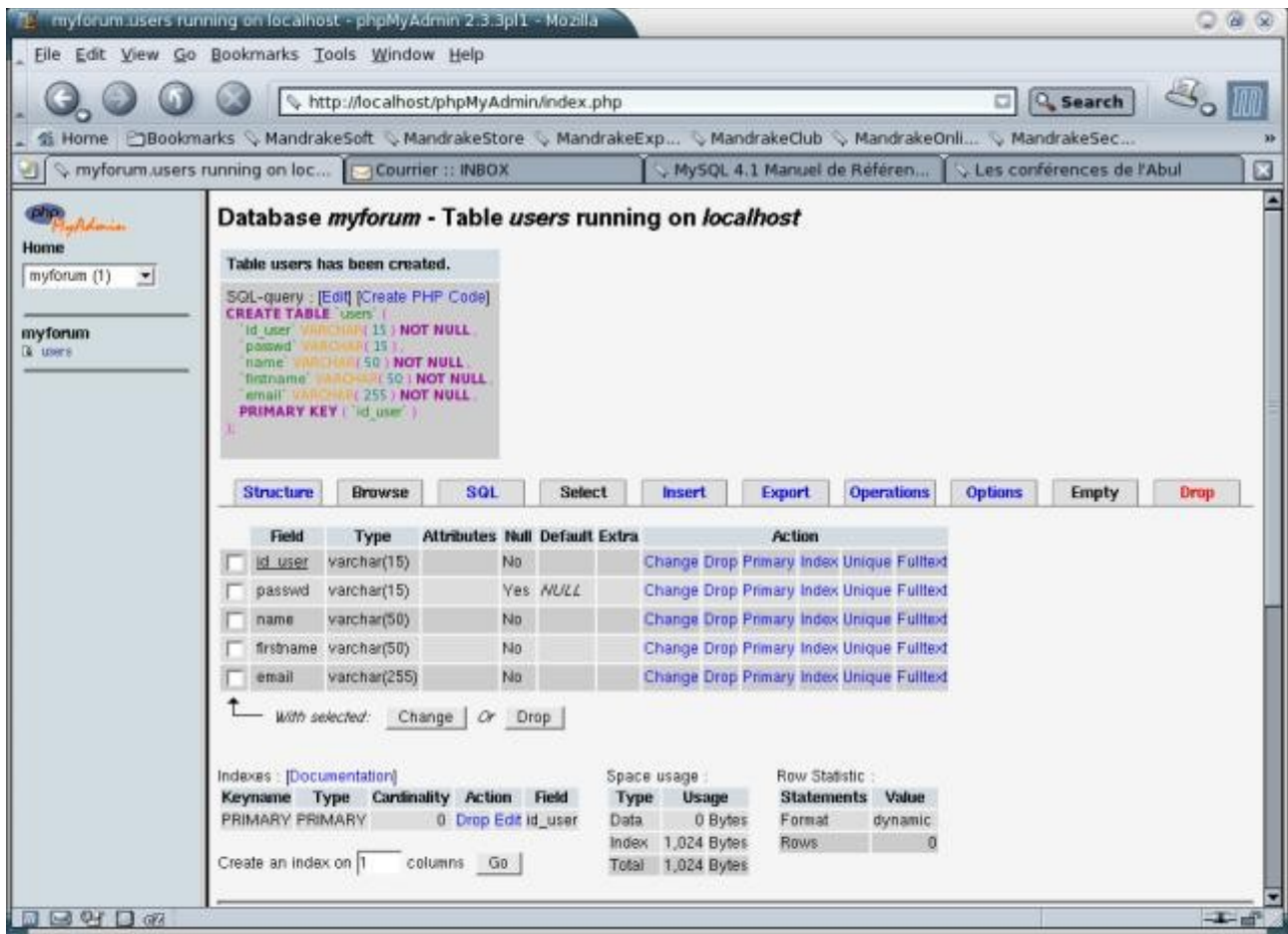
MySQL permet de gérer une multitude de types différents : chaînes de caractères, entiers, flottants, dates, ...

Voici un tableau présentant les types les plus courants et qui nous serviront pour nos tables :

Type	Description
VARCHAR	Chaîne de caractères à longueur variable (1-255)
TINYINT	Entier entre -127 et 128
TEXT	Texte de 1 à 65535 caractères
DATE	Date au format AAAA-MM-JJ
INT	Entier de -2147483648 à 2147483647
BIGINT	Entier de -9223372036854775808 à -9223372036854775807
FLOAT	Permet de stocker des nombres décimaux
DOUBLE	Permet de stocker des nombres décimaux
DECIMAL	Permet de stocker des nombres décimaux
TIME	Permet de stocker une heure au format HH:MM:SS
BLOB	Permet de stocker des objets binaires de grande taille

Nous allons donc créer notre table avec la structure suivante :

- **id\_user** : varchar(15)
- **passwd** : varchar(15)
- **name** : varchar(50)
- **firstname** : varchar(50)
- **email** : varchar(255)



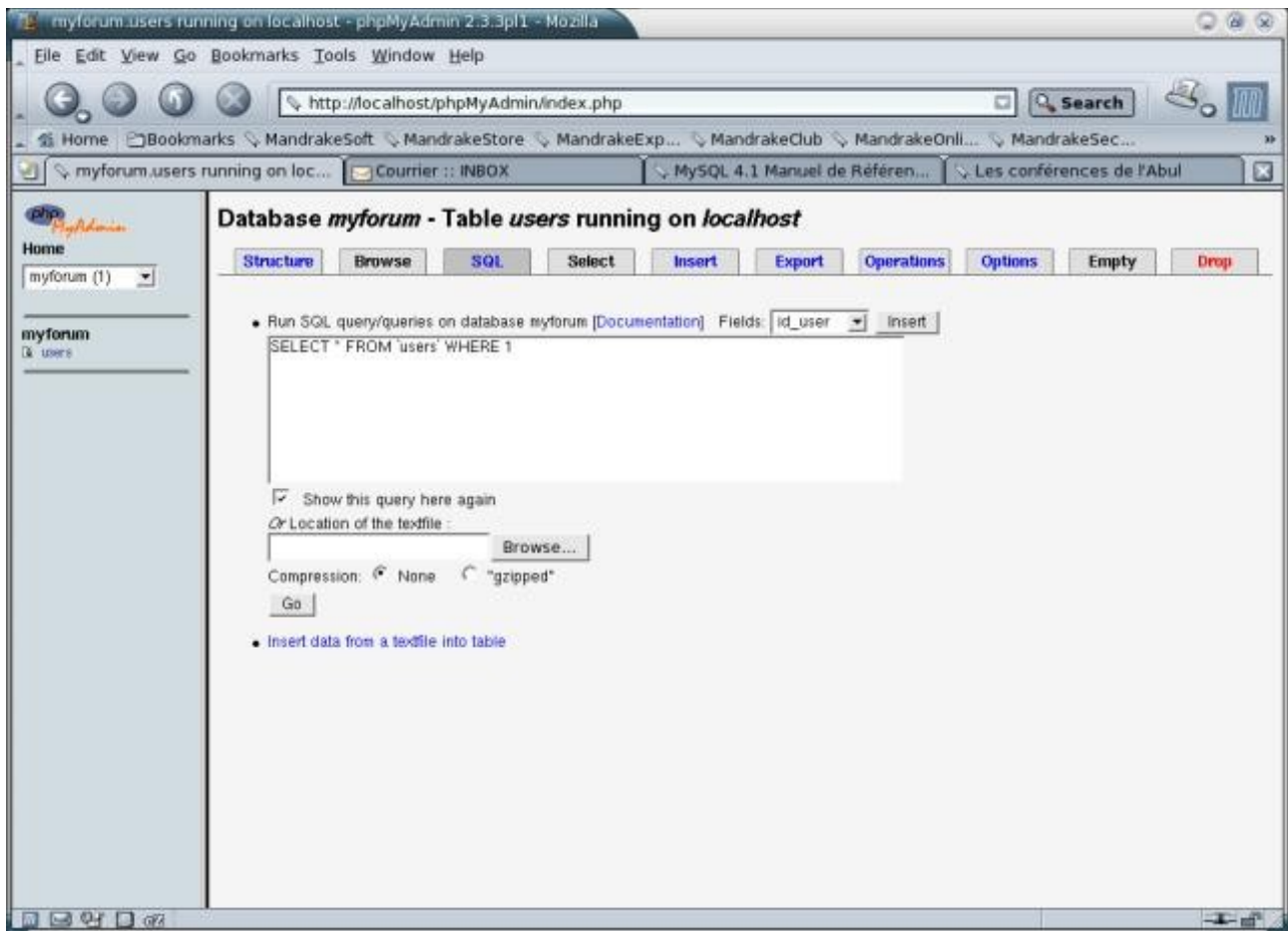
Lorsque l'on soumet le formulaire, on obtient en réponse le message "Table users has been created" qui indique que l'action s'est bien déroulée.

On obtient également la requête SQL qui vient d'être exécutée, ceci est une bonne chose lorsque l'on ne connaît pas le SQL et que l'on désire l'apprendre.



## Exécution de requêtes

Nous venons de créer notre table grâce à l'interface fournit par phpMyAdmin, on peut également lancer nos requêtes directement grâce à l'onglet SQL.

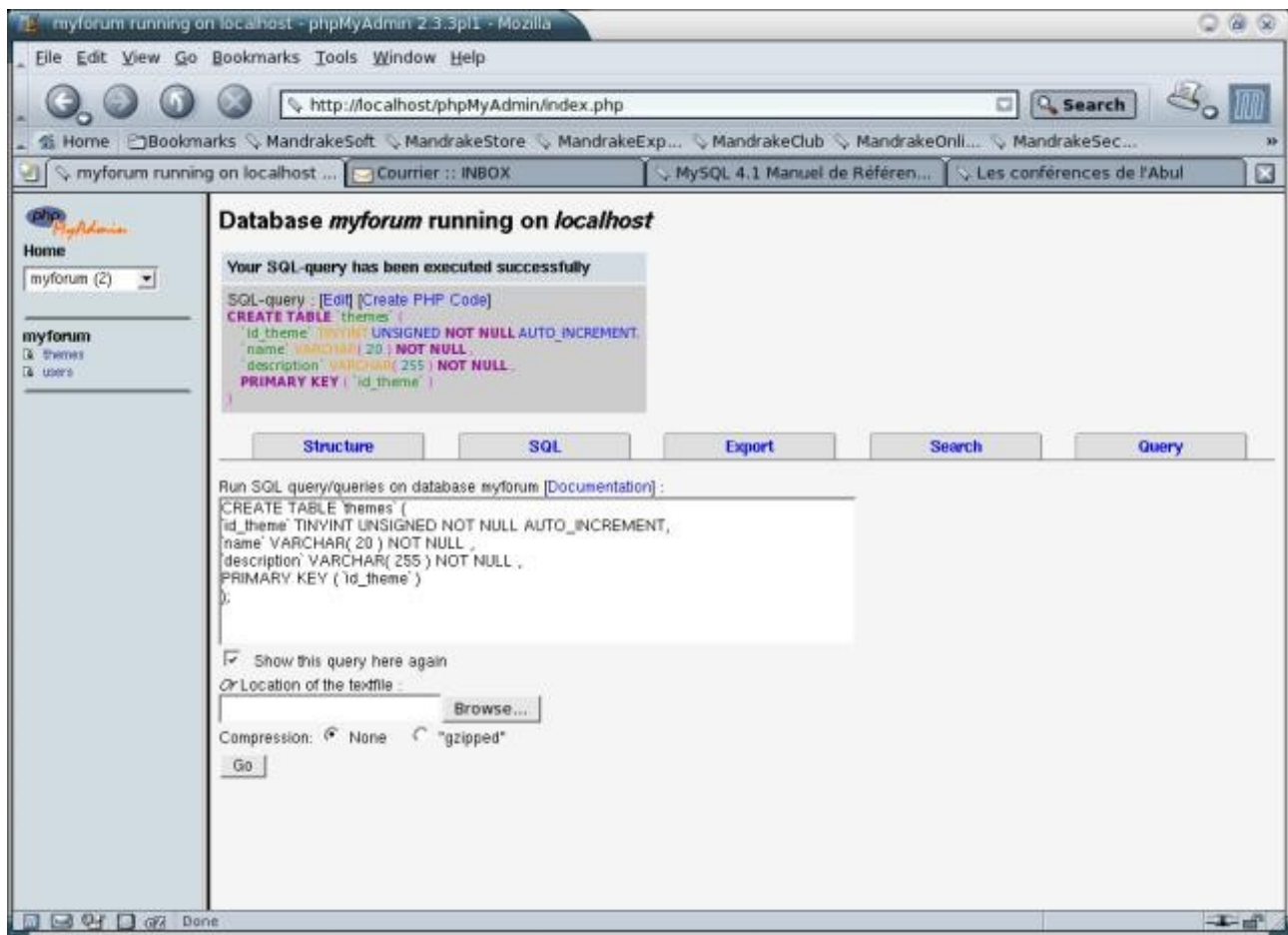


En effet nous allons créer la table thème de cette façon. Cette table est de la structure suivante :

- `id\_theme` qui est un nombre auto incrémenté
- `name` qui est une chaine de caractère
- `description` qui est également une chaine de caractère.

Nous allons donc utiliser la requête suivante :

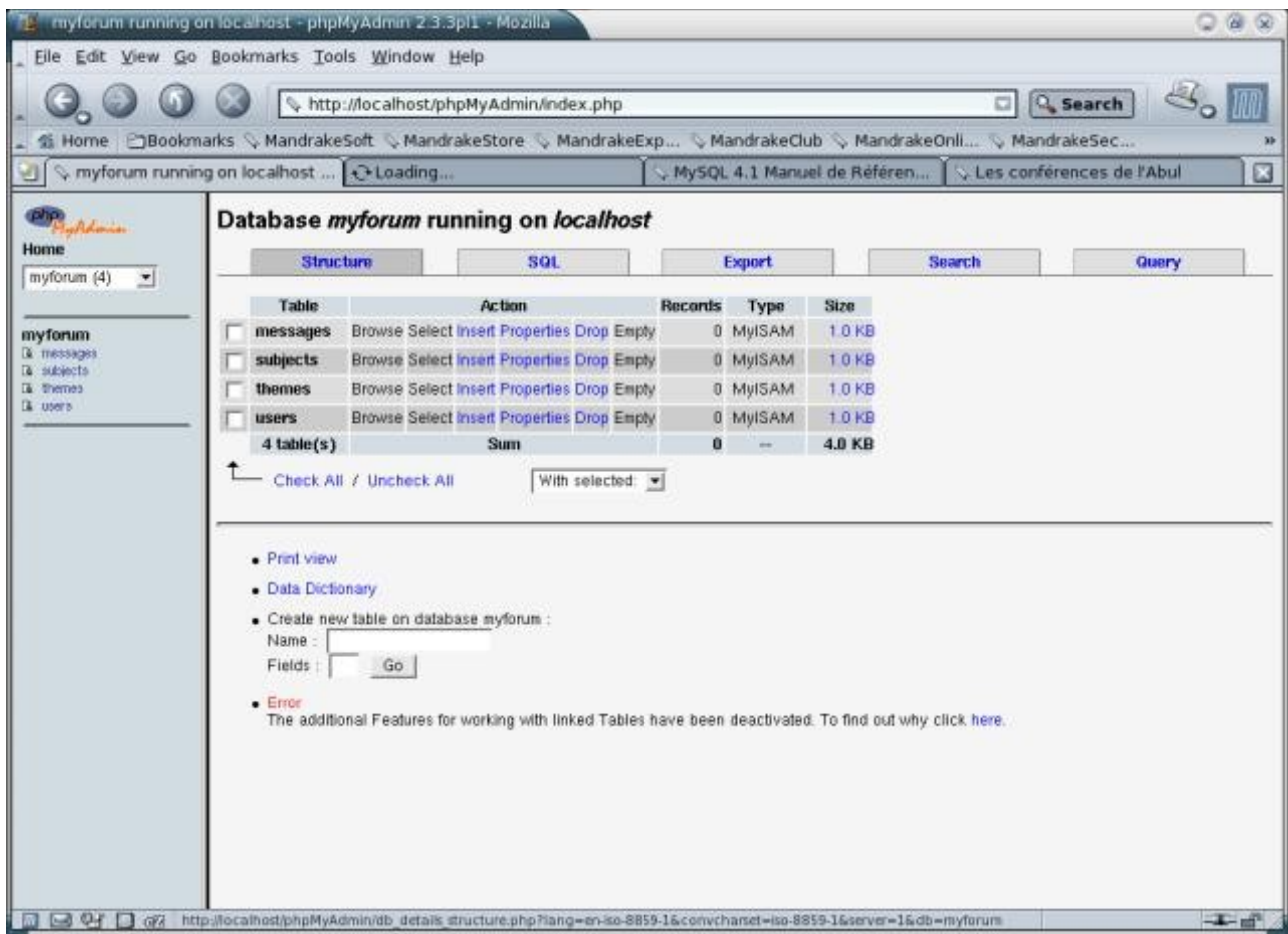
```
CREATE TABLE themes (  
id_theme TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,  
name VARCHAR(20) NOT NULL,  
description VARCHAR(255) NOT NULL,  
PRIMARY KEY (id_theme)  
);
```



Il faut donc maintenant créer les autres tables afin d'obtenir la structure complète de notre base de données :

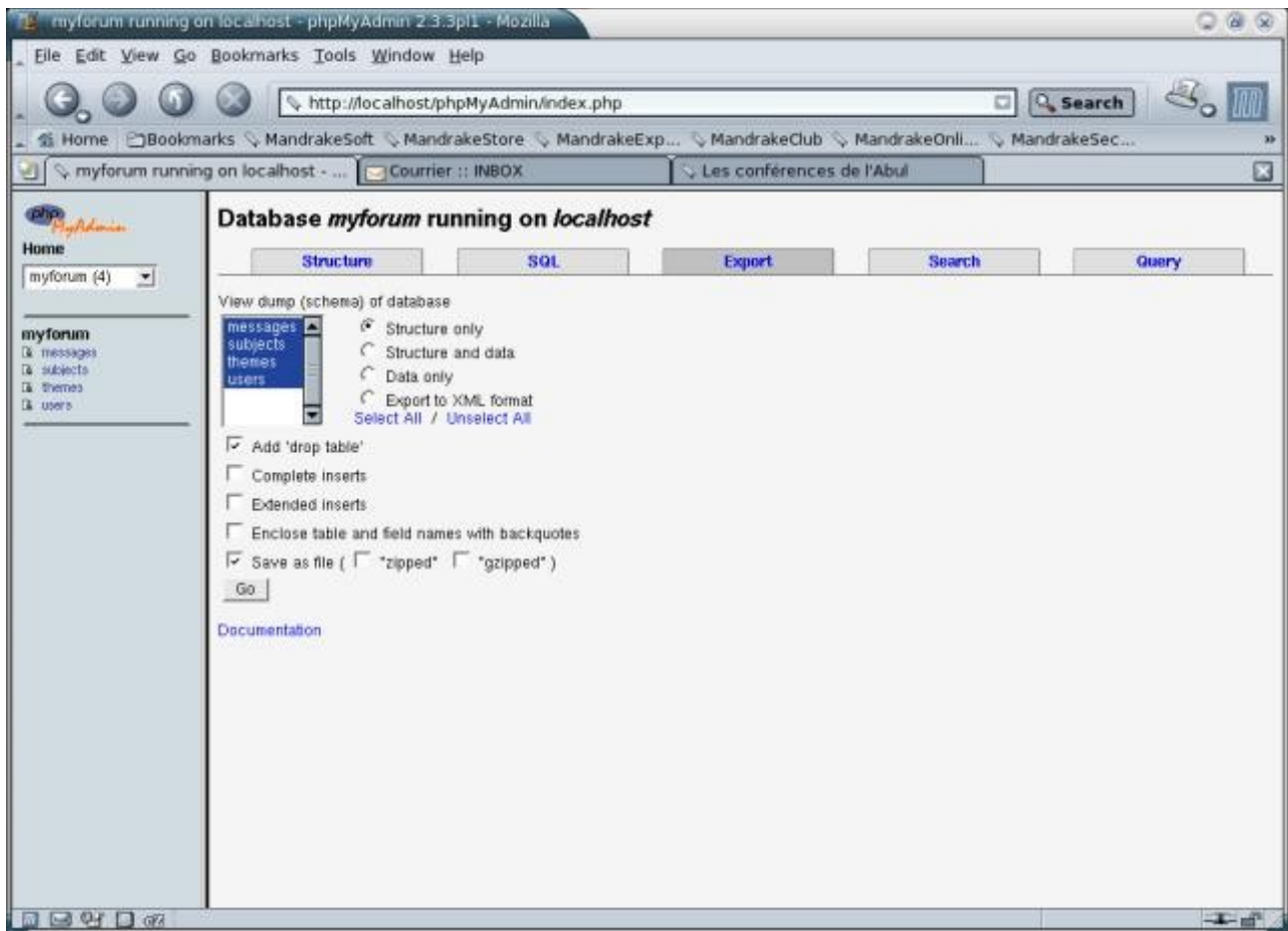
```
CREATE TABLE subjects (
id_subject INT UNSIGNED NOT NULL auto_increment,
title VARCHAR(255) NOT NULL,
date DATE NOT NULL,
time TIME NOT NULL,
id_user VARCHAR(15) NOT NULL,
id_theme TINYINT UNSIGNED NOT NULL,
PRIMARY KEY (id_subject),
KEY id_user (id_user),
KEY id_theme (id_theme)
);
```

```
CREATE TABLE messages (  
  id_message BIGINT UNSIGNED NOT NULL auto_increment,  
  title VARCHAR(255) NOT NULL,  
  text TEXT NOT NULL,  
  date DATE NOT NULL,  
  time TIME NOT NULL,  
  id_user VARCHAR(15) NOT NULL,  
  id_subject INT UNSIGNED NOT NULL,  
  PRIMARY KEY (id_message),  
  KEY id_user (id_user),  
  KEY id_subject (id_subject)  
);
```



## Sauvegarder la structure

Une fois, notre structure créée, il est bon de la sauvegarder. PhpMyAdmin permet d'exporter la structure ainsi que les données d'une base de données.



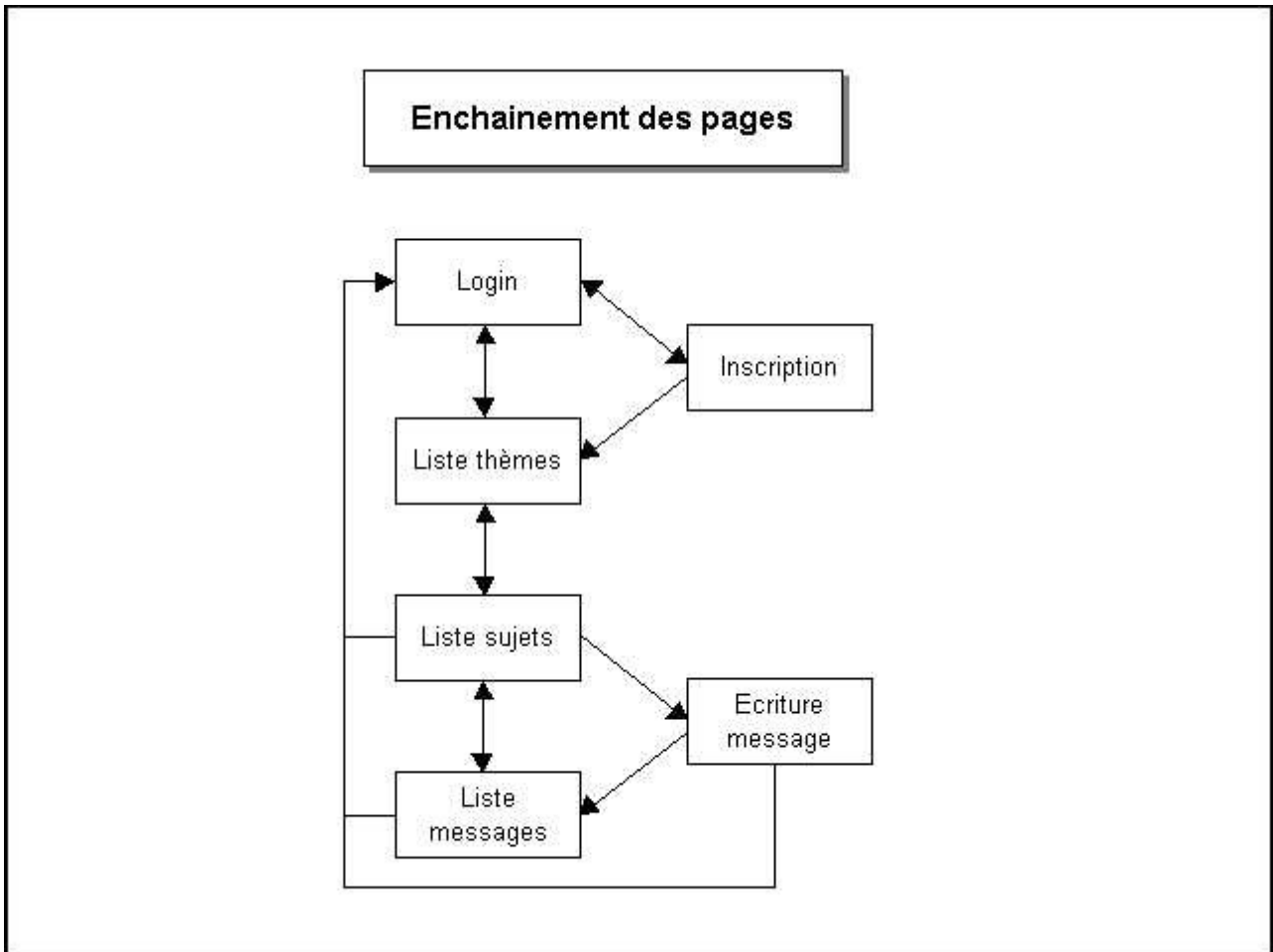
## IV. Maquettage de l'application

Avant de programmer les pages du forum en PHP avec les accès à la base de données MySQL, nous allons passer par une phase de maquettage de l'application. Il s'agit de créer les pages avec des valeurs statiques dans le but de réaliser l'interface et la navigation entre pages.

Pour notre forum de discussion, nous aurons besoin des pages suivantes :

- **Login** : cette page est la première page que l'utilisateur verra, il devra s'authentifier pour utiliser le forum
- **Inscription** : cette page permet à un nouvel utilisateur de s'inscrire, elle est accessible à partir de la page login
- **Liste thèmes** : cette page s'affiche lorsque l'utilisateur se sera authentifié, elle affiche la liste des thèmes disponibles sur le forum
- **Liste sujets** : cette page affiche l'ensemble des sujets de discussions appartenant à un thème.
- **Liste messages** : cette page affiche le fil de discussion d'un sujet.
- **Ecriture message** : cette page est appelée lorsque l'utilisateur veut répondre sur un sujet ou, créer un nouveau sujet de discussion.

Pour réaliser ces pages nous allons utiliser les technologies XHTML pour le contenu des pages et CSS pour la mise en forme de celles-ci.



## V.Le XHTML

Il s'agit du standard en vigueur datant du 26 janvier 2000. Le XHTML 1.0 est basé sur le langage XML.

Il s'agit uniquement du HTML 4 à la sauce XML. Donc si vous savez écrire du HTML 4, vous savez écrire du XHTML 1.0

Vous trouverez l'ensemble des spécifications du XHTML 1.0 à l'adresse suivante :

<http://www.w3.org/TR/xhtml1>

Parmi les points les plus importants de XML par rapport à SGML :

- Chaque élément ouvert doit être fermé
- les balises sont toujours orthographiées en minuscule
- les valeurs des attributs sont toujours renseignées et entre guillemets

### ***Les types de documents***

Il existe, comme pour le HTML 4.01, trois types de documents définissant le standard XHTML :

#### **Le type Strict**

Il s'agit du type qu'il faut choisir à chaque fois que cela est possible, en effet il permet de séparer complètement le contenu de la présentation.

Il interdit les balises HTML de type <font> ou <b> les effets de présentation devant être décrit uniquement à l'aide de CSS.

Pour implémenter ce type le document doit impérativement commencer par les lignes suivantes :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ....
  </head>
  <body>
    ....
  </body>
</html>
```

## **Le type Transitional**

Il permet de mélanger contenu et présentation et on peut l'utiliser si l'utilisation de CSS n'est pas possible ou satisfaisante (cas où les visiteurs ciblés ne disposent que de navigateurs anciens et le rendu compte plus que l'information, etc.).

Pour implémenter ce type le document doit impérativement commencer par les lignes suivantes :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ....
  </head>
  <body>
    ....
  </body>
</html>
```

---

## **Le type Frameset**

Ce type est à utiliser si vous souhaitez utiliser des cadres dans la présentation de vos pages HTML.

Pour implémenter ce type le document doit impérativement commencer par la ligne suivante :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ....
  </head>
  <body>
    ....
  </body>
</html>
```



## ***L'entête du document : <head></head>***

L'élément HEAD rassemble des informations sur le document courant, telles que son titre, des mots-clés pouvant être utilisés par des robots de recherche, et toute autre donnée considérée comme venant à part du contenu du document. Les éléments situés à l'intérieur de la déclaration HEAD ne devront pas être visualisées dans le cas d'agents utilisateurs conformes, sauf mention particulière.

### **La balise <title>**

Tout document HTML doit comporter un et un seul élément TITLE dans sa section HEAD. Les agents utilisateurs utilisent le contenu de l'élément titre pour donner à l'utilisateur une information concise sur le contenu probable du document, par exemple, en l'écrivant dans la barre de titre de fenêtre, ou en le citant par synthèse vocale.

Les titres peuvent contenir des entités caractères (pour les caractères accentués, les caractères spéciaux, etc.), mais ne pourront contenir aucune autre balise. Voici un exemple de titre de document :

```
<title>myForum : Login</title>
```

### **Les balises <meta>**

Ce type de balise n'est pas obligatoire mais permet de donner diverses informations telles que l'encodage de la page, le nom de l'auteur, les mots clés (pour les moteurs de recherches), une description du contenu de la page...

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="keywords" lang="fr" content="forum, discussion, logiciel, libre, PHP, mysql, xhtml, css" />
```

### **Les balises <link>**

La balise link permet d'effectuer des liens vers des ressources externes afin de définir par exemple les relations entre pages, mais aussi de définir le fichier de style CSS à utiliser (que nous verrons plus tard).

```
<link rel="next" href="Chapitre3.html" />
<link rel="stylesheet" href="default.css" type="text/css" />
```

## **Entête pour MyForum**

Pour notre application nous allons utiliser le code suivant :

```
<head>
  <title>myForum : Login</title>
  <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
  <meta name="keywords" lang="fr" content="forum, discussion, logiciel, libre, PHP, mysql, xhtml, css" />
  <meta name="author" lang="fr" content="Philippe Bousquet" />
  <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
  <meta name="license" content="GNU General Public License." />
  <link rel="stylesheet" href="default.css" type="text/css" />
</head>
```

Bien entendu nous allons changer le titre pour chaque page.

## ***Le contenu de la page : <body></body>***

Le corps d'un document contient son contenu effectif. Ce contenu pourra être représenté par un agent utilisateur d'une multitude de façons. Par exemple, pour les navigateurs visuels, vous pourrez considérer le corps comme un cadre dans lequel le contenu apparaît : texte, images, couleurs, dessins, etc. Pour les agents utilisateurs audio, le même contenu pourra être "parlé".

## **Balises de groupes <div></div> et <span></span>**

Les éléments DIV et SPAN, en conjonction avec les attributs id et class, proposent un mécanisme générique pour renforcer la structuration d'un document. Ce sont les deux seuls éléments du HTML qui n'ajoutent aucune information de présentation à leur contenu. De ce fait, il est possible de les utiliser associés à des identificateurs et des assignations de classe pour pouvoir formater l'apparence de zones de texte par application ultérieure de feuilles de style.

## **Balises d'entêtes <h1>, <h2>, <h3>, <h4>, <h5>, <h6>**

Un élément de titrage décrit brièvement le thème de la section qu'il introduit. Les informations de titrage pourront être récoltées par les agents utilisateurs pour, par exemple, construire une table des matières automatiques pour ce document.

Le HTML définit six niveaux de titrage depuis H1 affiché comme le plus imposant jusqu'à H6 le plus discret. Les navigateurs visuels utiliseront des polices plus ou moins grosses pour rendre compte du niveau de titre.

```
<h1>Niveau 1</h1>
<h6>Niveau 6</h6>
```

Le résultat obtenu :

**Niveau 1**  
Niveau 6

### **La balise de lien ou d'ancre <a></a>**

L'élément A peut être utilisé pour définir une ancre, un lien ou les deux.

L'exemple qui suit spécifie un lien :

```
Pour plus d'information sur du W3C, consultez le <a  
href="http://www.w3.org/">site Web du W3C</a>
```

Ce lien désigne la "home page" du World Wide Web Consortium. Lorsqu'un utilisateur active ce lien dans un agent utilisateur, ce dernier ira récupérer la ressource pointée, et dans notre cas, un document HTML.

L'exemple suivant illustre une définition d'ancre. Supposez que nous définissions une ancre nommée "ancre-une".

```
...texte précédent l'ancre...  
<a name="anchor-one">l'endroit de l'ancre une</a>  
...texte après l'ancre...
```

Cette définition assigne un nom d'ancre à la zone entière du document qui contient le texte "l'endroit de l'ancre une". Normalement, le contenu des éléments A ne seront pas affichés différemment que la normale lorsque l'élément A ne sert qu'à définir une ancre.

Une fois l'ancre définie, nous aimerions pouvoir nous y référer depuis un document (le même ou un autre). Les URL qui désignent les ancres se terminent par un "#" suivi du nom de l'ancre. Voici quelques exemples d'URL ainsi constituées :

- Une URL absolue : <http://www.mycompany.com/one.html#anchor-one>
- Une URL relative : ../one.html#anchor-one
- Lorsque le l'ancre est située dans le même document : #anchor-one

### **Texte préformatés <pre></pre>**

L'élément PRE indique aux agents utilisateurs visuels que le texte qui y est inclus est "préformaté".

Les agents utilisateurs visuels doivent traiter le texte préformaté comme suit :

- Ils doivent laisser les espaces blancs intacts.
- Ils devront écrire le texte avec une police à espacement fixe.
- Le retour automatique à la ligne pourra y être désactivé.
- Le traitement de la directionnalité doit rester opérationnel.

L'exemple suivant montre un texte préformaté :

```
<pre>
    Bonjour, j'aimerais réaliser un forum en PHP avec MySQL,
    Mais je ne sais pas du tout par ou commencer.

    Quelqu'un aurait il une idée ?
</pre>
```

Voici le rendu de votre navigateur :

```
Bonjour, j'aimerais réaliser un forum en PHP avec MySQL,
Mais je ne sais pas du tout par ou commencer.

    Quelqu'un aurait il une idée ?
```

### **Les tableaux <table></table>**

L'élément TABLE contient tous les autres éléments qui spécifient le titre, les lignes, le contenu, et le format d'un tableau.

```
<table>
    ... Le reste du tableau ...
</table>
```

### **Titres de tableaux :<caption></caption>**

Lorsque présent, le texte de l'élément CAPTION décrira usuellement la nature du tableau. L'élément CAPTION doit venir immédiatement après la balise de début de l'élément TABLE.

```
<table>
    <caption>Liste des themes du forum</caption>
    ... Le reste du tableau ...
</table>
```

### **Groupes de lignes : <thead>,<tbody>,<tfoot>**

Un tableau doit contenir au moins un groupe de lignes. Chaque groupe de lignes est divisé en trois parties : en-tête, corps, et piétement. Les en-têtes et piétements de tableaux sont optionnels. L'élément THEAD définit l'en-tête, l'élément TFOOT définit le piétement, et l'élément TBODY définit le corps de tableau.

Lorsqu'il existe, chacun des instances d'éléments THEAD, TFOOT, et TBODY DOIT contenir au moins une ligne (voir TR).

L'exemple suivant illustre l'ordre et la structure des en-têtes, pieds et corps de tableaux.

```
<table>
  <thead>
    <tr> ...information d'en-tête... </tr>
  </thead>
  <tfoot>
    <tr> ...information de piétement... </tr>
  </tfoot>
  <tbody>
    <tr> ...première ligne du bloc de données... </tr>
    <tr> ...deuxième ligne du bloc de données... </tr>
  </tbody>
</table>
```

### **Lignes de tableau : <tr></tr>**

L'élément TR agit comme un container accueillant des cellules d'une ligne d'un tableau.

### **Cellules de tableau : <th> et <td>**

Les éléments TH contiennent les en-têtes de colonnes (ou de ligne), tandis que les éléments TD définissent des cellules de données. Cette distinction permet aux agents utilisateurs de représenter différemment les en-têtes et les cellules de données, même en l'absence de feuilles de style.

Les cellules peuvent être vides (c.-à-d., ne contenir aucune données).

Exemple :

```
<table>
  <caption>MyForum : Liste des thèmes</caption>
  <thead>
    <tr><th>Thème</th><th>Description</th><th> Sujets
</th><th>Date</th></tr>
  </thead>
  <tbody>
    <tr>
      <td class="left"><a href="sujets.html">Débutants</a></td>
      <td class="left"><a href="sujets.html">Aide aux débutant sous
Linux</a></td>
      <td class="right">15</td>
      <td class="right">2003-12-21 12:45</td>
    </tr>
    <tr>
      <td class="left"><a href="sujets.html">Devel</a></td>
      <td class="left"><a href="sujets.html">Qestions relatives au
développement</a></td>
      <td class="right">2</td>
      <td class="right">2003-12-21 10:42</td>
    </tr>
  </tbody>
</table>
```

## Les objets externes

### Les images <img>

Cette balise permet d'intégrer des images dans un document XHTML.

Aujourd'hui, la plupart des navigateurs Web supportent les images, cependant Lynx (le navigateur le plus utilisé par les aveugles et les malvoyants) ne les supportent pas, en effet un aveugle ne peu en aucun cas apprécier le contenu d'une image, sauf si on lui décrit son contenu.

Dans cette optique la balise <img> donne la possibilité d'afficher un texte alternatif pour les navigateurs n'arrivant pas à afficher les images :

```

```

### **Les objets externes <object>**

Dans les anciennes versions du HTML intégrer des scripts externes (Applet Java, Objet ActiveX, ...), n'était pas toujours une chose aisée. Aujourd'hui une seule balise permet d'implémenter ce type de ressource, la balise <object>.

Celle ci permet en plus de proposer une solution alternative lorsque le navigateur ne peut afficher l'objet (plugin non activé, ...) :

```
<!-- En premier tenter l'applet Python -->
<object title="The Earth as seen from space"
classid="http://www.observer.mars/TheEarth.py">
  <!-- Sinon essayer une vidéo MPEG -->
  <object data="TheEarth.mpeg" type="application/mpeg">
    <!-- Sinon afficher un GIF animé -->
    <object data="TheEarth.gif" type="image/gif">
      <!-- Sinon afficher un texte alternatif -->
      The <strong>Earth</strong> as seen from space.
    </object>
  </object>
</object>
```

La balise OBJECT permet aussi d'intégrer des images (tout comme la balise IMG), aussi vaut il mieux préférer OBJECT (car une image est un objet externe).

### ***Les Formulaires <form></form>***

L'élément FORM fait office de container pour des contrôles. Il spécifie :

- Le programme qui traitera le formulaire une fois complété et dûment soumis (l'attribut action).
- La méthode par laquelle les données déposées par l'utilisateur seront transmises au serveur (l'attribut method, valeurs 'get' ou 'post').

Un formulaire peut contenir du texte et des balises (paragraphe, listes, etc.) ainsi que les contrôles détaillés ci-après.

La portée de l'attribut name associé à chacun des contrôles situé à l'intérieur d'un élément FORM est limitée à cet élément FORM.

## La balise <label>

L'élément LABEL peut être utilisé pour attacher une information à un autre contrôle (à l'exception des éléments LABEL eux-mêmes). Les labels pourront être affichés par les agents utilisateurs d'une multitude de façons (ex., visuellement, auditivement par des synthétiseurs vocaux, etc.)

```
<form action="..." method="post">
  <div class="field">
    <label for="cuser">Code user :</label> <input type="text"
name="cuser" id="cuser" value="" />
  </div>
  <div class="field">
    <label for="pass">Mot de passe :</label> <input type="password"
name="passwd" id="pass" value="" />
  </div>
  <div class="buttons">
    <input class="button" type="submit" name="action" value="Je me
connecte" /><br />
    <input class="button" type="submit" name="action" value="J'ai perdu
mon mot de passe" />
  </div>
</form>
```

Notez que les éléments LABEL sont associés aux éléments INPUT via l'attribut id.

## La balise <input>

La nature d'un contrôle défini par un élément INPUT dépend de la valeur de son attribut type.

### **Types de champs de saisie**

L'attribut type de l'élément INPUT détermine quel contrôle doit être créé.

#### **text**

Ce type crée un champ de texte mono ligne. La valeur émise lors de la soumission du formulaire est le texte entré.

#### **password**

Comme le type "text", mais le texte entré est visualisé à l'écran de sorte que les caractères ne puissent être reconnus (par exemple une série d'étoiles). Ce contrôle est utilisé pour l'entrée de données sensibles telles que des mots de passe. La valeur émise lors de la soumission du formulaire est le texte tapé (et non pas le texte affiché (!)).



### **checkbox**

Une case à cocher est une sorte de commutateur bipolaire. Lorsqu'elle est cochée, la case est dite "active". Lorsqu'elle est vide, la case est "inactive". La valeur de la case à cocher n'est émise que si cette dernière est active.

Plusieurs cases à cocher du même formulaire pourront partager le même nom. Au moment de la soumission, toute case à cocher "activée" y compris celles de nom semblable émet une paire nom/valeur dans laquelle le nom sera identique. Ceci permettra aux utilisateurs de choisir des valeurs multiples pour une propriété unique.

### **radio**

Un bouton radio est aussi un commutateur bipolaire. Lorsqu'il est marqué, le bouton radio est dit "actif". Lorsqu'il est vide, il est réputé "inactive". La valeur du bouton radio n'est émise que si ce dernier est actif.

Plusieurs boutons radio du même formulaire pourront partager le même nom. Cependant, seul l'un d'entre eux pourra être actif à la fois. Lorsque l'un des boutons radio est marqué, tous les autres portant le même nom sont automatiquement désélectionnés. Pour cet ensemble de boutons radio, il ne peut donc être émis qu'une seule paire nom/valeur.

### **submit**

Crée un bouton de soumission. Lorsque l'utilisateur clique sur ce bouton, le contenu du formulaire est soumis au programme spécifié par la localisation définie par l'attribut action de l'élément FORM englobant.

Un formulaire peut contenir plusieurs boutons de soumission. Cependant, seule la paire nom/valeur du bouton de soumission actionné sera envoyée dans le formulaire.

### **image**

Crée un bouton de soumission graphique. La valeur de l'attribut src spécifie l'URL de l'image qui servira de représentation graphique du bouton. Certains utilisateurs ne pourront visualiser cette image. Nous recommandons fortement d'adjoindre la définition d'un attribut alt valant pour alternative textuelle de l'image.

### **reset**

Crée un bouton de réinitialisation. Lorsque ce bouton est activé par l'utilisateur, les valeurs de tous les contrôles du formulaire sont remises à leur valeur initiale, telle que mentionnée dans l'attribut value. Le couple nom/valeur d'un bouton de réinitialisation n'est jamais envoyé lors de la soumission d'un formulaire.

### **hidden**

Crée un élément qui ne doit pas être représenté par l'agent utilisateur. Cependant, le couple nom/valeur de cet élément sera joint aux données envoyées lors de la soumission du formulaire.

Ce type de contrôle sera en général utilisé pour enregistrer des données d'échanges client/serveur qui seraient autrement perdues du fait de la nature volatile des processus HTTP.

## file

Demande à l'utilisateur de désigner un fichier. Lorsque le formulaire est soumis, le contenu de ce fichier sera transmis au serveur comme une valeur de n'importe quel autre contrôle.

## La balise <textarea>

L'élément TEXTAREA crée une zone de saisie de texte multi lignes (par opposition au champ de texte INPUT d'une seule ligne). Le contenu de cet élément fournit le texte initial qui est présenté dans la zone de saisie.

```
<form action="..." method="post">
  <div class="field2">
    <label for="titre">Titre : </label><br/>
    <input type="text" name="titre" id="titre" value="" />
  </div>
  <div class="field2">
    <label for="message"> Message : </message><br/>
    <textarea cols="50" rows="12" name="message" id="message">Entrez
ici votre texte...</textarea>
  </div>
  <div class="buttons">
    <input class="button" type="submit" name="action" value="Envoyer"
/>
    <input class="button" type="reset" name="reset" value="Reset" />
  </div>
</form>
```

## Les balises <select> et <option>

L'élément SELECT crée une liste d'options pouvant être sélectionnées par l'utilisateur. Chaque élément SELECT doit contenir au moins une définition d'option. Chaque option est spécifiée par une instance de l'élément OPTION.

Les agents utilisateurs se baseront sur le contenu de l'élément OPTION pour afficher le libellé de l'option dans la liste.

```
<form action="..." method="post">
  <select size="4" name="user">
    <option selected value="darken">Philippe BOUSQUET </option>
    <option value="alex">Alex</option>
    <option value="akira">Akira </option>
    <option value="tiana">Tatiana</option>
    <option value="chris">Christophe </option>
    <option value="liloune">Liloune</option>
  </select>
  <input type="submit" name="action" value="Valider" />
</form>
```

## **Vérifier la validité des pages**

Après l'écriture ou la génération de pages HTML il faut que les pages soient valides. Il ne suffit pas de vérifier le rendu des pages sur différents navigateurs, cette validation doit être faite auprès des validateurs de référence du W3C :

Service de validation HTML : <http://validator.w3.org/> (fourni par le W3C)

Lynx viewer est un service qui permet de vérifier le rendu de pages Web, telles qu'elles apparaissent dans le navigateur en mode texte Lynx. Lynx est très utilisé par les non-voyants et les malvoyants ainsi que sur certaines configurations informatiques ne disposant pas d'interface graphique évoluée.

## VI. Les pages du forum en XHTML

Pour notre forum, nous avons donc besoins de six pages XHTML, que nous allons décrire.

### ***La page d'identification***

Ceci est la page d'entrée du forum; toute personne voulant utiliser le forum doit se signer (et donc doit être inscrite).

Cette page contient donc un formulaire, avec deux champs saisissables :

- Code user : représentant le pseudo de l'utilisateur
- Mot de passe : son mot de passe lui permettant de se connecter.

Il y a deux boutons :

- Se connecter : qui permet d'effectuer la connexion proprement dit, et d'accéder au forum
- J'ai perdu mon mot de passe : qui permet de renvoyer à l'utilisateur son mot de passe par email.

Pour les personnes qui ne sont pas encore inscrites, il y a également un lien vers la page d'inscription.

### ***La page d'inscription***

Cette page est tout simplement le formulaire permettant à une personne de s'inscrire sur le forum.

Ce formulaire est composé de 6 champs saisissables et obligatoires :

- Le Nom de l'utilisateur
- Le Prénom de l'utilisateur
- L'email de l'utilisateur (pour pouvoir lui renvoyer son mot de passe)
- Le code user : l'utilisateur devra choisir un pseudo qui lui servira pour se connecter.
- Le mot de passe : l'utilisateur devra choisir un mot de passe qui lui servira pour se connecter.
- La confirmation : l'utilisateur devra ressaisir son mot de passe ici (vérifier l'égalité avec le champ précédent)

Il y a deux boutons :

- S'inscrire : qui permet d'enregistrer les données de l'utilisateur, de plus cela effectue la connexion au forum
- Rétablir : qui permet de réinitialiser le formulaire.

### ***La liste des thèmes***

Cette page liste les thèmes disponibles sur le forum, elle est constituée des éléments suivants :

- La barre de navigation : la seule option proposée est Quitter, qui renvoi vers la page d'identification.
- Le tableau dressant la liste des thèmes :
  - Nom du thème (lien vers la page des sujets)
  - Description du thème (lien vers la page des sujets)
  - Le nombre de sujets postés
  - La date du dernier message

### ***La liste des sujets***

Cette page liste les sujets postés pour un thème donné sur le forum, elle est constituée des éléments suivants :

- La barre de navigation :
  - Quitter, qui renvoi vers la page d'identification.
  - Thèmes, qui renvoi vers la page de liste des thèmes du forum
- Le tableau dressant la liste des sujets :
  - Auteur du sujet
  - Titre du sujet (lien vers la page des messages)
  - Le nombre de réponses au sujet
  - La date du dernier message concernant le sujet
- Un bouton Nouveau : qui permet de poster un nouveau sujet (lien vers Edition Message).

### ***Les messages concernant un sujet***

Cette page affiche la discussion concernant un sujet, elle est constituée des éléments suivants :

- La barre de navigation :
  - Quitter, qui renvoi vers la page d'identification.
  - Thèmes, qui renvoi vers la page de liste des thèmes du forum
  - Sujets, qui renvoi vers la page des sujets concernant le thème en cours
- Le tableau dressant la discussion, et ceci dans l'ordre d'arrivé :
  - Auteur du message
  - Titre du message
  - Date du message
  - Texte du message
- Un bouton Répondre : qui permet d'ajouter un message dans la discussion (lien vers Edition Message).

## L'édition d'un message

Cette page permet la création d'un sujet ou d'effectuer une réponse sur un sujet donné, elle est constituée des éléments suivants :

- La barre de navigation :
  - Quitter, qui renvoi vers la page d'identification.
  - Thèmes, qui renvoi vers la page de liste des thèmes du forum
  - Sujets, qui renvoi vers la page des sujets concernant le thème en cours
  - Messages, qui renvoi vers la page affichant la discussion en cours (dans le cas ou l'on répond à un message)
- Un formulaire composé de deux champs saisissables :
  - Un titre de message (pré alimenté avec "RE : titre sujet" dans le cas d'une réponse)
  - Une zone de texte, ou l'utilisateur saisira son message.
- Deux boutons :
  - Un bouton Envoyer qui permet d'enregistre un message et de revenir ver la page de la discussion.
  - Un bouton Rétablir qui permet de réinitialiser le formulaire.
- Le tableau dressant la discussion, dans l'ordre inverse d'arrivée :
  - Auteur du message
  - Titre du message
  - Date du message
  - Texte du message

## Une page XHTML valide

Voici par exemple le code source XHTML de la page d'édition d'un message au format XHTML (page validé par le validateur du W3C), sans mise en forme particulière.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>MyForum v0.1</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
    <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
    <meta name="author" lang="fr" content="Philippe Bousquet" />
    <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
    <meta name="license" content="GNU General Public License." />
  </head>
  <body>
    <!-- Entete de site -->
    <div class="header">
      
```

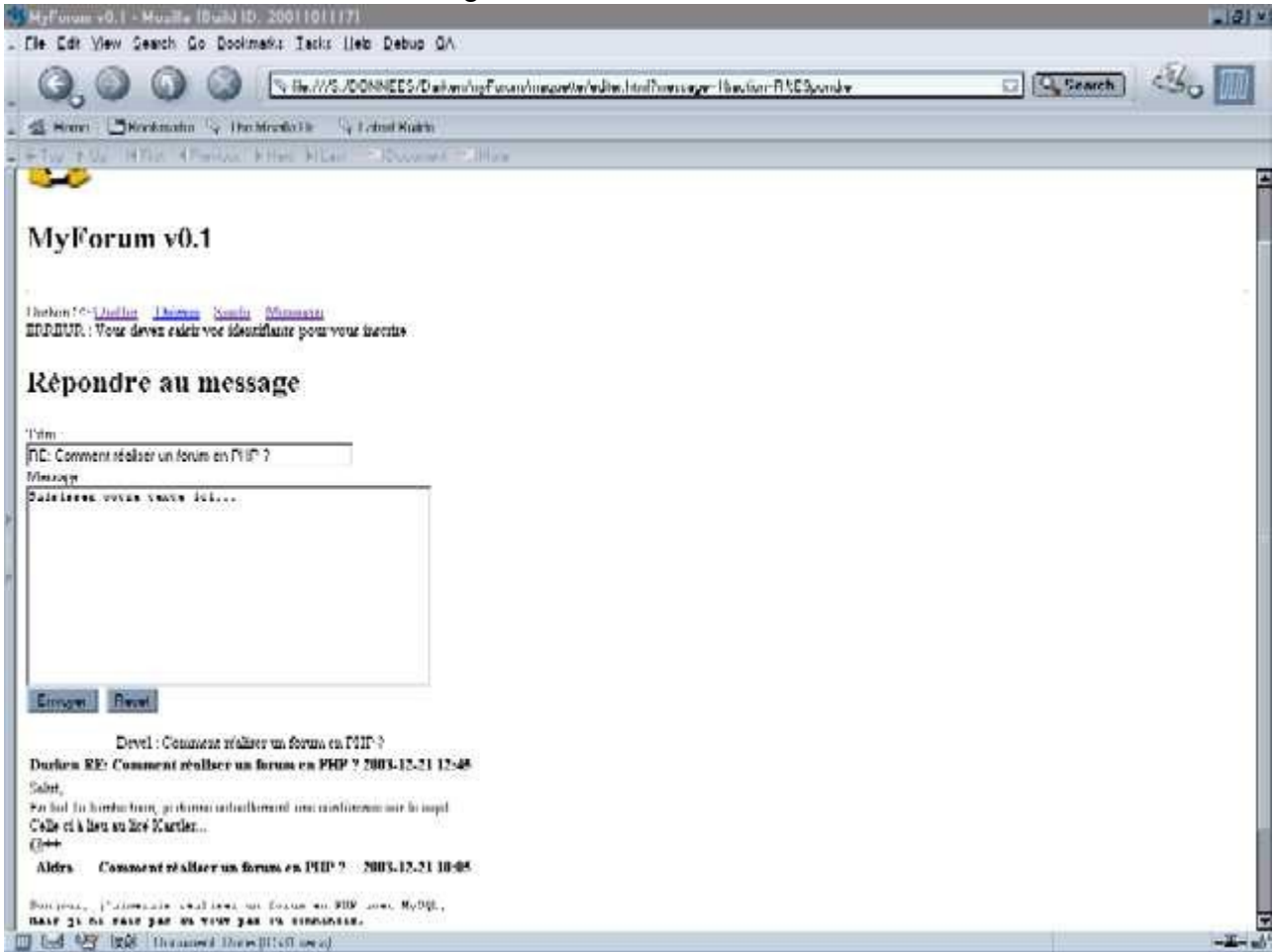
```

    <h1>MyForum v0.1</h1>
    <hr/>
</div>
<!-- La partie principale -->
<div class="main">
    <div class="pwd">
        Darken &gt;&gt; <a href="login.html"
title="quitter">Quitter</a> :: <a href="themes.html" title="retour aux
thèmes">Thèmes</a> :: <a href="sujets.html" title="retour sur
Devel">Sujets</a> :: <a href="messages.html" title="retour sur Comment
réaliser un forum en PHP ?">Messages</a>
    </div>
    <!-- Message d'erreur -->
    <div class="err">
        ERREUR : Vous devez saisir vos identifiants pour vous inscrire
    </div>
    <!-- Répondre au message -->
    <div class="box">
        <h1>Répondre au message</h1>
        <form action="messages.html" method="get">
            <div class="field2"><label for="title">Titre :
</label><br/><input type="text" id="title" name="titre" value="RE:
Comment réaliser un forum en PHP ?" size="50" /></div>
            <div class="field2"><label for="mess">Message :
</label><br/><textarea cols="50" id="mess" rows="12"
name="message">Saisissez votre texte ici...</textarea></div>
            <div class="buttons">
                <input class="button" type="submit" name="action"
value="Envoyer" />
                &nbsp;&nbsp;&nbsp;<input class="button" type="reset" />
            </div>
        </form>
    </div>
    <!-- La liste des themes -->
    <table class="old">
        <caption>Devel : Comment réaliser un forum en PHP ?</caption>
        <tbody>
            <!-- 1 message -->
            <tr class="row"><th class="left">Darken</th><th
class="left">RE: Comment réaliser un forum en PHP ?</th><th
class="right">2003-12-21 12:45</th></tr>
            <tr class="row"><td colspan="3" class="left">Salut,<br />En
fait tu tombe bien, je donne actuellement une conférence sur le
sujet.<br/>Celle ci à lieu au licé Kastler...<br/>@++</td></tr>
            <!-- 1 message -->
            <tr class="row"><th class="left">Akira</th><th
class="left">Comment réaliser un forum en PHP ?</th><th
class="right">2003-12-21 10:05</th></tr>
            <tr class="row"><td colspan="3" class="left"><pre>Bonjour,
j'aimerais réaliser un forum en PHP avec MySQL,<br/>Mais je ne sais pas
du tout par ou commencer.<br/>Quelqu'un aurait il une
idée....</pre></td></tr>
        </tbody>
    </table>
</div>
<!-- Le pied de page -->
<div class="footer">
    <hr/>
    Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
    Ce logiciel est sous licence Gnu Genral Public License
</div>

```

```
</body>  
</html>
```

Et voici le résultat dans votre navigateur internet :





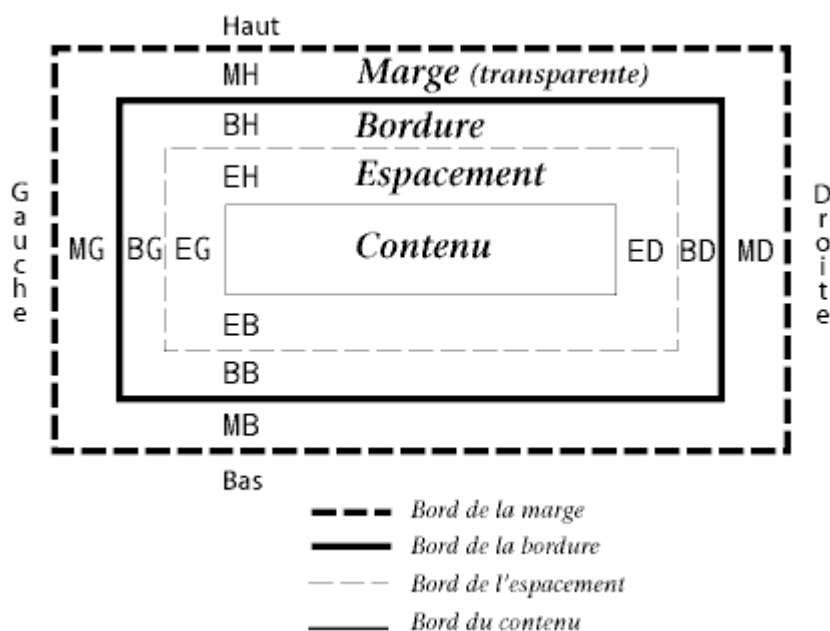
## VII. La mise en forme par CSS

Lors la création de nouveaux sites web, la séparation du contenu structuré de sa présentation est une aubaine pour les développeurs Web. En effet, il est possible de créer des feuilles de style qui s'appliqueront à un ou plusieurs documents, permettant ainsi de changer l'apparence d'un site en changeant simplement la feuille de style.

Il faut essayer de séparer au maximum le contenu de la présentation. Pour cela il est préférable d'écrire, de générer, des documents suivants la DTD XHTML 1.0 Strict plutôt que la DTD XHTML 1.0 Transitional.

### *Le modèle des boîtes*

Chaque boîte possède une aire de contenu (ex. un texte, une image, etc.) entourée en option par une aire d'espacement, une aire de bordure et une aire de marge ; Le schéma suivant illustre les relations entre ces aires et la terminologie employée pour les désigner :



On peut subdiviser la marge, la bordure et l'espacement selon qu'il s'agisse du côté gauche, droite, haut ou bas (ex. dans le schéma, "MG" mis pour marge gauche [ndt. margin-left], "ED" mis pour espacement droit [ndt. padding-right], "BH" mis pour bordure haute [ndt. border-top], etc.).

### **Les propriétés de marge : 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' et 'margin'**

Ces propriétés spécifient la largeur de l'aire de la marge d'une boîte. La propriété raccourcie 'margin' spécifie la valeur de la marge pour les quatre côtés à la fois, les autres propriétés fixant celle de leur côté respectif.

Les propriétés définies ici se réfèrent au type de valeur <marge-largeur>, celui-ci accepte une des valeurs suivantes :

**<longueur>** : Spécifie une largeur fixe ;

**<pourcentage>** : Le pourcentage est calculé par rapport à la largeur du bloc conteneur de la boîte générée. Ceci est vrai aussi pour les propriétés 'margin-top' et 'margin-bottom', sauf dans un contexte de page où le pourcentage se réfère à la hauteur de la boîte de la page.

**auto**

### **Les propriétés d'espacement : 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' et 'padding'**

Ces propriétés spécifient la largeur de l'aire d'espacement d'une boîte. La propriété raccourcie 'padding' spécifie la valeur de l'espacement pour les quatre côtés à la fois, les autres propriétés fixant celle de leur côté respectif.

Les propriétés définies ici se réfèrent au type de valeur <espacement-largeur>, celui-ci accepte une des valeurs suivantes :

**<longueur>** : Spécifie une largeur fixe.

**<pourcentage>** : Le pourcentage est calculé par rapport à la largeur du bloc conteneur de la boîte générée. De même pour les propriétés 'padding-top' et 'padding-bottom'.

À la différence des propriétés de marge, les valeurs d'espacement ne peuvent pas être négatives. Les valeurs de pourcentage des propriétés d'espacement, tout comme celles des propriétés de marge, se réfèrent à la largeur du bloc conteneur de la boîte générée.

## **Les propriétés de bordure**

Ces propriétés spécifient l'épaisseur, la couleur et le style de l'aire de bordure d'une boîte. Celles-ci s'appliquent à tous les éléments.

### **L'épaisseur de bordure : les propriétés 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' et 'border-width'**

Ces propriétés spécifient l'épaisseur de l'aire de bordure. Les propriétés définies ici se réfèrent au type de valeur <bordure-épaisseur>, celui-ci accepte l'une des valeurs suivantes :

**thin** : Une bordure mince.

**medium** : Une bordure moyenne.

**thick** : Une bordure épaisse.

**<longueur>** : L'épaisseur de la bordure a une valeur explicite. Cette valeur explicite ne peut être négative.

L'interprétation des trois premières valeurs dépend de l'agent utilisateur. Cependant, celui-ci doit respecter les directives suivantes

### **La couleur de bordure : les propriétés 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color' et 'border-color'**

Ces propriétés spécifient la couleur des bordures d'une boîte.

Voici la signification des valeurs :

**<couleur>** : Spécifie la valeur d'une couleur ;

**transparent** : La bordure est transparent (tout en ayant une épaisseur).

### **Le style de bordure : les propriétés 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style' et 'border-style'**

Ces propriétés spécifient le dessin des bordures d'une boîte (en trait plein, trait double, trait pointillé, etc.). Les propriétés définies ici se réfèrent au type de valeur <bordure-style>, celui-ci accepte l'une des valeurs suivantes :

- none** : Aucune bordure. Cette valeur force la valeur calculée de la propriété 'border-width' à 0 ;
- hidden** : Idem à 'none', sauf pour la résolution des conflits de bordure des éléments de table ;
- dotted** : La bordure est une ligne en pointillé ;
- dashed** : La bordure est une ligne en tirets ;
- solid** : La bordure est une ligne en trait plein.
- double** : La bordure est une ligne double, de deux traits pleins. La somme de ces lignes et de l'espace entre elles est égale à la valeur de 'border-width'.
- groove** : La bordure donne l'impression qu'elle est gravée dans le canevas ;
- ridge** : À l'opposé de 'groove', la bordure semble sortir du canevas ;
- inset** : La bordure donne l'impression que la boîte entière est incrustée dans le canevas ;
- outset** : À l'opposé de 'inset', la bordure donne l'impression que la boîte entière est extrudée du canevas

### **La largeur du contenu : la propriété 'width'**

Cette propriété spécifie la largeur du contenu des boîtes générées par les éléments de type bloc.

Les valeurs ont les significations suivantes :

- <longueur>** : Spécifie une largeur fixe ;
- <pourcentage>** : Spécifie une largeur en pourcentage. Le pourcentage est calculé en fonction de la largeur du bloc conteneur de la boîte générée ;
- auto**

La propriété 'width' n'admet pas de valeurs négatives.

### **La hauteur du contenu : la propriété 'height'**

Cette propriété spécifie la hauteur du contenu des boîtes générées par les éléments de type bloc.

Les significations des valeurs sont :

- <longueur>** : Spécifie une hauteur fixe ;
- <pourcentage>** : Spécifie une hauteur en pourcentage. Celui-ci est calculé en fonction du bloc conteneur de la boîte générée. Si la hauteur du bloc conteneur n'est pas spécifiée explicitement (c.à.d., celle-ci dépendant de la hauteur du contenu), sa valeur est considérée comme étant 'auto' ;

## auto

La propriété 'height' n'admet pas de valeur négative.

### ***Les couleurs et arrières plans***

Chaque boîte possède une aire de contenu (ex. un texte, une image, etc.) entourée en option par une aire d'espacement, une aire de bordure et une aire de marge ; Le schéma

#### **La couleur d'avant-plan : la propriété 'color'**

Cette propriété décrit la couleur d'avant-plan du contenu de texte d'un élément. Voici plusieurs façons d'indiquer la couleur rouge :

Exemple(s) :

```
em { color: red } /* nom de couleur prédéfini */  
em { color: rgb(255,0,0) } /* couleurs en RGB allant de 0 à 255 */  
em { color: #FF0000 } /* Couleur HTML RGB Hexadécimal */
```

#### **L'arrière-plan**

Les auteurs peuvent spécifier l'arrière-plan d'un élément (c.à.d. la surface où celui est rendu) comme étant une couleur ou bien une image. Selon le modèle de la boîte, l'arrière-plan correspond aux aires du contenu, de l'espacement et de bordure. Les couleurs et styles de bordure sont spécifiées par les propriétés de bordure. Les marges étant transparentes, l'arrière-plan du parent est toujours visible au travers de celles-ci.

Bien que les propriétés d'arrière-plan ne s'héritent pas, l'arrière-plan de la boîte du parent transparaîtra par défaut, du fait de la valeur initiale 'transparent' de la propriété 'background-color'.

L'arrière-plan de la boîte générée par l'élément racine recouvre la totalité du canevas.

Dans le cas de documents HTML, on recommande aux auteurs de spécifier un arrière-plan à l'élément BODY, plutôt qu'à l'élément HTML. Les agents utilisateurs devraient suivre les règles de préséance suivantes pour remplir le fond du canevas : quand la valeur de la propriété 'background' pour l'élément HTML diffère de 'transparent', alors utiliser la valeur spécifiée, autrement utiliser celle spécifiée par la propriété 'background' de l'élément BODY. Le rendu n'est pas défini si la valeur finale reste 'transparent'.

#### **Les propriétés : 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position' et 'background'**

### 'background-color'

Valeur : <couleur> | transparent | inherit

Initiale: transparent

Cette propriété donne la couleur d'arrière-plan d'un élément, avec une valeur de <couleur>, ou bien avec le mot-clé 'transparent', celui-ci laissant éventuellement voir les couleurs situées plus en-dessous.

### 'background-image'

Valeur : <uri> | none | inherit

Initiale : none

S'applique à : tous les éléments

Cette propriété spécifie l'image d'arrière-plan d'un élément. En même temps qu'une image, les auteurs devraient aussi spécifier une couleur d'arrière-plan, cette couleur étant employée en remplacement d'une image indisponible. Celle-ci, une fois disponible, vient se superposer sur le fond coloré. La couleur du fond étant ainsi visible au travers des zones transparentes de l'image.

### 'background-repeat'

Valeur : repeat | repeat-x | repeat-y | no-repeat | inherit

Initiale : repeat

Quand on spécifie une image d'arrière-plan, cette propriété indique si l'image est répétée (apposée) et la manière de la répétition. La mosaïque de fond résultante correspond aux aires de contenu, d'espacement et de bordure de la boîte de l'élément en question. Les significations des valeurs sont :

Un exemplaire de l'image d'arrière-plan est placé au centre, puis d'autres exemplaires de celles-ci se placent les unes au-dessus, les autres en-dessous, produisant une bande verticale derrière l'élément.

### 'background-attachment'

Valeur : scroll | fixed | inherit

Initiale : scroll

Quand on spécifie une image d'arrière-plan, cette propriété indique si l'image est fixe par rapport à l'espace de visualisation (pour la valeur 'fixed'), ou si celle-ci défile en même temps que le document (pour la valeur 'scroll').

### 'background-position'

Valeur : [ [<pourcentage> | <longueur> ]{1,2} | [ [top | center | bottom] || [left | center | right] ] ] | inherit

Initiale : 0% 0%

Quand on spécifie une image d'arrière-plan, cette propriété indique la position initiale de celle-ci.

Quand l'image d'arrière-plan est fixe par rapport à l'espace de visualisation (voir la propriété 'background-attachment'), cette image se place relativement à celui-ci, et non par rapport à l'aire d'espacement de l'élément.

## **Les polices**

Chaque boîte possède une aire de contenu (ex. un texte, une image, etc.) entourée en option par une aire d'espacement, une aire de bordure et une aire de marge ; Le schéma

### **La famille de polices : la propriété 'font-family'**

Cette propriété donne une liste, par ordre de priorité, de noms de familles de polices et/ou de noms de familles génériques. Pour contourner certaines difficultés, une police seule ne contenant pas nécessairement tous les glyphes pour le rendu de chacun des caractères d'un document, ou des polices étant absentes de certains systèmes, ainsi la propriété permet aux auteurs de spécifier une liste de polices, celles-ci de même style et taille, qui sont essayées successivement pour vérifier la correspondance d'un glyphe avec un caractère donné. On appelle cette liste un jeu de polices.

La famille de polices générique sera utilisée si une, ou plusieurs, des polices d'un jeu de polices n'étaient pas disponibles. Bien que certaines polices fournissent un glyphe de substitution pour indiquer un "caractère manquant", celui-ci prenant typiquement l'aspect d'un carré, on ne devrait pas considérer ce glyphe comme une correspondance valide.

Il existe deux types de noms de famille de polices :

#### **<famille-nom>**

Le nom d'une famille de polices choisi. Dans l'exemple précédent, "Baskerville", "Heisi Mincho W3" et "Symbol" sont des noms de famille de polices. Celles dont l'intitulé contient des blancs devraient être écrites entre guillemets. Si on omet les guillemets, chacun des caractères blancs avant et après le nom de la police sont ignorés et chaque séquence de blancs, dans celui-ci, est convertie en un seul espace ;

#### **<famille-générique>**

On définit les familles génériques suivantes : 'serif', 'sans-serif', 'cursive', 'fantasy' et 'monospace'. Voir le passage traitant des familles de polices génériques pour leur description. Les noms de famille de polices génériques sont des mots-clés, on ne doit pas les mettre entre guillemets.

On recommande aux auteurs de citer une famille de polices générique comme dernière alternative, pour une meilleure efficacité.

## Le style de police : les propriétés 'font-style', 'font-variant', 'font-weight' et 'font-stretch'

### 'font-style'

Valeur : normal | italic | oblique | inherit

Initiale : normal

La propriété 'font-style' sélectionne, parmi les polices d'une famille de polices, celles avec un dessin normal (aussi appelé "roman" ou "upright"), italique et oblique. Les significations des valeurs sont :

**normal** Spécifie une police dite normale dans la base de données de police de l'agent utilisateur ;

**oblique** : Spécifie une police dite oblique dans la base de données de police de l'agent utilisateur. Les polices dont le nom contient les mots Oblique, Slanted ou Incline sont typiquement étiquetées 'oblique' dans la base de données de police de l'agent utilisateur. Celles avec ce label ayant pu avoir été obtenues électroniquement en inclinant une police normale ;

**italic** : Spécifie une police dite italique dans la base de données de police de l'agent utilisateur, ou, s'il n'y en a pas, une avec un label 'oblique'. Les polices dont le nom contient les mots Italic, Cursive ou Kursiv seront typiquement étiquetées 'italic'.

Exemple(s) :

### 'font-variant'

Valeur : normal | small-caps | inherit

Initiale : normal

Dans une police en petites capitales, les glyphes des lettres minuscules ont un aspect similaire aux lettres majuscules, cependant avec une taille réduite et des proportions légèrement différentes. La propriété 'font-variant' appelle ce genre de police bicamérale (qui a deux casses, comme les écritures latines). Celle-ci ne produit aucun effet visible pour les écritures monocamérales (qui n'ont qu'une seule casse, comme la plupart des systèmes d'écriture mondiaux). Les significations des valeurs sont :

**normal** : Spécifie une police qui n'est pas étiquetée comme étant en petites capitales ;

**small-caps** : Spécifie une police étiquetée comme étant en petites capitales. S'il n'y a pas une telle police, les agents utilisateurs devraient en effectuer la simulation, par exemple en sélectionnant une police normale et y remplaçant les lettres minuscules par des majuscules mises à l'échelle. En dernier ressort, les lettres majuscules inchangées d'une police normale peuvent se substituer aux glyphes en petites capitales, ainsi le texte apparaîtrait entièrement en majuscule.



### 'font-weight'

Valeur : normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

Initiale : normal

S'applique à : tous les éléments

Héritée : oui

Pourcentage : sans objet

Médias : visuel

La propriété 'font-weight' spécifie la graisse de la police.

### 'font-stretch'

Valeur : normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit

La propriété 'font-stretch' sélectionne les dessins normal, comprimé ou élargi dans une famille de polices. Les valeurs des mots-clés absolus, du plus étroit au plus espacé, sont :

- ultra-condensed
- extra-condensed
- condensed
- semi-condensed
- normal
- semi-expanded
- expanded
- extra-expanded
- ultra-expanded

Le mot-clé relatif 'wider' spécifie la valeur d'expansion supérieure à celle héritée (sans effet si la valeur héritée est déjà 'ultra-expanded'), à l'inverse du mot-clé relatif 'narrower' celui-ci spécifiant la valeur de contraction inférieure à celle héritée (sans effet si la valeur héritée est déjà 'ultra-condensed').

## La taille de police : la propriété 'font-size'

'font-size'

Valeur : <taille-absolue> | <taille-relative> | <longueur> | <pourcentage> | inherit

Initiale : medium

Cette propriété décrit la taille d'une police spécifiée explicitement. Celle-ci correspond au carré em, un concept issu de la typographie. Noter que certains glyphes peuvent déborder de leur carré em. Les significations des valeurs sont :

<taille-absolue>

Un mot-clé <taille-absolue> se réfère à une entrée de la table des tailles de police, celle-ci étant dressée et mise en œuvre par l'agent utilisateur. Les valeurs possibles :

[ **xx-small** | **x-small** | **small** | **medium** | **large** | **x-large** | **xx-large** ]

Pour un écran de moniteur, on suggère un facteur d'échelle de 1.2 entre les valeurs consécutives de la table ; si la valeur 'medium' correspond à 12pt, la valeur 'large' devrait correspondre à 14.4pt. Les facteurs d'échelle peuvent différer selon les médias considérés. Par ailleurs, l'agent utilisateur devrait prendre en compte la qualité et la disponibilité des polices au moment du calcul de cette table. Celle-ci peut aussi différer d'une famille de polices à une autre.

<taille-relative>

Un mot-clé <taille-relative> s'entend par rapport à la table des tailles de police et par rapport à la taille de la police de l'élément parent. Les valeurs possibles :

[ larger | smaller ]

Par exemple, quand l'élément parent a une taille de police de valeur 'medium', l'élément en question ayant une valeur 'larger', la taille de police résultante de celui-ci correspondra à 'large'. Si la taille de police de l'élément parent a une valeur trop éloignée d'une des valeurs de la table, l'agent utilisateur est libre d'effectuer une interpolation entre les deux valeurs qui la circonscrivent ou un arrondi à la valeur la plus proche. L'agent utilisateur peut devoir procéder à une extrapolation des valeurs de la table quand une valeur numérique sort du champ des mots-clés.

<longueur>

Une valeur de longueur spécifie une taille de police absolue (indépendante de la table des tailles de police de l'agent utilisateur). Les valeurs négatives ne sont pas admises ;

<pourcentage>

Une valeur en pourcentage spécifie une taille de police absolue par rapport à celle de l'élément parent. Leur emploi, de même pour les valeurs exprimées en 'em', conduit à des feuilles de style plus fiables, en plein accord avec le principe de la cascade.

La valeur réelle, pour cette propriété, peut différer de la valeur calculée, la propriété 'font-size-adjust' ayant une valeur numérique et certaines tailles de police étant indisponibles.

## **Le texte**

### **L'alinéa : la propriété 'text-indent'**

'text-indent'

Valeur : <longueur> | <pourcentage> | inherit

Initiale : 0

Cette propriété spécifie un alinéa pour la première ligne du texte dans un bloc. Plus précisément, celui de la première boîte de la première rangée dans la première boîte de ligne de ce bloc. Cette boîte est indentée à partir du bord gauche (ou droit, pour une mise en page de droite à gauche) de la boîte de ligne. Les agents utilisateurs devraient représenter cette indentation comme un espace vide.

### **L'alignement : la propriété 'text-align'**

'text-align'

Valeur : left | right | center | justify | <chaîne> | inherit

Initiale : selon l'agent utilisateur et selon le sens d'écriture

Cette propriété décrit l'alignement d'un contenu en-ligne dans un élément de type bloc.

Les significations des valeurs sont :

**left, right, center** et **justify**

Respectivement, l'alignement à gauche, à droite, centré et justifié d'un texte ;

#### **<chaîne>**

Spécifie une chaîne sur laquelle les cellules d'une table vont s'aligner (voir le passage sur l'alignement horizontal dans une colonne pour le détail et pour un exemple). Cette valeur ne s'applique qu'aux cellules d'une table. Appliquée à un autre élément, celle-ci sera considérée comme étant 'left' ou 'right', en fonction de la valeur de la propriété 'direction', respectivement 'ltr' ou 'rtl'.

Un pavé de texte consiste en un empilement de boîtes de ligne. Pour les valeurs 'left', 'right' et 'center', cette propriété indique la façon dont les boîtes en-ligne, dans chacune des boîtes de ligne, s'alignent par rapport aux côtés gauche et droite de ces boîtes de ligne ;

l'alignement n'est pas effectué par rapport à l'espace de visualisation. Pour la valeur 'justify', l'agent utilisateur peut, en plus des ajustements de position, étirer les boîtes de ligne. Voir également les propriétés 'letter-spacing' et 'word-spacing').

### **La décoration: la propriété 'text-decoration'**

'text-decoration'

Valeur : none | [ underline || overline || line-through || blink ] | inherit

Initiale : none

S'applique à : tous les éléments

Cette propriété décrit les décorations qui sont ajoutées au texte d'un élément. Quand la propriété est appliquée à un élément de type bloc, elle agit sur tous les descendants de type en-ligne de celui-ci. Quand elle est appliquée à un élément de type en-ligne, ou agit sur celui-ci, cette propriété influence toutes les boîtes générées par cet élément. Quand les éléments sont vides ou sans contenu textuel (ex. l'élément IMG en HTML), les agents utilisateurs doivent ignorer cette propriété.

Les significations des valeurs sont :

**none** : Aucune décoration ;

**underline** : Chaque ligne de texte est soulignée ;

**overline** : Chaque ligne de texte reçoit un trait au-dessus ;

**line-through**: Chaque ligne de texte est rayée en son milieu ;

**blink** : Le texte clignote (une alternance entre visible et invisible). Les agents utilisateurs conformes ne sont pas tenus de reconnaître cette valeur.

Cette propriété n'est pas héritée, cependant les boîtes qui descendent d'une boîte de bloc donnée devraient recevoir la même décoration que celle-ci (ex. toutes devraient être soulignées). La couleur de la décoration devrait être conservée, même si la valeur de la propriété 'color' des éléments descendants était différente.

## **Divers**

### **Le positionnement des flottants : la propriété 'float'**

'float'

Valeur : left | right | none | inherit

Initiale : none

S'applique à : tous les éléments, sauf ceux positionnés et ceux dont le contenu est généré

Cette propriété spécifie le flottement d'une boîte à gauche, à droite ou pas du tout. On peut l'employer pour des éléments générant des boîtes qui ne sont pas en position absolue.

Voici la signification des valeurs que celle-ci admet :

**left** : L'élément génère une boîte de bloc qui flotte à gauche. Le contenu s'écoule sur son flanc droit en commençant en haut (en fonction de la valeur de la propriété 'clear'). En ignorant la valeur de la propriété 'display', sauf si cette valeur est 'none' ;

**right** : Identique à 'left', mais en inversant la gauche de la droite ;

**none** : La boîte ne flotte pas.

## **Le contrôle du flux autour des flottants : la propriété 'clear'**

'clear'

Valeur : none | left | right | both | inherit

Initiale : none

S'applique à : ceux des éléments de type bloc

Cette propriété indique quels côtés d'une ou des boîtes d'un élément ne doivent pas être adjacents à une boîte flottante précédente. (Il peut arriver que l'élément lui-même ait des descendants flottants, la propriété 'clear' n'a alors aucun effet sur eux).

Cette propriété ne peut s'appliquer qu'aux éléments de type bloc (dont les flottants). Dans le cas des boîtes compactes et en enfilade, la propriété s'applique à la boîte de bloc finale à laquelle celles-ci appartiennent.

Voici la signification des valeurs admises par la propriété quand on l'applique aux boîtes de bloc non flottantes :

**left** : La marge haute de la boîte générée est augmentée juste assez pour que le bord haut de sa bordure soit sous le bord externe bas d'une boîte flottant à gauche issue d'un élément précédent du document source ;

**right** : Reprendre le précédent en inversant gauche et droite ;

**both** : La boîte générée se déplace sous chacune des boîtes flottantes qui sont issues d'éléments précédents du document source ;

**none** : La boîte ne subit aucune contrainte de position vis-à-vis des flottants.

### La propriété 'display'

'display'

Valeur : inline | block | list-item | ..... | none | inherit

Initiale : inline

S'applique à : tous les éléments

Les valeurs de cette propriété ont le sens suivant :

**block** : induit un élément à générer une boîte de bloc principale ;

**inline** : induit un élément à générer une ou plusieurs boîtes en-ligne ;

**list-item** : induit un élément (ex. l'élément LI en HTML) à générer une boîte de bloc principale et une boîte en-ligne pour un item de liste ; Consulter la partie traitant des listes pour des informations et des exemples de mise en forme de celles-ci ;

**none** : cette valeur fait qu'aucune boîte n'est générée par l'élément dans la structure de formatage (c.à.d., cet élément n'a pas d'influence sur la mise en forme du document). Les éléments qui en descendent ne génèrent pas de boîtes non plus ; on ne peut plus modifier leur comportement avec la propriété 'display'.

Il est à noter qu'une valeur 'none' ne crée pas de boîte invisible, elle ne crée pas de boîte du tout. CSS comprend des mécanismes permettant la génération de boîtes dans la structure de formatage, boîtes qui influencent la mise en forme mais qui ne sont pas visibles.

### La visibilité : la propriété 'visibility'

'visibility'

Valeur : visible | hidden | collapse | inherit

Initiale : visible

S'applique à : tous les éléments

La propriété 'visibility' spécifie le rendu, ou non, des boîtes générées par un élément donné. Ces boîtes, bien qu'invisibles, influencent toujours la mise en forme du document (utiliser la propriété 'display' avec la valeur 'none' pour prohiber la génération d'une boîte, et ainsi toutes influences sur la mise en forme).

Les valeurs ont les significations suivantes :

**visible** : La boîte générée est visible ;

**hidden** : La boîte générée est invisible (entièrement transparente), mais celle-ci influençant toujours la mise en forme

## **Le débordement : la propriété 'overflow'**

'overflow'

Valeur : visible | hidden | scroll | auto | inherit

Initiale : visible

S'applique à : ceux des éléments de type bloc et ceux remplacés

Cette propriété spécifie si le contenu d'un élément de type bloc doit être rogné quand celui-ci déborde de la boîte de cet élément (qui se comporte comme un bloc conteneur).

Les significations des valeurs sont :

**visible** : Le contenu ne sera pas rogné, et celui-ci peut être représenté hors de la boîte du bloc ;

**hidden** : Le contenu sera rogné et aucun mécanisme de défilement ne devrait être fourni pour voir la partie qui aura été rognée. On spécifie la taille et la forme du reliquat du rognage avec la propriété 'clip' ;

**scroll** : Le contenu sera rogné et, si disponible, l'agent utilisateur fournit un mécanisme de défilement visible à l'écran (tel qu'une barre de défilement ou un dispositif panoramique), celui-ci devrait apparaître pour une boîte donnée, que le contenu de celle-ci soit rogné ou non. Ceci pour éviter l'inconvénient que représenterait des barres de défilement apparaissant et disparaissant dans un environnement dynamique. Quand cette valeur est spécifiée conjointement avec un type de média 'print', la partie du contenu ayant débordée devrait aussi être imprimée ;

**auto** : L'interprétation de cette valeur dépend de l'agent utilisateur, cependant, celui-ci devrait fournir un mécanisme de défilement quand les boîtes débordent.

Même quand la valeur de la propriété 'overflow' est 'visible', il peut arriver que le contenu soit rogné par le système d'exploitation, pour tenir dans la fenêtre du document de l'agent utilisateur.

## ***Valider sa feuille de style CSS***

Après l'écriture ou la génération de feuilles de styles CSS, on peut vérifier la validité de celles ci grâce à un service de validation du W3C.

Service de validation CSS : <http://jigsaw.w3.org/css-validator/> (fourni par le W3C)



## Exemple : MyForum avec CSS

Voici un exemple de feuille de style CSS (default.css) :

```
* {
    font-family: Verdana, Arial, sans-serif;
}
*.left {
    text-align: left;
}
*.center {
    text-align: center;
}
*.right {
    text-align: right;
}
body {
    background: #FFFFFF;
    color: #000000;
}

a:link {
    color: #800060;
    background: transparent;
    text-decoration: none;
}

a:visited {
    color: #800060;
    background: transparent;
    text-decoration: none;
}

a:hover {
    color: #FFFFFF;
    background: #800060;
    text-decoration: none;
}

div.header {
    border-bottom: 1px #800050 solid;
}
div.header h1 {
    font-size: 30pt;
    color: #800050;
}
hr {
    display: none;
}
img.icon {
    float: left;
}
div.main {
    margin: 20px;
}
div.err {
    text-align: center;
    font-size: 12pt;
    font-weight: bold;
}
```

```

        color: #FF0000;
    }
    div.box {
        margin-left : 20%;
        margin-right : 20%;
        margin-top: 15px;
        margin-bottom: 15px;
        background: #800050;
        color: #FFFFFF;
    }
    div.box h1 {
        font-size: 14pt;
        text-align: center;
        color: #FFFFFF;
    }
    div.field {
        margin-top: 10px;
        width:70%;
        text-align:right;
        font-size: 8pt;
        font-weight: bold;
    }
    div.field2 {
        margin-top: 10px;
        margin-left: 10%;
        text-align:left;
        font-size: 8pt;
        font-weight: bold;
    }
    div.buttons {
        margin-top: 10px;
        padding-top: 10px;
        border-top: 1px white solid;
        text-align: center;
        font-size: 8pt;
        font-weight: bold;
    }
    input.button {
        margin: 5px;
        background: #C00060;
        color: #ffffff;
        border-color: #C00060;
    }
    div.pwd {
        background: #C0C0C0;
        font-size: 12pt;
        font-weight: bold;
        margin-bottom: 20px;
    }
    table {
        margin-top: 15px;
        margin-left : 10%;
        width:80%;
        border-collapse: collapse;
        border-spacing: 0;
        border: 1px #C00060 solid;
        background: #800050;
        color: #FFFFFF;
    }
    caption {
        margin: 15px;

```

```
    text-align: center;
    color: #800050;
    font-size: 14pt;
    font-weight: bold;
}
table.old caption {
    display: none;
}
table a:link {
    color: #FFFFFF;
    background: transparent;
    text-decoration: none;
}

table a:visited {
    color: #FFFFFF;
    background: transparent;
    text-decoration: none;
}

table a:hover {
    color: #FFFF00;
    background: transparent;
    text-decoration: none;
}
td {
    border-top: 1px #C00060 solid;
    font-size: 8pt;
}
th {
    border-top: 1px #C00060 solid;
    background: #C00060;
    text-align: left;
    font-size: 8pt;
    font-weight: bold;
}
textarea {
    font-size: 8pt;
}
div.footer {
    border-top: 1px #800050 solid;
    font-size: 8pt;
}
```

Voici également ce que cela donne lorsque l'on applique la feuille de style à la page XHTML développée précédemment :



Pour appliquer la feuille CSS il faut rajouter la ligne suivante dans la partie <head></head> de la source XHTML :

```
<link rel="stylesheet" href="default.css" type="text/css" />
```

## **VIII.Programmation de l'application**

Une fois le modèle de données défini et le maquettage de l'application réalisé, on peut remplacer le contenu statique de nos page web par de la programmation PHP, Nous allons donc voir dans un premier temps les bases du langage PHP, puis, dans un deuxième temps la programmation que nous allons implémenter dans nos pages HTML pour rendre notre forum Opérationnel.

## IX. Le langage PHP

PHP est un langage de scripts multi plates-formes, embarqué dans des documents HTML. Plus simplement PHP vous offre un moyen de placer des instructions dans vos documents HTML en vue de créer des contenus dynamiques.

Ces instructions sont lues et analysées par le serveur web. Elles ne parviennent jamais jusqu'au navigateur qui affiche la page. Le serveur web remplace le code PHP par le contenu que le code avait pour but de générer.

### *Les bases du langage*

PHP est un véritable langage de programmation. Syntactiquement il tient du Perl et du C. Il s'agit d'un langage interprété, c'est à dire que les fichiers écrits dans ce langage ne nécessitent pas de compilations ou de traitement spécifique de la part d'une application tierce. Ce que vous codez en PHP est directement utilisable sur votre serveur Web.

### **Commentaires**

L'utilité de placer des commentaires judicieux dans votre code source n'est aujourd'hui plus à démontrer, en effet il permet de préciser les algorithmes utilisés dans un langage humain, ce qui simplifie grandement la maintenance ou le débogage du programme.

Le PHP fournit trois types de commentaires :

```
/* Commentaire de type C */  
// Commentaire de type C++  
# Commentaire de type Shell ou Perl
```

## **Variables**

Le PHP est un langage non typé. En clair, cela signifie qu'une variable peut contenir indifféremment une valeur numérique ou une chaîne de caractères. Par opposition, les langages typés comme le C obligent à définir une variable d'un certain type en fonction de ce qu'elle va contenir.

Les variables en PHP sont préfixées par le caractère '\$'. Le langage PHP permet de manipuler trois genres de variables :

- Les variables scalaires :

```
$toto=5
```

- Les tableaux :

```
$tab[0]="philippe"
```

- Les tableaux associatifs :

```
$age["philippe"]=27
```

## **Types de données**

PHP fournit trois types de données primaires : nombres entiers, nombres décimaux et chaînes.

### **Entiers**

La plage des entiers dans PHP est équivalente à la portée du type long du langage C. Sur les plateformes 32 bits, les valeurs entières vont de -2147483648 à +2147483647.

PHP convertit automatiquement les valeurs supérieures en nombre décimaux.

Un entier peut être exprimé en décimal (base 10), hexadécimal (base 16) ou en octal (base 8).

Par exemple :

```
$decimal = 16;  
$hex = 0x10;  
$octal=020;
```

### **Nombres décimaux**

La plage des nombres décimaux dans PHP est équivalente à la portée du type double en C. Les valeurs, sur la plupart des plateformes, vont de 1.7E-308 à 1.7E+308.

Un nombre décimal peut être exprimé sous forme de nombre normal avec un point décimal ou en notation scientifique.

Par exemple :

```
$var = 0.017;  
$var = 17.0E-3;
```

### Chaînes

Une chaîne est une séquence de caractères.

Une chaîne peut être délimité par des guillemets simples ou doubles :

```
'Bonjour à tous'  
"C'est cool le PHP"
```

Les chaînes placées entre doubles guillemets sont sujettes aux substitutions de variables et au traitement des séquences d'échappement, alors que celles placées entre guillemets simples ne le sont pas :

```
$a = "jour!";  
/* affiche : Bon jour! */  
echo "Bon\t$a";  
/* affiche : Bon\t$a */  
echo 'Bon\t$a';
```

Tableau des séquences d'échappements

\n	Nouvelle ligne
\t	Tabulation
\r	Retour chariot
\\	Anti slash
\\$	Signe dollar

### Valeurs Booléennes

Chaque valeur possède dans PHP une valeur booléenne dite de vérité (true ou false) qui lui est associé.

C'est typiquement utilisé dans les structures de contrôle, telles que if/else ou for.

### Expressions

L'expression est la pierre maîtresse du langage. Tout ce qui possède une valeur peut être considéré comme une expression, voici quelques exemples :

```
5  
5+5  
$a  
$a==5  
sqrt(9)
```



## **Opérateurs**

Les expressions sont combinées et manipulées à l'aide d'opérateurs.

Le tableau ci dessous dresse la liste des opérateurs disponibles ainsi que leur priorité et leur associativité.

Ces opérateurs devraient vous paraître familier si vous avez une expérience du C, Java ou de Perl :

<b>Opérateurs</b>	<b>Priorité</b>	<b>Associativité</b>
!, ~, ++, --, @, (opérateurs de transtypage)	16	Droite
*, /, %	15	Gauche
+, -, .	14	Gauche
<<, >>	13	Gauche
<, <=, >=, >	12	Non associatif
==, !=	11	Non associatif
&	10	Gauche
^	9	Gauche
	8	Gauche
&&	7	Gauche
	6	Gauche
? : (opérateur conditionnel)	5	Gauche
=, +=, -=, *=, /=, %=, ^=, .=", &=",  =	4	Gauche
And	3	Gauche
Xor	2	Gauche
Or	1	Gauche

## **Structures de contrôles**

Les structures de contrôle de PHP sont très similaires à celles utilisées par le langage C. Elles sont utilisées pour contrôler un flux logique dans un script PHP.

### **If**

L'instruction if est une condition standard que l'on trouve dans la plupart des langages. Elle force l'exécution d'un code particulier si l'expression sur laquelle elle agit est vraie.

```
if (expr) {  
    instructions  
}  
elseif (expr) {  
    instructions  
}  
else {  
    instructions  
}
```

### **Switch**

L'instruction switch peut être utilisée à la place d'une longue instruction if. L'expression de chaque instruction case est comparée à l'expression switch et, si elles correspondent, le code suivant cette instruction est exécuté.

```
switch (expr) {  
    case expr:  
        instructions  
        break;  
    default:  
        instructions  
        break;  
}
```

### **While**

L'instruction while est une construction de boucle qui exécute du code de façon répétitive aussi longtemps qu'une expression particulière est vraie. L'expression est vérifiée avant chaque démarrage de la boucle.

```
while (expr) {  
    instructions  
}
```

On peut arrêter le fonctionnement d'une boucle par l'instruction break. On peut également sauter le reste des instructions et recommencer l'itération par le mot clé continue.

### Do/while

L'instruction do/while est similaire à l'instruction while, exception faite que l'expression est vérifiée en fin d'itération au lieu du début :

```
do {  
instructions  
}  
while (expr)
```

### For

une boucle for offre une construction plus complexe que la simple boucle while.

Elle contient trois expressions :

- La première est l'expression de démarrage, elle n'est exécuter qu'en début de la première itération, on l'utilise en général pour initialiser un compteur.
- La seconde est une expression conditionnelle qui contrôle l'itération de la boucle. Cette expression est vérifiée en début de chaque itération.
- La troisième expression est évaluée en fin de chaque itération. Elle est en général utilisée pour incrémenter un compteur de boucle.

```
for (start_expr; cond_expr; iter_expr) {  
instructions  
}
```

## **Fonctions**

Une fonction est une séquence d'instructions de code, dotée d'un nom, qui peut accepter des paramètres et qui retourne une valeur.

Un appel de fonction est une expression ayant une valeur; cette valeur est la valeur retournée par la fonction. PHP propose un grand nombre de fonctions internes. PHP prend également en charge les fonctions définies par l'utilisateur.

Pour définir une fonction on utilise le mot clé function :

```
function bonjour ($prenom) {  
return "Bonjour $prenom !";  
}
```

Après avoir défini une fonction, on l'appelle à l'aide d'arguments appropriés :

```
echo bonjour("Philippe");
```

On peut également définir des fonctions avec des paramètres optionnels. Il faut pour cela donner une valeur par défaut au paramètre optionnel :

```
function bonjour ($prenom="tout le monde") {  
return "Bonjour $prenom !";  
}
```

## **Programmation objet**

Une classe est une collection de variables et de fonctions qui fonctionnent avec ces variables. Une classe est définie en utilisant la syntaxe suivante :

```
<?
class Caddie {
    var $items; // Eléments de notre panier

    // Ajout de $num articles de type $artnr au panier
    function add_item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }

    // Suppression de $num articles du type $artnr du panier
    function remove_item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return TRUE;
        } else {
            return FALSE;
        }
    }
}
?>
```

L'exemple ci-dessus définit la classe Caddie qui est composée d'un tableau associatif contenant les articles du panier et de deux fonctions, une pour ajouter et une pour enlever des éléments au panier.

Note : En PHP 4, seuls les initialiseurs constants pour les variables var sont autorisés. Utilisez les constructeurs pour les initialisations variables, ou utilisant des expressions.

```
<?
/* Aucune de ces syntaxes ne fonctionnera en PHP 4 */
class Caddie {
    var $date_du_jour = date("d/m/Y");
    var $name = $firstname;
    var $owner = 'Fred ' . 'Jones';
    /* etc... */
}

/* Voici comment cela doit se faire désormais. */
class Caddie {
    var $date_du_jour;
    var $name;
    var $owner;
    function Caddie() {
        $this->date_du_jour = date("d/m/Y");
        $this->name = $GLOBALS['firstname'];
        /* etc... */
    }
}
?>
```

Les classes forment un type de variable. Pour créer une variable du type désiré, vous devez utiliser l'opérateur new.

```
<?
$cart = new Caddie;
$cart->add_item("10", 1);
?>
```

L'instruction ci-dessus crée l'objet \$cart de la class Caddie. La fonction add\_item() est appelée afin d'ajouter l'article numéro 10 dans le panier.

Une classe peut être une extension d'une autre classe. Les classes "extended" ou "derived" héritent de toutes les variables et de toutes les fonctions de la classe père plus toutes les définitions que vous rajoutez à cette classe. Cela se fait avec le mot clef "extends".

L'héritage multiple n'est pas supporté.

```
<?
class Caddie_nomme extends Caddie {
    var $owner;
    function set_owner ($name) {
        $this->owner = $name;
    }
}
?>
```

L'exemple ci-dessus définit la classe `Caddie_nomme` qui possède les mêmes variables que la classe `Caddie` et la variable `$owner` en plus, ainsi que la fonction `set_owner()`. Vous créez un panier nominatif de la même manière que précédemment, et vous pouvez alors affecter un nom au panier ou en connaître le nom. Vous pouvez de toutes les façons utiliser les mêmes fonctions que sur un panier classique.

```
<?
$ncart = new Caddie_nomme; // Création d'un panier nominatif
$ncart->set_owner ("kris"); // Affectation du nom du panier
print $ncart->owner; // Affichage du nom du panier
$ncart->add_item ("10", 1); // (héritage des fonctions de la classe
père)
?>
```

Dans les fonctions d'une classe, la variable `$this` est égale à l'objet de la classe. Vous pouvez utiliser la forme `"$this->quelquechose"` pour accéder aux fonctions ou aux variables de l'objet courant. Aussi bien depuis l'objet lui-même que de l'extérieur, vous n'avez pas besoin de `$` pour accéder aux propriétés d'un objet.

```
<?
$ncart->owner = "chris"; // pas de '$'
$ncart->$owner = "chris";
// Ceci est invalide, car $ncart->$owner équivaut à $ncart->""
$myvar = 'owner';
$ncart->$myvar = "chris";
// Ceci est valide, car $ncart->$owner équivaut à $ncart->owner
?>
```

Le constructeur est la fonction qui est appelée automatiquement par la classe lorsque vous créez une nouvelle instance d'une classe. La fonction constructeur a le même nom que la classe.

```
<?
class Auto_Caddie extends Caddie {
    function Auto_Caddie () {
        $this->add_item ("10", 1);
    }
}
?>
```

L'exemple ci-dessus définit la classe `Auto_Caddie` qui hérite de la classe `Caddie` et définit le constructeur de la classe. Ce dernier initialise le panier avec 1 article de type numéro 10 dès que l'instruction "new" est appelée. La fonction constructeur peut prendre ou non, des paramètres optionnels, ce qui la rend beaucoup plus pratique.

```
<?
class Constructor_Caddie extends Caddie {
    function Constructor_Caddie ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
// Place dans le caddie toujours la même chose...
$default_cart = new Constructor_Caddie;
// Place dans le caddie des objets différents, comme dans la réalité
$different_cart = new Constructor_Caddie ("20", 17);
?>
```

Pour les classes qui utilisent l'héritage, le constructeur de la classe père n'est pas automatiquement appelé lorsque le constructeur de la classe dérivée est appelé.

## PHP dans les pages XHTML

Vous avez peut être remarqué, que tout ce qui est affiché à l'aide de l'instruction echo figure dans le code HTML final, envoyé au navigateur.

C'est la base du fonctionnement de PHP comme langage embarqué dans HTML et également sa grande force par rapport à l'utilisation de scripts ou d'exécutables CGI. L'imbrication de PHP dans HTML, permet ainsi de créer des fichiers mixtes. On comprend alors pleinement les avantages de PHP.

Ci dessous un petit exemple de script PHP :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Page de bienvenue</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1">
  </head>
  <body>
    <h1>Bienvenue sur ce site</h1>
    <hr />
    <?
  $action=$HTTP_POST_VARS["action"];
  if ($action!="OK") {
    echo "<form action=\"bienvenue.php\" method=\"POST\">\n";
    echo "<h2>Veuillez entrer votre Prénom et votre année de
naissance</h2>\n";
    echo "<strong>Prénom : </strong><input type=\"text\"
name=\"prenom\" value=\"\" /><br/>\n";
    echo "<strong>Année de naissance : </strong><input type=\"text\"
name=\"dnaiss\" value=\"\" />
    <br/>\n";
    echo "<input type=\"submit\" name=\"action\" value=\"OK\" />";
    echo "</form>";
  }
  else {
    $prenom=$HTTP_POST_VARS["prenom"];
    $dnaiss=$HTTP_POST_VARS["dnaiss"];
    $date=date("Y"); echo "<h2>Bonjour $prenom</h2>\n";
    echo "Aujourd'hui nous somme le ",date("d/m/Y"),"<br />\n";
    echo "Tu as <strong>",$($date-$dnaiss),"</strong> ans.<br />\n";
  }
?>
    <hr />
    Copyright (c) 2003 - Philippe BOUSQUET.<br />
    Ce logiciel est sous licence <strong>Gnu General Public
License</strong>.
  </body>
</html>
```



## Les fonctions prédéfinies

Nous ne verrons ici que les fonctions dont nous allons nous servir pour la création de notre forum, pour avoir la liste complète des fonctions disponibles en PHP veuillez consulter la page : [http://dev.nexen.net/docs/php/manuel\\_toc.html#functions](http://dev.nexen.net/docs/php/manuel_toc.html#functions)

### Sessions

La gestion des sessions avec PHP est un moyen de sauver des informations entre deux accès. Cela permet notamment de construire des applications personnalisées, et d'accroître l'attrait de votre site.

Chaque visiteur qui accède à votre site se voit assigner un numéro d'identifiant, appelé plus loin "identifiant de session". Celui ci est enregistré soit dans un cookie, chez le client, soit dans l'URL.

Les sessions vous permettront d'enregistrer des variables, pour les préserver et les réutiliser tout au long des requêtes. Lorsqu'un visiteur accède à votre site, PHP vérifiera automatiquement (si `session.auto_start` est à 1) ou manuellement (explicitement avec `session_start()` ou implicitement avec `session_register()`) si une session a déjà été ouverte. Si une telle session existe déjà, l'environnement précédent sera recréé.

Toutes les variables à enregistrer seront enregistrées sur le disque à la fin de chaque requête. Les variables enregistrées mais non définies seront marquées comme tel. Lors des accès ultérieurs, elles ne seront définies que si l'utilisateur le fait.

Il y a deux modes de propagation de l'identifiant de session :

- Cookies
- Paramètre URL

Le module de session supporte les deux techniques. La méthode par cookie est optimale, mais étant donné le peu de fiabilité (les clients peuvent les refuser, ou les effacer), on ne peut pas se contenter de cette technique. La deuxième méthode place l'identifiant de session directement dans l'URL.

#### *session\_start*

Initialise les données de session

```
bool session_start (void)
```

`session_start()` crée une session (ou continue la session courante, en fonction de l'identifiant de session passé par une variable GET ou par un cookie)  
`session_start()` retourne toujours TRUE.

#### *session\_destroy*

Détruit toutes les données enregistrées d'une session

```
bool session_destroy (void)
```

`session_destroy()` détruit toutes les données associées à la session courante.  
`session_destroy()` retourne TRUE en cas de succès, et FALSE sinon.

### session\_name

Affecte et/ou retourne le nom de la session courante

```
string session_name (string name )
```

session\_name() retourne le nom de la session courante. Si name est fourni, le nom de la session changera, et prendra la valeur fournie.

Le nom de session fait référence à l'identifiant de session dans les cookies. Il ne doit contenir que des caractères alphanumériques; il doit être court et descriptif. (i.e. surtout pour les utilisateurs d'alertes de cookie). Le nom de session est remis à une valeur par défaut, enregistrées dans session.name au moment du démarrage. Ainsi, vous devez appeler session\_name() à chaque requête (et avant session\_start() ou session\_register()).

Exemple:

```
<?
# Change le nom de la session à WebsiteID
$previous_name = session_name ("WebsiteID");
echo "L'ancien nom de la session était $previous_name<P>";
?>
```

### session\_id

Affecte et/ou retourne l'identifiant de session courante

```
string session_id (string id)
```

session\_id() retourne l'identifiant de session courante. Si id est fourni, il remplacera l'identifiant courant de la session.

La constante SID peut aussi être utilisée pour retrouver le nom de la session courante et son identifiant, comme chaîne à ajouter dans les URL.

### session\_register

Enregistre une variable dans la session courante

```
bool session_register (mixed name, mixed ...)
```

session\_register() enregistre une variable avec le nom name dans la session courante.

session\_register() accepte un nombre d'arguments variable, qui peuvent être soit des chaînes représentant le nom de la variable, soit un tableau, contenant des chaînes ou d'autres tableaux (cas d'un tableau récursif).

session\_register() retourne TRUE lorsque les variables sont correctement enregistrées.

### session\_unregister

Supprime une variable dans la session courante

```
bool session_unregister (mixed name)
```

session\_unregister() supprime la variable nommée name dans la session courante.

session\_unregister() retourne TRUE lorsque la variable a été correctement supprimée de la session.

## Envoi de mail

```
bool mail (string to, string subject, string message, string  
additional_headers, string additional_parameters )
```

mail() poste automatiquement le message message à destination de to. Les destinataires multiples doivent être séparés par des virgules.

### Envoi de courrier électronique (mail)

```
<?  
  mail("rasmus@lerdorf.on.ca", "Mon Sujet", "Ligne 1\nLigne 2\nLigne  
3");  
?>
```

Le quatrième argument passé sera inséré à la fin de l'entête. Typiquement, cela permet d'insérer des entêtes supplémentaires. Les entêtes multiples doivent être séparés par des virgules.

Si le cinquième argument additional\_parameters est fourni, PHP l'utilisera dans son appel du programme d'envoi de courrier électronique. Ceci est pratique pour passer une valeur correcte à l'entête Return-Path, avec sendmail.

Note : Le cinquième paramètre a été ajouté en PHP 4.0.5.

### Envoi de eMail avec des entêtes supplémentaires.

```
<?  
  mail("nobody@aol.com", "Le sujet", $message,  
        "From: webmaster@$SERVER_NAME\nReply-  
To:webmaster@$SERVER_NAME\nX-Mailer:PHP/" . phpversion());  
?>
```

Avec le cinquième paramètre, vous pouvez ajouter d'autres paramètres de ligne de commande qui seront utilisés par le programme d'envoi de courrier. Dans l'exemple ci-dessous, l'entête Return-Path est correctement paramétré. Normalement, sendmail ajoute automatiquement l'entête X-Authentication-Warning (paramètre -f), car l'utilisateur "serveur web" n'est probablement pas un de ses utilisateurs de confiance ("trusted users"). Pour supprimer cette alerte, ajoutez l'utilisateur du serveur web dans la configuration de sendmail.

### Envoi de eMail avec des entêtes supplémentaires et un paramètre de ligne de commande supplémentaire

```
<?  
  mail("nobody@aol.com", "the subject", $message,  
        "From: webmaster@$SERVER_NAME", "-fwebmaster@$SERVERNAME");  
?>
```

## **MySQL**

Ces fonctions vous permettent d'accéder aux bases de données MySQL. Afin de pouvoir les utiliser, vous devez compiler PHP avec le support MySQL, en utilisant l'option `--with-mysql`. Si vous utilisez cette fonction sans préciser le chemin d'accès à la base MySQL, PHP utilisera les bibliothèques cliente MySQL fournies en standard. Les utilisateurs qui font tourner d'autres applications qui utilisent elles-mêmes MySQL (par exemple, PHP 3 et PHP 4 utilisés comme des modules concurrents apache, ou encore `auth-mysql`), devrait toujours spécifier le chemin jusqu'à MySQL : `--with-mysql=/path/to/mysql`. Cela va forcer PHP à utiliser les bibliothèques clientes installées par MySQL et évitera les conflits.

Plus d'informations sont disponibles à <http://www.mysql.com/>.

La documentation de MySQL est disponibles à <http://www.mysql.com/documentation/>, ainsi qu'en français chez nexen.

### *mysql\_affected\_rows*

Retourne le nombre de lignes affectées lors de la dernière requête SQL.

```
int mysql_affected_rows (resource link_identifiant )
```

`mysql_affected_rows()` retourne le nombre de lignes affectées lors de la dernière requête INSERT, UPDATE ou DELETE sur le serveur associé à l'identifiant de connexion. Si cet identifiant n'est pas précisé, `mysql_affected_rows()` utilise la dernière connexion ouverte. Si la dernière requête a échoué, `mysql_affected_rows()` retourne -1.

### *mysql\_close*

Ferme la connexion MySQL

```
bool mysql_close (ressource link_identifiant )
```

`mysql_close()` retourne TRUE en cas de succès et FALSE sinon. `mysql_close()` ferme la connexion au serveur MySQL associée à l'identifiant `link_identifiant`. Si cet identifiant n'est pas spécifié, cette commande s'applique à la dernière connexion ouverte.

### *mysql\_connect*

Ouvre une connexion à un serveur MySQL

```
int mysql_connect (string hostname, string username, string password )
```

`mysql_connect()` retourne un identifiant positif de connexion en cas de succès, et sinon FALSE.

`mysql_connect()` établit une connexion à un serveur MySQL. Tous les arguments sont optionnels, et s'ils manquent, les valeurs par défaut sont utilisées ( 'localhost', nom du propriétaire du processus, mot de passe vide).

Si un second appel à `mysql_connect()` est fait avec les mêmes arguments, PHP ne va pas ouvrir une nouvelle connexion, mais va retourner l'identifiant de la connexion déjà ouverte. Le lien sera fermé automatiquement dès que l'exécution du script sera terminée, à moins d'être fermé explicitement avec `mysql_close()`.

## Exemple MySQL connect

```
<?PHP
$link = mysql_connect ("kraemer", "marliesle", "secret")
                        or die ("Connexion impossible");
print ("Connexion réussie");
mysql_close ($link);
?>
```

### mysql\_db\_query

Envoie une requête MySQL à un serveur MySQL

```
resource mysql_db_query (string database, string query, resource
link_identifieur)
```

mysql\_db\_query() retourne un identifiant de résultat si la requête réussit et FALSE sinon. mysql\_db\_query() sélectionne une base de données et exécute une requête. Si l'identifiant de lien link\_identifieur n'est pas précisé, mysql\_db\_query() prendra par défaut la dernière connexion ouverte sur le serveur et si elle n'en trouve pas, elle tentera de se connecter, en utilisant la fonction mysql\_connect(), sans arguments.

### mysql\_errno

Retourne le numéro de message d'erreur de la dernière opération MySQL.

```
int mysql_errno (resource link_identifieur )
```

mysql\_errno() retourne le numéro de message d'erreur de la dernière opération MySQL sur la connexion courante, ou sur la connexion spécifiée avec l'option link\_identifieur. Les erreurs qui sont remontées depuis le serveur MySQL ne sont plus des alertes. A la place, il faut utiliser mysql\_errno() pour obtenir le numéro d'erreur.

### mysql\_error

Retourne le texte associée avec l'erreur générée lors de la dernière requête.

```
string mysql_error (resource link_identifieur )
```

mysql\_error() retourne le dernier message d'erreur MySQL sur la connexion courante, ou sur la connexion spécifiée avec link\_identifieur.

Les erreurs générées par MySQL ne se transforment plus en alerte. A la place, elles sont accessibles via ces fonctions :

```
<?PHP
mysql_connect("marliesle");
echo mysql_errno().": ".mysql_error()."<BR>";
mysql_select_db("nonexistentdb");
echo mysql_errno().": ".mysql_error()."<BR>";
$conn = mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno().": ".mysql_error()."<BR>";
?>
```

### mysql\_fetch\_array

Retourne une ligne de résultat sous la forme d'un tableau associatif.

```
array mysql_fetch_array (resource result_identifieur, int result_type )
```

mysql\_fetch\_array() retourne un tableau qui contient la ligne demandée, ou FALSE si il ne reste plus de ligne.

mysql\_fetch\_array() est une version étendue de mysql\_fetch\_row(). En plus d'enregistrer les données sous forme d'un tableau à indice numérique, elle peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme indices.

Si plusieurs colonnes ont le même nom, la dernière colonne aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser l'index numériques, ou faire un alias pour chaque colonne.

Il est important de souligner que mysql\_fetch\_array() N'est PAS plus lente que mysql\_fetch\_row(), tandis qu'elle ajoute un confort d'utilisation notable. .

```
<?PHP
mysql_connect($host,$user,$password);
$result = mysql_db_query("database","select * from table");
while($row = mysql_fetch_array($result)) {
    echo$row["user_id"];
    echo $row["fullname"];
}
mysql_free_result($result);
?>
```

### mysql\_fetch\_row

Retourne une ligne de résultat sous la forme d'un tableau

```
array mysql_fetch_row (resource result_identifieur)
```

mysql\_fetch\_row() retourne un tableau énuméré qui correspond à la ligne demandée, ou FALSE, si il ne reste plus de ligne.

mysql\_fetch\_row() va rechercher une ligne dans le résultat associé à l'identifiant de résultat spécifié. La ligne est retournée sous la forme d'un tableau. Chaque colonne est enregistré sous la forme d'un tableau commençant à la position 0.

Les appels suivants à mysql\_fetch\_row() retourneront la ligne suivante dans le résultat, ou FALSE si il n'y a plus de ligne disponible.

### mysql\_free\_result

Efface le résultat de la mémoire

```
int mysql_free_result (resource result_identifieur)
```

mysql\_free\_result() n'est à appeler que si vous avez peur d'utiliser trop de mémoire durant l'exécution de votre script. Toute la mémoire associée à l'identifiant de résultat sera automatiquement libérée.

### mysql\_num\_rows

Retourne le nombre de ligne d'un résultat

```
int mysql_num_rows (resource result_identifieur)
```

mysql\_num\_rows() retourne le nombre de lignes d'un résultat. Cette commande n'est valide que pour les commandes SELECT . Pour connaître le nombre de lignes retournées par INSERT, UPDATE ou DELETE, utilisez mysql\_affected\_rows().

Exemple mysql\_num\_rows() par crubel@trilizio.org

```
<?PHP
    $conn = mysql_connect("adresse de l'hote", "utilisateur", "mot de
    passe");
    mysql_select_db("base",$conn); // nécessaire si vous avez plusieurs
    bases
    $Resultfornummembers = mysql_query("SELECT * FROM Accounts",$conn);
    $NumMembers = mysql_num_rows($Resultfornummembers);
    echo"$NumMembers Membres";
?>
```

### mysql\_query

Envoie une requête SQL à un serveur MySQL

```
resource mysql_query (string query, resource link_identifieur )
```

mysql\_query() envoie une requête SQL à la base de données actuellement active sur le serveur MySQL. Si link\_identifieur n'est pas précisé, la dernière connexion est utilisée. Si aucune connexion n'a été ouverte, la fonction tentera d'en ouvrir une, avec la fonction mysql\_connect() mais sans aucun paramètre (c'est à dire avec les valeurs par défaut). mysql\_query() retourne TRUE ou FALSE, pour indiquer le succès ou l'échec de la requête. En cas de retour TRUE, la requête était valide et a pu être exécuté sur le serveur. Cela n'indique pas le nombre de ligne affectées, ou retournées. Il est parfaitement possible qu'une requête valide n'affecte aucune ligne ou ne retourne aucune ligne.

### mysql\_result

Retourne un champ d'un résultat

```
mixed mysql_result (resource result_identifieur, int row, mixed field )
```

mysql\_result() retourne le contenu d'un champ dans le résultat MySQL result\_identifieur. L'argument row peut-être un offset de champs, ou le nom d'un champ, ou le nom de la table + point + le nom du champ ("table.champ"). Si la colonne a été aliasée, utilisez de préférence l'alias.

Lorsque vous travaillez sur des résultats de grande taille, vous devriez utiliser une des fonctions qui vont rechercher une ligne entière dans un tableau. Ces fonctions sont NETTEMENT plus rapides. De plus, l'utilisation d'un offset numériques est aussi beaucoup plus rapide que de spécifier un nom littéral.

Les appels mysql\_result() ne devraient pas être mélangés avec d'autres fonctions qui travaillent aussi sur le résultat.

### mysql\_select\_db

Sélectionne une base de données MySQL

```
int mysql_select_db (string database_name, resource link_identifieur )
```

mysql\_select\_db() retourne TRUE en cas de succès, FALSE sinon.

mysql\_select\_db() change la base de données active sur la connexion représentée par link\_identifieur. Si aucun identifiant n'est spécifié, la dernière connexion est utilisée. S'il n'y a pas de dernière connexion, la fonction tentera de se connecter seule, avec mysql\_connect() et les paramètres par défaut.

Toutes les requêtes suivantes avec mysql\_query() seront faites avec la base de données active.

## **Chaines de Caractères**

### strtoupper

Met tous les caractères en majuscules

```
string strtoupper (string string)
```

strtoupper() retourne string avec tous ses caractères alphabétiques mis en majuscule.

Notez que le caractère 'alphabétique' est déterminé par la table de caractères locale. Par exemple, dans la table des caractères par défaut du "C", des caractères tels que a-umlaut (ä) ne seront pas convertis.

Exemple avec strtoupper()

```
<?PHP
$str = "Marie A Un Petit Agneau, Et Elle L'Adore";
$str = strtoupper($str);
print $str;
# Affiche : MARIE A UN PETIT AGNEAU, ET ELLE L'ADORE
?>
```

### strtolower

Met tous les caractères en minuscules

```
string strtolower (string string)
```

strtolower() retourne string avec tous ses caractères alphabétiques mis en minuscule.

### ucfirst

Met tous le premier caractère de la chaîne en majuscule

```
string ucfirst (string string)
```

ucfirst() retourne string avec le premier caractère mis en majuscule.



## X.L'application my Forum

Voici la structure de l'application :

```
myForum/
+----- imgs/
|         +----- linux.png
+----- inc/
|         +----- config.inc.php
|         +----- session.inc.php
|         +----- user.inc.php
+----- default.css
+----- editer.php
+----- inscription.php
+----- inscrit.php
+----- login.php
+----- mail.php
+----- messages.php
+----- savemessage.php
+----- sujets.php
+----- themes.php
```

### **La classe Config**

Cette classe permet de stocker dans un objet les données permettant d'accéder à la base de données, elle permet également de vérifier que la connexion au serveur MySQL est OK.

```
<?
//-----
// config.inc.php
//-----

class Config {
    var $db_host;
    var $db_name;
    var $db_user;
    var $db_pass;

    function Config() {
        $this->db_host="localhost";
        $this->db_name="myforum";
        $this->db_user="root";
        $this->db_pass="";
    }

    function getDbHost() {
        return $this->db_host;
    }

    function getDbName() {
```

```
    return $this->db_name;
}

function getDbUser() {
    return $this->db_user;
}

function getDbPass() {
    return $this->db_pass;
}

function testConnection() {
    $dbh=mysql_connect($this->getDbHost(),$this->getDbUser(),
        $this->getDbPass());

    if ($dbh==false) {
        return false;
    }
    else {
        $dbh=mysql_select_db($this->getDbName());
        if ($dbh==false) {
            return false;
        }
        else {
            return true;
        }
    }
}
}
?>
```

## La classe Session

Cette classe crée un objet permettant de gérer les sessions : récupérations de données, sauvegarde de données (notez l'utilisation de `$_SESSION` au lieu de `session_(un)register`), modifications des URL (ajout des params de session).

```
<?
//-----
// session.inc.php
//-----

class Session {
    // Constructeur
    function Session() {
        session_start();
    }

    // Support Session pour les urls
    function parseURL($url,$vars="") {
        return $url."?".session_name()."."=".session_id().
            ($vars!="?"&".$vars:"");
    }

    // Sauvegarde d'une variable
    function save($name,$value) {
        $_SESSION[$name]=$value;
    }

    // Charger une variable sauvegardée
    function load($name) {
        return $_SESSION[$name];
    }

    // Fermer la session
    function close() {
        session_destroy();
    }
}
?>
```

## La classe User

Cette classe crée un objet permettant de gérer l'utilisateur.

```
//-----
// user.inc.php
//-----

class User {
    var $user_code;
    var $user_name;
    var $user_firstname;
    var $user_email;

    // Constructeur
    function User() {
        $user_code="";
        $user_name="";
        $user_firstname="";
        $user_email="";
    }

    // Verification de la session
    function isValid($session) {
        // La session est elle active
        $this->restoreData($session);
        if ($this->getId()=="") {
            return false;
        }
        else {
            // La session est active
            return true;
        }
    }

    function connect($cuser,$pass,$cfg,$session) {
        $this->user_code=$cuser;
        $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
            $cfg->getDbPass());
        $dbh=mysql_select_db($cfg->getDbName());
        $dbr=mysql_query("SELECT id_user, name, firstname, email FROM
            users WHERE id_user='$cuser' AND passwd='$pass'");
        if (mysql_num_rows($dbr)>0) {
            $nr=mysql_fetch_array($dbr);
            if ($nr['id_user']==$this->user_code) {
                $this->user_name = $nr['name'];
                $this->user_firstname = $nr['firstname'];
                $this->user_email = $nr['email'];
                $this->saveData($session);
                return true;
            }
            else {
                return false;
            }
        }
        else {
            return false;
        }
    }
}
```

```
    }  
  }  
  
  function saveData($session) {  
    $session->save("userId",$this->user_code);  
    $session->save("userName",$this->user_name);  
    $session->save("userFirstname",$this->user_firstname);  
    $session->save("userEmail",$this->user_email);  
    return true;  
  }  
  
  function restoreData($session) {  
    $this->user_code=$session->load("userId");  
    $this->user_name=$session->load("userName");  
    $this->user_firstname=$session->load("userFirstname");  
    $this->user_email=$session->load("userEmail");  
    return true;  
  }  
  
  function getId() {  
    return $this->user_code;  
  }  
  
  function getName() {  
    return $this->user_name;  
  }  
  
  function getFirstname() {  
    return $this->user_firstname;  
  }  
  
  function getEmail() {  
    return $this->user_email;  
  }  
}  
?>
```

## La page login

Ceci est la première page sur lequel l'utilisateur va pouvoir s'authentifier.

```

<?
//-----
// login.php
//-----

require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
// Fermer une session qui serait restée ouverte
$session=new Session;
$user=new User;
if ($user->isValid($session)) {
    $session->close();
}
?>
<? echo "<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>MyForum v0.1</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
    <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
    <meta name="author" lang="fr" content="Philippe Bousquet" />
    <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
    <meta name="license" content="GNU General Public License." />
    <link rel="stylesheet" href="default.css" type="text/css" />
  </head>
  <body>
    <!-- Entete de site -->
    <div class="header">
      
      <h1>MyForum v0.1</h1>
      <hr/>
    </div>
    <!-- La partie principale -->
    <div class="main">
      <!-- Message d'erreur -->
      <div class="err">
<?
        if ($ERROR!="") {
          echo "ERREUR : $ERROR";
        }
?>
      </div>
      <!-- La boite de connection -->
      <div class="box">
        <h1>Identifiez vous</h1>
        <?echo "<form action=\"".$session->parseURL("themes.php").\" "
method="post\">"; ?>
        <div class="field">

```

```
        <label for="cuser">Code user : </label><input type="text"
id="cuser" name="cuser" value="" />
    </div>
    <div class="field">
        <label for="passwd">Mot de passe : </label><input
type="password" id="passwd" name="passwd" value="" />
    </div>
    <div class="buttons">
        <input class="button" type="submit" name="action" value="Je
me connecte" /><br />
        <input class="button" type="submit" name="action"
value="J'ai perdu mon mot de passe" />
    </div>
</form>
</div>
<div class="center">
    Si vous n'êtes pas encore inscrit veuillez cliquer <a
href="inscrit.php">ici</a>.
</div>
</div>
<!-- Le pied de page -->
<div class="footer">
    <hr/>
    Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
    Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>
```

## La page inscrit

Ceci est la page d'inscription lorsque l'utilisateur veut s'inscrire sur le forum.

```

<?
//-----
// inscrit.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
// Fermer une session qui serait restée ouverte
$session=new Session;
$user=new User;
if ($user->isValid($session)) {
    $session->close();
}
?>
<? echo "<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
    <title>MyForum v0.1</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
    <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
    <meta name="author" lang="fr" content="Philippe Bousquet" />
    <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
    <meta name="license" content="GNU General Public License." />
    <link rel="stylesheet" href="default.css" type="text/css" />
</head>
<body>
    <!-- Entete de site -->
    <div class="header">
        
        <h1>MyForum v0.1</h1>
        <hr/>
    </div>
    <!-- La partie principale -->
    <div class="main">
        <!-- Message d'erreur -->
        <div class="err">
<?
            if ($ERROR!="") {
                echo "ERREUR : $ERROR";
            }
?>
        </div>
        <!-- La boite d'inscription -->
        <div class="box">
            <h1>Inscription</h1>
            <form action="inscription.php" method="post">
                <div class="field">
                    <label for="name">Nom :</label> <? echo '<input type="text"
id="name" name="name" value="'. $name.'" />'; ?>

```



```

        </div>
        <div class="field">
            <label for="first">Prénom :</label> <? echo '<input
type="text" id="first" name="firstname" value="'. $firstname.'" />'; ?>
        </div>
        <div class="field">
            <label for="email">Email :</label> <? echo '<input
type="text" id="email" name="email" value="'. $email.'" />'; ?>
        </div>
        <div class="field">
            <label for="cuser">Code user :</label> <?echo '<input
type="text" id="cuser" name="cuser" value="'. $cuser.'" />'; ?>
        </div>
        <div class="field">
            <label for="passwd">Password :</label> <input type="password"
id="passwd" name="passwd" value="" />
        </div>
        <div class="field">
            <label for="passwd2">Confirmation :</label> <input
type="password" id="passwd2" name="passwd2" value="" />
        </div>
        <div class="buttons">
            <input class="button" type="submit" name="action"
value="s'inscrire" />
            &nbsp;&nbsp;&nbsp;<input class="button" type="reset" />
        </div>
    </form>
</div>
</div>
<!-- Le pied de page -->
<div class="footer">
    <hr/>
    Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
    Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>

```

## La partie métier inscription

Ceci est purement un acte métier qui ajoute l'utilisateur dans la base mySQL, elle redirigera ensuite vers la page de sélection des thèmes.

```
<?
//-----
// inscription.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
// Fermer une session qui serait restée ouverte
$session=new Session;
$cfg=new Config;
$user=new User;
if ($user->isValid($session)) {
    $session->close();
}
if (!$cfg->testConnection()) {
    header("Location:
inscrit.php?ERROR=Erreur%20du%20serveur%20MySQL%20!");
    exit();
}
// Controles
$name=strtoupper($name);
$firstname=strtolower($firstname);
$cuser=strtolower($cuser);
if ($name=="") {
    header("Location:
inscrit.php?ERROR=Vous%20devez%20saisir%20votre%20nom.&name=$name&first
name=$firstname&email=$email&cuser=$cuser");
    exit();
}
if ($firstname=="") {
    header("Location:
inscrit.php?ERROR=Vous%20devez%20saisir%20votre%20prénom.&name=$name&fi
rstname=$firstname&email=$email&cuser=$cuser");
    exit();
}
if ($email=="") {
    header("Location:
inscrit.php?ERROR=Vous%20devez%20saisir%20votre%20email.&name=$name&fir
stname=$firstname&email=$email&cuser=$cuser");
    exit();
}
if ($cuser=="") {
    header("Location:
inscrit.php?ERROR=Vous%20devez%20choisir%20un%20code%20user.&name=$name
&firstname=$firstname&email=$email&cuser=$cuser");
    exit();
}
if ($passwd!=$passwd2) {
    header("Location:
inscrit.php?ERROR=Mot%20de%20passe%20incorrect,%20pensez%20à%20la%20con
firmation.&name=$name&firstname=$firstname&email=$email&cuser=$cuser");
    exit();
}
```

```
}
// Verification de l'inexistence du code user
$dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),$cfg-
>getDbPass());
$dbh=mysql_select_db($cfg->getDbName());
$dbr=mysql_query("SELECT * FROM users WHERE id_user='$cuser'");
if (mysql_num_rows($dbr)>0) {
    header("Location:
inscrit.php?ERROR=Code%20user%20déjà%20utilisé,%20veuillez%20en%20chois
ir%20un%20autre.&name=$name&firstname=$firstname&email=$email&cuser=");
    exit();
}
// Les controles sont ok inscription de l'utilisateur
$requete="INSERT INTO `users` ( `id_user` , `passwd` , `name` ,
`firstname` , `email` ) VALUES ('$cuser', '$passwd', '$name',
'$firstname', '$email');";
$dbr=mysql_query($requete);
// On se connecte
header("Location: themes.php?cuser=".$cuser."&passwd=".$passwd);
?>
```

## La partie métier mail

Ceci est purement un acte métier qui envoie à l'utilisateur son mot de passe par email, si celui ci l'a oublié.

```

<?
//-----
// mail.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$cfg=new Config;
$user=new User;
// Controles
if ($cuser=="") {
    header("Location:
login.php?ERROR=Vous%20devez%20saisir%20votre%20code%20user.");
    exit();
}
$requete="SELECT passwd, email FROM users WHERE id_user='$cuser'";
$dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
    $cfg->getDbPass());
$dbh=mysql_select_db($cfg->getDbName());
$dbr=mysql_query($requete);
if (mysql_num_rows($dbr)==0) {
    header("Location: login.php?ERROR=Cet%20
    utilisateur%20est%20inconnu,%20veuillez%20vous%20inscrire.");
    exit();
}
$enr=mysql_fetch_array($dbr);
$text="Bonjour vous avez demandé que l'on vous envoie votre mot de
    passe myForum.\n\nVotre mot de passe :".$enr['passwd'].
    "\n\nCordialement.";
if (mail($enr['email'], "mot de passe myforum", $text,
    "From: webmaster@$SERVER_NAME\n
    Reply-To:webmaster@$SERVER_NAME\nX-Mailer:PHP/".
    phpversion())) {
    header("Location: login.php?ERROR=Votre%20mot%20
    de%20passe%20vous%20à%20été%20envoyé%20par%20email.");
}
else {
    header("Location: login.php?ERROR=Il%20y%20a%20eu%20un%20
    problème%20lors%20de%20l'envoi%20de%20mail.");
}
?>

```

## La page de sélection de thème

Cette page permet à l'utilisateur de sélectionner le thèmes qu'il veut consulter.

```

<?
//-----
// themes.php
//-----
if ($action=="J'ai perdu mon mot de passe") {
    require("mail.php");
    exit();
}
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$cfg=new Config;
$user=new User;
if (!$cfg->testConnection()) {
    header("Location: login.php?ERROR=Erreur%20du%20
    serveur%20MySQL%20!");
    exit();
}
if (!$user->isValid($session) && !$user->connect($cuser,$passwd,
    $cfg,$session)) {
    header("Location: login.php?ERROR=Mot%20de%20passe%20incorrect,
    %20ou%20utilisateur%20inconnu.");
    exit();
}
?>
<? echo '<?xml version="1.0" encoding="ISO-8859-1"?>' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
    <title>MyForum v0.1</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
    <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
    <meta name="author" lang="fr" content="Philippe Bousquet" />
    <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
    <meta name="license" content="GNU General Public License." />
    <link rel="stylesheet" href="default.css" type="text/css" />
</head>
<body>
    <!-- Entete de site -->
    <div class="header">
        
        <h1>MyForum v0.1</h1>
        <hr/>
    </div>
    <!-- La partie principale -->
    <div class="main">
        <div class="pwd">
            <? echo $user->getId(); ?> &gt;&gt; <? echo "<a
href=\"\".$session->parseURL("login.php").\" title=\"quitter\">"; ?

```

```

>Quitter</a>
</div>
<!-- La liste des themes -->
<table>
  <caption>MyForum : Liste des thèmes</caption>
  <thead>
    <tr class="title"><th>Thème</th><th>Description</th><th>
Sujets </th><th>Date</th></tr>
  </thead>
  <tbody>
<?
  $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
                    $cfg->getDbPass());
  $dbh=mysql_select_db($cfg->getDbName());
  $dbr=mysql_query("SELECT id_theme, name, description FROM
                    themes;");
  while ($enr=mysql_fetch_array($dbr)) {
    echo '<tr>';
    echo "<td class=\"left\"><a href=\"".$session-
>parseURL("sujets.php","id_theme=".$enr['id_theme']).\">".$enr['name']
.</a></td>";
    echo "<td class=\"left\"><a href=\"".$session-
>parseURL("sujets.php","id_theme=".$enr['id_theme']).\">".$enr['descri
ption'].</a></td>";
    $dbr2=mysql_query("SELECT COUNT(*) as nb_sujets,
                      MAX(date) as date FROM subjects where
                      id_theme='".$enr['id_theme']."'");
    $enr2=array();
    if (mysql_num_rows($dbr2)>0) {
      $enr2=mysql_fetch_array($dbr2);
    }
    else {
      $enr2['nb_sujets']="0";
      $enr2['date']="0000-00-00 00:00:00";
    }
    echo "<td class=\"right\">".$enr2['nb_sujets']."</td>";
    echo "<td class=\"right\">".$enr2['date']."</td>";
    echo '</tr>';
  }
?>
  </tbody>
</table>
</div>
<!-- Le pied de page -->
<div class="footer">
  <hr/>
  Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
  Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>

```

## La page de sélection du sujet

Cette page permet à l'utilisateur de sélectionner le sujet sur lequel il veut consulter les messages, ou en créer un nouveau.

```

<?
//-----
// sujets.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$cfg=new Config;
$user=new User;
if (!($user->isValid($session))) {
    header("Location: login.php?ERROR=Utilisateur%20inconnu,%20
        ou%20la%20session%20a%20expirée.");
    exit();
}
if (!($cfg->testConnection())) {
    header("Location: login.php?ERROR=Erreur%20du%20serveur%20
        MySQL%20!");
    exit();
}
?>
<? echo '<?xml version="1.0" encoding="ISO-8859-1"?>' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <title>MyForum v0.1</title>
        <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
        <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
        <meta name="author" lang="fr" content="Philippe Bousquet" />
        <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
        <meta name="license" content="GNU General Public License." />
        <link rel="stylesheet" href="default.css" type="text/css" />
    </head>
    <body>
        <!-- Entete de site -->
        <div class="header">
            
            <h1>MyForum v0.1</h1>
            <hr/>
        </div>
        <!-- La partie principale -->
        <div class="main">
            <div class="pwd">
                <? echo $user->getId(); ?> &gt;&gt;
                <? echo "<a href=\"\".$session->parseURL("login.php").
                    \"\" title=\"quitter\">"; ?>Quitter</a> ::
                <? echo "<a href=\"\".$session->parseURL("themes.php").
                    \"\" title=\"retour aux thèmes\">"; ?>Thèmes</a>
            </div>

```

```

<!-- La liste des themes -->
<table>
  <caption>
    <?
      $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
                        $cfg->getDbPass());
      $dbh=mysql_select_db($cfg->getDbName());
      $dbr=mysql_query("SELECT name FROM themes WHERE
                      id_theme='$id_theme'");
      $enr=mysql_fetch_array($dbr);
      echo $enr['name'];
    ?>
    &nbsp;   :: Liste des sujets
  </caption>
  <thead>
    <tr class="title">
      <th>Auteur</th>
      <th>Titre</th>
      <th>Rep.</th>
      <th>Date</th>
    </tr>
  </thead>
  <tbody>
<?
  $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
                    $cfg->getDbPass());
  $dbh=mysql_select_db($cfg->getDbName());
  $dbr=mysql_query("SELECT id_subject, title, date, id_user FROM
                  subjects WHERE id_theme='$id_theme' ORDER BY date DESC;");
  while ($enr=mysql_fetch_array($dbr)) {
    echo '<tr>';
    echo "<td class=\"left\">".$enr['id_user'].</td>";
    echo "<td class=\"left\">
          <a href=\"".$session->parseURL("messages.php",
          "id_theme=".$id_theme."&id_subject=".$enr['id_subject']).
          \">".$enr['title'].</a></td>";
    $dbr2=mysql_query("SELECT COUNT(*) as nb_rep FROM messages
                      WHERE id_subject='".$enr['id_subject']."'");
    $enr2=array();
    if (mysql_num_rows($dbr2)>0) {
      $enr2=mysql_fetch_array($dbr2);
      $enr2['nb_rep']--;
    }
    else {
      $enr2['nb_rep']=0;
    }
    echo "<td class=\"right\">".$enr2['nb_rep'].</td>";
    echo "<td class=\"right\">".$enr['date'].</td>";
    echo "</tr>\n";
  }
?>

  <!-- Le bouton nouveau -->
  <tr class="row"><td colspan="4" class="center">
    <? echo "<form action=\"".$session->parseURL("editer.php",
          "id_theme=".$id_theme).\" method=\"post\"> ?>
      <input class="button" type="submit" name="action"
value="Nouveau" />
    </form>
  </td></tr>
  </tbody>
</table>

```



```
</div>
<!-- Le pied de page -->
<div class="footer">
  <hr/>
  Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
  Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>
```

## La page des messages

Cette page affiche la liste des messages concernant un sujet, l'utilisateur pourra également spécifier qu'il veut répondre sur ce sujet.

```

<?
//-----
// messages.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$cfg=new Config;
$user=new User;
if (!$user->isValid($session)) {
    header("Location: login.php?ERROR=Utilisateur%20inconnu,%20
        ou%20la%20session%20a%20expirée.");
    exit();
}
if (!$cfg->testConnection()) {
    header("Location: login.php?ERROR=Erreur%20du
        %20serveur%20MySQL%20!");
    exit();
}
?>
<? echo '<?xml version="1.0" encoding="ISO-8859-1"?>' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <title>MyForum v0.1</title>
        <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
        <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
        <meta name="author" lang="fr" content="Philippe Bousquet" />
        <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
        <meta name="license" content="GNU General Public License." />
        <link rel="stylesheet" href="default.css" type="text/css" />
    </head>
    <body>
        <!-- Entete de site -->
        <div class="header">
            
            <h1>MyForum v0.1</h1>
            <hr/>
        </div>
        <!-- La partie principale -->
        <div class="main">
            <div class="pwd">
                <? echo $user->getId(); ?> &gt;&gt;
                <? echo "<a href=\"". $session->parseURL("login.php").
                    "\" title=\"quitter\">"; ?>Quitter</a> ::
                <? echo "<a href=\"". $session->parseURL("themes.php").
                    "\" title=\"retour aux thèmes\">"; ?>Thèmes</a> ::
                <? echo "<a href=\"". $session->parseURL("sujets.php",

```

```

        "id_theme=".$id_theme)."\title=\"retour
        aux sujets\">"; ?>Sujets</a>
</div>
<!-- La liste des themes -->
<table>
  <caption>
    <? $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
                        $cfg->getDbPass());
    $dbh=mysql_select_db($cfg->getDbName());
    $dbr=mysql_query("SELECT name FROM themes WHERE
                    id_theme='$id_theme'");
    $enr=mysql_fetch_array($dbr);
    echo $enr['name']. " : ";
    $dbr=mysql_query("SELECT title FROM subjects WHERE
                    id_subject='$id_subject'");
    $enr=mysql_fetch_array($dbr);
    echo $enr['title']; ?>
  </caption>
  <tbody>
<? $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
                    $cfg->getDbPass());
$dbh=mysql_select_db($cfg->getDbName());
$dbr=mysql_query("SELECT id_user,title,text, date, id_user FROM
                messages WHERE id_subject='$id_subject' ORDER
                BY date, id_message;");
while ($enr=mysql_fetch_array($dbr)) {
  echo '<tr class="row">';
  echo '<th class="left">'.$enr['id_user'].'</th>';
  echo '<th class="left">'.$enr['title'].'</th>';
  echo '<th class="right">'.$enr['date'].'</th>';
  echo '</tr>'. "\n";
  echo '<tr class="row">';
  echo '<td colspan="3" class="left"><pre>'.$enr['text'].
    '</pre></td>';
  echo '</tr>'. "\n";
}
?>

  <!-- Le bouton répondre -->
  <tr class="row"><td colspan="3" class="center">
    <? echo '<form action="'. $session->parseURL("editer.php",
        "id_theme=".$id_theme."&id_subject=".$id_subject).
        '" method="post">'; ?>
      <input class="button" type="submit" name="action"
value="Répondre" />
    </form>
  </td></tr>
  <tbody>
</table>
</div>
<!-- Le pied de page -->
<div class="footer">
  <hr/>
  Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
  Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>

```

## La page de création de message

Cette page permet soit de répondre sur un sujet donné, soit de créer son propre message sur un nouveau sujet.

```

<?
//-----
// editer.php
//-----
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$cfg=new Config;
$user=new User;
if (!$user->isValid($session)) {
    header("Location: login.php?ERROR=Utilisateur%20inconnu,%20
        ou%20la%20session%20a%20expirée.");
    exit();
}
?>
<? echo '<?xml version="1.0" encoding="ISO-8859-1"?>' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <title>MyForum v0.1</title>
        <meta http-equiv="content-type" content="text/html; charset=ISO-
8859-1" />
        <meta name="keywords" lang="fr" content="forum, discussion,
logiciel, libre, php, mysql, xhtml, css" />
        <meta name="author" lang="fr" content="Philippe Bousquet" />
        <meta name="copyright" content="&copy; 2004 Philippe Bousquet." />
        <meta name="license" content="GNU General Public License." />
        <link rel="stylesheet" href="default.css" type="text/css" />
    </head>
    <body>
        <!-- Entete de site -->
        <div class="header">
            
            <h1>MyForum v0.1</h1>
            <hr />
        </div>
        <!-- La partie principale -->
        <div class="main">
            <div class="pwd">
                <? echo $user->getId(); ?> &gt;&gt;
                <? echo "<a href=\"". $session->parseURL("login.php").
                    \"\" title=\"quitter\">"; ?>Quitter</a> ::
                <? echo "<a href=\"". $session->parseURL("themes.php").
                    \"\" title=\"retour aux thèmes\">"; ?>Thèmes</a> ::
                <? echo "<a href=\"". $session->parseURL("sujets.php",
                    "id_theme=\".$id_theme).\"\"title=\"retour
                    aux sujets\">"; ?>Sujets</a>
                <?
                    if ($id_subject!="") {

```

```

        echo ' :: <a href="'. $session->parseURL("messages.php",
            "id_theme=". $id_theme."&id_subject=". $id_subject).
            "' title="retour sur les messages">Messages</a>';
    }
    ?>
</div>
<!-- Message d'erreur -->
<div class="err">
<?
    if ($ERROR!="") {
        echo "ERREUR : $ERROR";
    }
?>
</div>
<!-- Répondre au message -->
<div class="box">
<?
    $titre="";
    if ($id_subject!="") {
        $dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
            $cfg->getDbPass());
        $dbh=mysql_select_db($cfg->getDbName());
        $dbr=mysql_query("SELECT title FROM subjects WHERE
            id_subject=' $id_subject'");
        $enr=mysql_fetch_array($dbr);
        $titre="RE: ". $enr['title'];
    }
?>
<h1>Répondre au message</h1>
<? echo '<form action="'. $session->parseURL("savemessage.php",
    "id_theme=". $id_theme."&id_subject=". $id_subject).
    "' method="post">'; ?>
    <div class="field2"><label for="title">Titre : </label><br/>
<? echo '<input type="text" id="title" name="titre"
    value="'. $titre.'" size="50" />';?></div>
    <div class="field2"><label for="mess">Message : </label><br/>
<textarea cols="50" id="mess" rows="12" name="message">
    Saisissez votre texte ici...
</textarea></div>
    <div class="buttons">
    <input class="button" type="submit" name="action"
        value="Envoyer" />
    &nbsp;<input class="button" type="reset" />
    </div>
</form>
</div>
<!-- La liste des themes -->
<?
if ($id_subject!="") {
echo '<table class="old" summary="Devel : Comment
    réaliser un forum en PHP ?">';
echo '<caption>Devel : Comment réaliser un forum en
    PHP ?</caption>';
echo '<tbody>';
$dbh=mysql_connect($cfg->getDbHost(),$cfg->getDbUser(),
    $cfg->getDbPass());
$dbh=mysql_select_db($cfg->getDbName());
$dbr=mysql_query("SELECT id_user,title,text, date, id_user
    FROM messages WHERE id_subject=' $id_subject'
    ORDER BY date DESC, id_message DESC;");
while ($enr=mysql_fetch_array($dbr)) {

```

```
        echo '<tr class="row">';
        echo '<th class="left">'.$enr['id_user'].'</th>';
        echo '<th class="left">'.$enr['title'].'</th>';
        echo '<th class="right">'.$enr['date'].'</th>';
        echo '</tr>'. "\n";
        echo '<tr class="row">';
        echo '<td colspan="3" class="left"><pre>'.
            $enr['text'].'</pre></td>';
        echo '</tr>'. "\n";
    }
    echo '</tbody>';
    echo '</table>';
}
?>
</div>
<!-- Le pied de page -->
<div class="footer">
    <hr/>
    Copyright &copy; 2003-2004 Philippe BOUSQUET<br />
    Ce logiciel est sous licence Gnu Genral Public License
</div>
</body>
</html>
```

## La partie métier de sauvegarde d'un message

Cette partie permet de créer un sujet en base MySQL (si besoin), ainsi que d'enregistrer le message saisi par l'utilisateur.

```

<?
require("inc/config.inc.php");
require("inc/session.inc.php");
require("inc/user.inc.php");
$session=new Session;
$config=new Config;
$user=new User;
if (!$user->isValid($session)) {
    header("Location: login.php?ERROR=Utilisateur%20inconnu,%20
ou%20la%20session%20a%20expirée.");
    exit();
}
if ($titre=="") {
    header("Location: ".$session->parseURL("editer.php",
"ERROR=Vous%20devez%20saisir%20un%20titre.
&id_theme=".$id_theme."&id_subject=".$id_subject));
    exit();
}
if ($id_subject=="") {
    $requete="INSERT INTO `subjects` (`id_subject`, `title`, `date`,
`id_user`, `id_theme`) VALUES ('', '$titre',
'0000-00-00 00:00:00', '". $user->getId()."',
'$id_theme');";
    $dbh=mysql_connect($config->getDbHost(),$config->getDbUser(),
$config->getDbPass());
    $dbh=mysql_select_db($config->getDbName());
    $dbr=mysql_query($requete);
    $dbr=mysql_query("SELECT MAX(id_subject) as id_subject FROM
subjects WHERE id_theme='$id_theme'");
    $enr=mysql_fetch_array($dbr);
    $id_subject=$enr['id_subject'];
}
$date=date("Y-m-d")." ".date("H:i:s");
$requete="INSERT INTO `messages` (`id_message`, `title`, `text`,
`date`, `id_user`, `id_subject`) VALUES ('', '$titre',
'$message', '$date', '". $user->getId()."',
'$id_subject');";
if ($dbh=="") {
    $dbh=mysql_connect($config->getDbHost(),$config->getDbUser(),
$config->getDbPass());
    $dbh=mysql_select_db($config->getDbName());
}
$dbr=mysql_query($requete);
$requete="UPDATE `subjects` SET `date` = '$date' WHERE
`id_subject` = '$id_subject'";
$dbr=mysql_query($requete);
header("Location: ".$session->parseURL("messages.php","id_theme="
$id_theme."&id_subject=".$id_subject));
?>

```

## XI.Références

Pour plus d'informations sur MySQL :

- <http://www.mysql.com>
- <http://dev.nexen.net/docs/mysql/>

Pour plus d'informations sur phpMyAdmin :

- <http://www.phpmydamin.org>

Pour plus d'informations sur XHTML et CSS :

- <http://www.w3.org/> (Le site officiel du W3C)
- <http://openweb.eu.org/> (Pour les standards du web. En français)

Pour plus d'informations sur PHP :

- <http://dev.nexen.net/docs/php/>