

Premiers pas en PHP

Pour faciliter la programmation et le test des pages php il est préférable de pouvoir disposer :

- d'un éditeur de page : PHP Edit
- d'un outil de simulation du serveur pour pouvoir tester les pages réalisées : EasyPhp

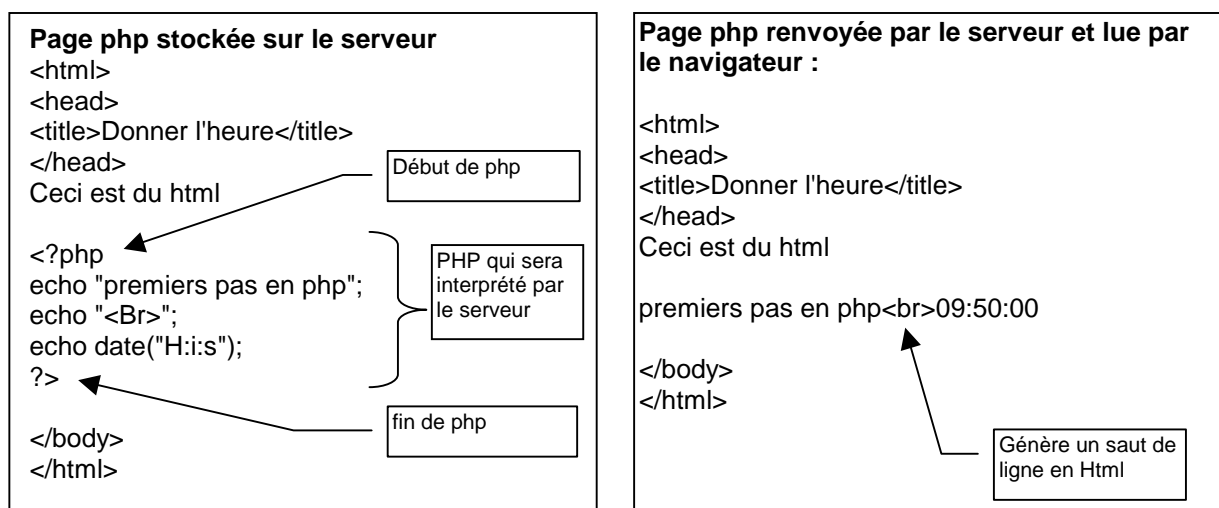
*Les pages réalisées sous PhpEdit sont copiées dans le dossier c:\program files\easyphp\www
Depuis un navigateur on saisit l'adresse de la page à visualiser : <http://localhost/page1.php>
(on s'adresse au serveur easyphp -<http://localhost> - qui pointe par défaut vers ce dossier)*

Ces logiciels sont téléchargeables gratuitement depuis Internet.

PHP est un langage dédié exclusivement à la production de pages HTML. Le code Php n'est pas interprété par le navigateur de l'utilisateur final(à la différence de javascript), mais par le serveur qui renvoie au navigateur une page au format html.

1 Ecrire une première page contenant du php

1.1 Exemple d'une page renvoyant l'heure



Le Séparateur d'instructions est constitué par un point-virgule

L'instruction **echo** permet d'afficher l'information souhaitée à l'écran (elle supporte les balises html qui sont insérées comme du texte entre des guillemets).

Comment faire, si l'on souhaite pouvoir afficher des guillemets dans un texte affiché à l'écran ?

Il faut placer les caractères \"

Ex : `echo "Ce personnage est dénommé \"Maurice\"";`

L'intérêt du php va surtout résider dans la possibilité de rendre les pages plus dynamiques en augmentant l'interactivité avec les utilisateurs.

Comme dans tout langage, on utilisera des variables pour stocker les données.

1.2 Les variables

Exemple :

```
<?
//affectation d'une variable
$salutations = "Veuillez agréer, Monsieur, l'expression de mon profond respect";
//affiche le contenu de la variable
echo $salutations;
?>
```

Les remarques qui servent à documenter le programme sont précédées de // afin qu'elles ne soient pas interprétées par le serveur.

Les variables sont donc composées du symbole \$ suivi du nom de variable. Elles peuvent contenir de chaînes de caractères, des entiers ou des nombres réels (attention, les décimales sont repérées par un point).

```
$prixht=12;
$prixttc=$prixht*1.196;
// après remise prixttc peut être calculé comme suit
$prixttc=$prixttc*0.9;
echo "<br> $prixttc";
```

1.3 Quelques structures de contrôle

1.3.1 La boucle pour...

La page php

```
<html>
<Body>
<?php
for($i=1;$i<=6;$i=$i+1)
{
    echo "<font size=$i>";
    echo "Une boucle en PHP";
    echo "<br>";
}
?>
</Body>
</html>
```

Attention accolades

La page envoyée par le serveur

```
<html>
<Body>
<font size=1>Une boucle en PHP<br>
<font size=2>Une boucle en PHP<br>
<font size=3>Une boucle en PHP<br>
<font size=4>Une boucle en PHP<br>
<font size=5>Une boucle en PHP<br>
<font size=6>Une boucle en PHP<br>
</Body>
/html>
```

Le résultat lu dans un navigateur

Une boucle en PHP
Une boucle en PHP
Une boucle en PHP
Une boucle en PHP
Une boucle en PHP
Une boucle en PHP

1.3.2 La boucle tant que...

Cette boucle fait la même chose que la précédente

```
while ( $j <= '6' ) // la boucle s'arrêtera lorsque la variable $i sera supérieure à 6
{
    echo "<font size=$j>";
    echo "Une boucle en PHP";
    echo "<br>";
    $j = $j+1; // ajouter 1 à chaque tour de boucle
}
```

1.3.3 La condition si, alors, sinon..

```
$jeu = 421;
if($jeu == 421)
{
    echo"gagné";
}
elseif($jeu >400 AND $jeu<430)
{
    echo"on en n'est pas loin";
}
else
{
    echo "on est loin du compte";
}
```

Le elseif est facultatif, il permet d'emboîter plusieurs conditions

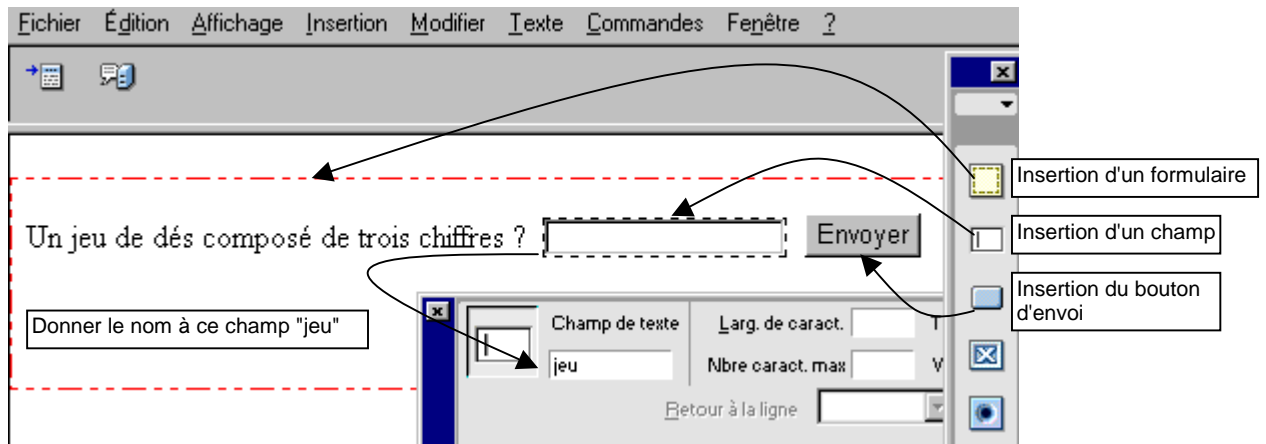
Tester cette page php et l'enregistrer sous le nom condition.php

Ce programme n'a d'intérêt que si l'on pose une question à l'utilisateur et qu'on lui offre la possibilité de soumettre une donnée pour la variable \$jeu. Il est nécessaire d'utiliser pour cela un formulaire.

2 Utilisation d'un formulaire pour acquérir une donnée saisie par l'utilisateur

2.1 Un premier exemple

Reprenons l'exemple du test précédent et construisons une page html avec Dreamweaver par exemple :



Enregistrer la page sous le nom jeu.htm

Visualiser le code html généré et modifier l'action associée à ce formulaire

```
<body bgcolor="#FFFFFF">
<form method="post" action="condition.php">
  <p>Un jeu de dés composé de trois chiffres ?
    <input type="text" name="jeu">
    <input type="submit" name="submit" value="Envoyer">
  </p>
  <p>  </p>
</form>
```

La donnée contenue dans la variable jeu sera envoyée à la page condition.php

La page condition.php doit être modifiée : il est nécessaire de supprimer l'affectation de la variable jeu (\$jeu=421) puisque celle-ci sera fournie par le formulaire appelant (jeu.htm)

Tester les échanges entre la page jeu.htm (qui envoie la valeur contenue dans la variable jeu du formulaire) et la page condition.php. Adresse à saisir dans le navigateur : <http://localhost/jeu.htm>

2.2 Application : Utilisation d'un formulaire pour l'envoi de courriels

2.2.1 Le principe d'envoi d'un courriel en php

Lorsque l'on souhaite pouvoir récupérer dans sa boîte aux lettres l'avis de nombreux utilisateurs, il faut leur faciliter la tâche. En effet, le recours à un logiciel de messagerie (en ligne ou hors ligne peut être dissuasif).

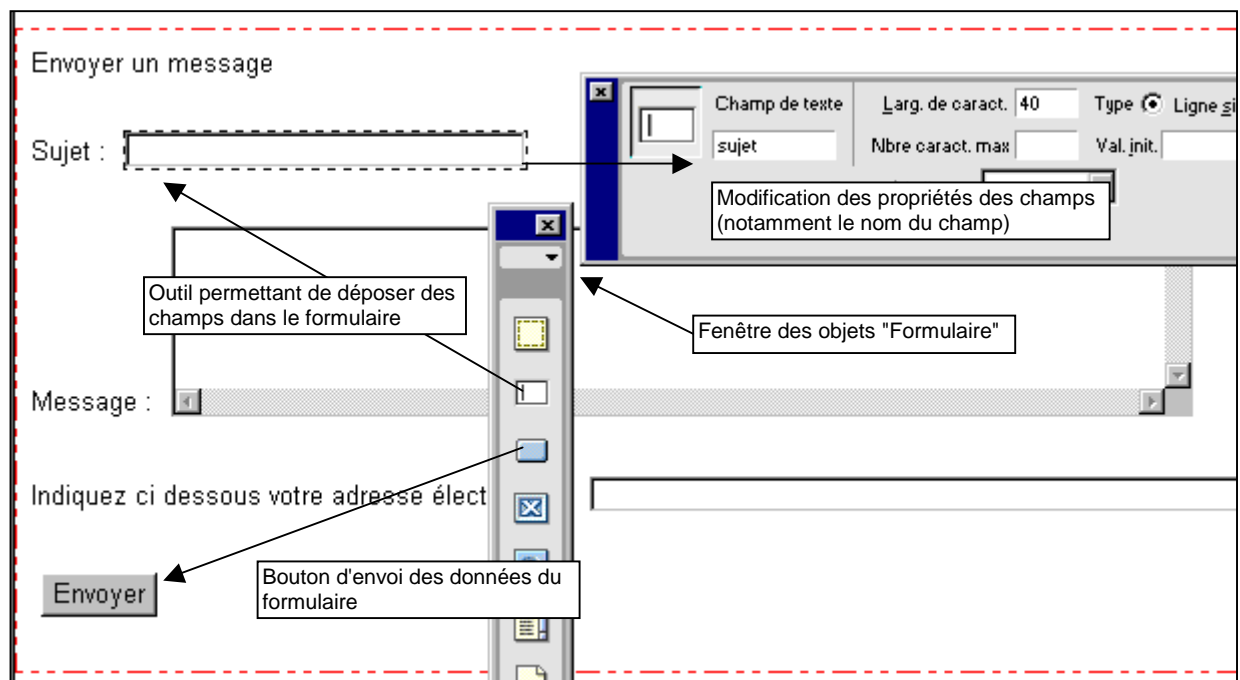
Voici la syntaxe php pour envoyer de l'information à une adresse mél :

```
<?php
$to = 'leon.marc@reims.iufm.fr' ; //l'email du destinataire
$sujet = 'Ceci est un mail de test'; //le sujet
$message = 'Bonjour, ceci est un envoi de mail test'; //le message
$from = 'From: votre@email.com'; //l'email de celui qui envoi le message
//maintenant on envoi le mail avec la fonction php
mail($to,$sujet,$message,$from);
?>
```

Mais cette syntaxe conduit à envoyer toujours le même contenu informatif ! Il convient donc de permettre à l'utilisateur final consultant une page sur le web de renseigner des champs de formulaires et de poster les informations.

2.2.2 Création du formulaire avec un éditeur de pages web habituel

Ici sous Dreamweaver



Voici le code généré en Html :

```
<form method="post" action=" ">
<p><font face="Arial, Helvetica, sans-serif" size="2">Envoyer un message</font></p>
<p><font face="Arial, Helvetica, sans-serif" size="2">Sujet :
  <input type="text" name="sujet" size="40">
</font></p>
<p><font face="Arial, Helvetica, sans-serif" size="2">Message :
  <textarea name="message" cols="100" rows="5"></textarea>
</font></p>
<p><font face="Arial, Helvetica, sans-serif" size="2">Indiquez ci dessous votre
  adresse &eacute;lectronique
  <input type="text" name="from" size="100">
</font></p>
<p> <font face="Arial, Helvetica, sans-serif" size="2">
  <input type="submit" name="submit" value="Envoyer">
</font></p>
</form>
```

La page html est enregistrée sous le nom mail.htm

Il faut désormais faire en sorte que les données soient envoyées à un programme qui s'exécutera sur le serveur et envoie ces informations au destinataire.

2.2.3 Le formulaire envoie ses données à une page php

Intéressons-nous à la première ligne de ce formulaire et modifions l'action :

```
<form method="post" action="poster.php">
```

Les instructions de la page php seront exécutées sur le serveur.

Sous le bloc-notes de Windows ou sous EasyPhp, il convient de créer cette page **poster.php** :

```
<?php
//affectation de la variable recevant le nom du destinataire
$to='leon.marc@reims.iufm.fr';
$from='from:$from';
//envoi de mail en utilisant les paramètres envoyés par la page html
mail($to,$sujet,$message,$from);
//affichage du message confirmant l'envoi
echo "Votre message a été correctement envoyé."
?>
```

Lorsque l'utilisateur clique sur le bouton "envoyer", les données contenues dans les champs sont envoyées au programme **poster.php** qui est exécuté sur le serveur, celui-ci génère un mél à l'attention de leon.marc@reims.iufm.fr et l'envoie.

Par ailleurs, ce même programme php affiche une page web sur le navigateur de l'utilisateur affichant **"Votre message a été correctement envoyé"**.

Pour tester ces deux pages (html et php), il convient de les télécharger sur un site disposant d'un serveur php (free, freesurf...)

Mais il est possible de les tester localement après les avoir copiées dans le dossier c:\program files\easyphp\www

Depuis un navigateur saisir l'adresse : <http://localhost/mail.htm> (on s'adresse au serveur easyphp - <http://localhost> - qui pointe par défaut vers ce dossier)

On obtient cependant un message d'erreur car le message n'a pu être envoyé depuis l'outil de simulation du serveur Apache local.

Attention, pour éviter les problèmes de spamming, certains fournisseurs d'espace d'hébergement refusent ce type de mail via des formulaires.

3 Comment constituer un fichier stocké sur le serveur ?

Il peut être utile de stocker dans un fichier texte, des informations saisies par l'utilisateur depuis une page web.

Exemple : ouvrir la page "controle.htm" (contexte : PLC2 STI voir la fiche de travail correspondante)

Nom :

Mentionnez les actions ou les états

0

1

2

Envoyer Rétablir

Dans cette page, le formulaire contient 6 champs nommés de haut vers bas : nom, champ1, champ2, champ3, champ4, champ5.

Les données du formulaire doivent être envoyées à une page ajoutresult.php qui aura pour tâche de stocker les données dans le fichier texte exercice1.txt stocké sur le serveur.

Voici la boîte à outils permettant de créer un fichier texte sur le serveur.

Ecriture dans le fichier

```
<?php
$fichier = "nom du fichier texte";
$ouvert_ecrit = fopen($fichier,'a+'); //ouvre le fichier en lecture et écriture
fwrite($ouvert_ecrit,"nom des champs \n");
// écriture des données de chaque champ dans une ligne du fichier
//le fichier est automatiquement créé s'il n'existe pas
fclose($ouvert_ecrit); //referme le fichier
?>
```

En local, l'écriture dans un fichier ne pose pas de problème particulier autre que celui de la syntaxe php. Mais à distance, s'ajoutent des problèmes de droits sur les fichiers. A cet égard consultez la remarque en bas de la page 11.

Lecture des données du fichier via le navigateur

```
<?
$fichier = 'exercice1.txt'; // déclare le nom du fichier à ouvrir
$taille = filesize($fichier); //donne la taille du fichier
$ouvert_lect = fopen($fichier,'r'); //ouverture du fichier en lecture seule
while(!feof($ouvert_lect)) //balayage jusqu'à la fin du fichier
{
$ligne = fgets($ouvert_lect,$taille); //lecture du fichier et stockage dans la variable ligne
echo $ligne."<br>"; //affiche la ligne à l'écran, et passe à la ligne suivante
}
fclose($ouvert_lect); //fermeture fichier en lecture
?>
```

Les données affichées ne sont guère mises en forme, mais pour cet exercice, elles seront traitées sous Excel via un programme en visual basic. (Cf. Formation Visual Basic Application pour Excel).

4 Comment renseigner une base de données à distance ?

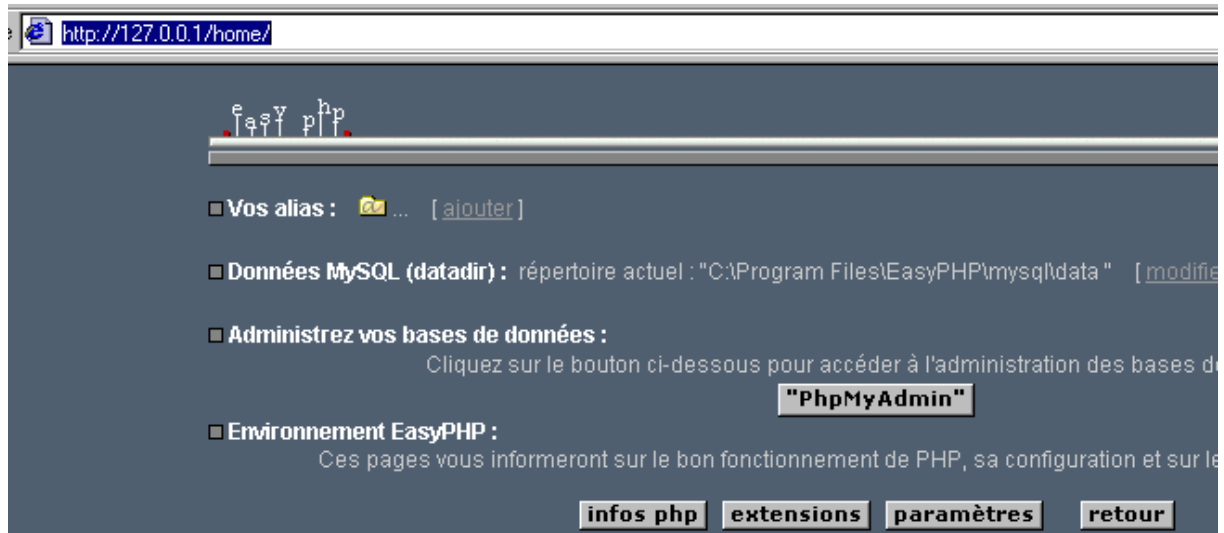
La manipulation des données est facilitée lorsqu'on les stocke dans une table.

Php permet d'accéder en lecture et en écriture à des **bases de données SQL** (langage d'interrogation structuré que l'on retrouve sous MSAccess, Oracle...)

Disons simplement qu'une base de données contient une ou plusieurs tables.

Le logiciel **Mysql** permet de gérer des bases de données, il est installé sur les serveurs de certains fournisseurs d'accès (free, freesurf ...) et il est installé localement sur votre machine si vous avez installé EasyPhp.

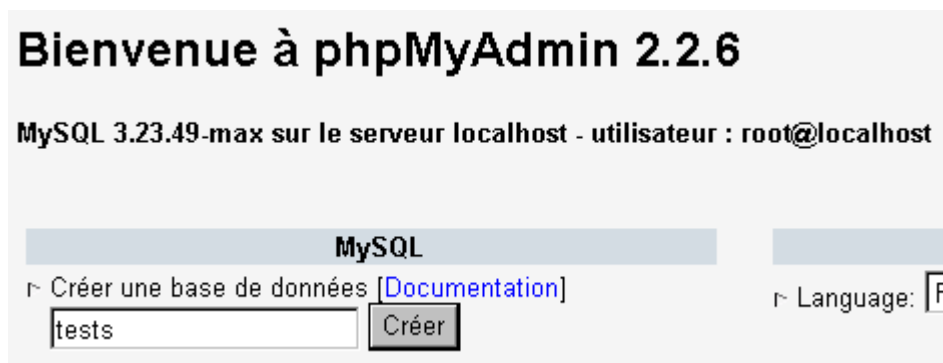
En cliquant avec le bouton droit sur l'icône représentatif d'EasyPhp dans la barre des tâches, il est possible d'activer l'affichage de la page d'administration du serveur



Depuis cette page, on accède à phpMyadmin qui permet de générer une base de données.

4.1 Création d'une base de données et d'une table dans cette base

4.1.1 Création de la base de données



4.1.2 Création d'une table dans la base de données

Il est possible de la créer via une instruction sql (requête)

```
CREATE TABLE table_test
(
id int NOT NULL
auto_increment,
date varchar (20) NOT
NULL,
champ1 varchar (50) NOT
NULL,
champ2 varchar(55) NOT
NULL,
PRIMARY KEY (id)
);
```

Mais il est possible également de la créer via l'interface de phpMyAdmin

- Créer une nouvelle table sur la base tests :

Nom :

Champs :

Dans ce cas il faut renseigner la structure de la table

	Champ	Type	Attributs	Null	Défaut	Extra
<input type="checkbox"/>	id	int(11)		Non		auto_increment
<input type="checkbox"/>	date	varchar(20)		Non	0000-00-00	
<input type="checkbox"/>	champ1	varchar(50)		Non		
<input type="checkbox"/>	champ2	varchar(55)		Non		

↑ Pour la sélection : Ou

Le id est de type auto-incrément dans notre exemple, c'est un compteur en quelque sorte qui permet d'identifier de manière unique et certaine un et un seul enregistrement de la table, on parle de clé primaire pour la table ; elle est utilisée pour créer des liens entre tables d'une même base de données (Cf. Bases de données relationnelles).

4.2 Accès à la table depuis une page php

Nous n'évoquerons pas ici comment alimenter la table via l'interface PhpMyAdmin, qui ne pose aucun problème.

L'objectif ici est de savoir comment alimenter en données cette table via une page lisible depuis un navigateur.

Il faut se connecter à la base, l' instruction est la suivante

```
include ("config.php");
//connexion à l'hôte
$dbase = mysql_connect($mysql_hostname, $mysql_username, $mysql_password);
//connexion à la base identifiée dans config.php
mysql_select_db($mysql_database, $dbase) or die ("erreur de connexion base");
```

Plutôt que de saisir les libellés exacts de serveur hôte, login et mot de passe, je préfère utiliser des noms de variables, qui sont elles mêmes affectées via un fichier config.php. ce qui facilite la portabilité des fichiers (du serveur local au serveur distant).

Le fichier config.php est appelé juste avant la commande de connexion à la base de donnée, grâce à l'instruction "include".

Voici le petit fichier de configuration config.php

```
<?

    $mysql_hostname = 'localhost';
    $mysql_username = 'login_de_connexion';
    $mysql_password = 'mdp_de_connexion';
    $mysql_database = 'nom_de_la_base';
    $mysql_tablename = 'nom_de_la_table';

?>
```

Ce fichier va être dans un premier temps configuré pour que je puisse accéder à ma base en local :

```
<?

    $mysql_hostname = 'localhost';
    $mysql_username = 'root';
    $mysql_password = '';
    $mysql_database = 'tests';
    $mysql_tablename = 'table_test';

?>
```

Il suffira ensuite de modifier ce fichier en fonction du login et du mot de passe nécessaires pour accéder au serveur distant (donc serveur en ligne sur lequel la base sera créée).

4.3 Insertion de données dans la table

La page web accueillera donc un formulaire permettant d'acquérir les données pour les champs1 et champ2 et qui enverra les contenus des variables champ1 et champ2 à une page insertion.php

```
include ("config.php");
//connexion à l'hôte
$database = mysql_connect($mysql_hostname, $mysql_username, $mysql_password);
//connexion à la base identifiée dans config.php
mysql_select_db($mysql_database, $database) or die ("erreur de connexion base");
$date = date("d-m-Y");
//on insère le contenu du formulaire
mysql_query("INSERT INTO table_test VALUES (',$date','$champ1','$champ2') ");
mysql_close(); //fermer la connexion
```

Remarques :

Le champ **id** est de type auto-incrément (voir supra) et n'a pas à être renseigné depuis la page web.

Le champ date doit être affecté automatiquement par une instruction faisant appel à une fonction :

```
$date = date("d-m-Y");
```

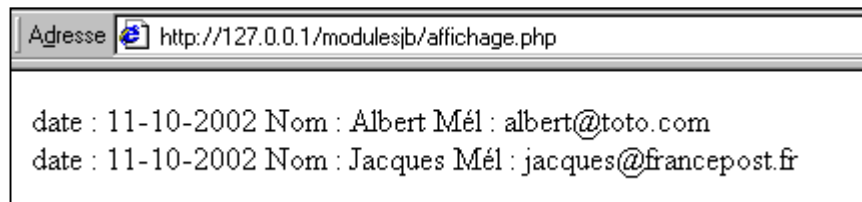
Attention, ne pas oublier la virgule après les " et avant '\$date' car elle prend en compte le champ **id**.

4.4 Lecture et affichage des données de la table

```
<?php
include ("config.php");
//connexion à l'hôte
$database = mysql_connect($mysql_hostname, $mysql_username, $mysql_password);
//connexion à la base identifiée dans config.php
mysql_select_db($mysql_database, $database) or die ("erreur de connexion base");

// on sélectionne uniquement les champs utiles (donc pas id)
$req = mysql_query("SELECT date,champ1,champ2 from table_test");
//balayage table avec while() et stockage données dans tableau associatif avec mysql_fetch_array()
while ( $resultat = mysql_fetch_array($req))
{
    Echo "date : ".$resultat[date]. " ". "Nom : ".$resultat[champ1]. " ". "Mél : ".$resultat[champ2]. "<br>";
}
mysql_close();
?>
```

Mais le résultat affiché n'est pas magnifique !

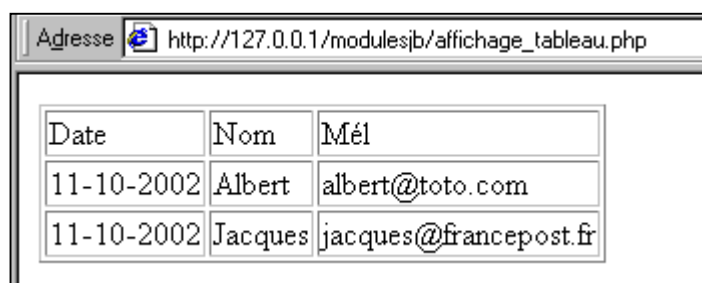


Il serait donc souhaitable d'afficher les résultats dans un tableau

On utilise les balises de tableau propres au langage html et le balayage devient alors :

```
echo "<table border =1>";
// première ligne du tableau
echo "<tr> <td> Date </td> <td> Nom </td> <td> Mél</td> </tr>";
//lignes suivantes
while ( $resultat = mysql_fetch_array($req))
{
    Echo "<tr>
    <td>".$resultat[date]. "</td>". "<td>".$resultat[champ1]. "</td>". "<td>".$resultat[champ2]. "</td></tr>";
}
echo "</table>";
```

Ce qui donne l'affichage suivant :



Date	Nom	Mél
11-10-2002	Albert	albert@toto.com
11-10-2002	Jacques	jacques@francepost.fr

4.5 Récupération des données d'une table dans un fichier texte

Il s'agit d'utiliser les commandes vues précédemment de lecture de table SQL et d'y associer l'insertion de lignes dans un fichier texte (vu au chapitre 3)

Ce qui donne le code suivant :

```
<?php
include ("config.php");
//connexion à l'hôte
$database = mysql_connect($mysql_hostname, $mysql_username, $mysql_password);
//connexion à la base identifiée dans config.php
mysql_select_db($mysql_database, $database) or die ("erreur de connexion base");
// on sélectionne tous les champs ordonnés selon le numéro d'identificateur'
$res=mysql_query("select * from table_test order by id");
//ouverture d'un fichier texte en lecture et écriture
$fichier = "enquete.txt";
$ouvert_ecrit = fopen($fichier,'a+');
//écriture des libellés de chaque champ dans une ligne du fichier
fwrite($ouvert_ecrit,"id,date,nom,mél\n");
//le fichier est automatiquement créé s'il n'existe pas
//balayage des données de la table
while ($ligne=mysql_fetch_object($res))
{
//stockage des données de chaque ligne avec un séparateur (la virgule)
//Curieuse syntaxe, la variable $ligne délivre la valeur du champ situé après la flèche
fwrite($ouvert_ecrit, "$ligne->id,$ligne->date,$ligne->champ1,$ligne->champ2\n");
}
fclose($ouvert_ecrit); //referme le fichier
mysql_close(); //clôture de la base
?>
```

Lorsqu'il n'y a plus de ligne à lire, mysql_fetch_object() renvoie FALSE.

Il resterait à mettre un hyperlien dans une page web pour pointer vers le fichier texte pour pouvoir récupérer les données de la table localement.

Attention, ce qui fonctionne en local peut très bien ne pas fonctionner à distance du fait des droits sur les fichiers :

Sur serveur Unix, les fichiers textes doivent avoir les droits pour autoriser l'accès en lecture et écriture par tout le monde. C'est à cela que sert la commande Unix chmod.

Avec Ws_Ftp il est possible de modifier ces propriétés (commande CHMOD) pour tel ou tel fichier :

Le 1er chiffre spécifie les droits pour le propriétaire du fichier (celui qui l'a déposé en principe).

Le 2ème chiffre spécifie les droits pour le groupe propriétaire du fichier.

Le 3ème chiffre spécifie les droits pour tout le monde sur le fichier (c'est à dire un visiteur Internet).

Valeur du chiffre :

0 : aucun droit en lecture, écriture, exécution.
1 : droit d'exécution
2 : droit d'écriture.
4 : droit de lecture.

Le propriétaire et le groupe propriétaire doivent avoir un droit d'écriture et de lecture (2+4 et 2+4 donc 66)
Et pour tout le monde un droit de lecture seulement (4)
Donc on parle d'un CHMOD 664 pour les fichiers qui devront être accédés, comme par exemple les fichiers textes des chapitres 3 et 4.5.

Ces quelques pages ne constituent qu'une première approche de PHP et de MySql, la syntaxe est lourde lorsque l'on n'est pas aguerri à ce type d'exercice. Il existe fort heureusement de nombreux sites proposant des scripts déjà saisis pour répondre aux besoins les plus courants.

Une recherche sur Internet avec un moteur vous donnera de quoi alimenter largement votre construction de compétences nouvelles en la matière.

**Bon courage,
Jacques Bresson.**