

Cours de PHP

Sommaire

I. Commencer	page 2
II. Les différentes variables	page 3
III. Les variables d'environnement	page 5
IV. Les conditions	page 6
V. Les boucles	page 8
VI. Les cookies	page 10
VII. Les dates en PHP	page 12
VIII. Les tableaux en PHP	page 14
IX. Travailler avec les fichiers	page 17
X. Traiter les chaînes de caractères	page 20
XI. Récupérer et vérifier les données d'un formulaire	page 23
XII. Envoyer des mails	page 24
XIII. Les pseudos frames	page 26
XIV. Débuter avec MySQL	page 29
XV. Update et Delete	page 33
XVI. Utiliser la clause WHERE	page 36
XVII. Liste de quelques fonctions MySQL	page 39
XVIII. Utilisation des fonctions PHP	page 40
XIX. Exemple de script pour un livre d'or	page 45
XX. Exemple de script pour un mini moteur de recherche	page 49
XXI. Afficher les résultats sur plusieurs pages	page 54
XXII. Exemple de script pour un forum	page 59
XXIII. Exemple de script pour un système de vote (sondage)	page 66
XXIV. Exemple de script pour un système de vote (sondage)	page 71

Chapitre I

Commencer

Avant de vous lancer dans le codage et l'apprentissage de php, vous devez avoir un éditeur de texte vous permettant de créer et d'enregistrer vos pages, une recherche rapide sur <http://www.google.fr> vous permettra de trouver votre bonheur, cependant je vous conseille Phped qui est gratuit. Il vous faut aussi un serveur Apache. Le cours n'étant pas consacré à la configuration des serveurs apaches, télécharger le logiciel Easyphp qui installe automatiquement le serveur.

I. Etape 1.

Php est un langage exécuté coté serveur, à la différence de java script et html, donc votre code n'apparaît pas dans la source de votre page Web, seule les informations que vous souhaitez affichées seront visible par les visiteurs.

II. Les balises

Lorsque vous créer un code, vous devez le placer entre balise php pour que celui-ci soit interprété, comme ceci:

```
<?
echo 'bonjour';
?>
```

ou encore

```
<?PHP
echo 'bonjour';
?>
```

Ce qui affichera à l'écran ' bonjour '.

Ne poser pas encore vos question sur le echo etc..., tout est expliqué dans les cours suivant ;-)

Si vous souhaitez tester vos scripts en local sur votre PC, télécharger Apache, PHP4 (voir dans la rubrique installer)

Bon jusque la tout va bien, passons maintenant à l'étape suivante, le cours sur les variables.

Chapitre II

Les différentes variables

Nous allons rentrer maintenant dans le vif du sujet, connaître et se servir des différentes variables.

I. Définition

Il existe plusieurs type de variables en php, celle que vous définissez vous même en donnant une valeur, les variables d'environnement; les variables de sessions etc...

Nous allons traiter dans ce cours, des variables que vous définissez et des variables d'environnements, les variables de sessions seront expliquées dans un cours réservé à ce type de manipulation.

II. Définir une variable

Vous devez garder en mémoire tout au long de ce cours, comment associé un nom de variable et y donner une valeur, regarder l'exemple suivant.

```
<?
$variable = 'bonjour'; //on défini une variable
```

```
echo $variable; //on affiche le résultat
?>
```

Affiche bonjour à l'écran

Explication:

Une variable fonctionne sous la forme `variable_nom = variable_valeur`, mémoriser bien ce fonctionnement, cela fait parti de la structure d'un langage de programmation. Lorsque vous attribuez une valeur à un nom de variable, pour afficher celle ci à l'écran, appeler le nom de cette variable et sa valeur sera affichée.

III. Affichage de variables et de texte à l'écran

Vous pouvez afficher des phrases, variables etc... de plusieurs manières en php.

Exemple 1:

```
<?
$nom = 'visiteur'; //on donne une valeur à la variable appelé nom
echo "bonjour $nom"; //on affiche le résultat avec echo et les "
?>
```

Affichera à l'écran

bonjour visiteur

Exemple 2:

```
<?
$nom = 'visiteur'; //on donne une valeur à la variable appelé nom
echo "bonjour ".$nom; //on affiche le résultat avec echo et les "
```

```
?>
```

Affichera à l'écran

```
bonjour visiteur
```

Attention: Remarquez qu'il y'a un . entre le ' et la variable \$nom, le point sert à concaténer la variable à la chaîne. N'oubliez jamais ce point sinon vous aurez un message d'erreur pour cette page

Exemple 3:

```
<?
```

```
$nom = 'visiteur'; //on donne une valeur à la variable appelé nom
```

```
echo'bonjour $nom'; //on affiche le résultat avec echo et les "
```

```
?>
```

Affichera à l'écran

```
bonjour $nom
```

Attention :

- si vous utilisez les ' au lieu des " , la variable n'est pas interprétée comme une variable mais comme une chaîne de caractère.

-Lorsque vous utilisez les ' si dans votre texte vous avez des apostrophes, vous devez les faire précéder d'antislash par exemple, la phrase echo'vous n'êtes pas inscrit'; affichera un message d'erreur,

vous devrez l'écrire comme ceci echo'vous n\êtes pas inscrit';

Même chose pour les guillemets, la phrase

```
echo"<a href="http://www-etu">Apprendre php</a>"; affichera une erreur
```

vous devrez alors l'écrire de cette manière

```
echo"<a href=\"http://www-etu\">Apprendre php</a>";
```

Pour afficher des variables, ou textes, vous pouvez aussi utiliser print 'ce que vous voulez';

Allez cours suivant, les variables d'environnements.

Chapitre III

Les variables d'environnement

Le plus efficace pour connaître les différentes variables d'environnements, c'est tout simplement de faire une page php contenant ceux-ci

```
<?
phpinfo();
?>
```

Dans le résultat de la page affichée, vous avez bon nombre d'infos concernant ces fameuses variables, par exemple:

`$HTTP_HOST` //affiche l'hôte, c'est à dire le nom de votre espace Web
`$HTTP_REFERER` //affiche la provenance de votre visiteur, l'url d'ou celui ci arrive
`$DOCUMENT_ROOT` // le répertoire racine de l'arborescence des document sur le serveur
`$QUERY_STRING` //le contenu de ce qui suit l'url de la page, par exemple page.php?url=az-php.com , affichera url=az-php.com
`$REQUEST_METHOD` //la méthode utilisée, GET, POST, pratique pour vérifier les variable provenant d'un formulaire

Je ne vais pas trop m'étendre sur ces variables, je vous conseil de faire un phpinfo, vous trouverez certainement dont vous avez besoin.

Chapitre IV

Les conditions

Si vous avez tous déjà regardé un script, et vous avez sans doute remarqué des choses du genre,

```
<?
if($var == 'ok')
{
echo 'test';
}
else{
echo 'refusé';
}
?>
```

C'est ce que l'on appelle une condition. Celle ci s'exécute de la manière suivante:

```
if($var == 'ok') //teste si la variable nommée $var est égale à ok
{
echo 'test'; //on affiche le résultat
}
else{ // sinon on affiche un autre message
echo 'refusé';
}
?>
```

Jusque là tout va bien ?

Voici quelques conditions

Exemple 1, avec if else elseif

```
$variable = 'voiture';

if($variable == 'voiture'){
echo 'bravo vous avez trouvé';
}
elseif($variable == 'automobile'){
echo 'c'est presque ça';
}
else {
echo 'ce n'est pas ça veuillez réessayer';
}
```

Exemple 2, switch()

```
switch($operation)
```

```

{
case '1': // si la variable opération est égale à 1
echo 'operation numero 1'; // on affiche cette phrase
break; // on referme cette condition

case '2': // si la variable opération est égale à 2
echo 'operation numero 2';
break;

default: // si la variable opération n'est pas égale à 1 ni à 2 ou si elle n'est pas définie
echo 'operation par défaut'; // on affiche une phrase par défaut
}

```

Vous comprendrez mieux l'avantage de la fonction switch plus loin et du déroulement d'un script avec if else elseif etc...

Les opérateurs de contrôle sont les suivants :

==	strictement égale
!=	différent
>	plus grand que
<	inférieur à
>=	supérieur à
<=	inférieur à
&&	et
	ou
AND	et
OR	ou
TRUE 1	ou oui
FALSE 0	ou non

Les opérateurs mathématiques sont les suivants :

+	addition
-	soustraction
/	division
*	multiplication

Chapitre V

Les boucles

Les boucles vous permettent de parcourir un ensemble d'informations stockées dans un tableau, un fichier texte, une base de donnée, et de les afficher ou de les traiter.

Les différentes boucles sont :

```
while()
for()
foreach() ( cette commande ne peut être utilisée que sous PHP4 )
```

Je vous donne un exemple de boucle mais vous trouverez des explications dans les cours suivants, lorsque nous allons concrètement travailler sur un code.

Exemple avec while ()

```
$i= 0; // on défini une variable à 0 pour le compteur de boucle

while ( $i < '4' ) // la boucle s'arrêtera lorsque la variable $i sera égale à 4
{
echo 'boucle numéro '.$i.'<br>'; // on affiche une phrase avec le numéro de la boucle

$i++; // le ++ sert à ajouter 1 à chaque tour de boucle, ne l'oubliez pas sinon la boucle sera
infini donc affichera une erreur !

}
```

Affichera à l'écran

```
boucle numéro 0
boucle numéro 1
boucle numéro 2
boucle numéro 3
boucle numéro 4
```

Exemple avec for()

```
for ($i=0;$i<4;$i++) // c'est exactement la même chose que dans l'exemple while, mais
l'avantage de for est que vous avez tout sur la même ligne, cela évite les oublies
d'incrémentatior ☺ !!
{
print 'boucle numéro '.$i.'<br>'; // on affiche une phrase avec le numéro de la boucle
}
```

Affichera à l'écran

boucle numéro 1

boucle numéro 2

boucle numéro 3

boucle numéro 4

Vous trouverez des exemples de boucles et leurs utilisations dans les cours suivants.

Chapitre VI Les Cookies

Ils sont stockés sur l'ordinateur de votre visiteur, et permettent d'y insérer un petit nombre d'information.

Les cookies conservent le schéma des variables, comme expliqué dans un cours précédent, `nom_de_variable = valeur_de_variable`, cependant d'autres paramètres sont également définis dans un cookie:

- * la durée pour l'expiration
- * le chemin du répertoire ou doit être lu ce cookie
- * le nom domaine
- * et le type d'entête (http, https)

Pour envoyer un cookie, il faut utiliser la fonction de php nommé `setcookie()`

Exemple:

Nous allons envoyer un cookie avec la variable `$visiteur` ayant pour valeur `utt`

```
<?
setcookie("visiteur","utt","/");
?>
```

Dans cette exemple, le cookie peut être affiché sur toutes les pages du site, grâce à `"/"`

Nous allons maintenant envoyer un cookie, en gardant la même variable, mais en mettant l'expiration à 1 an

Exemple :

```
<?
setcookie("visiteur","utt",time()+365 * 24 * 3600 , "/");
?>
```

Explication :

En premier, vous avez `visiteur`, c'est le nom de la variable pour ce cookie, `utt` est sa valeur, ensuite vous avez `time()`, ceci est l'heure en local (voir le cours sur les dates qui a lieu juste après), donc dans cette exemple, `time() +` c'est l'heure et date du jour, `+ 365 jours X 24 heures X 3600 seconde` (60 secondes multiplié par 60 minutes pour donner 1 heure), ce cookie expirera donc dans 1an

.
Comment récupérer le cookie ?

Vous récupérer ce cookie tout simplement avec son nom

Exemple de récupération de cookie :

On garde toujours le même cookie, soit visiteur, qui a pour valeur utt.

```
<?
echo $visiteur.' est mon cookie';
?>
```

Affichera à l'écran

utt est mon cookie

Simple non ? ☺

Comment supprimer un cookie ?

C'est aussi simple, vous envoyez un cookie avec une variable sans valeur et une expiration dépassée.

Exemple:

```
<?
setcookie("visiteur","",time()-100,"/");
?>
```

Voilà, maintenant le cookie n'a plus de valeur et est supprimé de l'ordinateur de votre visiteur.

Chapitre VII

Les dates en PHP

Nous allons apprendre maintenant à manipuler les dates sous différents formats et comment les afficher.

Code à utiliser avec la fonction date() :

format	description	exemple
a	"am" ou "pm" minuscules	pm
A	"AM" ou "PM" majuscules	PM
d	jour du mois	01/20
D	jour de la semaine en 3 lettres	Mon
F	nom du mois	Janvier
h	heure (format 12 heures avec 0 en entête)	12
H	heure (format 24 heures avec 0 en entête)	08
g	heure (format 12 heures sans 0 en entête)	4
G	heure (format 24 heures sans 0 en entête)	10
i	minutes	44
j	jours du mois (sans 0 en entête)	3
m	mois de l'année (0 en entête)	04
M	mois de l'année en 3 lettres	jui
n	mois de l'année (sans 0 en entête)	4
s	secondes	30
y	année à deux chiffres	03
Y	année à 4 chiffres	2003

Exemple, afficher la date du jour :

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

echo 'Nous sommes le '.$date_du_jour;
?>
```

Affiche à l'écran si la date est le 2 avril 2002

Nous sommes le 02-04-2002

La date au format système :

Exemple d'affichage avec la fonction time

```
<?
$date_du_jour = time();

echo 'la date système du jour est '.$date_du_jour;
?>
```

Affichera à l'écran quelque chose comme ceci

la date système du jour est 1012566581

Code de tableau à utiliser avec la fonction getdate() :

clés	description	Exemple
seconds	Secondes	30
minutes	minutes	5
hours	heure de la journée de 0 à 23	15
mday	jour du mois de 1 à 31	12
wday	jours de la semaine de 0 à 6	2
mon	moi de l'année	4
year	année en 4 chiffres	2003
yday	jours de l'année de 0 à 365	180
weekday	nom du jour de la semaine (en anglais)	Monday
month	mois de l'année (en anglais)	January

Exemple de travail avec getdate():

```
<?

$time = time(); // la date au format système

$date = getdate($time); // passage de la variable time dans getdate pour ressortir les infos

print 'Nous sommes le '.$date[mday].'- '.$date[mon].'- '.$date[year].' il est
'.$date[hours].':'.$date[minutes];
?>
```

Affichera à l'écran

Nous sommes le 2 - 4 - 2002 il est 20:05

Chapitre VIII

Les tableaux en php

Il existe 2 types de tableaux, les tableaux nominatifs et les associatifs.
Pour créer un tableau, utiliser la fonction array();

I. Exemple de tableau nominatif:

```
<?
$tableau = array( 'utt', 'php4', 'mysql' ); // on déclare les valeurs du tableau
echo $tableau;
?>
```

Affiche à l'écran

Array

Et oui, pour afficher un tableau, il faut le parcourir, c'est ici que vous allez avoir besoin des cours sur les boucles ☺

II. Exemple de tableau associatif:

```
<?
$tableau_ass = array(site=>'utt', language=>'php4', base=>'mysql');
// On déclare le tableau sous le format nom_de_variable=>'valeur'

//pour afficher le site par exemple
echo 'le nom du site est '.$tableau_ass[site];

?>
```

Affiche à l'écran

Le nom du site est utt

III. Comprendre le fonctionnement des tableaux:

Les lignes des tableaux ne sont pas numérotées comme vous le pensez, le premier enregistrement porte le numéro de ligne 0, donc un tableau comme ceci:

```
$tableau = array('valeur1', 'valeur2', 'valeur3');
```

Le 1er élément, valeur1 porte le numéro de ligne 0, si vous souhaitez l'afficher, procéder comme suit:

```
<?
echo $tableau[0];
?>
```

Affichera à l'écran

valeur1

IV. Se servir des boucles pour parcourir un tableau :

Avec foreach(), uniquement sous php4

```
<?
$tableau = array('utt','php4','mysql'); //on défini le tableau et ses éléments

foreach ( $tableau as $contenu ) //on parcourt le tableau
{
echo $contenu.'<br>'; //on affiche le contenu
}
```

Affiche à l'écran

utt
php4
mysql

V.La même chose mais en tableau associatif

```
<?
$tableau = array(site=>'utt',language=>'php4',base=>'mysql'); //on défini le tableau et ses éléments

foreach ( $tableau as $key=>$contenu ) //on parcourt le tableau en attribuant la clé à la variable
{
echo 'clé '.$key.' valeur '.$contenu.'<br>'; //on affiche la clé et son contenu
}
```

Affiche à l'écran

clé site valeur utt
clé language valeur php4
clé base valeur mysql

VI. Ajouter un élément dans un tableau

```
<?
```

```
$tableau = array( 'utt', 'php4', 'mysql' ); //on définit le tableau et ses éléments
```

```
array_push($tableau, "javascript"); //on ajoute avec la fonction array_push()
```

```
//le nouveau tableau
```

```
$tableau = array( 'utt', 'php4', 'mysql', 'javascript' );
```

Chapitre IX

Travailler avec les fichiers

Vous pourrez être amené à travailler avec des fichiers texte pour stocker des informations diverses pour votre site, ce cours vous servira à mieux comprendre les différentes utilisations des fonctions. Si vous ne désirez pas apprendre à utiliser les fichiers et que vous préférez utiliser une base de donnée Mysql, sautez ce chapitre.

Les fonctions pour ouvrir un fichier :

`file(); //parcours un fichier et le stock dans un tableau`

`fopen(); //ouvre un fichier`

Les associations d'ouvertures liées à `fopen()` :

'w' : ouvre le fichier pour écriture, et détruit les données si le fichier en contient

'w+' : comme ci dessus mais crée le fichier si celui ci n'existe pas

'a' : ouvre le fichier pour ajouter de nouvelles données

'a+' : comme ci-dessus mais crée le fichier si celui n'existe pas

'r' : ouvre le fichier en lecture seulement

Lire le contenu stocké dans un fichier :

```
<?
```

```
$fichier = 'monfichier.txt'; // on déclare le nom du fichier à ouvrir
```

```
$taille = filesize($fichier); //donne la taille du fichier
```

```
$fp = fopen($fichier,'r'); //ouverture du fichier en lecture seulement
```

```
while(!feof($fp)) //feof indiquera la fin du fichier, donc dans cette boucle, le fichier est  
parcouru jusqu'à la fin
```

```
{
```

```
$ligne = fgets($fp,$taille); //lecture du fichier et stockage dans la variable ligne
```

```
echo $ligne.<br>; //affiche la ligne à l'écran, n'oubliez pas le <br> qui est le retour à la ligne  
html
```

```
}
```

```
fclose($fp); //pensez à refermer à la fin du script
```

```
?>
```

Commentaires:

Pour lire le contenu, vous pouvez utiliser également la fonction `fread()` qui fonctionne comme la fonction `fgets()`

Ecrire dans un fichier:

On reprend le même exemple que ci-dessus mais on ouvre le fichier en écriture

```
<?
$fichier = 'monfichier.txt'; // on déclare le nom du fichier à ouvrir

$f = fopen($fichier,'w'); //ouverture du fichier en écriture, en supprimant les données qu'il
contient déjà

fwrite($f,"ici la phrase que vous souhaitez\n"); //écriture dans le fichier, le \n à la fin de la
phrase indique le retour à la ligne dans le fichier

fclose($f); //pensez à refermer à la fin du script
?>
```

Commentaires:

Pour écrire dans un fichier vous pouvez utiliser la fonction fputs(); qui fonctionne comme fwrite.

Je vous laisse faire pour l'ajout de nouvelles données dans un fichier sans effacer ce qu'il contient ;-)

Un exemple d'utilisation de la manipulation des fichiers:

```
<?
//*****
//
// Votre mini compteur de visite utt
//
//*****

$fichier = 'compteur.txt';

$f = file($fichier); // ici j'utilise file() cette fonction stocke le contenu du fichier sous forme
d'un tableau (array) dans la variable $f

//puisque c'est un compteur, on ne lit qu'une seule ligne, celle où le numéro se trouve
donc dans ce code, le total du compteur se trouve dans la ligne 0 du fichier soit $f[0]

$compteur = $f[0] + 1; // on ajoute 1 au compteur actuel
//----- ouverture du fichier en écriture pour insérer le nouveau nombre de visites

$nouveau = fopen($fichier,'w+'); //on ouvre le fichier en écriture

fwrite($nouveau,"$compteur\n"); // on écrit le nouveau numéro de compteur dans le fichier,
en supprimant les anciennes données, le fichier est créé s'il n'existe pas
```

```
fclose($nouveau); //on referme le fichier
```

```
?>
```

Maintenant à vous de jouer

Et oui chacun son tour, maintenant à vous d'afficher le contenu du fichier sur votre site, pour voir le nombre de visiteurs.

Chapitre X

Traiter les chaînes de caractères

Vous allez découvrir dans ce cours comment travailler sur les chaînes de caractères, ainsi que la liste des fonctions.

Les fonctions :

Fonction Description

strtolower() Mise en minuscule
strtoupper() Mise en majuscule
ucfirst() Mise en majuscule de l'initiale
nl2br() Remplace le \n par
 pour affichage
htmlspecialchars() Convertit les caractères html
addslashes() Fait précéder les caractères spéciaux d'un \
stripslashes() Supprime les \
ltrim() Supprime les espaces initiaux
trim() Supprime les espaces en début et fin

Fonctions de remplacement : str_replace()

//la phrase modèle

```
$chaine = "Le php est un langage payant";
```

```
str_replace()
```

```
//Nous allons remplacer payant par gratuit
```

```
$chaine1 = str_replace("payant","gratuit",$chaine) ;
```

```
//on affiche le résultat
```

```
echo $chaine1;
```

Affiche à l'écran

```
Le php est un langage gratuit
```

```
ereg_replace()  
$chaine1 = ereg_replace("payant","gratuit",$chaine) ;
```

//on affiche le résultat

```
echo $chaine1;
```

Affiche à l'écran

Le php est un langage gratuit

Rechercher un mot :

//la phrase modèle

```
$chaine = "Le php est un langage gratuit";
```

```
if( ereg('un',$chaine)) //on recherche le mot un dans la phrase
```

```
{  
echo 'le mot un est bien dans la phrase';  
}
```

```
else{
```

```
echo 'le mot n'est pas trouvé';
```

```
}
```

Découper une chaîne:

//la phrase modèle

```
$chaine = "Le php est un langage gratuit";
```

//on utilise split() pour découper la phrase

```
$resultat = split(" ",$chaine); //on découpe la chaîne avec comme modèle, l'espace
```

//Attention: le résultat obtenu est sous forme de tableau

```
print $resultat[1];
```

Affiche à l'écran

php

Pour mémoire, le tableau obtenu se présente comme ceci

Le php est un langage gratuit

```
0 1 2 3 4 5
```

Vérifier si une variable est vide:

```
if( empty($variable_a_verifier)) //empty vérifie si la variable est vide
{
echo "La variable est vide"; //message erreur variable vide
}
else{
echo "c'est ok"; //la variable est attribuée c'est donc ok
}
```

Le contraire de empty() est isset().

```
if( isset($variable_a_verifier)) //isset vérifie si la variable est attribué
{
print "ok la variable est attribuée";
}
else{
print "la variable est vide";
}
```

Chapitre XI

Récupérer et vérifier les données d'un formulaire

Vous avez déjà vu la déclaration de variable, les manipulations et autres, nous allons traiter dans ce cours, comment récupérer et vérifier les données des formulaires.

Récupérer des valeurs de champs de texte

Nom
Prenom

Le code html du formulaire

```
<form method="post" action="verif.php">
<input type="text" name="nom">
<input type="text" name="prenom">
<input type="submit" name="bouton" value="Envoyer">
</form>
```

La page verif.php

```
<?
if(empty($nom) || empty($prenom)) //on vérifie avec empty voir si les champs sont vide
{
echo "le champ nom ou le champ prenom est vide"; //si un des 2 champs n'est pas rempli,
message d'erreur
}
else{ //sinon message de confirmation
echo "les champs sont ok";
}
?>
```

Vérifier une adresse email

```
<?
if(!ereg("\.", $email) || !ereg("@", $email)) //on vérifie avec un regex, il y a d'autres
possibilités, mais nous verrons cela dans un cours sur les expressions régulières
{
echo "adresse email non valide"; //le . ou @ ou les 2 ne sont pas dans le champ $email
}
else{
echo "adresse email ok";
}
?>
```

Chapitre XII

Envoyer des mails

Pour envoyer des mails en php, vous devez utiliser la fonction mail() , le principe de base est comme suit:

Envoi simple d'email

```
<?
mail('mon@email','sujet','message'); //voici la fonction basique pour envoyer un mail
?>
```

L'exemple ci-dessus vous montre la structure pour se servir de la fonction mail() , en 1er le destinataire, en second, le sujet et ensuite le message. Vous verrez comment se servir de cette fonction plus loin, et ajouter un email de provenance (from), ou envoyer un message au format html

Envoyer un mail avec l'email de provenance (from)

```
<?
$to = 'mon@email' ; //l'email du destinataire
$from = 'From: votre@email.com'; //l'email de celui qui envoi le message

$sujet = 'Ceci est un mail de test'; //le sujet
$message = 'Bonjour, ceci est un envoi de mail test'; //le message

//maintenant on envoi le mail avec la fonction php
mail($to,$sujet,$message,$from);

?>
```

Vous avez sans doute remarqué que j'ai ajouter le from à la suite de sujet dans la fonction mail, noté bien que c'est le seul champ supplémentaire que l'on peut ajouter, cette fonction ne prendra que 4 paramètres, si vous en ajoutez d'autre, vous obtiendrez un message d'erreur.

Envoyer un mail au format html

```
<?
$to = 'monr@email'; // email du destinataire
$sujet ='test mail html'; //le sujet
```

```
$from ="From: votre@email.com \r\n"; //votre email  
$from .= "MIME-Version: 1.0\r\n"; //on défini le type mime pour le message  
$from .= "Content-Type: text/html; charset=us-ascii\r\n"; //ici on défini le format , soit html  
  
$message = " "; //entrez ici un code html  
  
mail($to,$sujet,$message,$from); // la fonction mail  
  
?>
```

Vous avez sans doute remarqué le . devant le égale de la variable \$from, cela s'appelle la concaténation, cela permet de définir une variable sur plusieurs lignes, la concaténation à d'autre avantage ;-)

Chapitre XIII

Les pseudos frames

La fonction include vous permet d'insérer une page, ou plusieurs, dans une page php choisi, par exemple:

Une page index.php avec menu

Le code html de la page index.php

```
<html>
<head><title>mon site</title>
</head>
<body color="#000000">

<table border="0" width="750" align="center">
<tr>
<td width="150" valign="top">
<? include('menu.php'); ?> //on insert la page menu
</td>

<td width="600" valign="top">
<? include('centre.php'); ?> //on insert le contenu
</td>
</tr>
</table>
```

Attention: lorsque vous insérer une page, celle ci ne doit pas contenir de balise html comme <html><head><body>, uniquement le tableau et les images , ou simplement le contenu de la page, sinon vous aurez plusieurs fois les balise html dans votre page!!

Page menu.php

```
<table border="0" width="150" align="center">
<tr>
<td align="center">lige menu 1</td>
</tr>

<tr>
<td align="center">ligne menu 2</td>
</tr>
</table>
```

Je vous laisse faire pour la page centre.php, procéder comme dans la page menu, uniquement le contenu, je le rappelle mais j'ai vu pas mal d'erreur à cause de ça.

Passons à quelque chose qui va vous intéresser. Vous avez sans doute déjà vu des sites, comme celui ci par exemple, avec des page qui ne bouge pas et seulement une url dur genre <http://ledomaine/?page=news>, vous voulez savoir comment faire, je vais vous expliquer le fonctionnement ci-après.

On conserve le début du cours avec la page index.php, la page menu et la page centre

Les liens dans la page menu.php

```
<table border="0" width="150" align="center">
<tr>
<td align="center"><a href="/?page=news">Les news</a> </td>
</tr>

<tr>
<td align="center"><a href="/?page=livre_dor">Livre d'or</a> </td>
</tr>
</table>
```

J'ai mis le liens avec un / mais vous pouvez remplacer par index.php?page=news, car sous le / c'est la page index qui est caché, mais je pense que vous le saviez ;-)

Nous allons passer à la page centre.php, c'est ici que vous allez avoir besoin du cours sur les conditions

```
<?
if($page == 'news'){ //si la variable $page est égale à news
include('news.php'); //on insert la page news
}
elseif( $page == 'livre_dor'){ //idem pour livre d'or
include('livre.php');
}
?>
```

Attention: si la page se trouve dans un sous répertoire, pensé que vous devez aller chercher la page avec son chemin, donc si la page news.php se trouve dans le répertoire info par exemple, vous devrez faire `include('info/news.php');`

Pour ma part, je vous conseil d'utiliser switch si vous avez un grand nombre de page à insérer, procéder comme ceci:

Page centre.php avec switch()

```
<?
switch($page)
{

case'news':
include('news.php');
break;

case'livre_dor':
include('livre.php');
break;

default:
//pensez toujours à mettre une page par défaut pour que votre visiteur voie une page si la
variable n'est pas attribuée

}
?>
```

Chapitre XIV

Débuter avec mysql

Un cours qui va vous intéresser et vous montrer que mysql va devenir votre amie. Vous pouvez tout faire avec une base donnée, et de plus celle-ci vous simplifie la vie pour la manipulation des données, vous pourrez construire rapidement toute sorte de scripts, tels que livre d'or, news, forum, boutique en ligne etc.....
Mais avant de se lancer, il faut comprendre le fonctionnement, alors allons y ;-)

Construire une table

```
CREATE TABLE test_tbl (  
id int NOT NULL auto_increment,  
date varchar(20) NOT NULL,  
prenom varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
PRIMARY KEY (id)  
);
```

Explications:

Pour créer un table dans mysql, via phpmyadmin ou autre, vous devez utiliser l'expression CREATE TABLE et ensuite donné le nom à celle ci, dans l'exemple test_tbl, attention, les . , - ne sont pas pris en compte !!

Dans la 2eme ligne, id int auto_increment, cela veut dire que l'on crée une colonne id, qui sera incrémenté d'un chiffre à chaque enregistrement, auto increment veut dire que cela se fera tout seul lors de l'insertion.

Ensuite on déclare les autres colonnes, date, avec une longueur maxi de 20 caractères (varchar (20)) etc...

La dernière ligne, indique que l'id est la clé primaire de la table en question.

Exemple d'insertion

Pour insérer un enregistrement dans une table, vous devez utiliser la fonction insert into()

Dans nôtre exemple, ça donnerait ceci:

```
INSERT INTO test_tbl (id,date,prenom,email) values('4-04-  
2002','webmestre','webmaster@az-php.com');
```

Vous pouvez aussi faire plus court avec seulement les données à insérer

```
INSERT INTO test_tbl values('4-04-2002','webmestre','webmaster@az-php.com');
```

Se connecter à une base depuis une page Web

Pour vous connecter depuis une page php de votre site, vous devez utiliser les paramètres suivants:

```
<?
$db = mysql_connect('hote de connexion','votre login','votre mot de passe') or die ("erreur de connexion");
mysql_select_db('nom de la base',$db) or die ("erreur de connexion base");
?>
```

Si vous avez rempli correctement l'hote, login, password et base, aucun message d'erreur ne devrait apparaître.

Passons à un exemple concret, allez dans phpmyadmin, puis insérer les données suivantes:

Table et données

#la structure de la table test_tbl

```
CREATE TABLE test_tbl (
id int NOT NULL auto_increment,
date varchar(20) NOT NULL,
prenom varchar (50) NOT NULL,
email varchar(55) NOT NULL,
PRIMARY KEY (id)
);
```

#insertion des données que l'on souhaite voir dans la table

```
INSERT INTO test_tbl VALUES('1','4-04-2002','webmestre','webmaster@az-php.com');
INSERT INTO test_tbl VALUES ( '2', '4-04-2002', 'pierre', 'email@pierre.com');
INSERT INTO test_tbl values('3','4-04-2002','paul','email@paul.com');
INSERT INTO test_tbl values('4','4-04-2002','jacques','email@jacques.com');
```

Voilà, c'est inséré dans la base et la table est créée ainsi que les données.
Comment afficher le résultat depuis votre site?

Afficher le contenu de la table test_tbl

```
<?
//on se connecte à la base de donnée
```

```
$db = mysql_connect('hote de connexion','votre login','votre mot de passe') or die ("erreur de connexion");
```

```
//on sélectionne votre base
```

```
mysql_select_db('nom de la base',$db) or die ("erreur de connexion base");
```

```
// Pour récupérer les données, on utilise la fonction select comme suit
```

```
$req = mysql_query("SELECT id,date,prenom,email from test_tbl "); // on sélectionne les colonnes id,date,prenom,email de la table test_tbl
```

```
//pour afficher on se sert de 2 fonctions, une boucle while() et mysql_fetch_array() qui regroupe les données de la requête dans un tableau associatif
```

```
while ( $resultat = mysql_fetch_array($req))  
{  
print "\'id ->'. $resultat[id]. ' date ->'. $resultat[date]. ' prenom->'. $resultat[prenom]. ' email ->'. $resultat[email]. '<br>';  
}
```

```
mysql_close(); //pensez toujours à refermer la connexion  
>
```

Ce qui affichera à l'écran

```
l'id ->1 date ->4-04-2002 prenom-> webmestre email-> webmaster@az-php.com
```

```
l'id ->2 date ->4-04-2002 prenom-> pierre email-> email@pierre.com
```

```
l'id ->3 date ->4-04-2002 prenom-> paul email-> email@paul.com
```

```
l'id ->4 date ->4-04-2002 prenom-> jacques email-> email@jacques.com
```

Voilà comment ressortir des infos d'une table, maintenant nous allons insérer des données depuis une page Web

Code html de la page Web

```
<html><head><title>formulaire</title>
```

```
</head>
```

```
<body>
```

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

```
<input type="text" name="prenom"><br>
```

```
<input tupe="text" name="email"><br>
```

```
<input type="submit" name="submit" value="Insérer dans la base">
```

```
</form>
```

```
</body>
```

```
</html>
```

La page d'insertion inserer.php

```
<?
//on se connecte à la base de donnée
$db = mysql_connect('hote de connexion','votre login','votre mot de passe') or die ("erreur de
connexion");
//on sélectionne votre base
mysql_select_db('nom de la base',$db) or die ("erreur de connexion base");

//on défini une date
$date = date("d-m-Y");
//on insère le contenu du formulaire
mysql_query("INSERT INTO test_tbl VALUES (',$date','$prenom','$email' ");

//et on referme la connexion
mysql_close();

?>
```

Si vous avez suivi, on peut passer au cours suivant concernant les fonctions update et delete de mysql

Chapitre XV Update et Delete

Update et delete sont 2 fonctions qui vont vous servir dans vos futures applications,

Update: la fonction update vous permet de modifier les enregistrements d'une ou plusieurs colonnes dans votre table.

Delete: vous permet de vider les enregistrements d'une table ou de supprimer un seul enregistrement avec une clause where.

La fonction Update

On garde notre table test_tbl du cours précédent

```
CREATE TABLE test_tbl (  
id int NOT NULL auto_increment,  
date varchar(20) NOT NULL,  
prenom varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
PRIMARY KEY (id)  
);
```

Celle-ci contient les enregistrements suivants :

- 1 / 30-09-2003 / webmestre / webmaster@utt.fr
- 2 / 30-09-2003 / tomaka / tomaka@utt.fr
- 3 / 30-09-2003 / ady / ady@utt.fr
- 4 / 30-09-2003 / ken / ken@utt.fr

Nous allons remplacer webmestre par jeck.

```
<?  
//on se connecte à la base de donnée  
$db = mysql_connect('hote de connexion','votre login','votre mot de  
passe') or die ("erreur de connexion");  
//on sélectionne votre base  
mysql_select_db('nom de la base',$db) or die ("erreur de connexion  
base");  
  
//on modifie le prenom de l'id n°1  
mysql_query("Update test_tbl Set prenom = 'jeck' where id = '1' ");  
?>
```

Explication: la structure de la fonction update est

```
Update //le nom de la fonction
test_tbl //le nom de la table à modifier
Set //sélection de la colonne
prenom //le nom de la colonne à modifier
= 'jeck' // est égale à la nouvelle valeur soit jean
where //pour le choix
id // on choisi ici l'id d'insertion pour modifier
= '1' // égale au numéro choisi soit 1
```

Vous pouvez mettre plusieurs paramètres pour SET séparés par des virgules, comme ceci SET colonne1 = 'valeur1', colonne2 = 'valeur2' etc...

Passons à la fonction Delete pour supprimer un enregistrement

La fonction Delete

Toujours avec notre exemple de la table test_tbl

Si vous souhaitez vider une table de son contenu, la requête à utiliser est comme ceci :

```
<?
//on se connecte à la base de donnée
$db = mysql_connect('hote de connexion','votre login','votre mot de
passe') or die ("erreur de connexion");
//on sélectionne votre base
mysql_select_db('nom de la base',$db) or die ("erreur de connexion
base");

//on vide la table complètement
mysql_query("Delete from test_tbl ");

mysql_close();

?>
```

Supprimer 1 seul enregistrement, on choisi l'id numéro 1

```
<?
//on se connecte à la base de donnée
$db = mysql_connect('hote de connexion','votre login','votre mot de
passe') or die ("erreur de connexion");
//on sélectionne votre base
mysql_select_db('nom de la base',$db) or die ("erreur de connexion
base");

//on vide la table complètement
mysql_query("Delete from test_tbl where id ='1'");
```

```
mysql_close();
```

```
?>
```

Explications : La `close where` vous permet de choisir l'enregistrement que vous souhaitez supprimer, si vous n'utilisez pas cette `close`, la table sera complètement vidée de son contenu, donc à manipuler en faisant attention !

Pour la suite, on va se concentrer sur la `close where` et comment choisir ce que l'on veut voir apparaître à l'écran.

Chapitre XVI

Utiliser la clause where

La clause where vous permet de choisir ce que vous voulez afficher en sortie de requête, where vous sert d'élément de comparaison.

Débutez avec la clause where

On garde notre table test_tbl du cours précédent

```
CREATE TABLE test_tbl (  
id int NOT NULL auto_increment,  
date varchar(20) NOT NULL,  
prenom varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
PRIMARY KEY (id)  
);
```

Celle-ci contient les enregistrements suivants:

```
1 / 30-09-2003 / webmestre / webmaster@utt.fr  
2 / 30-09-2003 / tomaka / tomaka@utt.fr  
3 / 30-09-2003 / ady / ady@utt.fr  
4 / 30-09-2003 / ken / ken@utt.fr
```

Nous allons sélectionner seulement les données pour paul
<?

```
//on se connecte à la base de donnée
```

```
$db = mysql_connect('hote de connexion','votre login','votre mot de  
passe') or die ("erreur de connexion");
```

```
//on sélectionne votre base
```

```
mysql_select_db('nom de la base',$db) or die ("erreur de connexion  
base");
```

```
//on recherche avec un nom comme correspondance, ici paul pour l'exemple
```

```
$req = mysql_query("SELECT id,email,date from test_tbl where prenom ='ady' ");
```

```
//on ressort la requête dans un tableau
```

```
$result = mysql_fetch_array($req);
```

```
// affichage du resultat
```

```
print "\nid -> '$result[id].' email -> '.result[email].' -> '$result[date].'<br>";
```

```
?>
```

// Affichera à l'écran

l'id 3 email -> ady@utt.fr date -> 30-09-2003

Explications : Vous pouvez utiliser plusieurs instructions avec la clause where, par exemple :

```
$req = mysql_query("SELECT id,email from test_tbl where prenom ='ady' AND date = '30-09-2003' ");
```

Vous auriez eu le même résultat.

Utiliser where et LIKE comme élément de comparaison

<?

//on se connecte à la base de donnée

```
$db = mysql_connect('hote de connexion','votre login','votre mot de passe') or die ("erreur de connexion");
```

//on sélectionne votre base

```
mysql_select_db('nom de la base',$db) or die ("erreur de connexion base");
```

//on recherche avec la lettre t en début de chaîne

```
$req = mysql_query("SELECT email from test_tbl where prenom LIKE 't '");
```

//on ressort la requête dans un tableau

```
while ( $result = mysql_fetch_array($req) )
```

```
{
```

// Affichage du resultat

```
print 'le resultat de la requête est '.$result[email]. '<br>';
```

```
}
```

?>

Affichera à l'écran tous les prénoms commençant par p

le resultat de la requête est tomaka@utt.fr

Autre requete LIKE

<?

//on se connecte à la base de donnée

```
$db = mysql_connect('hote de connexion','votre login','votre mot de passe') or die ("erreur de connexion");
```

//on sélectionne votre base

```
mysql_select_db('nom de la base',$db) or die ("erreur de connexion base");
```

//on recherche avec une lettre, le a qui se trouve dans les prénoms

```
$req = mysql_query("SELECT email from test_tbl where prenom LIKE '%a%' ");
```

```
//on ressort la requete dans un tableau
```

```
while ( $result = mysql_fetch_array($req) )
```

```
{
```

```
// Affichage du resultat
```

```
Print 'le resultat de la requête est '.$result[email].'<br>';
```

```
}
```

```
?>
```

Cette requête vous permet de faire une recherche rapide lettre ou mot dans une phrase, ici on recherche la lettre a dans le prenom, donc affichera uniquement les prenom contenant un 'a' gardez ça en mémoire ça va vous servir un peu plus tard.

Les opérateurs de comparaison mysql

| Opérateur | Description | Exemple |
|-----------|-------------------|----------------------|
| = | Egal | where prenom = 'ken' |
| != | Inégal | where prenom !='ken' |
| <= | Inférieur ou égal | where id <= '2' |
| < | Inférieur | where id < '2' |
| >= | Supérieur ou égal | where id >= '2' |
| > | Supérieur | where id > '2' |

Les opérateurs logiques mysql

| Opérateur | Description | Exemple |
|-----------|-------------|-----------------------------------|
| OR / | ou | where prenom = 'ken' id = '2' |
| AND / && | et | where prenom = 'ken' AND id = '3' |

Chapitre XVII

Liste de quelques fonctions mysql

Voici donc une liste (non exhaustive) des fonctions de mysql.

Vous trouverez beaucoup plus de fonction en consultant le manuel de php

<http://www.php.net/manual/fr>

Si vous avez bien suivi les cours précédents vous devriez en connaître déjà plusieurs !

Fonction Description

mysql_close() Ferme la connexion au serveur sql

mysql_connect() Ouvre une connexion au serveur sql

mysql_fetch_array() Fournit une sortie comme tableau associatif

mysql_fetch_row() Fourni une sortie comme tableau indexé

mysql_insert_id() Fourni le numéro du dernier id inséré

mysql_numrows() Fourni le nombre d'enregistrement pour la requête

mysql_query() Envoi une requête au serveur sql

mysql_select_db() Sélectionne une base de donnée

D'autres fonctions de sortie très utiles

'Order by' vous permet de choisir un ordre de sortie, en précisant 'Asc' vous aurez une sortie en ordre croissant, et avec 'Desc', la sortie sera en ordre décroissant.

Un exemple de requête avec 'order' :

```
$req = mysql_query("Select prenom,email,date from test_tbl Order by id Asc");
```

Affichera le contenu de la table de l'id 1 à l'id 4, si vous remplacez 'Asc' par 'Desc', la sortie sera de 4 à 1

'Limit' vous permet de limiter le nombre de ligne en sortie de requête, exemple :

```
$req = mysql_query("Select prenom,email,date from test_tbl Limit 0, 20 ");
```

Sélectionne à partir de la ligne 0 (la 1ere enregistrée) et en affiche 20, vous pouvez sélectionner les 20 enregistrement à partir de n'importe quelle ligne, 'Limit 10, 20' par exemple.

Pour le prochain cours, vous allez trouver un exemple complet pour le codage d'un livre d'or, cela vous permettra de mieux comprendre le fonctionnement de mysql et de php.

Chapitre XVIII

Utilisation des fonctions PHP

Nous allons voir dans ce cours, l'utilisation des fonctions.

Les fonctions vont sûrement vous faire changer votre façon de programmer. Lors de la l'exécution / création de votre première fonction, vous allez tout remettre en cause.

Vos premières questions seront sûrement :

- Vais-je refaire tout mon site via les fonctions ?
- Est ce que je vais pouvoir créer de nouveaux services sur mon site ?
- etc...

Bref, comme vous l'aurez peut-être remarqué dans ces quelques lignes de présentation, nous allons aborder un thème très important de php.

Notre première fonction

Notre première fonction va consister à afficher un menu sur notre page accueil.php . Nous allons appeler cette fonction: `display_menu()`. Les parenthèses sont l'élément essentiel qui nous permet d'identifier une fonction.

Le but de ce code est comme je l'ai dis dans les lignes précédentes, d'afficher un menu sur notre page accueil.php.

Nous allons taper le code suivant à l'endroit où nous voulons faire apparaître le menu :

Fichier accueil.php

```
<html>
<title>Les fonctions en PHP</title>
<body>

<h1>Bienvenue sur mon site !</h1>
<br><br>
<center>

<?
// Appel de la fonction display_menu()
display_menu();
?>

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

Après avoir tapé ce petit bout de code, vous commencez à vous dire que vous auriez mieux fait d'écrire votre menu à la main 😊

Passons maintenant à une phase un peu plus intéressante, celle de la création de la fonction et du fichier qui va comporter toutes les fonctions de notre site.

Fichier fct.php

```
<?
// Fichier contenant toutes les fonctions du site

// Display_menu()
function display_menu()
{
?>
<!-- Le menu écrit en html -->
<a href="accueil.php">Accueil</a>
<?
}
?>
```

Après avoir écrit ce code, nous pouvons dire que :

- toute fonction doit être placée entre des accolades ({et}).
- on peut bien évidemment mélanger Php/Html.
- pas besoin de ; à la fin de notre code.

Si vous exécutez votre page accueil.php dans votre navigateur, nous obtiendrez un message d'erreur vous disant que la fonction display_menu() n'a pas été trouvée.

Pour que le fichier fonctionne, il vous suffit simplement de rajouter le code suivant dans votre accueil.php

```
<?
// Fichier contenant les fonctions
require_once("fct.php");
?>
```

Si vous exécutez à nouveau votre fichier dans le navigateur, vous verrez votre menu apparaître.

Il est temps de passer à l'étape suivante.

Passage de paramètre dans une fonction.

Une fonction ne sert bien évidemment pas seulement qu'à afficher des valeurs définit, mais également des valeurs dynamiques via des paramètres que nous déclarons dans les parenthèses de la fonction.

L'exemple que nous allons prendre, est celui d'une fonction qui permet d'afficher un lien hypertexte avec diverses informations sur ce dernier.
Nous allons appeler cette fonction: display_url()

```
<?
/* Informations:
- le paramètre $url contient l'url du lien
- le paramètre $target définit le cadre de sortie
- le paramètre $info affiche une info bulle concernant le lien
- le paramètre $name contient le mot qui va avoir le lien hypertext/
*/

function display_url($url, $target, $info, $name)
{
// Url
?>
<a href="<?=$url?>" target="<?=$target?>" title="<?=$info?>"><?=$name?></a>
<?
}
?>
```

Comme vous pouvez le remarquer, les paramètres s'écrivent exactement comme des variables via le symbole "\$".
Maintenant, nous allons voir comment utiliser notre fonction.

```
<? display_url("accueil.php", "_top", "Accueil du site", "Accueil");?>
```

Cette fonction affichera donc un lien nommé "Accueil" qui pointera vers une page "accueil.php" et qui s'ouvrira dans le même cadre (_top = cadre entier), le lien aura comme légende: "Accueil du site".
N'oubliez pas d'inclure votre fichier de fonction au début de vos pages.

Retour des valeurs.

Les retours des valeurs, seront des valeurs retournées (logique ;o)). Les valeurs peuvent être TRUE (vrai) ou FALSE (fausse). Pour écrire ces retours, il vous faudra utiliser l'instruction "return".

La meilleure explication de l'instruction return, est d'après moi un exemple concret.

Dans ce premier exemple, nous allons retourner une variable.

```
<?
// La fonction db () va vous permettre de vous connectez à une base de donnée et de retourner
un résultat
function db()
{
```

```
// Déclaration d'une variable $result
$result = mysql_connect("nommduserveur", "login", "motdepasse");
// On teste la connexion
if (!$result)
// Erreur
return false;
// On teste la sélection de la base
if (!mysql_select_db("notredevotrebse"))
// Erreur
return false;
// C'est bon :-)
return $result;
}
?>
```

Dans ce deuxième exemple, nous allons retourner différentes valeurs.

```
<?
// La fonction read_news () permet de lire des news enregistrés dans une table
function read_news()
{
// Connexion à la base de donnée (cf. fonction précédente)
$con = db();
// On teste la variable $con
if (!$con)
// Erreur
return "Erreur ! --> Impossible de se connecter à la base. Réessayer plus tard...";
// Requete
$result = mysql_query("SELECT ..... FROM .....");
// On teste $result
if (!$result)
// Erreur
return "Erreur ! --> Impossible d'afficher les news en ce moment. Réessayer plus tard...";
// Affichage des enregistrements
while($row = mysql_fetch_array($result))
{
// Ici un code pour afficher vos enregistrements
}
// On retourne la valeur TRUE, donc c'est bon :-)
return true;
}
?>
```

Pour ainsi utiliser votre fonction read_news() vous procéderez comme dans les cours précédents:

```
<? read_news(); ?>
```

Pour informations, vous pouvez également retourner des valeurs décimales avec un code ce genre par exemple:

```
<?  
// .....  
  
if (mysql_num_rows($result)>0)  
return 1;  
else  
return 0;  
  
// .....  
>
```

Et voilà ce cours touche à sa fin. Vous posséder maintenant largement de quoi faire votre propre site php. En effet si vous avez tout compris vous pouvez être fier d'affirmer que le langage ne vous est pas inconnu. Il reste cependant pour ceux qui désirent aller plus loin plusieurs choses à voir, comme par exemple les sessions...

Les derniers chapitres seront consacrés à donner quelques exemples de script php. J'espère que vous l'avez appréciés. Si vous avez des remarques le concernant, des idées pour l'améliorer envoyer un mail à l'UNG (ung@utt.fr).

Chapitre XIX

Exemple de script pour un livre d'or

Dans ce cours, vous allez voir en détail comment programmer vos applications, l'exemple suivant vous servira de livre d'or.

Bon surtout pas de copier collé, le but n'étant pas de pomper inutilement, mais de comprendre et d'apprendre à programmer.

Maintenant au boulot.

Création de la table

```
#  
#structure de la table livre d'or  
#  
  
CREATE TABLE livre_tbl (  
id int NOT NULL auto_increment,  
date_verif varchar (20) NOT NULL,  
date varchar(20) NOT NULL,  
pseudo varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
message text NOT NULL,  
PRIMARY KEY (id)  
);
```

On a choisi plusieurs paramètres pour la structure de table, l'id, date_verif, la date au format US c'est à dire inversée, par exemple 2003-09-09 20:30, ensuite la date, celle au format français pour l'affichage, ensuite le pseudo, l'email, et le message, on déclare que l'id est la clé primaire.

Le formulaire d'ajout de message pour le visiteur (formulaire.html)

```
<html>  
<head>  
<title>Votre livre d'or</title>  
</head>  
<body bgcolor="#ffffff">  
<form method="post" action="ajouter.php">  
<input type="text" name="pseudo"><br>  
<input type="text" name="email"><br>  
<textarea name="message" wrap="VIRTUAL" cols="40" rows="10"></textarea><br>
```

```
<input type="submit" name="submit" value="Enregistrer">
</form>

</body>
</html>
```

Pas de détail ici, c'est un formulaire html !!

La page pour ajouter les données dans la base (ajouter.php)

```
<?
//-----
//
// Page d'insertion des données dans mysql
//
// Pourquoi je détaille l'entête ?? Pour que le code soit propre
// et la page identifiée rapidement
//
//-----

$mysql_host = 'localhost'; // entrer votre hote ici
$mysql_user = 'votre login'; // entrer votre login
$mysql_password = 'password'; // entrer votre password
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----
// Début du script, vérification des champs venant du formulaire

if(empty( $pseudo) || empty($message)) // si les champ pseudo ou message sont vides
{

echo<a href="javascript:history.back();">Cliquez ici pour compléter le formulaire</a>;

}
else{ // les champs sont ok, on insert dans la base

//on prépare la date au format US pour l'ordre de sortie
$date_verif = date("Y-m-d H:i");

//la date au format français
$date = date("d-m-Y");

// On supprime les caractères spéciaux
$pseudo = addslashes (htmlspecialchars($pseudo));
$email = addslashes (htmlspecialchars($email));
$message = addslashes (htmlspecialchars($message));

//on se connecte à la base de donnée
```

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

```
//sélection de votre base de donnée
```

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

```
//requête d'insertion dans la table
```

```
mysql_query("INSERT Into livre_tbl VALUES ('','$date_verif','$date','$pseudo','$email','$message') ") or die ("erreur requête");
```

```
//on ferme la connexion
```

```
mysql_close();
```

```
//on redirige le visiteur sur la page d'accueil, là ou s'afficheront les messages
```

```
header('location: accueil.php');
```

```
}  
?>
```

La page où s'afficheront les messages (accueil.php)

```
<?
```

```
//-----
```

```
//
```

```
// Page d'affichage des messages
```

```
//
```

```
//-----
```

```
$mysql_host = 'localhost'; // entrer votre hote ici
```

```
$mysql_user = 'votre login'; // entrer votre login
```

```
$mysql_password = 'password'; // entrer votre password
```

```
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base
```

```
//-----
```

```
//on se connecte à la base de donnée
```

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

```
//sélection de votre base de donnée
```

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

```
//la requête de sélection des messages
```

```
$req = mysql_query("SELECT date,pseudo,email,message from livre_tbl Order by date_verif Desc ") or die ("erreur requete");
```

```
//on ressort le contenu en boucle, et on stock dans un tableau avec mysql_fetch_array
while( $contenu = mysql_fetch_array ($req))
{

//on affiche le résultat
echo 'Posté le : '.$contenu[date].'\n';
echo 'Auteur : <a href="mailto:'.$contenu[email].'">'.$contenu[pseudo].'\n';
echo 'Message : \n'.$contenu[message].'\n';
echo '\n';

}

//on referme la connexion
mysql_close();

//----- FIN DU SCRIPT -----

?>
```

Et voila, c'est fini, vous avez un code complet, et la méthode pour créer vos scripts, c'est déjà ça de gagner ;-)

Je précise quand même pour les petits malins qui se seraient contenté uniquement de copier le code sans faire l'effort de comprendre un minimum, parce que c'est mieux quand c'est tout fait, que celui-ci n'est pas complet à 100%, et que si vous souhaitez que le livre d'or s'affiche comme il faut, il faudra chercher un peu dans les cours et faire marche un peu vos cellules grises même si elles font grève, et oui, dommage ;-))

Chapitre XX

Exemple de script pour un mini moteur de recherche

Le but de ce cours n'est pas de vous apprendre à construire un moteur performant, mais de vous montrer le fonctionnement d'une requête de recherche, donc il sera simple et détaillé. Nous allons considérer qu'il s'agit d'un annuaire.

Création de la table

```
#  
#structure de la table annuaire  
#  
  
CREATE TABLE annuaire_tbl (  
id int NOT NULL auto_increment,  
date varchar(20) NOT NULL,  
url varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
description varchar(255)NOT NULL,  
mots_cles varchar (255) NOT NULL,  
PRIMARY KEY (id)  
);
```

On choisi les paramètres classiques d'un annuaire soit l'url, l'email de l'auteur, la description du site (255 caractères maxi) , les mots clés idem, 255 caractères, et on déclare l'id comme clé primaire.

On crée un formulaire d'insertion :

Url du site
Email de l'auteur
Description du site
Mots clés

Code Html

```
<form method="post" action="ajouter.php">  
<table width="400" border="0" cellspacing="0" cellpadding="0" align="center">  
<tr>  
<td width="160" height="17"><b>url du site</b></td>
```

```
<td width="240" height="17">
<input type="text" name="url">
</td>
</tr>
<tr>
<td width="160"><b>E-mail de l'auteur</b></td>
<td width="240">
<input type="text" name="email">
</td>
</tr>
<tr>
<td width="160"><b>Description du site</b></td>
<td width="240">
<input type="text" name="description" maxlength="255" size="30">
</td>
</tr>
<tr>
<td width="160"><b>Mots cl&eacute;s</b></td>
<td width="240">
<input type="text" name="mots_cles" maxlength="255" size="30">
</td>
</tr>
<tr>
<td width="160">&nbsp;</td>
<td width="240">
<input type="submit" name="button" value="Enregistrer">
</td>
</tr>
</table>
</form>
```

Comme d'habitude on évitera les détails sur un formulaire html !

La page ajouter.php

```
<?
//-----
//
// Page d'insertion des données dans mysql
//
//Page ajouter.php
//
//-----

$mysql_host = 'localhost'; // entrer votre hote ici
$mysql_user = 'votre login'; // entrer votre login
$mysql_password = 'password'; // entrer votre password
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----
```

//Attention, je ne vais pas reprendre les vérifications de formulaires, vous les avez vu dans les cours précédents.

//on se connecte à la base de donnée

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

//sélection de votre base de donnée

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

//la date pour l'insertion

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

//remplacement des caractères

```
$url = addslashes (htmlspecialchars($url));  
$email = addslashes (htmlspecialchars($email));  
$description = addslashes (htmlspecialchars($description));  
$mots_cles = addslashes (htmlspecialchars($mots_cles));
```

//requête d'insertion dans la table

```
mysql_query("INSERT Into annuaire_tbl VALUES  
(','$date','$url','$email','$description','$mots_cles') ") or die ("erreur requête");
```

//on ferme la connexion

```
mysql_close();
```

//on redirige le visiteur sur une page de remerciement, je vous en laisse les soins

```
header('location: merci.php');
```

```
?>
```

Bon on a inséré des données, supprimé les caractères spéciaux, on est enfin prêt à se servir de l'annuaire, plutôt presque, il ne vous reste qu'à remplir la base ;-)

Le formulaire de recherche (Rechercher dans notre base)

Entrez un mot clé

```
<form method="post" action="recherche.php">
<table width="400" border="0" cellspacing="0" cellpadding="0" align="center">
<tr bgcolor="#000066">
<td colspan="2">
<div align="center"><font color="#FFFFFF"><b>Rechercher
dans n&ocirc;tre base </b></font></div>
</td>
</tr>
<tr>
<td width="164">&nbsp;</td>
<td width="236">&nbsp;</td>
</tr>
<tr>
<td width="164">Entrez mot cl&eacute;</td>
<td width="236">
<input type="text" name="recherche">
</td>
</tr>
<tr>
<td width="164">
<input type="button" name="Submit2" value="Rechercher">
</td>
<td width="236">&nbsp;</td>
</tr>
<tr>
<td width="164">&nbsp;</td>
<td width="236">&nbsp;</td>
</tr>
</table>
</form>
```

Toujours pas d'explications sur le html

La page recherche.php

```
<?
//-----
//
// Page de recherche avec le mot ou les mots clés
//
// recherche.php
//
//-----

$mysql_host = 'localhost'; // entrer votre hote ici
$mysql_user = 'votre login'; // entrer votre login
$mysql_password = 'password'; // entrer votre password
```

```
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----
//Attention, je ne vais pas reprendre les vérifications de formulaires, vous les avez vu dans les
cours précédents.

//on se connecte à la base de donnée
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de
connexion");

//sélection de votre base de donnée
mysql_select_db($mysql_base,$db) or die ("erreur de connexion
base");

//requête de recherche
$req = mysql_query("SELECT url,description, email from annuaire_tbl where mots_cles
LIKE '%$recherche%' Order by url Asc ");

//on compte les résultats
$res= mysql_numrows($req);

//on affiche le nombre de résultats trouvés
print 'Pour le mot '.$recherche.' il y\'a '.$res.' résultats <br>';

//-----
//on ressort les infos par ordres alphabétiques, voir order dans la requête

while( $sortie = mysql_fetch_array($req))
{

print '<b>Description du site:</b><br>';
print $sortie[description].<br>';
print '<a href=""'.$sortie[url].">'.$sortie[url].</a><br>';
print '-----'; // c'est juste une ligne de
séparation ;-)

} //fin de boucle

//on ferme la connexion
mysql_close();

?>
```

Bon j'ai comme l'impression que c'est terminé ! Alors inutile de répéter que comme dans le cours précédent, j'ai oublié certaines choses, si vous avez lu tous les cours, vous ne rencontrerez pas de problèmes, et attention, pas de copié collé, on tape sur le clavier avec ses p'tits doigts ! ;-)

Chapitre XXI

Afficher les résultats sur plusieurs pages

Pour améliorer vos codes, je vais vous expliquer simplement comment installer un lien pour afficher les pages suivantes. On reprend le code du livre d'or (cours n°18).

La requete originale de la page d'affichage

//on se connecte à la base de donnée

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

//sélection de votre base de donnée

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

//la requête de sélection des messages

```
$req = mysql_query("SELECT date,pseudo,email,message from livre_tbl Order by date_verif Desc ") or die ("erreur requete");
```

Avec ce type de requête, vous affichez toutes les données contenues dans le livre d'or, pour limiter la sortie, procéder comme suit.

1°/ On compte les messages que le livre contient

//on se connecte à la base de donnée

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

//sélection de votre base de donnée

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

//sélection total des données

```
$req_limit = mysql_query("Select id from livre_tbl");
```

```
$resultat = mysql_numrows($req_limit); //mysql_numrows donne le résultat de la requete
```

// Maintenant on va se servir du résultat pour limiter la sortie

```
$limite_page = '10'; //choisissez le nombre de messages par pages
```

```
$nombre_page = $resultat / $limite_page; // ici on divise le total par le nombre de messages que vous avez choisi
```

```
$nombre_total = ceil($nombre_page); // on arrondi le nombre de page pour ne pas avoir de virgules, ici au chiffre supérieur
```

```
$nombre = $nombre_total - 1; // on enlève 1 au nombre de pages, car la 1ere page est celle affichée ne fait pas partie des pages suivantes
```

```
if(isset($numero_page) || $numero_page != '0') // si la variable numéro de page a une valeur ou est différente de 0
```

```
{  
$limite_mysql = $limite_page * $numero_page; // on multiplie votre limite par le numéro de la page passée par l'url
```

```
}  
else{ // pas de variable numero_page
```

```
$limite_mysql = '0'; // la limite est de 0
```

```
}
```

Voilà, on a fait le plus gros, on a compté le nombre page et préparé les limites pour l'affichage, passons maintenant à la requete de sortie

La requete pour la page d'affichage

```
//la requête originale de sélection des messages
```

```
$req = mysql_query("SELECT date,pseudo,email,message from livre_tbl Order by date_verif Desc ") or die ("erreur requete");
```

```
//la requête modifiée de sélection des messages
```

```
$req = mysql_query("SELECT date,pseudo,email,message from livre_tbl Order by date_verif Desc Limit $limite_mysql , $limite_page") or die ("erreur requete");
```

Dans la nouvelle requête, nous fixons les limites, avec votre limite de messages par pages (\$limite_page) et la limite qui sera définie par le compteur (\$limite_mysql).

La dernière étape, le lien pour afficher les pages

```
if( $nombre != '0' && empty($numero_page)) // si le nombre de page n'est pas 0 et si la variable numero_page n'est pas définie
```

```
{  
echo '<a href="affichage.php?numero_page=1">Page suivante</a>'; // on passe la variable numéro_page à 1  
}
```

```
// Dans cette condition, la variable numero_page est incrémenté et est inférieure à $nombre
```

```
elseif($nombre !='0' && isset($numero_page) && $numero_page < $nombre)
{
    $suivant = $numero_page + 1; // on ajoute 1 au numéro de page en cours

    echo '<a href="affichage.php?numero_page='.$suivant.'">Page suivante</a>'; //le lien pour
    les pages suivantes

    echo '&nbsp;&nbsp; <a href="javascript: history.back();">Page précédente</a>'; // retour page
    précédente, ici version javascript
}

// Dans cette condition, le lien qui sera affiché lorsque le nombre de page a été atteint
elseif( $nombre !='0' && isset($numero_page) && $numero_page >= $nombre )
{
    echo '<a href="javascript: history.back();">Page précédente</a>';
}
}
```

Et voilà, vous avez tous les éléments entre vos mains, il vous suffit juste de l'insérer dans le code du livre d'or.

Ca ressemblera à ça :

//on se connecte à la base de donnée

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

//sélection de votre base de donnée

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

//sélection total des données

```
$req_limit = mysql_query("Select id from livre_tbl");
$resultat = mysql_numrows($req_limit); //mysql_numrows donne le resultat de la requête
```

// Maintenant on va se servir du résultat pour limiter la sortie

```
$limite_page = '10'; //choisissez le nombre de messages par pages
$nombre_page = $resultat / $limite_page; // ici on divise le total par le nombre de messages
que vous avez choisi
```

```
$nombre = ceil($nombre_page); // on arrondi le nombre de page pour ne pas avoir de
virgules, ici au chiffre supérieur
```

```
$nombre = $nombre_total - 1; // on enlève 1 au nombre de pages, car la 1ere page est celle
affiché ne fait pas partie des pages suivantes
```

```
if(isset($numero_page) || $numero_page != '0') // si la variable numéro de page a une valeur
ou est différente de 0
```

```
{
```

```
$limite_mysql = $limite_page * $numero_page; // on multiplie votre limite par le numéro de la page passée par l'url
```

```
}  
else{ // pas de variable numero_page  
  
$limite_mysql = '0'; // la limite est de 0  
  
}
```

```
/*-----*/
```

On met les liens pour éviter de les oublier ;-)

```
-----*/
```

```
if( $nombre != '0' && empty($numero_page)) // si le nombre de page n'est pas 0 et si la variable numero_page n'est pas définie  
{  
print '<a href="affichage.php?numero_page=1">Page suivante</a>'; // on passe la variable numero page à 1  
}
```

```
// Dans cette condition, la variable numero_page est incrémenté et est inférieure à $nombre  
elseif($nombre !='0' && isset($numero_page) && $numero_page < $nombre)  
{  
$suivant = $numero_page + 1; // on ajoute 1 au numero de page en cours
```

```
print '<a href="affichage.php?numero_page='.$suivant.'">Page suivante</a>'; //le lien pour les pages suivantes
```

```
print '&nbsp;&nbsp;&nbsp;<a href="javascript: history.back();">Page précédente</a>'; // retour page précédente, ici version javascript  
}
```

```
// Dans cette condition, le lien qui sera affiché lorsque le nombre de page a été atteint  
elseif( $nombre !='0' && isset($numero_page) && $numero_page >= $nombre )  
{  
print '<a href="javascript: history.back();">Page précédente</a>';  
}
```

```
/*-----*/
```

La suite du code livre d'or avec la nouvelle requete de sélection

```
-----*/
```

```
$req = mysql_query("SELECT date,pseudo,email,message from livre_tbl Order by date_verif  
Desc Limit $limite_mysql , $limite_page") or die ("erreur requete");
```

// La suite se trouve dans le chapitre XIX

C'est terminé, il ne manque plus qu'à mettre le code en application, alors au boulot, bon courage ;-)

Chapitre XXII

Exemple de script pour un forum

Vous êtes nombreux à vouloir réaliser vos applications en php, nous allons maintenant passer à une étape qui va vous intéresser : la construction d'un forum.

Bon toujours pas de copier collé, le but n'étant pas de pomper inutilement, mais de comprendre et d'apprendre à programmer.

On respire, et on y va

Création de la table

```
#  
#structure de la table forum  
#  
  
CREATE TABLE forum_tbl (  
id int NOT NULL auto_increment,  
reponse_id varchar (30) not null,  
date_verif varchar (20) NOT NULL,  
date varchar(20) NOT NULL,  
pseudo varchar (50) NOT NULL,  
email varchar(55) NOT NULL,  
sujet varchar (60) NOT NULL,  
message text NOT NULL,  
PRIMARY KEY (id)  
);
```

Plusieurs paramètres pour la table du forum, l'id qui sera incrémentée automatiquement à chaque message, reponse_id recevra l'id du message original à chaque nouvelle réponse, date_verif recevra la date au format us pour classer les sorties, date servira à afficher la date du post, le pseudo, l'email et message.

Le formulaire d'ajout de nouveau sujet pour le visiteur qu'on appellera nouveau.php

```
<form method="post" action="ajouter.php">  
<table border="0" width="350" align="center">  
<tr>  
<td width="100"><b>Pseudo</b></td>  
<td width="250"><input type="text" name="pseudo"></td>
```

```

</tr>
<tr>
<td width="100"><b>E-Mail</b></td>
<td width="250"><input type="text" name="email"></td>
</tr>
<td width="100"><b>Sujet</b></td>
<td width="250"><input type="text" name="sujet"></td>
</tr>
<td colspan="2" align="center"><b>Message</b></td>
</tr>
<tr>
<td colspan="2" align="center">
<textarea name="message" wrap="VIRTUAL" cols="40" rows="10"></textarea>
</td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="submit"
value="Enregistrer"></td>
</tr>
</table>
</form>

```

Pas de détail ici, c'est un formulaire html !! Remarquez cependant qu'il n'y a pas de balise `<html><body><head>` car nous allons nous servir de la fonction include.

Le formulaire d'ajout de réponse au sujet par le visiteur qu'on appellera reponse.php
`<form method="post" action="ajouter.php">`

`<input type="hidden" name="reponse_id" value="<? print $id; ?>">` // on ajoute dans ce formulaire un champ caché nommé `reponse_id` de façon à déterminer à quel id correspond la réponse apportée

```

<table border="0" width="350" align="center">
<tr>
<td width="100"><b>Pseudo</b></td>
<td width="250"><input type="text" name="pseudo"></td>
</tr>
<tr>
<td width="100"><b>E-Mail</b></td>
<td width="250"><input type="text" name="email"></td>
</tr>
<td width="100"><b>Sujet</b></td>
<td width="250"><input type="text" name="sujet"></td>
</tr>
<td colspan="2" align="center"><b>Message</b></td>
</tr>
<tr>

```

```
<td colspan="2" align="center">
<textarea name="message" wrap="VIRTUAL" cols="40" rows="10"></textarea>
</td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="submit"
value="Enregistrer"></td>
</tr>
</table>
</form>
```

Ne pas oublier le champ caché `reponse_id`, celui ci est très important dans la conception du forum

La page d'ajout des messages, nommée `ajouter.php`

```
<?
//-----
//
// Page ajouter.php
//
// Ajout des messages et réponses dans mysql
// Vérification des champs obligatoires et traitement des données
//
//-----

$mysql_host = 'localhost'; // entrer votre hote ici
$mysql_user = 'votre login'; // entrer votre login
$mysql_password = 'password'; // entrer votre password
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----
// Début du script, vérification des champs venant du formulaire

if( empty( $pseudo) || empty($sujet) || empty($message) ) // si les champ pseudo ,sujet ou
message sont vides
{

print'<a href="javascript:history.back();">Cliquez ici pour compèter le formulaire</a>';

}
else{ // les champs sont ok, on insert dans la base

//on prépare la date au format US pour l'ordre de sortie
$date_verif = date("Y-m-d H:i");

//la date au format français
```

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

// On supprime les caractères spéciaux
$pseudo = addslashes (htmlspecialchars($pseudo));
$email = addslashes (htmlspecialchars($email));
$message = addslashes (htmlspecialchars($message));

//on se connecte à la base de donnée
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de
connexion");

//sélection de votre base de donnée
mysql_select_db($mysql_base,$db) or die ("erreur de connexion
base");

//requête d'insertion dans la table
mysql_query("INSERT Into forum_tbl VALUES
('$reponse_id','$date_verif','$date','$pseudo','$email','$sujet','$message') ") or die ("erreur
requête");

//on ferme la connexion
mysql_close();

//on redirige le visiteur sur la page d'accueil, là ou s'afficheront les messages soit index.php
header('location: index.php');

}
?>
```

La page index de votre forum (index.php)

```
<?
//-----
//
// Page d'affichage des sujets, auteurs, dates
// index.php
//
//-----

$mysql_host = 'localhost'; // entrer votre hote ici
$mysql_user = 'votre login'; // entrer votre login
$mysql_password = 'password'; // entrer votre password
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----

//on se connecte à la base de donnée
```

```
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de connexion");
```

```
//sélection de votre base de donnée
```

```
mysql_select_db($mysql_base,$db) or die ("erreur de connexion base");
```

```
//la requête de sélection des messages, remarqué la clause where, on sélectionne seulement les sujets dont le champ reponse_id est vide, j'ai fixé ici une limite de sortie à 50 sujets, à vous de choisir celle que vous souhaitez, ou de vous servir du cours n°20 ;-)
```

```
$req = mysql_query("SELECT id,date,pseudo,email,sujet from forum_tbl where reponse_id = " Order by date_verif Desc Limit 0, 50 ") or die ("erreur requete");
```

```
//on ressort le contenu en boucle, et on stock dans un tableau avec mysql_fetch_array
```

```
while( $forum = mysql_fetch_array ($req))  
{
```

```
//on affiche le résultat, dans le lien hypertexte du sujet, on passe par l'url l'id original du sujet, de façon à sélectionner celui et les réponses dans la page lecture.php
```

```
print '<a href="lecture.php?id='.$forum[id]."'>'. $forum[sujet]. '</a> -Auteur : <a href="mailto:'.$forum[email]."'>'. $forum[pseudo]. '</a> posté le : '.$forum[date]. '<br>';  
print '<hr>';
```

```
}
```

```
//on referme la connexion
```

```
mysql_close();
```

```
include('nouveau.php'); // on inclus le formulaire pour les nouveaux sujets.
```

```
//----- FIN DE LA PAGE INDEX-----
```

```
?>
```

L'affichage se fait de façon basique, vous devrez vous même insérer votre code html, pour que le forum soit plus agréable à voir bien sur.

La page de lecture du message original et des réponses apportées, nommé lecture.php

```
<?
```

```
//-----
```

```
//
```

```
// Page d'affichage des messages et des réponses
```

```
// lecture.php
```

```
//
```

```
//-----
```

```
$mysql_host = 'localhost'; // entrer votre hote ici
```

```
$mysql_user = 'votre login'; // entrer votre login
```

```
$mysql_password = 'password'; // entrer votre password
```

```
$mysql_base = 'le nom de votre base de donnée'; // entrer le nom de votre base

//-----

//on se connecte à la base de donnée
$db = mysql_connect($mysql_host,$mysql_user,$mysql_password) or die ("erreur de
connexion");

//sélection de votre base de donnée
mysql_select_db($mysql_base,$db) or die ("erreur de connexion
base");

//-----
//
// La requête de sélection du message original dont l'id est celle passée par l'url
//
//-----
$req = mysql_query("SELECT id,date,pseudo,email,sujet,message from forum_tbl where id
=$id' ") or die ("erreur requete");

//on ressort le contenu et on stock dans un tableau avec mysql_fetch_array
$original = mysql_fetch_array ($req);

//on affiche le message original
print $original[sujet].<br>;
print'Auteur : <a href="mailto:'. $original[email]."'>'. $original[pseudo].</a> posté le :
'. $original[date].<br>;
print 'message: <br>';
print $original[message];

print '<hr>'; // ligne html de séparation

//-----
//
// On ressort maintenant les reponse_id ayant le même numéro id
// que le message original
//
//-----
$rep = mysql_query("SELECT id,date,pseudo,email,sujet,message from forum_tbl where
reponse_id ='$id' ") or die ("erreur requete reponse ");

//on ressort le contenu en boucle et on stock dans un tableau avec mysql_fetch_array
while ( $reponse = mysql_fetch_array ($rep)){

// on affiche les réponses

print $reponse[sujet].<br>;
print'Auteur : <a href="mailto:'. $reponse[email]."'>'. $reponse[pseudo].</a> posté le :
'. $reponse[date].<br>;
```

```
print 'message: <br>';
print $reponse[message];

print '<hr>'; // ligne html de séparation pour chaque réponses

}

//on referme la connexion
mysql_close();

include('reponse.php'); //on inclu le formulaire pour les réponses à ce sujet
//----- FIN DU SCRIPT LECTURE-----

?>
```

L'affichage est plutôt basique, à vous de personnaliser.

Vous avez remarqué que nous avons inséré le formulaire de réponse en bas de la page lecture, souvenez vous du champ caché `reponse_id`, celui-ci porte maintenant le numero de l'id du message original, qui est passé par l'url, donc à chaque réponse, ce message sera classé de façon à pouvoir le ressortir à la suite des autres réponses. Ce champ est très important, c'est lui qui fait fonctionner le script correctement.

Voilà, encore un code de plus, il ne vous reste qu'à personnaliser l'ensemble et à le mettre en ligne, vous avez maintenant votre forum, il est donc unique .

Je précise quand même pour les petits malins qui se seraient contentés uniquement de copier le code sans faire l'effort de comprendre un minimum, parce que c'est mieux quand c'est tout fait, que celui-ci n'est pas complet à 100%, et que si vous souhaitez que le forum s'affiche comme il faut, il faudra chercher un peu dans les cours et se creuser les méninges, et oui, dommage ;-))

Chapitre XXIII

Exemple de script pour un système de vote (Sondage)

Dans ce cours, je tenterais de vous expliquer comment créer votre propre sondage sans chercher à installer d'autres scripts, car il est toujours préférable de créer le votre afin de mieux l'adapter à vos besoins et vos préférences.
Alors on prend son souffle et on y va ! :-)

Des choses à savoir :

Afin de bien réussir ce script vous aurez besoin de bien assimiler certaines choses simples que je pense tout le monde pourra facilement comprendre et qui se rapporte essentiellement à mysql:

Les variables d'environnement.

La requête Mysql INSERT, SELECT et WHERE

Si vous n'arrivez pas à comprendre ça je vous conseille de voir la documentation Mysql de Nexen.

Commençons par le tout début - PhpMyAdmin :

Avant toute autre chose, il nous faudra créer notre table dans laquelle seront stockés les votes, pour cela nous utiliserons PhpMyadmin en y appliquant la requête suivante :

Créer la table (sous PhpMyAdmin par exemple)

```
CREATE TABLE sondage (  
  id int(11) NOT NULL auto_increment, // Champs indexé  
  vote tinyint(4) NOT NULL, // Champs de stockage des votes des visiteurs  
  ip char(50) NOT NULL, // Stockage de l'ip du voteur (système anti-triche)  
  UNIQUE KEY (id)  
) Type=MyISAM;
```

Création de l'interface de vote :

Ce dont nous aurons besoin ici n'est que du code html que je n'expliquerais pas car là n'est pas le but de ce cours. Vous avez sûrement du voir des sondages sur le net, il nous faudra des radios (autant de radios que de réponse possible) et un bouton de validation comme ceci :

Formulaire de vote (HTML)

```
<form method='post' action='votiez.php'>  
<input type=radio name="vote" value="1"> // Première réponse possible  
Très cool<br>  
<input type=radio name="vote" value="2"> // deuxième réponse possible
```

```
Assez Cool<br>
<input type=radio name="vote" value="3"> // troisième réponse possible
Booof<br>
<input type=radio name="vote" value="4"> // quatrième réponse possible
Nuul à chier<br>
<br>
<center>
<input type=submit value="Valider" name="submit"> // bouton de validation
</center>
<br>
</form>
```

Notons que les valeurs des votes sont stockées dans les value des radios, ici pour ne pas trop nous compliquer la vie, nous les nommerons 1, 2, 3 et 4 et ce sont ces mêmes valeurs que nous insérons dans la table et que nous pourrons récupérer pour afficher les résultats grâce à la clause WHERE.

Voyons ensemble ce que donne ce code en visuel :

Comment trouvez vous le site ?

Très cool
Assez Cool
Booof
Nuul

Ici nous proposons au visiteur de donner son avis sur notre site grâce à des radios où il choisit son vote. Le bouton valider mène vers le script d'ajout contenant les requêtes mysql nécessaire à l'insertion dans la table.

Requête d'insertion dans la table

Passons maintenant au code d'insertion dans la table, il se fera de la manière suivante :

Voici le code de notre fichier votez.php

```
<?php
// On inclut le fichier de configuration avant tout
require ("config.inc.php");
// On ouvre une connections à notre serveur
```

```
mysql_connect("$hote","$user","$pass");

// On sélectionne notre base de données
mysql_select_db("$base");

// On définit la variable ip adresse
$ip = $REMOTE_ADDR;

// On créé la requête d'insertion dans la table
$sql = "INSERT INTO (vote,ip) VALUES ('$vote','$ip)";

// On exécute la requête SQL
mysql_query($sql);

// On redirige vers la page de sondage
// http_referer est l'adresse de la page qui nous menés à votez.php
header(location:"$HTTP_REFERER");

// On ferme la connections à mysql
mysql_close();

?>
```

Affichage des résultats

Dans cette partie, vous verrez comment afficher les résultats de manière basique car les améliorations, c'est à vous de les faire, le but ici est de vous expliquer le fonctionnement pour le reste c'est à vous d'imaginer. ! :p

Ici pour récupérer les résultats nous aurons recours à la requete SELECT, car nous sélectionnerons tous les champs où vote = 1 ou 2 etc et nous procéderons au comptage des lignes grâce à mysql_numrows. Regarder le code pour comprendre :

Code du fichier result.php

```
<? php

// On inclut le fichier de configuration
include ("config.inc.php");

// On définit le nom de la table
$table = "sondage";

// On se connecte au serveur sql
mysql_connect("$hote","$user","$pass");

// On sélectionne notre base de données
```

```
mysql_select_db("$base");

// On sélectionne les lignes où vote est 1 --> "très bien" dans notre exemple
$s1 = "SELECT id,vote,nbr_vote FROM $table where vote='1'";

// On exécute la requete de sélection
$r1 = mysql_query($s1);

// On compte le nombre de lignes où vote est égal à 1
$numvote1 = mysql_numrows($r1);

// On fait la même opération pour les autres votes.
$s2 = "SELECT id,vote,nbr_vote FROM $table where vote='2'";
$r2 = mysql_query($s2);
$numvote2 = mysql_numrows($r2);

$s3 = "SELECT id,vote,nbr_vote FROM $table where vote='3'";
$r3 = mysql_query($s3);
$numvote3 = mysql_numrows($r3);

$s4 = "SELECT id,vote,nbr_vote FROM $table where vote='4'";
$r4 = mysql_query($s4);
$numvote4 = mysql_numrows($r4);

/*
 * Dans le reste du code, nous allons afficher les résultats sous
 * forme de table, à vous d'arranger l'apparence que vous voulez
 */

// Ceci est la table où apparaissent le nombre de vote pour chaque réponse
echo "<table width=\"40%\" cellspacing=\"0\" cellpadding=\"0\">";
echo "<tr><td>Très bien : </td><td> $numvote1</td></tr>";
echo "<tr><td>Assez bien : </td><td> $numvote2</td></tr>";
echo "<tr><td>Booof : </td><td> $numvote3</td></tr>";
echo "<tr><td>Nul à chier : </td><td> $numvote4</td></tr>";
echo "</table>";

// On sélectionne toute la table
$sql = "SELECT id FROM $table";

// On compte le nombre total de lignes
$total = mysql_numrows(mysql_query($sql));

// Affichage du nombre total de votes
echo "<table width=\"40%\" cellspacing=\"0\" cellpadding=\"0\">";
echo "<tr><td>Nbr total des votes : $total</td></tr></table>";

// Fermeture de la connections à mysql
mysql_close();
?>
```

Le fichier de configuration standard

Il n'y a rien à expliquer, ceci est le fichier de configuration qui contiendra nos variables de connections.

```
<?php

// Adresse de votre serveur sql (ex : sql.free.fr pour free)
$hote = "localhost";

// Votre login chez votre hébergeur
$user = "";

// Votre pass correspondant à ce login
$pass = "";

// Nom de votre base
$base = "az-php";

?>
```

Eh voilà ce chapitre touche à sa fin, j'espère que vous saurez après ça créer votre propre système de vote, car toutes les explications sont données et normalement il ne devrait pas y avoir de problèmes.

Bien sûr ce cours ne vous donnera pas un script parfait car il reste plusieurs choses à faire mais là encore c'est à vous d'imaginer les diverses possibilités à ajouter, tel que :

Affichage graphique des résultats

Zone d'administration des votes

Système anti-triche (vérification des ips)

Chapitre XXIV

Exemple de script pour un système de vote (Sondage)

J'espère qu'en lisant les cours du site, vous aurez acquis un certain niveau en codage PHP, ce qui nous permettra d'aborder quelques sujets, je ne dirais pas avancés mais bon. Dans le cas de ce cours nous étudierons les sessions (apparues avec PHP4) qui nous permettent de faire passer nos variables à travers les différentes pages. Certes il y a d'autres façons tel que les querystrings (passage des variables par l'url) mais il est plus élégant et plus sécurisé d'utiliser les sessions.

Voyons voir maintenant comment fonctionnent ces sessions par un petit exemple concret :

Gestion d'un espace membres.

Sommaire

Formulaire d'identification

Script d'identification

Ouverture de la session

Enregistrement des variables

Vérification de l'identification du membre

Destruction de la session

Récapitulatif de toutes les fonctions

Pour que vous compreniez mieux le fonctionnement des sessions nous créerons un espace membre, mais nous n'utiliserons que les trucs basiques.

Commençons par le principal : la table "membres".

Nous ne créerons pas une multitude de champs dans notre cas, nous n'aurons besoin que de 3 champs : id, login et pass qui nous serviront à vérifier l'identité du membre.

```
CREATE TABLE membres ( // création de la table
id tinyint(4) NOT NULL auto_increment, // le champs id qui sera indexé
login varchar(50) NOT NULL, // champs pour stocker le login du membre
pass varchar(20) NOT NULL, // pour stocker le pass du membre
UNIQUE KEY (id)
);
```

Le formulaire d'identification

Jusque là tout va bien pour vous, c'est du déjà vu, le formulaire d'identification ne risque pas non plus de poser de problème vu que ce n'est que du html. Voici le code :

```
<table width="300" cellspacing="1" border="1"><form method="post"
action="identifier.php">
```

```

<tr><td bgcolor="darkblue"><center><font color="white">Formulaire
d'identification</font></center></td></tr>
<tr><td><center><input type="text" name="login"></center></td></tr>
<tr><td><center><input type="password" name="pass"></center></td></tr>
<tr><td><center><input type="submit" value="Identifier"></center></td></tr>
</form></table>

```

Passons à notre premier code : l'identification

Nos données concernant les membres sont stockées dans une table, il nous faudra donc nous connecter à la base, et sélectionner les login et les pass et voir s'il y en a un qui concorde avec ceux qu'a entré le visiteur si oui on le redirige vers la zone membre et on ouvre la session, sinon on lui réaffiche le formulaire d'identification.

NB : Je vous conseille de lire les commentaires du code c'est là que tout se passe.

```
<?php
```

```
// ON inclut le fichier de configuration
require ("config.inc.php");
```

```
// On se connecte à notre serveur Sql
mysql_connect($server,$user,$pass);
```

```
// On sélectionne la base de données
mysql_select_db($base);
```

```
// On sélectionne quand le champs login correspond au login entré
// et le champs passe au pass entré.
$sql = "SELECT * FROM membres WHERE login=$login and pass=$pass";
```

```
// On exécute la requête de sélection
$res = mysql_query($sql);
```

```
// On compte le nombre de ligne des résultats
// 1 : si valide 0 si aucun login ne correspond
$exist = mysql_num_rows($res);
```

```
// Si la variable $exist = 0 --> login inexistant ou faux pass
if(!$exist) {
    // On affiche ce message d'erreur
    echo "<center>Veuillez vérifier vos données</center>";
```

```
    // On inclut le formulaire d'identification
    include("formulaire.php");
}
```

```
// Sinon, si le login et pass sont valides
else {
```

```
// On ouvre la session
session_start();

// On enregistre la variable login qu'on fera passer sur toutes les pages
// ATTENTION : Notez bien l'absence de $ devant login
session_register("login");

// On affiche un message de bienvenu
echo "<center>Bienvenu sur votre espace membre : $login</center>";

// Notons que la session a un nom par défaut et un identifiant
// Nom par défaut : PHPSESSID
// Id Aléatoire généré lors de l'ouverture de la session

// Définissons d'abord les variables
$sess_nom = session_name();
$sess_id = session_id();

// Affichons si on veut le nom est l'id de la session
echo "Le nom de la session est :<br> ";
echo $sess_nom." et l'Id est : ".$sess_id."";

// On affiche maintenant un lien vers une page d'essai
// Notez qu'on ne passe pas les variables par l'url
echo "<a href=\"page.php\">Page Essai</a>";

// A la fin on offre au membre la possibilité de se déconnecter
echo "<a href=\"deconnexion.php\">Déconnexion</a>";
}

?>
```

Décortiquons le code ci dessus

Nous allons maintenant récapituler le code ci dessus afin que tout soit vraiment clair dans votre petite tête de petit programmeur ! :-)

Alors on affiche au visiteur le formulaire d'identification où il entre son login et son pass.

On cherche dans notre base de données si le login entré existe, et si oui on vérifie si le password entré est aussi juste.

En cas d'erreur, on affiche le message d'erreur et on réaffiche le formulaire pour que le visiteur puisse réessayer (bien sûr dans une application avancée on pourrait limiter le nombre d'essais à 3).

Si les login et pass sont corrects, on ouvre la session (avec `session_start();`) et on enregistre les variables à passer entre les pages (avec `session_register(variable);`) notez qu'il n'y a pas de '\$' devant la variable lors de l'enregistrement.

Maintenant nous allons étudier de près cette "page.php" et voir comment nous ferons pour vérifier si le visiteur est identifié, et les actions à effectuer selon le cas.

Comment vérifier si le visiteur est identifié

Vérifier si le visiteur est identifié, c'est tout simplement vérifier si la variable \$login est déclarée (isset) la façon la plus simple de procéder est celle ci :

Code du fichier page.php (1ère façon)

```
<?php

// On commence par inclure le fichier de configuration
require ("config.inc.php");

// On vérifie si la variable login est déclarée
// Si elle ne l'est pas on redirige vers le formulaire
if(!$login) {
    echo "<center>Veuillez vous identifier d'\abord</center>";
    include ("formulaire.php");
}

// Si la variable est déclarée on affiche ce message
else {
    echo "Bienvenu sur votre espace membre $login";
}

?>
```

Cependant, le code ci dessus contient une importante faille de sécurité (j'espère que vous l'aviez deviné) car n'importe quel visiteur peut entrer l'url avec la variable login déclarée et de ce fait le script php voyant la variable, n'y verra que du feu et les données de votre site se trouveront menacées (Url trompeuse <http://www.votresite.com/page.php?login=quelconque>).

Corrigeons ce bug

Nous allons voir la deuxième méthode possible ou on utilisera la vérification des sessions afin d'éviter la faille exposée ci dessus. Pour cela on aura recours à la fonction :
if(!session_is_registered(variable))

```
<?php

/*****
* Voici la version non défectueuse
* du fichier page.php, c'est ce même
* code que vous devrez copier sur toutes
* les pages nécessitent un enregistrement.
*****/
```

```
// On inclut le fichier de configuration
require ("config.inc.php");
```

```
// On ouvre la session avant tout
```

```
session_start();

// On vérifie si la session "login" est enregistrée
// Si la session "login" n'est pas enregistrée
if(!session_is_registered(login)) {
// On affiche ce message d'erreur
echo "<center>Vous devez vous identifier pour accéder à cette page</center>";

// On affiche le formulaire d'identification
include ("formulaire.php");
}

// sinon, la session "login" est enregistrée
else {
// On affiche le message de bienvenue
echo "<center>Bienvenu sur votre espace membre $login</center>";
}

?>
```

Et voilà vous saurez maintenant comment vérifier que votre visiteur est identifié, nous reste maintenant le fichier deconnexion.php qui comme son nom l'indique permettra au visiteur de se déconnecter (destruction de la session).

```
<?php

// On commence par ouvrir la session
session_start();

// On désenregistre la session login
session_unregister("login");

// On supprime toutes les variables de la session
session_unset();

// On détruit totalement la session
session_destroy();

// On affiche le message de réussite
echo "Vous n'êtes plus connecté !";

// On réaffiche le formulaire d'identification
include ("formulaire.php");

?>
```

Pour configurer le fichier de configuration je pense que vous n'aurez pas trop de mal :) on voit ça à chaque fois donc !

```
<?php  
  
// On déclare les variables de connections  
$server = "localhost";  
$user = "";  
$pass = "";  
$base = "az-php";  
  
?>
```

! ATTENTION !

la fonction `session_start()`; doit être lancée avant tout envoi de code html, print echo ou autre affichage, si vous ne respectez pas cette condition, vous aurez des erreurs de HEADERS que l'on voit souvent ;-)

Nous allons maintenant reprendre toutes les fonctions de sessions que nous avons vues et utilisées au cours de la création de notre script.

Récapitulation des fonctions de sessions

- `session_start()` démarre une session
- `session_name()` affiche le nom de la session (par défaut PHPSESSID)
- `session_id()` affiche le numéro de session
- `session_register()` enregistre une variable pour la session en cours
- `session_unregister()` supprime une variable de session
- `session_is_registered()` vérifie que la variable de session est enregistrée
- `session_unset()` détruit toutes les variables de session enregistrées
- `session_destroy()` détruit la session en cours

Vous avez maintenant toutes les cartes en mains pour réaliser vos scripts et manipuler les sessions.

On peut bien sûr améliorer ce script de zone membre en ajoutant une interface d'administration et autres options utiles.