

Pourquoi ce supplément ?

Le manuel que vous avez acheté va vous permettre d'exploiter avec efficacité les bases de données MySQL et le PHP pour gérer un site web dynamique.

Pour des raisons évidentes de place, l'auteur a donc choisi de ne pas trop s'attarder sur la partie « initiation » au PHP et à MySQL. Nous vous proposons ici de découvrir le PHP et MySQL en prenant le temps.

Vous trouverez dans ce (long) supplément :

- une approche simple et progressive du PHP et de MySQL
- La présentation de quelques concepts et de fonctions plus élaborés qui ne sont pas abordés dans ce manuel.
- Enfin, vous y trouverez aussi le guide d'installation et de configuration d'un serveur web local Apache à l'aide de EasyPHP.
- Une liste de fonctions PHP, une liste des hébergeurs et de leur offre sur le PHP et MySQL.

Contrairement à un manuel à part entière, les points développés dans le supplément ne sont pas nécessairement liés entre eux : les exemples ne se suivent pas et chaque titre du sommaire peut-être lu indépendamment des autres.

Ce supplément peut-être lu avant et après le manuel que vous avez acheté.

Nous vous conseillons même de l'imprimer pour l'avoir sous la main... Avec l'habitude, vous finirez par ne plus jamais devoir l'ouvrir ;-)

En espérant que ce supplément vous sera utile, et en vous souhaitant une excellente lecture,

David Bosman

Pourquoi PHP?

Il existe d'autres langages tels Perl et ASP, ...

Mais :

- PHP est gratuit, alors que ASP ne l'est pas.
- L'avantage de PHP par rapport à Perl est qu'il est très répandu (vous trouverez énormément d'exemples gratuits), ses fonctions sont intégrées (vous pouvez utiliser toutes les fonctions sans vous soucier de les charger à l'avance), et que les fonctions d'accès aux différents logiciels de bases de données sont quasiment identiques.
- PHP est très simple et vous pourrez très rapidement mettre en pratiques vos nouvelles connaissances !
- Même si PHP est très simple, il propose des **extensions** très performantes qui permettent la manipulation d'images (création, modification), la gestion de bases de données, la génération de fichiers PDF ou d'applications ShockWave Flash, etc.

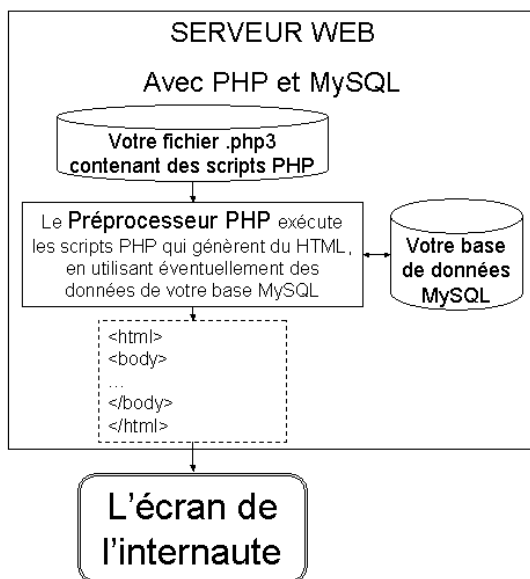


Les hébergeurs ne fournissent qu'un certain nombre de ces extensions avec PHP. C'est pourquoi il est important de bien choisir son hébergeur en fonction de ses besoins (voyez en annexe notre récapitulatif des offres)

Pourquoi MySQL ?

pour les mêmes raisons...

Principe de fonctionnement de PHP/MySQL



Les programmes PHP sont des fichiers textes ayant pour extension .php3 ou .php.

Lorsqu'un internaute entre l'URL de votre fichier PHP (par exemple `http://www.votre_site.com/index.php3`), celui-ci est lu et exécuté sur le serveur de l'hébergeur qui fait appel au module PHP.EXE

Le programme génère d'abord le code HTML grâce aux fonctions **echo** et **print**.

Enfin ce code HTML est envoyé vers le navigateur de l'internaute, qui affiche le résultat.

Voici un exemple de programme PHP :

```
<?
echo "Bonjour !";
?>
```

Une fois interprété, ce code afficherait dans un navigateur la phrase : Bonjour !
Ouvrez votre éditeur de texte et tapez le code qui suit (enregistrez le sous le nom : text.php3):

```
<?
echo"<html><head><title>exemple de PHP</title></head><body>";
echo "Numéros de 1 à 5:<br>";
for ( $i = 1 ; $i <= 5 ; $i ++ )
    {
        echo "$i<br>";
    }
echo"</body></html>";
?>
```

Ce script se décompose comme suit:

- les balises d'ouverture et de fermeture qui indiquent le script PHP : `<? et ?>`
- La fonction **echo** génère le code HTML qui sera affiché à l'écran
- **\$i** représente une variable. En PHP les variables sont toujours précédées du signe \$. Ici, la variable *i* prend une valeur qui va de 1 à 5 par incrémentation de un (`$i++` signifie `$i + 1`).
- **For** est une instruction de contrôle qui va répéter une instruction tant qu'une condition sera vérifiée. Ici tant que *\$i* sera plus petit ou égal (`<=`) à la valeur 5.
- lisez la boucle :
- pour *\$i* est égal à 1 ET *\$* plus petit ou égal à 5, augmenter *\$i* de un ;
- afficher « `$i
` » ;
- etc jusqu'à *\$i*=5 où on affiche une dernière fois sa valeur.
- à la boucle suivante *\$i*=6, la condition n'est donc plus valable : la boucle est interrompue : on affiche plus rien.
- Toutes ces opérations ont été exécutées de manière transparente sur le serveur de votre site, et le visiteur n'en percevra que le résultat (le code HTML généré est envoyé au navigateur):

```
Numéros de 1 à 5:
1
2
3
4
5
```



Si, depuis votre navigateur vous affichez le code source de la page, vous ne verrez que ceci :

```
<html><head><title>exemple de PHP</title></head><body>
Numéros de 1 à 5:<br>1<br>2<br>3<br>4<br>5<br></body></html>
```

Comme nous vous le disions, le navigateur ne reçoit du serveur que le code HTML qui a été généré par le programme PHP.

Malheureusement, pour pouvoir tester ce programme vous devez disposer d'un interpréteur PHP... ce n'est pas le cas normalement.

Si vous déposez d'un site qui supporte le PHP vous pouvez par exemple uploader votre page et l'afficher dans votre navigateur. Mais cela deviendrait vite pénible de faire ça pour chaque page et chaque fois que vous modifiez une page, non ?

Exécuter du PHP

Soit sur votre site web:

1. Vérifiez que votre hébergeur accepte le PHP et que votre compte PHP est activé.
2. Uploadez la page sur votre serveur.
3. Chargez dans votre navigateur en tapant le chemin d'accès, comme ceci :
`http ://www.votre_hébergeur.com/votre_site/ test.php3`

Soit sur votre PC :

- Très utile si vous comptez beaucoup travailler avec le PHP : vous réaliserez vos tests sans devoir passer par Internet, *en local*. ***Pour réaliser les tests en local, vous devez configurer votre ordinateur en SERVEUR WEB.***

CONFIGURATION SERVEUR WEB PERSONNEL

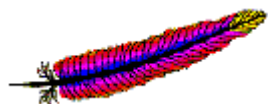


Il existe d'autres solutions que la nôtre : nous avons pris deux critères en compte : gratuité et simplicité.

Installer les logiciels

L'utilisation de PHP nécessite différents logiciels, en plus de PHP lui-même, dans les pages qui suivent nous vous proposons une solution clé en main : un programme qui installe tous les composants automatiquement, enfin presque ;-)

Les logiciels requis sont:



Apache

Transforme votre ordinateur en serveur Web. Il est gratuit et c'est probablement le serveur le plus utilisé sur Internet. Il existe des alternatives gratuites (Xitami, par exemple) ou payantes (Microsoft SSI)

[Http://www.apache.org](http://www.apache.org)



PHP

C'est le pré-processeur lui-même. Il va être appelé par Apache pour interpréter les scripts PHP

[Http://](http://)



MySQL

Pour gérer vos bases de données.

[Http://](http://)

phpMyAdmin

phpMyAdmin

phpMyAdmin n'est pas nécessaire pour faire tourner votre serveur web local avec PHP et MySQL, mais il est tellement pratique !

[Http://](http://)

Il existe deux manières d'installer ces logiciels :

- La façon simple c'est d'utiliser *easyphp*, qui les installe tous avec très peu d'interventions manuelles.

Recommandé aux utilisateurs débutants.

- La façon "bricoleur" : installer chaque logiciel individuellement, et configurer sur mesure votre installation, ce qui nécessite un peu plus d'interventions manuelles mais vous permet de profiter des dernières évolutions des logiciels.

Conseillé aux utilisateurs plus avertis.

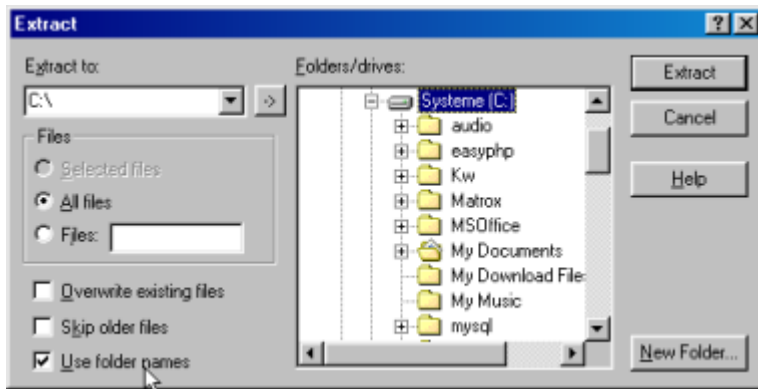
Installer easyPHP

Téléchargez le fichier d'installation sur <http://www.manucorp.com/>

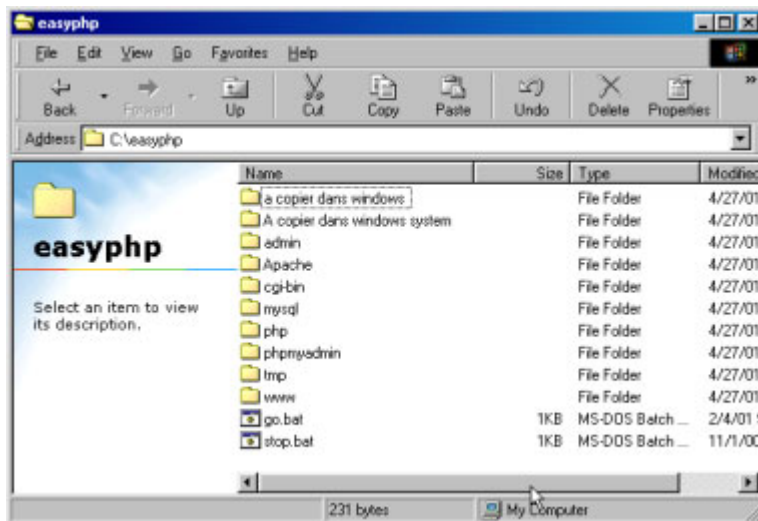


Lors de l'installation des fichiers, Windows peut signaler que **MSVCRT.DLL** est en cours d'utilisation par Windows lui-même et ne peut être remplacé. *Ignorez cette erreur !*

1. Décompressez les fichiers dans le répertoire **C:**,





2. Vérifiez bien que la case **Use Folder Name** est cochée (voir illustration ci-dessus) : un répertoire **easyphp** sera automatiquement créé dans **c:**, contenant des sous-répertoires tels Apache, PHP, mysql et phpmyadmin.
3. Quand l'extraction est achevée, ouvrez le répertoire **C:\easyphp**.



4. Copiez les fichiers du répertoire **a copier dans windows** (en fait, il ne contient que le fichier **php.ini**) dans le répertoire Windows (**c:\windows** pour Windows 95/98/ME, **c:\winnt** pour Windows 2000).
5. Copiez les fichiers du répertoire **A copier dans windows system** dans le répertoire système de Windows (**c:\windows\system** pour Windows 95/98/ME, **c:\winnt\system32** pour Windows 2000).

Démarrer et arrêter Apache et MySQL

-  **go.bat** • Pour démarrer Apache et MySQL, lancez le programme **C:\EASYPHP\GO.BAT** à partir de l'explorateur Windows
-  **stop.bat** • Pour arrêter Apache et MySQL, lancez le programme **C:\EASYPHP\STOP.BAT** à partir de l'explorateur Windows

Pour plus de facilité, créez un raccourcis vers ces deux fichiers dans votre **menu Démarrer** ou sur le **Bureau**

Tout d'abord, le programme démarre MySQL : vous voyez apparaître une première fenêtre DOS qui se ferme automatiquement après un bref instant.



En fait, MySQL tourne en arrière plan, et attend d'être appelé par PHP, comme nous le verrons plus loin.

Ensuite, le programme démarre Apache sous la forme d'une fenêtre DOS. Si tout va bien, vous voyez ceci :

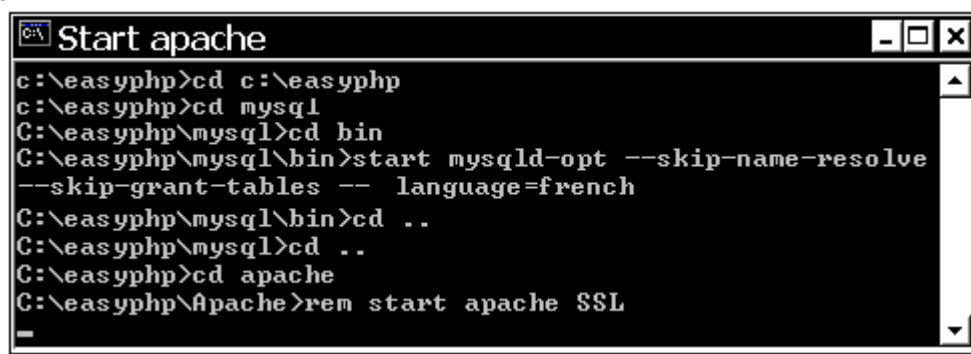


```
C:\easyphp\Apache\Apache.exe
[Sat Apr 21 21:03:35 2001] [warn] pid file
c:/easyphp/apache/logs/httpd.pid overwritten
-- Unclean shutdown of previous Apache run?
Apache/1.3.17 (Win32) PHP/4.0.5-dev running...
```



Vous ne devez pas fermer cette fenêtre, mais vous pouvez la *minimiser* dans la barre des tâches

Une autre fenêtre est affichée : il s'agit de la fenêtre d'exécution du programme `GO.BAT`. Le programme ayant terminé son travail de démarrage de MySQL et de Apache, *vous pouvez fermer cette fenêtre*.



```
C:\ Start apache
c:\easyphp>cd c:\easyphp
c:\easyphp>cd mysql
C:\easyphp\mysql>cd bin
C:\easyphp\mysql\bin>start mysqld-opt --skip-name-resolve
--skip-grant-tables -- language=french
C:\easyphp\mysql\bin>cd ..
C:\easyphp\mysql>cd ..
C:\easyphp>cd apache
C:\easyphp\Apache>rem start apache SSL
```



MySQL semble ne pas fonctionner si le PC possède plusieurs cartes ethernet (MySQL identifie l'adresse du PC par l'une des 2 adresses IP au hasard). Je n'ai pas trouvé de solution à ce problème avec la version actuelle de MySQL (si vous l'avez n'hésitez pas à nous la faire connaître afin que tous puissent en profiter !)



Pour vous simplifier la vie lors du lancement de Apache, il existe un petit utilitaire qui permet de faire tourner Apache en service : une icône s'affichera en bas à droite dans la barre des tâches (vous n'aurez plus cette fenêtre DOS dans la barre des tâches). Depuis cette icône vous pourrez facilement commander Apache. Le plus simple est d'installer cet utilitaire directement dans le répertoire **Démarrer|Programmes|Démarrage**

Vous pouvez le télécharger ici : <http://brian.threadnet.com/apmgr.html>


Vérifier la conformité de l'installation

Après avoir lancé GO.BAT, lancez votre navigateur Internet et saisissez <http://localhost>. Vous devez voir apparaître la page suivante :



Si cette page est bien affichée, cela signifie que Apache et PHP fonctionnent bien.

Si ce n'est pas le cas, il vous faut re-vérifier toutes les étapes précédentes.

 Vérifiez surtout que vous avez bien copié le contenu des répertoires **A copier dans Windows** et **A copier dans Windows Système**

Vérifier l'installation de MySQL

Pour vérifier que MySQL fonctionne, cliquez sur le titre "**Administration de la base de données**" en haut à gauche. L'écran suivant doit s'afficher, avec en particulier sous **Accueil**, la base de données "**mysql**". Si elle n'apparaît pas ou s'il y a un message d'erreur, recommencez toutes les étapes depuis le début.



Voilà, vous en avez fini avec l'installation de PHP et MySQL.

Vous pouvez boire un café avant de passer à la partie la plus intéressante : utiliser tout ça !!!

Installation sur Macintosh et sur Unix

<http://www.zend.com/manual/install.macosx.php>

Il existe une version Mac OSX pour MySQL.

Le site suivant propose aussi des informations :

<http://www.xclave.com>

PHP Pas à Pas

Premier programme

Ce qu'il faut savoir avant de commencer un programme

Avant toute chose,

- quelle extension de fichier est acceptée par l'hébergeur ? (free impose l'extension .php3 et n'accepte pas .php)
- quels délimiteurs sont acceptés par l'hébergeur ? (free n'accepte pas <% %>)
- quelle version de php est proposée par l'hébergeur ?
- est-ce que l'hébergeur accepte MySQL ?

Et ensuite,

- quelles fonctions php sont acceptées par l'hébergeur ? (disponibilité de fonctions graphiques, etc.)
- quelle est la méthode de restriction des accès ? (**.HTACCESS**)

Dans ce premier chapitre nous allons créer la page d'accueil d'un catalogue de bibliothèque.

Script PHP

Un fichier PHP est un simple fichier texte avec une extension .php ou .php3. ce fichier sera interprété (lu) par le serveur avant d'être affiché dans le navigateur Internet. Pour créer un fichier PHP vous n'avez besoin de rien d'autre qu'un éditeur de texte.

1. Démarrez l'application **Bloc-notes** : cliquez sur **Démarrer | Programmes | Accessoires | Bloc-Notes**.
2. Sauvegardons tout de suite notre future première page : **C:/EASYPHP/www/test.php3**



A noter qu'une page HTML peut se passer des balises `<html>` , `<body>`, `</body>` et `</html>`. Aussi, tous nos exemples peuvent être saisis tels quels et très bien fonctionner. Par soucis de clarté, indiquons les balises HTML

1. La première partie de la page ne pose pas de problème c'est du code HTML :

```
<html>
<head>
<title>Ma bibliothèque</title>
</head>
<body>
<p align="center"></p>
```

2. Insérons le premier script :

```
<? echo "Bienvenue dans le catalogue de ma bibliothèque. Aujourd'hui,
nous sommes le"; ?>
```

3. Nous voulons que la date du jour s'affiche. Pour cela, nous allons insérer le script PHP suivant :

```
<?
echo strftime("%d/%m/%Y");
?>
```

4. Pour afficher l'heure exacte, nous tapons:

```
Il est <? echo strftime( "%H:%M:%S" ); ?>.
```

5. La page est terminée comme toute page HTML:

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

Pour finir, enregistrez votre document !

Début et fin d'instructions PHP

Nous avons inséré le code PHP dans une page HTML traditionnelle : vous noterez que le code PHP se place **TOUJOURS** entre des balises: `<? ?>`



Si vous oubliez les balises `<? ?>` qui encapsulent les scripts PHP, le navigateur affichera tout simplement votre script à l'écran !

Le programme PHP peut être composé de sections d'instructions PHP mélangées à des sections de code HTML.

Chaque section de PHP doit commencer par les deux caractères `<?>` et se terminer par les mêmes deux caractères inversés `?>`. A l'intérieur de la section, il est également possible de générer du HTML. On peut en particulier conditionner cette génération, et insérer des variables dans le HTML.

Le code qui se trouve en dehors de ces balises est du HTML.

Il est possible de n'avoir qu'une seule section d'instructions PHP.

D'autres balises existent (selon la configuration du serveur de votre hébergeur) mais `<? ?>` sont les plus répandues :

```
<?php ?>
<% %>
```

Vérifiez avec votre hébergeur quel type de balise utiliser.

Entre ces balises vous insérez vos lignes de script PHP. Remarquez les `;` à la fin de chaque instruction.

Nous verrons plus loin comment le code HTML peut être inséré dans une page PHP



Lorsque le script PHP ne comporte qu'une instruction, je conseille d'écrire le script sans saut de ligne. Nous obtenons ainsi :

```
Aujourd'hui, nous sommes le <? echo strftime( "%d/%m/%Y" ); ?>.
```

Au final, nous obtenons cette page :

```
<html>
<head>
<title>Ma bibliotheque</title>
</head>
<body>
<p>
<? echo "Bienvenue sur ma page PHP. Aujourd'hui, nous sommes le"; ?>
<? echo strftime( "%d/%m/%Y" ); ?>. Il est <? echo
strftime("%H:%M:%S"); ?>.</p>
</body>
</html>
```

Si vous avez bien enregistré votre document dans le répertoire racine de votre serveur (<C:/EASYPHP/www/test.php3>) comme nous l'avons indiqué, il ne vous reste plus qu'à tester votre page!

1. Vérifiez que Apache est bien lancé.
2. Ouvrez votre navigateur favori et tapez l'adresse suivante : [HTTP://LOCALHOST/test.php3](http://localhost/test.php3)

La page devrait s'afficher comme ceci :



Bienvenue dans ma bibliothèque. Aujourd'hui, nous sommes le 28/04/2001. Il est 17:20:43.

Analyse du script

1. `<? ouverture du script PHP`
2. `echo` est une *fonction* (voyez plus bas dans ce chapitre pour une définition) qui va afficher ce qui la suit directement : ici, le message de bienvenue jusqu'à `"le"`.
3. Le second `echo` est suivi par une autre fonction (`strftime`) : avant d'afficher quelque chose PHP va exécuter cette fonction et c'est son résultat qui sera affiché à l'écran.
4. la fonction `strftime` indique la date en jour-mois-année, selon les paramètres qu'elle a reçus : (`"%d/%m/%Y/"`) d=jour, m=mois Y(majuscule = année sur 4 chiffres)
5. et le `;` qui clos l'instruction `echo`
6. La même opération est répétée aux points 2 et 3 pour afficher cette fois l'heure exacte, en fonction des nouveaux paramètres passés à la fonction `strftime` : (`"%H:%M:%S"`).
7. `?>` fermeture du script PHP.



A retenir : `echo` est suivi de guillemets qui lui indiquent ce qu'il doit afficher. Dans le cas où il doit afficher le résultat d'une fonction, il n'y a pas de guillemets.

Une instruction se termine toujours par un `;`

Les variables, les constantes, les fonctions, les tableaux

Nous n'allons pas vous faire une initiation à la programmation, ce n'est pas le but, mais il y a quelques notions fondamentales à connaître avant d'aller plus loin.

Si vous connaissez les notions de variables, de boucles, de tableaux ce chapitre n'est pas pour vous.

Si ces noms ne vous inspirent rien : prenez le temps de lire ce chapitre ! Ces quelques pages vous permettront de comprendre la suite du manuel, mais comprenez bien qu'il ne s'agit que d'un "couteau de survie" de la programmation !

Les commentaires

Cela peut paraître étonnant de parler d'abord de "commentaires", mais vous apprendrez très vite combien ils sont essentiels pour faire du bon travail.

// Voici un commentaire

Les commentaires permettent d'identifier les différentes parties de votre script PHP sans "gêner" l'exécution du programme.

Aujourd'hui, ce script que vous écrivez vous est familier, mais dans un mois ou deux ? vous souviendrez-vous encore de son sens aussi clairement ? Si votre script est un petit peu compliqué, les commentaires vous permettront de "repérer" les différentes parties dans votre code. Ils vous aideront, ainsi que les autres personnes amenées à lire votre code, à vous y retrouver rapidement.

Un commentaire, c'est donc une ou plusieurs lignes que vous insérez dans le script en indiquant qu'il ne faut pas les interpréter lors de l'exécution du script. Vous le faites ainsi :

Sur une seule ligne:

Il doit commencer par un double slash // et se terminer sans saut de ligne (sans appuyer sur la touche **Entrée**). Ce type de commentaire peut être placé derrière une instruction :

```
print("bonjour"); // cette instruction affiche "bonjour"
```

Sur plusieurs lignes:

Parfois il est nécessaire de développer un commentaire. Pour faciliter la lisibilité, il est permis d'écrire les longs commentaires sur plusieurs lignes.

Il doit être précédé par les caractères /* et se terminer avec les caractères */, comme ceci :

```
/* voici un exemple de commentaire  
qui s'étale  
sans complexe  
sur plusieurs lignes */
```

Les variables

Un programme, un script,... qui s'exécutent sur un ordinateur manipulent des informations et des valeurs qui peuvent changer : ce sont des variables. Pour les identifier, ces variables sont nommées (par vous), par exemple :

Un compteur chargé de mémoriser le nombre de visites d'une page doit être incrémenté à chaque nouvelle visite : ce compteur est appelé une variable et est caractérisé par son nom. L'ordinateur associe à ce nom de variable la valeur du compteur. (voyez la partie "écriture dans un fichier" où nous créons un exemple de compteur très basique)

```
$compteur = 0 // initialise la variable $compteur à la valeur 0.
```

```
$compteur +=1 // incrémentation de la variable quand la page est chargée...
```

! Le signe \$ indique à l'interpréteur PHP qu'il s'agit d'une variable. Considérez qu'un nom de variable doit toujours être précédé par le signe \$... Vous découvrirez quand vous irez plus loin dans le PHP que ce n'est pas *absolument* vrai : mais au moins vous êtes certains de ne pas faire de bêtise !

Une variable peut contenir aussi bien du texte que des nombres. Elle peut aussi être vide.

Voici quelques exemples d'assignation de valeurs à des variables :

```
$saison = "été";  
$SAISON = "ETE";  
$_saison = "hiver";  
$prix = 150; //notez que l'absence des guillemets  
$jourJ = "4 juin 1945";  
$couleur = ""; // c'est une variable contenant la valeur vide. "echo  
$couleur="" //afficherait un écran vide
```

Des exemples nous pouvons tirer ceci :

- le ; termine chaque ligne
- les " " sont inutiles pour des valeurs numériques mais ils sont obligatoires dans les autres cas
- **PHP est sensible à la casse : \$bellesaison est différent de \$BELLESAISON : faites bien attention lors de vos manipulation si vous appelez une variable en ne respectant pas la casse vous obtiendrez une valeur "vide" et pas forcément un message d'erreur.**

! *Nous vous conseillons donc de vous fixer une règle claire dès le début : choisissez de n'utiliser que des minuscules et de séparer les mots d'une variable par un tiret bas, etc. C'est un effort que vous ne regretterez jamais d'avoir fait !*

Vous pouvez donc affecter une valeur à une variable dans votre script, mais vous pouvez aussi demander à l'utilisateur d'entrer lui-même des valeurs pour les variables par l'intermédiaires d'un formulaire (voyez plus loin).

Les variables se caractérisent par leur type (caractère, entier, entier long, ...).... Il y a encore beaucoup de choses importantes et intéressantes mais ce n'est pas l'endroit !

Portées des variables

Constantes

Les constantes ne sont pas précédées par le signe \$, contrairement aux variables. Elles se définissent par la fonction **define**. L'usage fait qu'on les écrive en majuscules.

Par exemple, la constante PI se définit et s'utilise comme suit :

```
<?  
define("PI" , 3.14);  
$diametre = 10 ;  
$circonference = $diametre * PI;  
echo $circonference ;  
>
```

Il existe des constantes créées par PHP au lancement de votre programme. Par exemple, TRUE, FALSE, M_PI, __FILE__.

Si vous désirez inclure une constante dans un littéral, vous devez utiliser le caractère de concaténation (voyez plus loin pour une explication).

TEST DE `__FILE__`.PHP

```
<?
print( "Le programme actuel est " . __FILE__ );
print( "Le programme actuel est __FILE__ " ); // ʘ
?>
```

```
-----
Le programme actuel est C:\test de __FILE__.php
-----
```

```
Le programme actuel est __FILE__
```

Littéraux

Un littéral correspond à un texte que l'on veut utiliser dans une instruction du programme.

Un littéral doit être encadré par deux apostrophes ou deux guillemets.

L'utilisation de deux guillemets permet d'insérer des variables à l'intérieur du littéral alors que cela n'est pas possible en utilisant deux apostrophes.

L'insertion de variables dans un littéral est tellement pratique que l'on utilise finalement toujours les guillemets.

Les tableaux

Dans le langage commun un tableau, comme une feuille de tableur, est constitué de lignes et de colonnes. En programmation un tableau est une variable contenant plusieurs valeurs qui sont indexées. Regardons de plus près un tableau :

Créer un tableau :

```
$jour=array("lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche");
echo $jour[0];
echo $jour[1];
```

Si c'était exécuté dans un script, cela afficherait : "lundimardi".

`$jour` est une variable de type tableau (*array*, en anglais) et elle est initialisée par `array(...)`
`echo $jour[0]` indique qu'il faut afficher l'index 0 ([0]) du tableau, c'est-à-dire la première valeur contenue dans le tableau (lundi).



Vous l'aurez noté, le premier élément du tableau (lundi) correspond à l'index [0] !

En programmation, le comptage commence à zéro : le premier élément d'une série est toujours numéroté "0".

Les 7 jours de la semaine sont donc enregistrés de 0 à 6...



Une variable de type array, ou tout simplement un array, contient plusieurs éléments distincts. On accède à la valeur de chacun de ces éléments par la syntaxe suivante :

```
$nom_du_tableau[index].
```

L'*index* (aussi appelé *clé*) peut être un *entier* ou une chaîne de caractère (*string*). Il permet de pointer un élément précis du tableau.

Par exemple :

```

$jour = array( "Monday" => "lundi", "Tuesday" => "mardi",
"Wednesday"=>"mercredi", "Thursday");
echo $jour[Monday];
echo $jour[0];
affiche "lundilundi".

```

Nous avons complété l'indexation par défaut par une seconde clé d'index, ici les nom des jours de la semaine en anglais.

Les valeurs du tableau peuvent alors indifféremment être appelée par l'une ou l'autre clé.

Modifier un tableau:

Vous pouvez initialiser ou modifier une valeur d'un tableau durant l'exécution d'un script, par exemple :

```

$jour[0]="vacances"; //echo "$jour[0]<br>"; afficherait "vacances"
$jour[0]=1; // echo "$jour[0]<br>"; afficherait "1"

```

vous modifiez l'entrée du tableau en indiquant le n° d'index blabla ///

Où vous donnez successivement la valeur "vacances" et "1" à la première valeur du tableau (index 0).

Pour ajouter un enregistrement à un tableau existant ... ///



Pour initialiser une variable avec une chaîne de caractère vous devez utiliser des "", mais pas lorsque vous initialisez avec un nombre.

Afficher un tableau:

Pour le debug de vos scripts, la fonction `print_r(nom_du_tableau)` ; vous permet d'afficher toutes les valeurs du tableau, avec leur clé, à la suite les unes des autres.

Nous verrons plus loin comment afficher à l'écran et manipuler toutes les valeurs d'un tableau.

Les nombres

Les nombres sont soit des entiers, soit des flottants, des réels. Mais ce qu'il importe de savoir c'est que lorsque vous travaillez avec des décimales vous devez l'indiquer avec un séparateur de décimale.

PHP exige le point comme séparateur décimal (écriture à l'anglaise).

<?

```

$pi = 3.14 ; // $pi = 3,14 n'est pas autorisé
$negatif = -100 ;
$million = 1e6 ; // est identique à $million = 1000000
$un_million_et_demi = 1.5e6 ; //1.5 exposant 6
?>

```

Les opérateurs

Les opérateurs permettent de manipuler les variables. Ils sont de deux types : opérateurs arithmétiques et opérateurs de comparaison.

| | | |
|----|-----------------------------|--------|
| + | addition | a + b |
| - | soustraction | a - b |
| * | multiplication | a * b |
| / | division | a / b |
| < | est plus petit que | a < b |
| <= | est plus petit ou égal à | a <= b |
| > | est plus grand que | a > b |
| >= | est plus grand ou égal à | a >= b |
| == | est égal à (double signe =) | a == b |
| != | n'est pas égal à | a != b |
| | ou | a b |
| & | et | a & b |

Par facilité d'écriture les programmeurs utilisent souvent des "raccourcis" pour effectuer les opérations les plus simples, voici les principaux :

```

$i++ ; // est équivalent à $i = $i + 1
$i-- ; // est équivalent à $i = $i - 1
$i += x ; // est équivalent à $i = $i + x
$i -= x ; // est équivalent à $i = $i - x
$i *= x ; // est équivalent à $i = $i * x
$i /= x ; // est équivalent à $i = $i / x
$i /= x ; // est équivalent à $i = $i / x

```

Si vous ne vous sentez pas à l'aise avec cette syntaxe, prenez le temps de lire un manuel d'initiation à la programmation.

L'opérateur de concaténation

Permet de combiner plusieurs variables les unes avec les autres pour un affichage à l'écran (ou la sortie définie).

Ainsi, dans l'exemple suivant, on veut afficher le futur du verbe régulier "manger" :

```

// initialisation de la variable : la racine du verbe
$verbe = "manger" ;

```

si vous écrivez ceci :

```

//essai de conjugaison du verbe...

```

```

echo "Le futur de $verbe est : je $verbeai, tu $verbeas, il $verbea,
nous $verbeons, vous $verbeez, ils $verbeont.";

```

vous obtiendrez cela :

```

Le futur de manger est : je , tu , il , nous , vous , ils.

```

\$verbeai , ainsi que toutes les autres personnes du verbe ont été interprétés comme une variable. Or ces variable n'ont pas de contenu (vides).

Il aurait fallu utiliser le caractère de concaténation . (point), comme ceci :

```

$verbe = "manger" ;

```

```
echo "Le futur de $verbe est : je $verbe"."ai, tu $verbe"."as, il
$verbe"."a, nous $verbe"."ons, vous $verbe"."ez, ils $verbe"."ont.";
```

```
Le futur de manger est : je mangerai, tu mangeras, il mangera, nous mangerons, vous
mangerez, ils mangeront.
```

Les caractères spéciaux

Si l'on désire afficher les caractères \, \$, " et ', il faut respecter les règles suivantes :

- Lorsque la chaîne qui est encadrée par des guillemets contient elle-même des guillemets, ces derniers doivent s'écrire : \" (antislash + ")

```
echo " L'élève répondit : \"Ce n'est pas moi Monsieur !\". "; //affiche
L'élève répondit : "Ce n'est pas moi Monsieur !".
```

- Lorsque la chaîne qui est encadrée par des apostrophes contient elle-même des apostrophes, ces derniers doivent s'écrire : \'. Si l'apostrophe est encadrée

```
echo 'Ce n\'est pas cher' ; // affiche Ce n'est pas cher
```

- Vous pouvez combiner les guillemets et les apostrophes:

```
echo " L'élève répondit : 'Ce n'est pas moi Monsieur !'. ";
//affiche L'élève répondit : 'Ce n'est pas moi Monsieur !'.
```

- Si le caractère après un \$ est une chaîne de caractères, le système considère qu'il s'agit d'un nom de variable. Pour afficher tels quels le \$ et la chaîne de caractères, faites précéder le \$ de l'antislash

```
echo "Le prix est de $pi" ; // affiche Le prix est de 3.1415926535
```

```
echo "Le prix est de \$pi" ; // affiche Le prix est de $pi
```

- ❗ Notez que quand le signe \$ est suivi d'un nombre il n'y a pas besoin de l'antislash pour l'afficher. Egalement si le signe \$ n'est suivi de rien (*attention : un espace est quelque chose!*).

```
echo "Le prix est de $10" ; // affiche Le prix est de $10
```

```
echo "Un nom de variable doit commencer par $" ; // affiche Un
nom de variable doit commencer par $
```

- Si vous voulez afficher à l'écran le caractère antislash lui-même, vous devez le doubler :

```
echo "Le répertoire est C:\\Windows" ;
// affiche Le répertoire est C:\Windows
```

- Si le caractère antislash précède un caractère "normal", il sera affiché tout simplement.

```
echo "\a"; // affiche \a
```

Vous travaillerez au choix avec des "" ou des '' pour manipuler les chaînes de caractères, vous pourrez changer de système dans un même script. C'est une question de clarté et de facilité.

Exemples de syntaxes correctes :

```
echo "c'est juste"; // cette ligne affiche le texte c'est juste
```

```
echo 'c\'est juste'; // cette ligne affiche le texte c'est juste
```

Exemples de syntaxes incorrectes :

```
echo 'c'est faux'; // cette ligne affiche le caractère c puis génère un
message d'erreur
```

```
echo '$ signifie dollar'; // cette ligne n'affiche pas $ car il est
suivi par un espace
```

Les fonctions

Les fonctions sont des fragments de codes (des blocs d'instructions en langage technique) que vous aller rédiger à part pour pouvoir les réutiliser très facilement et aussi souvent que vous le souhaitez. Une fonction, comme une variable, est nommée et quand vous l'utiliserez dans votre script, vous l'invoquerez par son nom.

Structure d'une fonction:

```
nom_de_la_fonction (arguments_éventuels);
{
    instructions_exécutées ;
}
```

Un exemple :

```
<?
function carre($nombre)
{
    $valeur=($nombre * $nombre);
    return $valeur;
}
echo "Le carré de 5 est : ".carre(5) ;
?>
```

la fonction attend un argument (**\$nombre**) quelle mettra au carré (**\$valeur**) et elle affichera ensuite : **Le carré de 5 est 25.**



Pour afficher le résultat d'une fonction vous devez **NECESSAIREMENT** la mettre hors des " " de echo et la concaténer avec l'opérateur de concaténation " . ". Comme ceci:

```
<?
Fonction_geniale (){
    1+1=2
}
echo "le résultat de ma fonction géniale est ".fonction_geniale ;
?>
```

Une fonction peut contenir autant d'instructions que vous le souhaitez ! C'est son avantage : car vous n'avez pas à répéter le code à chaque fois que vous en aurez besoin : vous n'aurez qu'à invoquer la fonction qui contient le code !

Bloc d'instructions

Un bloc d'instructions est...un bloc d'instructions ;-)

C'est-à-dire un ensemble logique d'instructions.

Un bloc peut ne comporter qu'une seule instruction

```
if( a==5 ) //si a est égal à 5
    echo "bonjour"; //affiche bonjour
```

Un bloc peut comporter plusieurs instructions, il faut alors encadrer les instructions par les caractères { }.

```
if( a==5 ) {
```

```
    echo "bonjour ";
    echo "le monde";
}
```

❗ La position des {}, la disposition de votre code et l'utilisation ou non de tabulation est sans impact sur l'exécution du code : c'est uniquement une question de lisibilité pour vous !

/ exemples d'écriture corrects*/*

```
if( a==5 ){
    echo "bonjour " ;}
```

```
if( a==5 )
{
echo "bonjour " ;
}
if( a==5 )
{
    echo "bonjour " ;
}
```

sont toutes des notations correctes et vous devez choisir celle qui vous convient le mieux.

Les structures de contrôles

"\$total1<\$total2" est vrai, mais seulement parce que nous avons donné des valeurs adéquates aux 2 variables. Si nous avons indiqué:

```
<?
$total1 = 100;
$total2=15;
echo "$total1<$total2";
?>
```

afficherait à l'écran "100 < 15" !! ce ne serait pas une bonne chose pour la crédibilité du programme...

Un bon moyen de se prémunir de ce genre de problèmes c'est de créer une *structure de contrôle* qui va vérifier qu'une condition est remplie avant de permettre l'exécution d'une tâche déterminée.

Dans notre cas, nous devrions vérifier que \$total1 est effectivement plus petit que \$total2 et dans ce cas seulement afficher le résultat.

En résumé : si \$total1 est plus petit que \$total2 affiche le.

En programmation le "si" se dit "if"

if

```
if ($total1<$total2)
    echo "$total1<$total2";
```

Si \$total est plus petit que \$total2 le script PHP affichera notre phrase ! Si la condition n'est pas vérifiée : rien ne sera affiché.

If else

Nous pourrions sophistiquer l'exemple pour que l'ordinateur affiche toujours la bonne réponse , la plus petite valeur et puis la plus grande, comme ceci:

```
if ($total1<$total2)
    echo "$total1<$total2";
else echo "$total2<$total1";
```

le "else" signifie "sinon" :

Si la condition est vérifiée tu affiches "ceci", sinon tu affiches "cela".



Pour enchaîner plusieurs structures de contrôles les unes aux autres utilisez **elseif** : "sinon si...".

Nous voulons également vérifier que les 2 variables n'ont pas la même valeur, auquel cas s'afficherait le message "les totaux sont identiques".

Il peut y avoir autant de **elseif** que vous le souhaitez.

La dernière instruction sera un **else** simple.

Comme ceci :

```
<?
$total1= 100;
$total2=15;
if ($total1<$total2)
    echo "$total1<$total2";
elseif ($total1>$total2)
    echo "$total1>$total2";
else echo "les totaux sont identiques !";
?>
```

Exercice:

Créez un script contenant 2 variables : **\$grande** de valeur 100 et **\$petite** de valeur 50. Rédigez une structure de contrôle qui permette d'afficher les variables dans l'ordre croissant : la plus grande en premier, la plus petite en dernier.(La solution est disponible dans le supplément en ligne)

Portées des variables dans les fonctions

Nous avons attendu ce moment pour aborder la question. En programmation, les variables sont caractérisées par leur "portée". C'est-à-dire l'espace de code dans lequel elles "existent".

Les variable définies dans une fonction n'existeront que dans "l'espace" de cette fonction (entre les {}).

Par exemple:

```
<?
function ma_fonction() {
    $a="maison" ;
    echo $a ; //affiche maison
    /*Dans la ligne suivante, on appelle la variable GLOBALE
$a($globals)*/
```

```

        echo $globals["a"]; // affiche 15
    }
    $a="univers"; // on déclare une variable globale.
    ma_fonction();
    /*l'appel de la fonction affiche le résultat de $a local, puis de $a
    global : maison univers.*/
?>

$global est une variable de type array qui contient toutes les variables globales du script. Dans notre
cas une seule : $global["a"]="univers";
<?
function ma_fonction(){
    global $a;
    echo $a; //affichera la valeur de la variable GLOBALE $a
}
$a="univers";
ma_fonction();
?>

```

Vous le voyez ici concrètement une variable déclarée dans une fonction à une portée locale : elle n'a de valeur que dans la fonction.

Une variable déclarée dans le corps du script (on parlera de 'main' : principal en anglais, en référence au main dans la programmation C) existera aussi dans tous le script mais pas dans les fonctions SAUF si vous appelez cette variable globale explicitement. C'est pour cela qu'une variable peut avoir le même nom et une valeur différente !! (prenez-y garde et de toute façon évitez de faire cela...c'était pour les besoins de l'exemple).

C'est compliqué, mais c'est simplement une forme de logique. Imaginez que les fonctions sont des caissons étanches que vous pouvez parfois ouvrir...

Ne soyez pas découragé si cela vous semble confus : utilisez ce chapitre comme un aide mémoire et au fur et à mesure que vous manipulerez le langage vous en aurez de moins en moins besoin ;-)

Créer un cookie.



Un cookie est un fichier texte qui est écrit par le serveur sur l'ordinateur du visiteur. Il contient des informations sur l'utilisateur (un identifiant, par exemple ou un compteur du nombre de visites effectuées...). A chaque chargement d'une page du site, le serveur vérifie automatiquement la présence ou non de ce cookie.

Une page PHP va vérifier l'existence de notre cookie sur le PC du visiteur : s'il existe il le lit et envoie le visiteur sur la page des visiteurs déjà enregistrés, s'il n'existe pas le visiteur est envoyé sur la page d'enregistrement et une fois qu'il s'est enregistré il est renvoyé sur la page des visiteurs enregistrés.



Toutes les pages seront maintenant créées à la racine de votre serveur local dans le répertoire de travail 'monsieur', que vous allez créer.

Si vous utilisez Easyphp le chemin d'accès est : "C:/easyphp/www/monsieur".

Créez les pages suivantes:

index.php3 qui vérifie l'existence du cookie et redirige en conséquence votre visiteur
back.php3 page d'accueil des visiteurs déjà enregistrés
new.html la page qui contient le formulaire d'inscription
cookie.php3 la page de création du cookie ET de renvoi du visiteur sur la page des utilisateurs enregistrés.



ATTENTION pour pouvoir suivre vos modifications plus facilement je vous conseille de modifier un paramètre d'affichage votre navigateur : mettre à jour à chaque chargement de page...faire un screen en FRANÇAIS

Renommez votre page index.php3 que vous avez créée en back.php3 pour ne pas perdre ce que vous avez déjà écrit.

La page **index.php3**:

```
<?
if ($prenom=="")
    header("location: new.htm");
else header("location: back.php3");
?>
```

Enregistrez votre page : index.php3

Explication:



Par défaut, le serveur vérifie automatiquement au chargement de la page que le cookie existe sur l'ordinateur du surfeur.

Si **\$prenom** est vide (=="") (c'est à dire que le cookie n'a pas été trouvé sur l'ordinateur du visiteur) : le cookie n'existe pas.

Dans ce cas, PHP génère en en-tête (header) de redirection vers la page d'inscription du visiteur :
header("location: new.htm");

Si **\$prenom** n'est pas vide : le cookie existe et le visiteur est automatiquement renvoyé sur la page d'accueil personnalisée "back.php3" : **else header("location: back.php3");**

back.php3:

Prenez votre page 'back.php3' et modifiez là comme ci-dessous:

```
<html>
<head>
    <title>Accueil</title>
</head>
<body>
<?
// Ici les modifications suite à l'utilisation du cookie
echo "bonjour $prenom &nbsp;$nom<br>";
//Fin des modifications.
echo "Bienvenue dans le catalogue de ma bibliothèque. Aujourd'hui, nous
sommes le";
echo "strftime( "%d/%m/%Y" ); ?>. Il est <? echo strftime( "%H:%M:%S" );
?>
</body>
</html>
```

Enregistrez votre page : back.php3

lecture:

La page d'accueil s'affiche avec le prénom et le nom du visiteur

new.htm

```
<html>
<head>
<title>
</title>formulaire</head>
<body>
<form action="cookie.php3" name="identification" id="identification">
Votre prenom :<input type="text" name="prenom"><br>
Votre nom :<input type="text" name="nom"><br>
<input type="submit" value="m'inscrire">&nbsp;<input type="Reset"
value="annuler l'inscription">
</form>
</body>
</html>
```

Enregistrez votre page : new.htm

lecture:

Le formulaire initialise les variables 'prenom' et 'nom' et les envoie vers la page 'new.php3'

❗ un script PHP lit automatiquement les variables d'un formulaire en y ajoutant le signe \$: voyez la page 'new.php3'

cookie.php3


```
<?
setcookie("prenom","$prenom", time()+5000);
setcookie("nom","$nom", time()+5000);
header("location: back.php3");
?>
```

Enregistrez votre page

Explication:

Un cookie est créé avec la variable `$prenom` (qui correspond au champ `prenom` du formulaire) et une durée de vie fixée (5000 secondes : libre à vous de modifier cette durée)

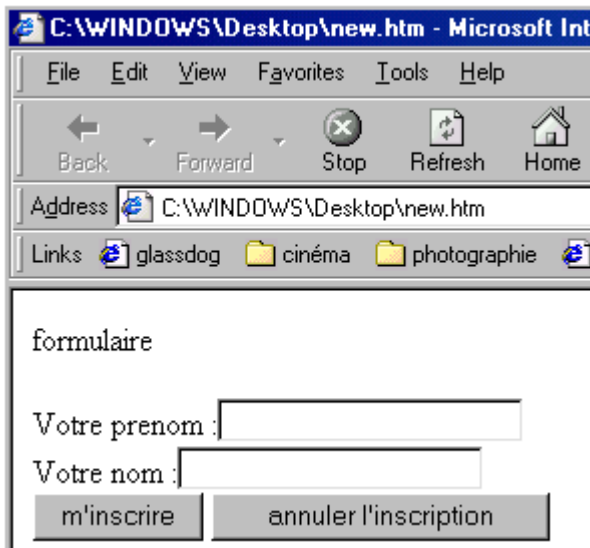
Un cookie est créé avec la variable `$nom` (qui correspond au champ `nom` du formulaire) et une durée de vie fixée (5000 secondes)

Après l'enregistrement des variables dans le cookie, le visiteur est renvoyé sur la page des utilisateurs enregistrés : 'back.php3')

❗ Plusieurs cookies peuvent être créés : nom, prénom, etc mais tous les cookies d'un même site web sont enregistrés dans un seul fichier texte sur l'ordinateur du visiteur.

Lancez votre page d'accueil depuis votre navigateur web : <http://localhost/index.php3>

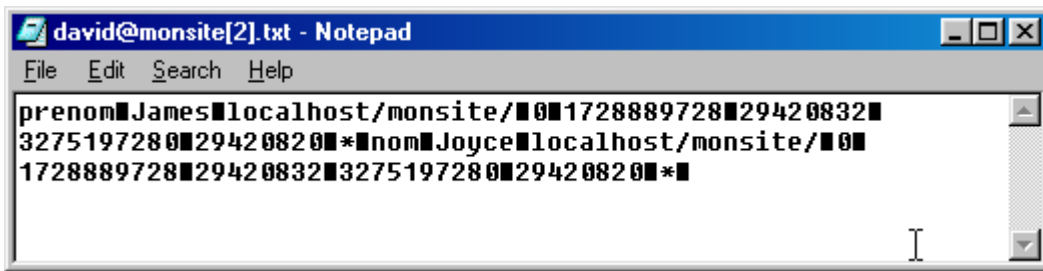
Si c'est la première fois que vous vous "connectez" sur votre catalogue, vous allez être renvoyé sur la page d'inscription : vous devriez avoir un écran d'invite de ce type :



The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "C:\WINDOWS\Desktop\new.htm - Microsoft Int". The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The navigation bar contains "Back", "Forward", "Stop", "Refresh", and "Home" buttons. The address bar shows "C:\WINDOWS\Desktop\new.htm". The Links bar has "glassdog", "cinéma", and "photographie". The main content area displays a form titled "formulaire" with two input fields: "Votre prenom :" and "Votre nom :". Below the fields are two buttons: "m'inscrire" and "annuler l'inscription".

Si ce n'est pas votre première visite, vous serez redirigé sur la page d'accueil du catalogue...

❗ Vous pouvez vérifier l'existence du cookie une fois le test exécuté. Allez dans "c:/windows/temporary internet files" et repérez un cookie du nom `quelquechose@localhost...` Double cliquez dessus pour l'ouvrir.



Supprimez le sans hésiter : vous devrez à nouveau vous identifier !

Récupérer les zones d'un formulaire HTML dans PHP

Les zones d'un formulaire HTML peuvent être transmises à un programme PHP.

Dans votre formulaire toutes les zones sont nommées, et lorsqu'un script PHP est exécuté (submit) des variables seront initialisées avec les valeurs et les noms des zones du formulaire dans le script PHP qui reçoit le formulaire.

Voici un exemple de formulaire HTML associé à du PHP :

```
...
// action envoie les données du formulaire à la page reception.php3
<form action="reception.php3">
<input type=text name=texte_saisi>
<input type=submit>
</form>
...
```

Le script `reception.php3` pourrait contenir le code suivant et `$texte_saisi` est initialisé avec la valeur du camp correspondant du formulaire.

```
<?
echo "Vous venez de saisir le texte suivant : $texte_saisi" ;
?>
```

Le code suivant affichera la même page différemment selon que le formulaire a déjà été complété ou pas.

```
<?
//initialisation de la variable qui 'renvoie' à la feuille active
$MYSELF = __FILE__ ;
//si texte_saisi est vide alors affiche le formulaire
if( $texte_saisi == "" )
{
    $MYSELF = __FILE__ ;
    echo "<form action='$MYSELF'>";
    echo "<input type=text name=texte_saisi>";
    echo "<input type=submit>";
    echo "</form>";
}
```

```

    }
//si texte_saisi n'est pas vide, alors affiche la phrase qui suit
if( $texte_saisi != "" )
    {
        echo "Vous venez de saisir le texte suivant : $texte_saisi" ;
    }
?>

```

Si l'on saisit le texte "" dans le formulaire et que l'on clique le bouton de validation, champ `champ_de_saisie`, et qu'un programme PHP est enchaîné, ce programme "recevra" la variable `$champ_de_saisie` qui vaut "", c-à-d rien



Attention : lorsque le programme reçoit des variables contenant les caractères ' , " , / , \ , ceux-ci sont précédés par le caractère antislash \ (voyez le point caractères spéciaux plus haut).

Si le texte `c:/code d'entrée.txt` est saisi dans la zone fichier du formulaire, la variable `$fichier` contiendra `c:\code d\'entrée.txt`.

La commande suivante affiche `c:/code d'entrée.txt`

```
echo $fichier; // affiche c:/code d'entrée.txt
```

Il est possible de supprimer ces caractères ajoutés en utilisant la fonction `stripslashes`. Voyez en annexe.

Voilà, vous savez maintenant récupérer les informations personnelles du visiteur, les stocker dans un cookie et modifier la page d'accueil en conséquence !

Ceci dit, l'inconvénient du cookie c'est que vous ne savez rien de vos visiteurs : impossible de vous faire une idée de qui visite le site encore moins de les contacter par email, par exemple. En plus, le visiteur peut refuser les cookies ou le supprimer lui-même ...

L'idéal serait donc de stocker sur votre serveur les informations collectées. Gardez ou non le cookie pour identifier automatiquement votre visiteur (ou proposez le choix au visiteur, c'est encore mieux), mais dans tous les cas enregistrez les informations sur votre serveur !

Nous allons créer et utiliser une base de données pour stocker ces renseignements.

Dans un premier temps nous allons créer le catalogue de la bibliothèque puis, à la fin, nous traiterons de l'enregistrement des informations personnelles des visiteurs (quoi demander et comment le faire). Prêt?

MYSQL

MySQL est un langage de consultation et de mise à jour de base de données. On l'appelle également un langage de requête. Il sera contrôlé depuis PHP grâce aux fonctions fournies avec l'extension MySQL de PHP. *Pour que cela fonctionne vous devez donc veiller à ce que les extensions MySQL soient disponibles sur votre serveur web : demandez à votre hébergeur s'il ne l'indique pas clairement sur le site (en local c'est le cas si vous installez easyPHP).*

Pourquoi une base de donnée?

Vous pourriez stocker les informations dans un fichier texte tout simple, mais l'avantage de la base de données c'est de pouvoir très facilement extraire des données par une requête, les manipuler et les mettre à jour.

Qu'est-ce qu'une base de données

Une base de donnée contient des *données* organisées selon vos indications.

Elle est composée d'une ou de plusieurs tables. Chaque table est composée de champs. Les champs contiennent les enregistrements : les données.

Les enregistrements sont très facilement accessibles à l'aide de requêtes formulées dans un script PHP. par exemple la requête suivante permet de lister les livres de Conan Doyle enregistrés dans le catalogue:

```
SELECT * FROM TABLE_LIVRES WHERE AUTEUR = "Conan Doyle".
```

Qui se lit : "Choisi tous les champs de la table_livre où auteur est égal à Conan Doyle"

Nous verrons comment insérer ce genre de requête dans un script PHP, mais d'abord créons notre base de données !

Création d'une base de données avec PHPMyAdmin

Il y a deux moyens pour le faire :

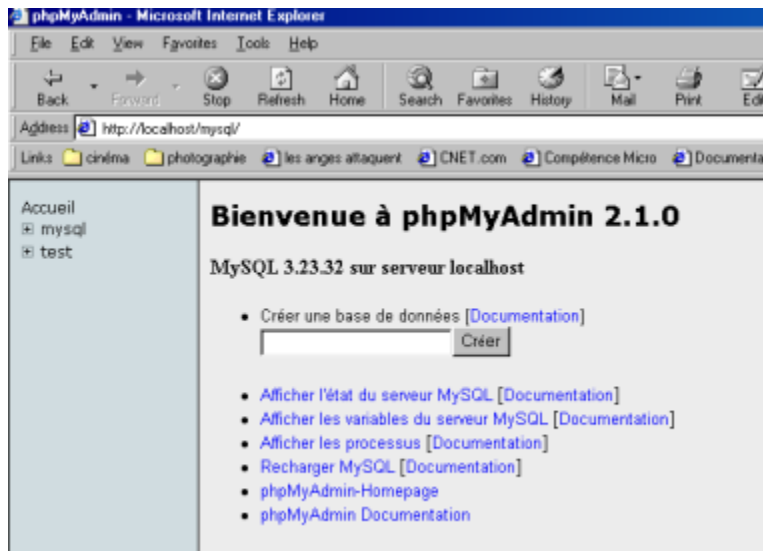
- Utiliser phpMyAdmin : un utilitaire conçu en PHP qui permet de manipuler très facilement une base de données MySQL.
- Créer manuellement un programme PHP qui appelle les fonctions de création de champs et de données

La base de données

La base de données (bdd) n'est que le conteneur plus ou moins "compartimenté" des informations que nous voulons stocker. Ces informations sont compartimentées dans des tables qui les rassemblent de façon "logique".

Par exemple : une table pour les livres (titre, auteur, etc) et une autre table pour les abonnés(nom, prénom, adresse, livres empruntés, etc.).

1. Ouvrez PHPMyAdmin (depuis votre navigateur Internet, après avoir vérifié que GO.BAT a été lancé : tapez <http://localhost>).
2. Choisissez **Accueil** dans le menu à gauche. La page d'administration de votre base de données s'affiche:



3. Pour notre projet de catalogue de bibliothèque, nous allons créer la base de données "biblio". Pour cela, tapez **biblio** dans le champ de saisie et cliquez sur **Créer**.

Vous voyez alors apparaître **biblio** sous "Accueil" dans le menu sur la gauche de l'écran.

4. Cliquez sur **biblio**. L'écran de gestion de la base apparaît alors :



! Lorsque le moment sera venu de mettre votre site sur Internet, il vous faudra peut-être changer le nom de votre base. En effet, certains hébergeurs ne vous offrent qu'une seule base de données ayant obligatoirement pour nom votre nom d'utilisateur, Free par exemple.

Création des champs de la table

Alors que la base de données n'est qu'un conteneur de tables, la table permet, elle d'enregistrer les données. En final, nous aurons une seule base de données "biblio", mais nous aurons plusieurs tables.

Imaginons que nous mettions en ligne notre catalogue de livres et bandes dessinées : nous allons créer la table "livres" qui contiendra les informations nécessaires pour chaque livre.

Cette table comportera quatre champs : titre du livre, description courte du livre, auteur du livre, disponibilité du livre (prêté ou non).

Lorsqu'on crée un champ, il faut obligatoirement lui attribuer un nom, un type et une longueur.

Nom identifiant du champ

Type indiquer si le champ contient des nombres ou des caractères

longueur indiquer le nombre maximum de caractères



Un nom ne doit pas contenir d'espace. Nous utiliserons uniquement des minuscules, des lettres non accentuées et des caractères de liaison (le soulignement "_") par souci de standardisation avec PHP, car ce langage est un peu plus exigeant en matière de noms de variables.

C'est de toute une bonne habitude à prendre pour éviter les erreurs en programmation PHP.

1. Ouvrez PHPMyAdmin si ce n'est déjà fait choisissez votre base **biblio**
2. Dans la partie droite de l'écran choisissez **Créer une nouvelle table sur la base biblio** :

- Créer une nouvelle table sur la base biblio:

Nom:

Champs:

3. Cliquez sur **EXECUTER**. L'écran suivant permet de saisir les champs de la table :

Base de données biblio - table livres

Champ	Type	Longueur	Attributes	Nu
<input type="text" value="titre"/>	<input type="text" value="TEXT"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="not nu"/>
<input type="text" value="description_courte"/>	<input type="text" value="TEXT"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="not nu"/>
<input type="text" value="auteur"/>	<input type="text" value="TEXT"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="not nu"/>

4. Cliquez sur le bouton **SAUVEGARDER**.
5. Sur la page suivante, notez le message de bon déroulement "**table livres a été créée.**"
Vous voyez aussi le code exact de la requête SQL qui a été exécutée. Cette ligne d'information vous permet d'apprendre le langage MySQL, elle apparaît à chaque opération sur la base de données.

Base de données biblio - table livres

table livres a été créé.

requête SQL:

```
CREATE TABLE livres (titre TEXT not null , description_courte TEXT not null , auteur TEXT not null )
```

Champ	Type	Attributes	Null	Defaut	Extra	Action
titre	text		Non			Modifier Effacer Primaire Index Unique
description_courte	text		Non			Modifier Effacer Primaire Index Unique
auteur	text		Non			Modifier Effacer Primaire Index Unique

- [Afficher](#)
- [Sélectionner](#)
- [Insérer](#)

• Ajouter un champ:

• [Insérer un fichier texte dans la table](#)

• Afficher le schéma de la table

Structure seule Ajouter des énoncés 'drop table'

Structure et données transmettre

Données CSV terminé(e)s par

• Changer le nom de la table pour:

• Copier la table vers:

Structure seule Structure et données

Enregistrer des données dans la table

1. Cliquez sur le lien [INSERER UN FICHIER TEXTE DANS LA TABLE](#), et saisissez les informations dans les champs appropriés comme dans l'illustration ci-dessous :

Champ	Type	Fonction	Valeur
titre	text	<input type="text"/>	Bételgeuse - 1. La planète
description_courte	text	<input type="text"/>	Bételgeuse, planète interdite ? Les colons ont de bonnes raisons de croire au pire.
auteur	text	<input type="text"/>	LEO

2. Cliquez sur [Sauvegarder](#)

Les modifications ont été sauvegardées.

requête SQL:

```
INSERT INTO livres (titre, description_courte, auteur) VALUES ('Bételgeuse - 1. La planète', 'Bételgeuse, planète interdite ? Les colons ont de bonnes raisons de croire au pire.', 'LEO') LIMIT 0, 30
```

3. Insérez les trois autres livres de la même façon:

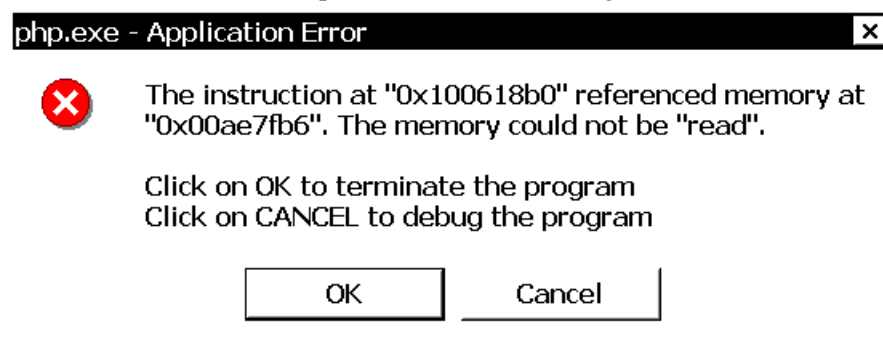
titre	description_courte	auteur
Calvin & Hobbes, Faites place à Hyperman !	Les aventures de Calvin et de son tigre en peluche Hobbes	Bill Watterson
Le Crépuscule des Idoles	Nietzsche se penche sur la fin de quelque chose	Friedrich Nietzsche
Le guide Michelin 2001	Le guide des restaurants 2001	Michelin

A l'avenir, il vous sera facile d'ajouter des livres.

Un problème connu de phpMyAdmin



A certains moments, vous pouvez obtenir le message suivant :



Cela se produit lorsque phpMyAdmin essaie de faire une opération sur une table dont on n'a pas défini de clé d'index primaire, et dont un des enregistrements contient une apostrophe.

Nous ne sommes pas certains de la cause exacte de ce problème, mais pour le contourner évitez d'utiliser le caractère d'apostrophe dans les données que vous introduisez...

Vous pouvez introduire d'autres titres dans votre catalogue, mais nous en resterons là pour les besoins de notre exemple...et puis croyez moi ces trois derniers titres suffisent ;-)

Créer une table avec un script PHP

Connexion à la base de données MySQL

Avant de pouvoir lire le contenu d'une base, avant de pouvoir effectuer n'importe quelle requête sur une base, vous devez d'abord ouvrir une connexion au serveur MySQL et sélectionner la base de données.

<?

```
mysql_connect("nom_de_serveur_mysql","user","password");
```



```
mysql_select_db( "nom_de_la_base" );
?>
```

`mysql_connect("nom_de_serveur_mysql","user","password");` est la fonction PHP qui démarre la connexion avec l'application MySQL.

| Variable | Usage | Sur un serveur web | En local |
|-----------------------------|--------------------------------------|--|-----------------------|
| <i>nom_du_serveur_mysql</i> | Engage la connexion avec la base SQL | adresse du serveur MySQL
ex. sur Free : sql.free.fr | localhost |
| <i>user</i> | valide l'utilisateur qui se connecte | votre nom d'utilisateur | peut-être laissé vide |
| <i>password</i> | authentification de l'utilisateur | votre mot de passe | peut être laissé vide |

Si vous utilisez easyPHP en local et si votre serveur est nommé Localhost, vous devez indiquer "Localhost" et donner un profil d'utilisateur.

`mysql_select_db("nom_de_la_bdd");` est la fonction PHP qui indique que toutes les opérations MySQL suivantes s'exécuteront sur la bdd indiquée en paramètre.

Dans notre cas, cela donne ceci :

```
/* Première étape : connexion à votre serveur MySQL sur Localhost, en
tant que root avec un mot de passe vide ("").
Si la connexion échoue (or die) un message d'erreur est affiché "ERREUR
connexion" (adaptez le à vos besoins)et le script est interrompu
*/
$mysql_link = mysql_connect("localhost","root","") or die("ERREUR
connexion");

/* Seconde étape : connexion à votre base de données
Si la connexion échoue (or die) un message d'erreur est affiché "ERREUR
connexion base de donnée" (adaptez le à vos besoins)et le script est
interrompu
*/
mysql_select_db("biblio") or die("ERREUR base de donnée");
```



Avant d'uploader votre script PHP chez l'hébergeur, vous devez modifier les paramètres de la fonction `mysql_connect`.

Par exemple, si votre hébergeur est Free, que votre nom d'utilisateur est `laurent_martin` et votre mot de passe est `arg546hh`, l'instruction s'écrira :

```
mysql_connect( "sql.free.fr", "laurent_martin", "arg546hh" );
```

De plus, certains hébergeurs vous imposent un nom de base de données MySQL, qui a souvent pour nom votre nom d'utilisateur. Il faut alors aussi modifier le nom de la base de données :

```
mysql_select_db( "laurent_martin" );
```

Pour ne pas se soucier de réécrire ces fonctions à chaque fois que l'on remonte les scripts PHP vers le site de l'hébergeur, nous allons détecter le lieu d'hébergement (local ou chez l'hébergeur

Web) et choisir ainsi les paramètres de connexion adéquats.

De plus, comme on doit écrire exactement les deux mêmes lignes (`mysql_connect` et `mysql_select_db`) dans chaque script PHP, nous allons externaliser ces lignes dans un script PHP `CONNEXION_BDD.PHP` dédié exclusivement à cette tâche, que l'on inclura dans chaque script PHP par la fonction :

```
require( "connexion_bdd.php" );
```

Création d'une nouvelle table

Quand la connexion est bien établie à votre base de données, vous pouvez créer une table. Créons la table des utilisateurs car elle va nous permettre de toucher à de nombreux aspects de la base de données. Création de table et de champs multiples, mais aussi création d'une clé primaire.

Pour cela, vous "chargez" la variable `$Query` avec les informations nécessaires à la création de la table `nom_de_la_table` et du champ `nom_du_champ` de type `x` (ici `varchar`) et de longueur `z` (ici `64`). Si vous souhaitez créer plusieurs champs dans votre table, indiquez les chacun avec leur caractéristiques, les uns à la suite des autres séparés par des virgules.

Le tout entre " " :

```
$Query = "CREATE TABLE nom_de_la_table (nom_du_champ VARCHAR(64))";
```

Cette variable est ensuite passée en paramètre à la fonction `mysql_query`, en même temps que les coordonnées de la base. Cette fonction crée la table ou bien retourne un message d'erreur :

```
mysql_query($Query, $mysql_link) or die("message d'erreur");
```



`mysql_query` est la fonction de base de manipulation d'une base de données, vous le verrez au fur et à mesure c'est avec cette fonction que vous créerez des tables et les manipulerez.

Dans notre cas, cela donne ceci :

```
/*crée la table "membres" qui se compose des champs suivants : "ID"
(numérique sur 10 chiffres) qui n'est pas nul et s'incrémente
automatiquement ; "nom" de type varchar et de longueur 64, "prenom" de
type varchar et de longueur 64. Définit comme clé primaire le champ
"ID".
```

```
$Query est passé à la fonction mysql-query qui l'exécute...
```

```
*/
```

```
$Query = "CREATE TABLE membres (ID INT (10) NOT NULL AUTO_INCREMENT,
nom VARCHAR(64), prenom VARCHAR (64), PRIMARY KEY (ID))";
```

```
mysql_query($Query, $mysql_link) or die("ERREUR creation de la table");
```

Au bout du compte, la table "membres" doit être créée avec les champs indiqués:

table des membres

| ID | nom | prenom | login | password |
|----|-----|--------|-------|----------|
|----|-----|--------|-------|----------|

ID : est un numéro non nul généré automatiquement.

nom: son nom ;-)

prénom : son prénom ;-)

login : identifiant ou surnom qu'il indiquera pour entrer dans le site

password : le mot de passe qui confirmera son identité(en anglais parce que il n'y a qu'un mot au lieu de trois en français).

Insertion des enregistrements

Une fois la table créée, nous allons lui donner un premier enregistrement.

Dans la table **membres** y a 5 champs, mais nous ne devons en remplir que 4 : **nom** et **prenom**, **login** et **password**, car **ID** sera automatiquement rempli (un chiffre) pour chaque enregistrement que nous entrerons !



Quand votre table comporte plusieurs champs, vous devez obligatoirement entrer un enregistrement pour chaque champs, même si vous n'avez rien à mettre dedans ET vous devez entrer les enregistrements dans le même ordre où ils ont été créés dans la table.

Quand il y a un champ auto-incrémenté vous devez.....

...

```
$Query = "INSERT INTO membres  
VALUES(0, 'Benoît', 'Spinoza', 'ethique', '12ah75b')";  
mysql_query($Query, $mysql_link) or die("ERREUR");
```

...

Récapitulons tout notre code pour la création de la table **membres** :

```
<?
```

```
/*
```

```
script complet de connexion à une base de donnée
```

```
et
```

```
Création d'une table, des champs et insertion d'enregistrements
```

```
*/
```

```
$mysql_link = mysql_connect("localhost","root","") or die("ERREUR  
connection");  
mysql_select_db("biblio") or die("ERREUR base de donnée");  
$Query = "CREATE TABLE membres (ID INT (10) NOT NULL AUTO_INCREMENT,  
nom VARCHAR(64), prenom VARCHAR (64), login VARCHAR(50), password  
VARCHAR(50), PRIMARY KEY (ID))";  
mysql_query($Query, $mysql_link) or die("ERREUR creation table");  
$Query = "INSERT INTO membres  
VALUES(0, 'Benoît', 'Spinoza', 'ethique', '12ah75b')";  
mysql_query($Query, $mysql_link) or die("ERREUR");  
?>
```

Formulaire HTML : récupérer des données

Nous avons créé nos tables, mais elle ne contiennent encore presque rien. Avouez qu'il serait pénible d'entrer tous les enregistrements en tapant le code PHP correspondant.

Lire un enregistrement dans la table MySQL

Maintenant que la connexion au serveur MySQL est démarrée et que la base de données "biblio" est sélectionnée, nous pouvons lire un enregistrement de la table "livres". Cela se fait avec 2 instructions successives :

```
$liste_livres = mysql_query( "select * from livres");  
$livre = mysql_fetch_array( $liste_livres );
```

`query` est l'ordre de requête passé avec en paramètre la sélection des champs : * (c'est à dire, ici, tous les champs) de la table livres.

Le script complet ressemble à ceci:

```
<?  
mysql_connect( "localhost" , "" , "" );  
mysql_select_db( "biblio" );  
$liste_livres = mysql_query( "select * from livres");  
$livre = mysql_fetch_array( $liste_livres );  
>
```

1. Nous nous connectons à la base de données
2. Nous nous connectons à la table Biblio
3. Nous lisons les enregistrement de la table livres
4. Nous les stockons dans le tableau \$livre.

Insérer un script pour chaque champ à afficher

Nous devons modifier encore notre script afin qu'il puisse afficher tous les champs.

Pour afficher la valeur du champ "titre", on insère le script suivant :

```
<? echo $livre[titre]; ?>
```

Pour insérer la valeur du champ "auteur", on insère le script suivant :

```
<? echo $livre[auteur]; ?>
```

Pour insérer la valeur du champ "description_courte", on insère le script suivant :

```
<? echo $livre[description_courte]; ?>
```

Trier les enregistrements

Nous voulons afficher seulement les deux derniers livres afin de générer une page HTML de taille limitée. Si l'on ne faisait pas de sélection, on afficherait tous les livres de la bibliothèque, et la taille de la page HTML s'exprimerait probablement en centaines de Ko ! et nous voulons que l'affichage soit le plus rapide possible.

- Le livre le plus récent sera affiché en premier.

Pour connaître les derniers livres, il faut ajouter la notion de date d'arrivée dans le catalogue de livres. Pour cela, on ajoute le champ `date_entree` de type DATE, qui devra être renseigné à chaque nouvel ajout de livre dans la table livres.

Le programme PHP doit maintenant sélectionner les derniers livres. Pour cela, nous allons devoir utiliser la commande SQL qui permet de trier :

```
ORDER BY nom_du_champ [DESC] [, nom_du_champ_2 [DESC] [,etc.] ]
```

Elle s'intègre dans la fonction **SELECT**.



Souvenez-vous en travaillant que les dates sont gérées sur le modèle américains : année-mois-jour ! Ne confondez pas en manipulant les dates !

Nous voulons trier les livres par la date d'entrée : de la plus grande (exemple 2001-04-16) à la plus petite (exemple 1999-10-30). Il s'agit donc d'un tri décroissant (du plus grand au plus petit), qui est renseigné par l'expressions **DESC** après le nom du champ.



Si vous voulez un tri croissant : il suffit de ne pas indiquer **DESC** après le nom du champ.

De plus, nous voulons sélectionner seulement 2 livres. Il suffit d'ajouter la commande SQL **LIMIT(nombre_maximum_d_enregistrements)** à la fonction **SELECT**.

Comme ceci :

```
select * from livres order by date_entree desc limit(2)
```

Répéter les lignes d'un tableau HTML grâce à PHP

Vous pouvez créer automatiquement les lignes et les cellules d'une table autant de fois que nécessaire, grâce à un script PHP.

```
<table>
<?
mysql_connect( "localhost" , "" , "" );
mysql_select_db( "biblio" );
$liste_livres = mysql_query( "select * from livres" );
$nombre_livres = mysql_num_records( $liste_livres );
for( $i = 1 ; $i <= $nombre_livres ; $i++ )
    {
    $livre = mysql_fetch_array( $liste_livres ) )
    ?>
    <tr><td colspan=2><hr></td></tr>
    <tr>
    <td></td>
    <td>
    <? echo "<p> $livre[titre] </p>" ; ?>
    <? echo "<p> $livre[auteur] </p>"; ?>
    <? echo "<p> $livre[description_courte] </p>"; ?>
    </td>
    </tr>
    <?    } // fin de la boucle "for" ?>
</table>
```

Fichiers et répertoires

Ouvrir et lire un fichier

Avec PHP vous pouvez lire et écrire dans un fichier. Pour cela vous devez d'abord l'ouvrir

fopen()

La fonction `fopen()` vous permet d'ouvrir un fichier avec certains droits que vous lui communiquez en argument (voyez le tableau ci-dessous).

Par exemple:

```
fopen("http://www.mon.site.com/mon_fichier", "r")
fopen("ftp://www.mon.site.com/mon_fichier", "r")
fopen("c:/mon_dossier/mon_fichier", "r")
```

Principaux modes d'accès:

- r Ouvrir en lecture seule
- r+ Ouvrir en lecture et écriture
- w Ouvrir en écriture, si le fichier existe déjà il sera effacé !
- w+ Ouvrir en lecture et écriture, si le fichier existe déjà il sera effacé!
- a Ouvrir en écriture, si le fichier existe les données seront ajoutées à la suite des autres. Sinon le fichier sera créé
- a+ Ouvrir en lecture et écriture, si le fichier existe les données seront ajoutées à la suite des autres. Sinon le fichier sera créé

Le fichier ouvert un pointeur pointe sur le début du fichier. Vous pouvez maintenant lire le contenu du fichier.

fgets()

La fonction `fgets()` permet de lire ligne après ligne le fichier jusqu'à la fin de ce fichier qui est détectée par `feof` (file end of file).

```
//ouvrir un fichier texte enregistrés dans "C:/un_fichier.txt"
<?
//on vérifie que le fichier mon_fichier.txt existe et peut-être ouvert
//sinon, on affiche le message d'erreur et le script se termine
if(!$mon_fichier= fopen("C:/un_fichier.txt", "r")) {
    print("Le fichier n'a pas pu être ouvert");
    exit;
}
//Si on est à la fin du fichier la boucle n'est pas exécutée,
//mais tant qu'on est pas à la fin du fichier...
while (!feof($mon_fichier)) {
    //...on lit une ligne du fichier. Vous devez donner une longueur
    //maximale à la ligne qui sera lue, nous avons indiqué 255
    $ligne=fgets($mon_fichier,255);
    //on affiche chaque ligne lue
    print("$ligne<br>\n");
}
```

```
//on referme le fichier
fclose($mon_fichier);
?>
```

fgetss(strip)

Écrire dans un fichier

Pour écrire dans un fichier, vous devez l'ouvrir avec les droits correspondants (reportez-vous au tableau ci-dessus) : "a" ou "w".

Vous écrivez avec la fonction fwrite.

fermez le fichier avec fclose.

Créer un fichier

Pour créer un fichier vous devez ouvrir un fichier qui n'existe pas encore en lui donnant les droits d'écriture w.

vous le remplissez avec fwrite

et vous le fermez avec fclose

Lire un répertoire

Vous pouvez lire un répertoire comme vous lisez un fichier, le résultat sera une liste des fichiers et, éventuellement, des sous-répertoire contenus.

Comme pour un fichier, il faut indiquer l'adresse du répertoire et ordonner son ouverture pour ensuite le "lire" ligne après ligne son contenu. La dernière ligne lue et affichée, vous ordonnerez la fermeture du répertoire.

```
<?
//Définissez le chemin du répertoire
$repertoire_lu="C:/mon_repertoire";
//ouvrez le répertoire à lire
$dir=opendir($repertoire_lu);
//Pointez sur le premier fichier du répertoire
while($file= readdir($dir)){
//affiche à l'écran le nom du fichier
    echo "$file<br>";
}
//on sort et on ferme le répertoire quand il a été entièrement lu
closedir($dir);
?>
```

A l'affichage, vous obtiendrez

.
..

plus des noms de fichiers ou de répertoires si il y en a

"." c'est le répertoire courant (celui qui est lu)

".." c'est le répertoire supérieur (qui contient celui qui est lu): dans notre cas C:/



Pour ne pas afficher ces deux répertoires particuliers, il suffit d'ajouter un peu de code:


```

while($file= readdir($dir)){
if (($file!=".")&&($file!="..")) echo "$file<br>";
}

```

Un peu d'histoire. Les répertoires sont traités comme ils le sont dans les systèmes Unix/Linux : ce sont des fichiers comme les autres mais qui contiennent des les adresses d'autres fichiers, simplement.

Pour lire un répertoire et tous les sous-répertoire il faut appliquer le même script à chaque fois qu'on rencontre un sous-répertoire. Le plus simple c'est alors de faire une fonction de notre script !

```

<?
//création de la fonction de lecture de répertoire
function liste_rep($rep_lu){
//on ouvre le repertoire racine
$dir=opendir($rep_lu);
while($file= readdir($dir)){
if (($file!=".")&&($file!="..")){
//$file est-elle un répertoire ? on le demande à la
fonction is_dir
//Si elle retourne vrai $file est un répertoire, sinon
un fichier
//Il faut lui donner l'adresse exacte de ce qu'elle
doit tester :
//tout le chemin d'accès
if(is_dir("$rep_lu/$file")){
// C'est un répertoire alors il faut le lister
son contenu
liste_rep("$rep_lu/$file");
}
else{
//Sinon c'est un fichier que l'on doit afficher!
echo "$rep_lu/$file<br>";
}
}
}
}
closedir ($dir);
}
//adresse du répertoire racine
$racine="C:/mon_repertoire";
//appel de la fonction
liste_rep($racine);
?>

```

La date et l'heure

Les dates et les heures sont stockées dans des variables de type nombre entier.

Ce nombre correspond au nombre de secondes écoulées depuis le 1er janvier 1970 (qui se veut être le début de l'ère UNIX). On appelle ce nombre "timestamp".

Un timestamp a la particularité de stocker à la fois une date et une heure, puisqu'il est exprimé en nombre de secondes.

Cependant, les deux parties peuvent être traitées distinctement.

Opérations sur les dates

L'exemple suivant montre comment ajouter des jours à une date. On crée la fonction `ajuste_date_heure` car la fonction système `mktime` est insuffisante pour les opérations sur les mois :

AJUSTE_DATE_HEURE.PHP

```
<?
function ajuste_date_heure ( $date_heure , $nombre_de_jours = 0 ,
$nombre_de_mois = 0 , $nombre_d_annees = 0 , $nombre_d_heures = 0 ,
$nombre_de_minutes = 0 , $nombre_de_secondes = 0 )
{
    $date_heure_tab = getdate( $date_heure ) ; // récupère la date
décomposée sous forme de tableau
    if( $nombre_de_mois != 0 || $nombre_d_annees != 0 )
    {
        $date_heure2 = mktime( 0 , 0 , 0 , $date_heure_tab[
"mon" ] + $nombre_de_mois + 1 , 0 , $date_heure_tab[ "year" ] +
$nombre_d_annees ) ;
        $date_heure2_tab = getdate( $date_heure2 ) ;
        if ( $date_heure_tab[ "mday" ] > $date_heure2_tab[
"mday" ] )
        {
            $date_heure = mktime( $date_heure_tab[
"hours" ] , $date_heure_tab[ "minutes" ] , $date_heure_tab[ "seconds" ] ,
$date_heure_tab[ "mon" ] , $date_heure2_tab[ "mday" ] , $date_heure_tab[
"year" ] ) ;
            $date_heure_tab = getdate( $date_heure ) ;
// récupère la date décomposée sous forme de tableau
        }
    }
    return( mktime( $date_heure_tab[ "hours" ] + $nombre_d_heures ,
$date_heure_tab[ "minutes" ] + $nombre_de_minutes , $date_heure_tab[
"seconds" ] + $nombre_de_secondes , $date_heure_tab[ "mon" ] +
$nombre_de_mois , $date_heure_tab[ "mday" ] + $nombre_de_jours ,
$date_heure_tab[ "year" ] + $nombre_d_annees ) ) ;
}

$a = time() ; // récupération aujourd'hui et heure courante
setlocale ( "LC_TIME", "fr" ); // utilisation du français pour %a, %A, %b,
%B de strftime

print( "Aujourd'hui, nous sommes le : " . strftime( "%A %d %B %Y" , $a ) )
;
print( "<br>" ) ;

$demain = ajuste_date_heure( $a , +1 ) ;
print( "Demain, nous serons le : " . strftime( "%A %d %B %Y" , $demain ) )
;
print( "<br>" ) ;

$dans_un_mois = ajuste_date_heure( $a , 0 , +1 ) ;
print( "Dans 1 mois, nous serons le : " ) ;
print( strftime( "%A %d %B %Y" , $dans_un_mois ) ) ;
```

```

print( "<br>" ) ;

$b = mktime( 0 , 0 , 0 , 3 , 31 , 2001 ) ;
$b2 = ajuster_date_heure( $b , 0 , -1 ) ;
print( "31 mars 2001 - 1 mois donne : " ) ;
print( strftime( "%A %d %B %Y" , $b2 ) ) ;
print( "<br>" ) ;

$b = mktime( 0 , 0 , 0 , 3 , 31 , 2000 ) ;
$b2 = ajuster_date_heure( $b , 0 , -1 ) ;
print( "31 mars 2000 - 1 mois donne : " ) ;
print( strftime( "%A %d %B %Y" , $b2 ) ) ;
print( "<br>" ) ;

$b = mktime( 22 , 50 , 0 , 3 , 31 , 2001 ) ;
$b2 = ajuster_date_heure( $b , 0 , -1 , 0 , 1 , 20 , 0 ) ;
print( "(31 mars 2001 22:50) - (1 mois + 1h20) donne : " ) ;
print( strftime( "%A %d %B %Y" , $b2 ) ) ;
print( "<br>" ) ;

```

?>

```

-----
Aujourd'hui, nous sommes le : vendredi 16 mars 2001
Demain, nous serons le : samedi 17 mars 2001
Dans 1 mois, nous serons le : lundi 16 avril 2001
31 mars 2001 - 1 mois donne : mercredi 28 février 2001
31 mars 2000 - 1 mois donne : mardi 29 février 2000
(31 mars 2001 22:50) - (1 mois + 1h20) donne : jeudi 01 mars 200
-----

```

Les fichiers PDF

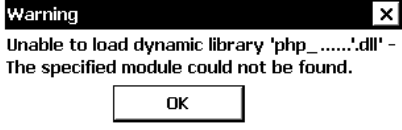
Ce script permet de générer une page

<?

```
// format A4
$pagewidth=595.35;
$pageheight=841.95;
$font = "Times-BoldItalic";
$fontsize = $fontsizedefault = 10;
$text="le lapin des plaines.";
$cursorx=100;
$ cursory=770;
$x=400;
$lineheight=50;
$alignment = $alignmentdefault = "left";
$fp = fopen( "fichier.pdf", "w" ); //
$pdf = pdf_open( $fp );
pdf_begin_page($pdf, $pagewidth, $pageheight);
pdf_set_font($pdf, $font, $fontsize, "host");
pdf_show_boxed($pdf, $text, $cursorx, $ cursory, $x, $lineheight,
$alignment);
pdf_end_page( $pdf );
pdf_close( $pdf );
fclose( $fp );
?>
```

Erreurs fréquentes

Suite à la saisie d'une adresse URL, vous obtenez un des messages suivants sur votre navigateur internet :

| N° erreur | Message d'erreur | Pourquoi... Que faire... |
|-----------|--|---|
| 1 | page non trouvée | Si l'adresse URL entrée est <code>http://localhost</code> , vérifiez les paramètres <code>DirectoryIndex</code> et <code>DocumentRoot</code> dans le fichier <code>HTTPD.CONF</code> (voir page Erreur ! Signet non défini.) |
| 2 | | Consultez le fichier <code>C:/PROGRAM FILES/APACHE GROUP/APACHE/LOGS/ERROR.LOG</code> |
| 3 | Warning: <fonction()> is not supported in this PHP build in <nom du script PHP où la fonction est appelée> on line <numéro ligne> | La fonction PHP exécutée est reconnue par PHP, mais elle n'est pas contenue dans cette version de PHP (<i>build</i>). L'application ne peut pas fonctionner à moins que vous n'installiez une nouvelle version de PHP avec l'extension demandée. |
| 4 |  <p>Fatal error: Call to undefined function: <fonction()> in <nom du script PHP où la fonction est appelée> on line <numéro ligne></p> | <p>Plusieurs boîtes de messages peuvent s'enchaîner indiquant que plusieurs fichiers <code>php_....dll</code> ne sont pas trouvés.</p> <p>Certaines versions de PHP ne reconnaissent pas la ligne suivante du fichier <code>PHP.INI</code> :</p> <pre>extension_dir=c:/progra~1/php/extension</pre> <p>à cause du caractère tilde (~). Cette ligne indique le répertoire où se trouve les fichiers dll. Pour corriger, voir page Erreur ! Signet non défini.</p> |

Erreurs fréquentes dans `c:/Program Files/Apache Group/Apache/logs/error.log`

Les erreurs sont écrites dans l'ordre chronologique. Aussi, ne prenez en compte que la dernière erreur du fichier.


















| Message d'erreur dans <code>php.ini</code> | Pourquoi... Que faire... |
|--|---------------------------|
| File does not exist: <nom fichier> | |
| FATAL: <code>emalloc(): Unable to allocate 1292894208 bytes</code> | |
| Directory index forbidden by rule: <répertoire> | |
| PHP Warning: Unable to load dynamic library <nom dll> - The specified module could not be found. | Conséquence de l'erreur 4 |

| Message d'erreur dans php.ini | Pourquoi... Que faire... |
|---|---|
| Premature end of script headers:
c:/program files/php/php.exe | Conséquence de l'erreur 4 |
| PHP: Error parsing <php.ini> on line
<ligne en erreur dans php.ini> | |
| Invalid URI in request GET <adresse
URL entrée dans le navigateur>
HTTP/1.1 | Apache ne trouve pas d'association à l'adresse URL
faisant référence à un programme PHP.
Vérifiez les paramètres ScriptAlias , AddType et
Action dans le fichier HTTPD.CONF . Voir page Erreur !
Signet non défini. |


Exécution de programme php

| Effet | Cause possible | Pourquoi... Que faire... |
|---|---|--|
| Programme interrompu
brutalement sans
explication | | Temps maximum d'exécution dépassé.
Augmentez la valeur du paramètre
max_execution_time dans le fichier
PHP.INI . Il est recommandé de vérifier si le
programme ne perd pas de temps
inutilement (optimiser le programme si
nécessaire) |
| Fonctionnement
incompréhensible | | La casse du nom d'une variable est
inconstante. Choisissez une casse et
appliquez-la à la variable partout dans le
programme |
| Fonctionnement
incompréhensible | | Un \$ devant une variable a été oublié.
Ajoutez le \$ |
| Fonctionnement
incompréhensible | La casse du nom d'une variable
est inconstante | La casse du nom d'une variable est
inconstante. Choisissez une orthographe
(casse) et appliquez-la à la variable
partout dans le programme |
| Fonctionnement
incompréhensible | | Test d'égalité avec = au lieu de ==.
Mettez == |

Sites internet parlant du PHP

| Site
(faire précéder par http://) | Langue | Contient |
|--|---|---|
| php-gratuit.forez.com |  | installation |
| www.nexen.net |  | manuel PHP en français |
| www.phpfrance.com |  | |
| www.zend.com | | |
| www.php.net |  | snaps.php.net offre un manuel PHP en anglais |
| www.mysql.com |  | manuel MySQL en anglais |
| www.manucorp.com/phpdev.exe |  | easyphp : un installateur de apache, PHP et MySQL |
| www.phpbuilder.com |  | |
| www.phpinfo.com |  | |
| php.faqts.com |  | Questions et réponses |
| www.xclave.com |  | Macintosh OSX |
| phpscripts.free.fr |  | |
| php.resourceindex.com |  | |
| www.hotscripts.com/PHP |  | |
| www.ilovephp.com |  | |
| php.weblogs.com |  | |
| www.phpdebutant.com |  | |
| openphp.free.fr |  | |

Autres sites

| Site
(faire précéder par http://) | Langue | Contient |
|--|---|--|
| www.w3c.com |  | Standards des langages de programmation sur internet (HTML, CSS, etc.) |

Sites internet contenant les applications nécessaires pour programmer en PHP (à télécharger)

| Site | Contient |
|---|----------|
| http://www.php.net/ | PHP |
| http://www.mysql.net/ | MySQL |
| http://httpd.apache.org/ | apache |

Les fonctions suivantes ne sont disponibles que si l'interface associée est installée :

| fonctions (ayant pour préfixe les caractères suivants) | application | description |
|--|-------------|--|
| gz | zlib | compression de fichiers |
| image | GD | gestion d'images |
| mysql_ | MySQL | base de données |
| preg_ | pcre | expressions régulières Perl |
| imageps | t1lib | gestion de fontes postscript dans les images |
| pdf_ | pdflib | création de documents PDF |

Liste de fonctions :

| | | | | |
|-----------------------|-------|-------|---------|--|
| abs | 3.0.0 | 4.0.0 | mixed | mixed number |
| acos | 3.0.0 | 4.0.0 | float | float arg |
| AddCSlashes | | 4.0b4 | string | string str, string charlist |
| AddSlashes | 3.0.0 | 4.0.0 | string | string str |
| apache_lookup_uri | 3.0.4 | 4.0.0 | class | string fichier |
| apache_note | 3.0.2 | 4.0.0 | string | string note_name [, string note_value] |
| array_count_values | | 4.0b4 | array | array input |
| array_diff | | 4.0.1 | array | array array1, array array2 [, array ...] |
| array_flip | | 4.0b4 | array | array trans |
| array_intersect | | 4.0.1 | array | array array1, array array2 [, array ...] |
| array_keys | | 4.0.0 | array | array input [, mixed search_value] |
| array_merge_recursive | | 4.0.1 | array | array array array1, array array2 [, array ...] |
| array_merge | | 4.0.0 | array | array array1, array array2 [, array ...] |
| array_multisort | | 4.0b4 | bool | array ar1 [, mixed arg [, mixed ... [, array ...]]] |
| array_pad | | 4.0b4 | array | array input, int pad_size, mixed pad_value |
| array_pop | | 4.0.0 | mixed | array array |
| array_push | | 4.0.0 | int | array array, mixed var [, mixed ...] |
| array_rand | | 4.0.0 | mixed | array input [, int num_req] |
| array_reverse | | 4.0b4 | array | array array |
| array_shift | | 4.0.0 | mixed | array array |
| array_slice | | 4.0.0 | array | array array, int offset [, int length] |
| array_splice | | 4.0.0 | array | array input, int offset [, int length [, array replacement]] |
| array_unique | | 4.0.1 | array | array array |
| array_unshift | | 4.0.0 | int | array array, mixed var [, mixed ...] |
| array_values | | 4.0.0 | array | array input |
| array_walk | 3.0.3 | 4.0.0 | int | array arr, string func, mixed userdata |
| array | 3.0.0 | 4.0.0 | array | [mixed ...] |
| arsort | 3.0.0 | 4.0.0 | void | array array [, int sort_flags] |
| asort | 3.0.0 | 4.0.0 | void | array array [, int sort_flags] |
| aspell_check-raw | 3.0.7 | 4.0.0 | boolean | int dictionary_link, string word |
| aspell_check | 3.0.7 | 4.0.0 | boolean | int dictionary_link, string word |

| | | | | |
|--------------------|-------|-------|--------|--|
| aspell_new | 3.0.7 | 4.0.0 | int | string master, string personal |
| aspell_suggest | 3.0.7 | 4.0.0 | array | int dictionary_link, string word |
| assert | | 4.0b4 | int | string bool assertion |
| assert-options | | 4.0b4 | mixed | int what [, mixed value] |
| base_convert | 3.0.6 | 4.0.0 | string | string number, int frombase, int tobase |
| base64_decode | 3.0.0 | 4.0.0 | string | string encoded_data |
| base64_encode | 3.0.0 | 4.0.0 | string | string data |
| basename | 3.0.0 | 4.0.0 | string | string chemin |
| bindtextdomain | 3.0.7 | 4.0.0 | string | string chemin_fichier |
| call_user_func | 3.0.3 | 4.0.0 | string | string domaine, string répertoire |
| | | | mixed | string function_name [, mixed parameter [, mixed ...]] |
| call_user_method | 3.0.3 | 4.0.0 | mixed | string method_name, object obj [, mixed parameter [, mixed ...]] |
| ceil | 3.0.0 | 4.0.0 | int | float number |
| chdir | 3.0.0 | 4.0.0 | int | string répertoire |
| checkdate | 3.0.0 | 4.0.0 | int | int month, int day, int year |
| checkdnsrr | 3.0.0 | 4.0.0 | int | string host [, string type] |
| chop | 3.0.0 | 4.0.0 | string | string texte |
| chr | 3.0.0 | 4.0.0 | string | int code_ascii_caractère |
| chunk_split | 3.0.6 | 4.0.0 | string | string caractère |
| class_exists | | 4.0b4 | bool | int code_ascii_caractère |
| clearstatcache | 3.0.0 | 4.0.0 | void | string texte [, int chunklen [, string end]] |
| closedir | 3.0.0 | 4.0.0 | void | string class_name |
| closelog | 3.0.0 | 4.0.0 | int | void |
| connection_aborted | 3.0.7 | 4.0b4 | int | int dir_handle |
| connection_status | 3.0.7 | 4.0b4 | int | mixed varname [, mixed ...] |
| connection_timeout | 3.0.7 | 4.0b4 | int | void |
| copy | 3.0.0 | 4.0.0 | int | void |
| count_chars | | 4.0b4 | mixed | string source, string destination |
| count | 3.0.0 | 4.0.0 | int | string texte [, mode] |
| crc32 | | 4.0.1 | int | mixed var |
| create_function | | 4.0.1 | string | string str |
| crypt | 3.0.0 | 4.0.0 | string | string args, string code |
| current | 3.0.0 | 4.0.0 | mixed | string str [, string salt] |
| date | 3.0.0 | 4.0.0 | string | array array |
| dblist | 3.0.0 | 4.0.0 | string | string format [, int timestamp] |
| dcgettext | 3.0.7 | 4.0.0 | string | void |
| debugger_off | 3.0.0 | non | int | string domaine, string message, int category |
| decbin | 3.0.0 | 4.0.0 | string | string adresse |
| dechex | 3.0.0 | 4.0.0 | string | string binaire |
| decoct | 3.0.0 | 4.0.0 | string | int nombre |
| define | 3.0.0 | 4.0.0 | int | int nombre |
| defined | 3.0.0 | 4.0.0 | int | string name |
| deg2rad | 3.0.4 | 4.0.0 | double | string name, mixed value [, int case_insensitive] |
| delete | 3.0.0 | 4.0.0 | void | double number |
| dgettext | 3.0.7 | 4.0.0 | string | string file |
| die | 3.0.0 | 4.0.0 | void | string domain, string message |
| dirname | 3.0.0 | 4.0.0 | string | string message |
| diskfreespace | 3.0.7 | 4.0b4 | float | string path |
| dl | 3.0.0 | 4.0.0 | int | string directory |
| doubleval | 3.0.0 | 4.0.0 | double | string library |
| | | | | mixed var |

| | | | | |
|-----------------------------|--------|--------|---------------|---|
| each | 3.0.0 | 4.0.0 | array | array array |
| easter_date | 3.0.9 | 4.0RC2 | int | int year |
| easter_days | 3.0.9 | 4.0RC2 | int | int year |
| echo | 3.0.0 | 3.0.0 | echo | mixed var |
| end | 3.0.0 | 4.0.0 | end | string pattern, string replacement, string string |
| ereg | 3.0.0 | 4.0.0 | int | string pattern, string string [, array regs] |
| eregi_replace | 3.0.0 | 4.0.0 | string | string pattern, string replacement, string string |
| eregi | 3.0.0 | 4.0.0 | int | string pattern, string string [, array regs] |
| error_log | 3.0.0 | 4.0.0 | int | string message, int message_type [, string destination [, string extra_headers]] |
| error_reporting | 3.0.0 | 4.0.0 | int | [int level] |
| escapeshellcmd | 3.0.0 | 4.0.0 | string | string commande |
| eval | 3.0.0 | 4.0.0 | mixed | string code_str |
| exec | 3.0.0 | 4.0.0 | string | string commande [, string array [, int return_var]] |
| exit | 3.0.0 | 4.0.0 | void | float arg |
| explode | 3.0.0 | 4.0.0 | array | string séparateur, string string [, int limit] |
| extension_loaded | 3.0.10 | 4.0b4 | bool | string name |
| extract | 3.0.7 | 4.0.0 | void | array var_array [, int extract_type [, string prefix]] |
| fclose | 3.0.0 | 4.0.0 | int | int fichierID |
| feof | 3.0.0 | 4.0.0 | int | int fichierID |
| fgetc | 3.0.0 | 4.0.0 | string | int fichierID |
| fgetcsv | 3.0.8 | 4.0.0 | array | int fichierID, int length [, string delimiter] |
| fgets | 3.0.0 | 4.0.0 | string | int fichierID, int length |
| fgetss | 3.0.0 | 4.0.0 | string | int fichierID, int length [, string allowable_tags] |
| file_exists | 3.0.0 | 4.0.0 | int | string fichier |
| file | 3.0.0 | 4.0.0 | array | string fichier [, int use_include_path] |
| fileatime | 3.0.0 | 4.0.0 | int | string fichier |
| filectime | 3.0.0 | 4.0.0 | int | string fichier |
| filemtime | 3.0.0 | 4.0.0 | int | string fichier |
| fileperms | 3.0.0 | 4.0.0 | int | string fichier |
| filesize | 3.0.0 | 4.0.0 | int | string fichier |
| filetype | 3.0.0 | 4.0.0 | string | string fichier |
| flock | 3.0.7 | 4.0.0 | bool | int fichierID, int operation [, int wouldblock] |
| floor | 3.0.0 | 4.0.0 | int | float number |
| flush | 3.0.0 | 4.0.0 | void | string fichier, string mode [, int use_include_path] |
| fopen | 3.0.0 | 4.0.0 | int fichierID | string fichier, string mode = r r+ w w+ a a+ [, int chercher_dans_liste_repertoires = 1]) |
| fpassthru | 3.0.0 | 4.0.0 | int | int fichierID |
| fputs | 3.0.0 | 4.0.0 | int | int fichierID, string str [, int longueur] |
| fread | 3.0.0 | 4.0.0 | string | int fichierID, int longueur |
| FrenchToJD | 3.0.0 | 4.0.0 | int | int mois, int jour, int année |
| fscanf | | 4.0.1 | mixed | int handle, string format [, string var1...] |
| fseek | 3.0.0 | 4.0.0 | int | int fichierID, int offset [, int whence] |
| fsockopen | 3.0.0 | 4.0.0 | int | string [udp://]hostname, int port [, int errno [, string errstr [, double timeout]]] |
| fstat | | 4.0rc1 | array | int fichierID |
| ftell | 3.0.0 | 4.0.0 | int | int fichierID |
| <u>FTP fonctions</u> | | | | |
| ftp_cdup | 3.0.13 | 4.0b4 | int | int ftp_stream |

| | | | | |
|----------------------------|--------|--------|--------|---|
| ftp_chdir | 3.0.13 | 4.0b4 | int | int ftp_stream, string directory |
| ftp_connect | 3.0.13 | 4.0b4 | int | string host [, int port] |
| ftp_delete | 3.0.13 | 4.0b4 | int | int ftp_stream, string path |
| ftp_fget | 3.0.13 | 4.0b4 | int | int ftp_stream, int fp, string remote_file, int mode |
| ftp_fput | 3.0.13 | 4.0b4 | int | int ftp_stream, string remote_file, int fp, int mode |
| ftp_get | 3.0.13 | 4.0b4 | int | int ftp_stream, string local_file, string remote_file, int mode |
| ftp_login | 3.0.13 | 4.0b4 | int | int ftp_stream, string username, string password |
| ftp_mdtm | 3.0.13 | 4.0b4 | int | int ftp_stream, string remote_file |
| ftp_mkdir | 3.0.13 | 4.0b4 | string | int ftp_stream, string directory |
| ftp_nlist | 3.0.13 | 4.0b4 | int | int ftp_stream, string directory |
| ftp_pasv | 3.0.13 | 4.0b4 | int | int ftp_stream, int pasv |
| ftp_put | 3.0.13 | 4.0b4 | int | int ftp_stream, string remote_file, string local_file, int mode |
| ftp_pwd | 3.0.13 | 4.0b4 | int | int ftp_stream |
| ftp_quit | 3.0.13 | 4.0b4 | int | int ftp_stream |
| ftp_rawlist | 3.0.13 | 4.0b4 | int | int ftp_stream, string directory |
| ftp_rename | 3.0.13 | 4.0b4 | int | int ftp_stream, string from, string to |
| ftp_rmdir | 3.0.13 | 4.0b4 | int | int ftp_stream, string directory |
| ftp_site | 3.0.15 | 4.0RC1 | int | int ftp_stream, string cmd |
| ftp_size | 3.0.13 | 4.0b4 | int | int ftp_stream, string remote_file |
| ftp_systype | 3.0.13 | 4.0b4 | int | int ftp_stream |
| ftruncate | | 4.0rc1 | int | int fp, int size |
| func_get_arg | | 4.0b4 | int | int arg_num |
| func_get_args | | 4.0b4 | int | void |
| func_num_args | | 4.0b4 | int | void |
| function_exists | 3.0.7 | 4.0.0 | int | string function_name |
| fwrite | 3.0.0 | 4.0.0 | int | int fp, string string [, int length] |
| get_browser | 3.0.0 | 4.0.0 | object | [string user_agent] |
| get_cfg_var | 3.0.0 | 4.0.0 | string | string varname |
| get_class_methods | | 4.0rc1 | array | string class_name |
| get_class_vars | | 4.0rc1 | array | string class_name |
| get_class | | 4.0b2 | string | object obj |
| get_current_user | 3.0.0 | 4.0.0 | string | void |
| get_declared_classes | | 4.0rc2 | array | void |
| get_extension_funcs | | 4.0b4 | array | string module_name |
| get_html_translation_table | | 4.0b4 | string | int table |
| get_included_files | | 4.0rc1 | array | void |
| get_loaded_extensions | | 4.0b4 | array | void |
| get_magic_quotes_gpc | 3.0.6 | 4.0.0 | long | void |
| get_magic_quotes_runtime | 3.0.6 | 4.0.0 | long | void |
| get_meta_tags | 3.0.4 | 4.0.0 | array | string fichier [, int use_include_path] |
| get_object_vars | | 4.0rc1 | array | object obj |
| get_parent_class | | 4.0b2 | string | object obj |
| get_required_files | | 4.0rc2 | array | void |
| GetAllHeaders | 3.0.0 | 4.0.0 | array | void |
| GetDate | 3.0.0 | 4.0.0 | array | int timestamp |
| GetEnv | 3.0.0 | 4.0.0 | string | string varname |
| GetHostByAddr | 3.0.0 | 4.0.0 | string | string ip_address |
| GetHostByName | 3.0.0 | 4.0.0 | string | string hostname |
| GetHostByNameI | 3.0.0 | 4.0.0 | array | string hostname |

| | | | | |
|-------------------------------|-------|-------|---------------|--|
| GetImageSize | 3.0.0 | 4.0.0 | array | string fichier [, array imageinfo] |
| GetLastMod | 3.0.0 | 4.0.0 | int | void |
| getmxrr | 3.0.0 | 4.0.0 | int | string hostname, array mxhosts [, array weight] |
| GetProtoByName | | 4.0b4 | int | string name |
| GetProtoByNumber | | 4.0b4 | string | int number |
| GetRandMax | 3.0.0 | 4.0.0 | int | void |
| GetRUsage | 3.0.7 | 4.0b2 | array | [int who] |
| GetServByName | | 4.0b4 | int | string service, string protocol |
| GetServByPort | | 4.0b4 | string | int port, string protocol |
| GetText | 3.0.7 | 4.0.0 | string | string message |
| GetTimeOfDay | 3.0.7 | 4.0b4 | array | void |
| GetType | 3.0.0 | 4.0.0 | string | mixed var |
| GregorianToJD | 3.0.0 | 4.0.0 | int | int month, int day, int year |
| <u>GZ fonctions</u> | | | | |
| gzclose | 3.0.0 | 4.0.0 | int | int fichierID |
| gzcompress | | 4.0.1 | string | string data [, int level] |
| gzEOF | 3.0.0 | 4.0.0 | int | int fichierID |
| gzfile | 3.0.0 | 4.0.0 | array | string fichier [, int use_include_path] |
| gzgetc | 3.0.0 | 4.0.0 | string | int fichierID |
| gzgets | 3.0.0 | 4.0.0 | string | int fichierID, int length |
| gzgetss | 3.0.0 | 4.0.0 | string | int zp, int length [, string allowable_tags] |
| gzopen | 3.0.0 | 4.0.0 | int fichierID | string fichier, string mode [, int use_include_path] |
| gzpassthru | 3.0.0 | 4.0.0 | int | int fichierID |
| gzputs | 3.0.0 | 4.0.0 | int | int fichierID, string str [, int length] |
| gzread | 3.0.0 | 4.0.0 | string | int fichierID, int length |
| gzrewind | 3.0.0 | 4.0.0 | int | int fichierID |
| gzseek | 3.0.0 | 4.0.0 | int | int fichierID, int offset |
| gztell | 3.0.0 | 4.0.0 | int | int fichierID |
| gzuncompress | | 4.0.1 | string | string donnees [, int longueur_donnees] |
| gzwrite | 3.0.0 | 4.0.0 | int | int fichierID, string string [, int longueur] |
| header | 3.0.0 | 4.0.0 | int | string texte |
| header_sent | ? | ? | boolean | void |
| highlight_file | | 4.0.0 | void | string fichier |
| highlight_string | | 4.0.0 | void | string texte |
| HtmlEntities | 3.0.0 | 4.0.0 | string | string texte [(3.0.17/4.0.3), ENT_COMPAT ENT_QUOTES ENT_NOQUOTES] |
| HtmlSpecialChars | 3.0.0 | 4.0.0 | string | string string [(3.0.17/4.0.3), ENT_COMPAT ENT_QUOTES ENT_NOQUOTES] |
| ignore_user_abort | 3.0.7 | 4.0b4 | int | [int setting] |
| <u>IMAGE fonctions</u> | | | | |
| ImageArc | 3.0.0 | 4.0.0 | int erreur | int imageID, int centre_x, int centre_y, int largeur, int hauteur, int degre_debut, int degre_fin, int couleurID |
| ImageChar | 3.0.0 | 4.0.0 | int erreur | int imageID, int fonteID, int x, int y, string c, int couleurID |
| ImageCharUp | 3.0.0 | 4.0.0 | int erreur | int imageID, int fonteID, int x, int y, string c, int couleurID |
| ImageColorAllocate | 3.0.0 | 4.0.0 | int couleurID | int imageID, int rouge, int vert, int bleu |
| ImageColorAt | 3.0.0 | 4.0.0 | int couleurID | int imageID, int x, int y |
| ImageColorClosest | 3.0.0 | 4.0.0 | int couleurID | int imageID, int rouge, int vert, int bleu |

| | | | | |
|-----------------------|--------|--------|-----------------------------|--|
| ImageColorDeAllocate | 3.0.6 | 4.0.0 | int erreur | int imageID, int couleurID |
| ImageColorExact | 3.0.0 | 4.0.0 | int couleurID | int imageID, int rouge, int vert, int bleu |
| ImageColorResolve | 3.0.2 | 4.0.0 | int couleurID | int imageID, int rouge, int vert, int bleu |
| ImageColorSet | 3.0.0 | 4.0.0 | bool erreur | int imageID, int index, int rouge, int vert, int bleu |
| ImageColorsForIndex | 3.0.0 | 4.0.0 | array rouge, vert, bleu | int imageID, int couleurID |
| ImageColorsTotal | 3.0.0 | 4.0.0 | int nombre_couleurs_palette | int imageID |
| ImageColorTransparent | 3.0.0 | 4.0.0 | int couleurID | int imageID [, int couleurID] |
| ImageCopy | 3.0.6 | 4.0.0 | int erreur | int dst_im, int src_im, int dst_x, int dst_y, int src_x, int src_y, int src_w, int src_h |
| ImageCopyResized | 3.0.0 | 4.0.0 | int erreur | int imageID_cible, int imageID_source, int X_cible, int Y_cible, int X_source, int Y_source, int largeur_cible, int hauteur_cible, int largeur_source, int hauteur_source) |
| ImageCreate | 3.0.0 | 4.0.0 | int imageID | int largeur, int hauteur |
| ImageCreateFromGIF | 3.0.0 | 4.0.0 | int imageID | string fichier |
| ImageCreateFromJPEG | 3.0.16 | 4.0rc1 | int imageID | string fichier |
| ImageCreateFromPNG | 3.0.13 | 4.0b4 | int imageID | string fichier |
| ImageDashedLine | 3.0.0 | 4.0.0 | int erreur | int imageID, int x1, int y1, int x2, int y2, int couleurID |
| ImageDestroy | 3.0.0 | 4.0.0 | int erreur | int imageID |
| ImageFill | 3.0.0 | 4.0.0 | int erreur | int imageID, int x, int y, int couleurID |
| ImageFilledPolygon | 3.0.0 | 4.0.0 | int | int imageID, array points(x1,y1,x2,y2[,x3,y3]), int nombre_points, int couleurID |
| ImageFilledRectangle | 3.0.0 | 4.0.0 | int | int imageID, int x1, int y1, int x2, int y2, int couleurID |
| ImageFillToBorder | 3.0.0 | 4.0.0 | int | int imageID, int x, int y, int couleurID_bordure, int couleurID |
| ImageGammaCorrect | 3.0.13 | 4.0.0 | int erreur | int imageID, double gamma_entree, double gamma_sortie |
| ImageGIF | 3.0.0 | 4.0.0 | int | int imageID [, string fichier] |
| ImageInterlace | 3.0.0 | 4.0.0 | int | int imageID [, int interlace] |
| ImageJPEG | 3.0.16 | 4.0rc1 | int | int imageID [, string fichier [, int qualité]] |
| ImageLine | 3.0.0 | 4.0.0 | int | int imageID, int x1, int y1, int x2, int y2, int couleurID |
| ImagePNG | 3.0.13 | 4.0b4 | int | int imageID [, string fichier] |
| ImagePolygon | 3.0.0 | 4.0.0 | int erreur | int imageID, array points, int nombre_points, int couleurID |
| ImageRectangle | 3.0.0 | 4.0.0 | int erreur | int imageID, int x1, int y1, int x2, int y2, int couleurID |
| ImageSetPixel | 3.0.0 | 4.0.0 | int erreur | int imageID, int x, int y, int couleurID |
| ImageString | 3.0.0 | 4.0.0 | int erreur | int imageID, int fonteID, int x, int y, string texte, int couleurID |
| ImageStringUp | 3.0.0 | 4.0.0 | int erreur | int imageID, int fonteID, int x, int y, string texte, int couleurID |
| ImageSX | 3.0.0 | 4.0.0 | int largeur | int imageID |
| ImageSY | 3.0.0 | 4.0.0 | int hauteur | int imageID |
| ImageTTFBBox | 3.0.1 | 4.0.0 | array 4_coins_texte | int taille, int angle, string fichier_fonte, string texte |
| ImageTTFText | 3.0.0 | 4.0.0 | array 4_coins_texte | int imageID, int taille, int angle, int x, int y, int couleurID, string fichier_fonte, string texte |

| | | | | |
|----------------|--------|--------|--------|--|
| implode | 3.0.0 | 4.0.0 | string | string glue, array pieces |
| in_array | | 4.0.0 | bool | mixed needle, array haystack |
| intval | 3.0.0 | 4.0.0 | int | mixed var [, int base] |
| ip2long | | 4.0rc1 | int | string ip_address |
| iptcp | 3.0.6 | 4.0.0 | array | string iptcblock |
| is_array | 3.0.0 | 4.0.0 | int | mixed var |
| is_bool | | 4.0b4 | int | mixed var |
| is_dir | 3.0.0 | 4.0.0 | bool | string fichier |
| is_double | 3.0.0 | 4.0.0 | int | mixed var |
| is_executable | 3.0.0 | 4.0.0 | bool | string fichier |
| is_file | 3.0.0 | 4.0.0 | bool | string fichier |
| is_float | 3.0.0 | 4.0.0 | int | mixed var |
| is_int | 3.0.0 | 4.0.0 | int | mixed var |
| is_integer | 3.0.0 | 4.0.0 | int | mixed var |
| is_long | 3.0.0 | 4.0.0 | int | mixed var |
| is_numeric | | 4.0rc1 | int | mixed var |
| is_object | 3.0.0 | 4.0.0 | int | mixed var |
| is_readable | 3.0.0 | 4.0.0 | bool | string fichier |
| is_real | 3.0.0 | 4.0.0 | int | mixed var |
| is_resource | | 4.0b4 | int | mixed var |
| is_string | 3.0.0 | 4.0.0 | int | mixed var |
| is_subclass_of | | 4.0b4 | bool | object obj, string superclass |
| is_writable | 3.0.0 | 4.0.0 | bool | string fichier |
| isset | | | int | mixed var |
| JDDayOfWeek | 3.0.0 | 4.0.0 | mixed | int julianday, int mode |
| JDMonthName | 3.0.0 | 4.0.0 | string | int julianday, int mode |
| JDToFrench | 3.0.0 | 4.0.0 | string | int month, int day, int year |
| JDToGregorian | 3.0.0 | 4.0.0 | string | int julianday |
| JDToJewish | 3.0.0 | 4.0.0 | string | int julianday |
| JDToJulian | 3.0.0 | 4.0.0 | string | int julianday |
| jdtounix | | 4.0rc2 | int | int jday |
| join | 3.0.0 | 4.0.0 | string | éléments joints_espacés_par_séparateur string
séparateur, array éléments |
| JulianToJD | 3.0.0 | 4.0.0 | int | int month, int day, int year |
| key | 3.0.0 | 4.0.0 | mixed | array array |
| krsort | 3.0.13 | 4.0b4 | int | array array [, int sort_flags] |
| ksort | 3.0.0 | 4.0.0 | int | array array [, int sort_flags] |
| leak | 3.0.0 | 4.0.0 | void | int bytes |
| list | | | void | [int timestamp [, bool is_associative]] |
| log | 3.0.0 | 4.0.0 | float | float arg |
| log10 | 3.0.0 | 4.0.0 | float | float arg |
| long2ip | | 4.0rc1 | string | int proper_address |
| lstat | 3.0.4 | 4.0.0 | array | string fichier |
| ltrim | 3.0.0 | 4.0.0 | string | string str |
| mail | 3.0.0 | 4.0.0 | bool | string destinataire, string sujet, string message [,
string additional_headers] |
| max | 3.0.0 | 4.0.0 | mixed | mixed arg1, mixed arg2, mixed argn |
| md5 | 3.0.0 | 4.0.0 | string | string str |
| metaphone | | 4.0b4 | string | string str |
| method_exists | | 4.0b2 | bool | object object, string method_name |
| mkdir | 3.0.0 | 4.0.0 | int | string pathname, int mode |

| | | | | |
|-------------------------------|--------|-------|--------|--|
| mktime | 3.0.0 | 4.0.0 | int | int hour, int minute, int second, int month, int day, int year [, int is_dst]), int serveurID] |
| mt_getrandmax | 3.0.6 | 4.0.0 | int | void |
| mt_rand | 3.0.6 | 4.0.0 | int | [int min [, int max]] |
| mt_srand | 3.0.6 | 4.0.0 | void | int seed |
| <u>MySQL fonctions</u> | | | | |
| mysql_affected_rows | 3.0.0 | 4.0.0 | int | [int connexionID] |
| mysql_change_user | 3.0.13 | | int | string utilisateur, string mot_de_passe [, string base_données [, int connexionID]] |
| mysql_close | 3.0.0 | 4.0.0 | int | [int connexionID] |
| mysql_connect | 3.0.0 | 4.0.0 | int | int connexionID [string serveur_mysql] |
| mysql_create_db | 3.0.0 | 4.0.0 | int | string base_données [, int connexionID] |
| mysql_data_seek | 3.0.0 | 4.0.0 | int | int résultatID, int numéro_enreg |
| mysql_drop_db | 3.0.0 | 4.0.0 | int | string base_données [, int connexionID] |
| mysql_errno | 3.0.0 | 4.0.0 | int | [int connexionID] |
| mysql_error | 3.0.0 | 4.0.0 | string | [int connexionID] |
| mysql_fetch_array | 3.0.0 | 4.0.0 | array | int résultatID [, int résultatID_type] |
| mysql_fetch_field | 3.0.0 | 4.0.0 | object | int résultatID [, int field_offset] |
| mysql_fetch_lengths | 3.0.0 | 4.0.0 | array | int résultatID |
| mysql_fetch_object | 3.0.0 | 4.0.0 | object | int résultatID [, int résultatID_type] |
| mysql_field_flags | 3.0.0 | 4.0.0 | string | int résultatID, int field_offset |
| mysql_field_len | 3.0.0 | 4.0.0 | int | int résultatID, int field_offset |
| mysql_field_name | 3.0.0 | 4.0.0 | string | int résultatID, int field_index |
| mysql_field_seek | 3.0.0 | 4.0.0 | int | int résultatID, int field_offset |
| mysql_field_table | 3.0.0 | 4.0.0 | string | int résultatID, int field_offset |
| mysql_field_type | 3.0.0 | 4.0.0 | string | int résultatID, int field_offset |
| mysql_free_result | 3.0.0 | 4.0.0 | int | int résultatID |
| mysql_insert_id | 3.0.0 | 4.0.0 | int | [int connexionID] |
| mysql_list_dbs | 3.0.0 | 4.0.0 | int | [int connexionID] |
| mysql_list_fields | 3.0.0 | 4.0.0 | int | string base_données, string nom_table [, int connexionID] |
| mysql_list_tables | 3.0.0 | 4.0.0 | int | string base_données [, int connexionID] |
| mysql_num_fields | 3.0.0 | 4.0.0 | int | int résultatID |
| mysql_num_rows | 3.0.0 | 4.0.0 | int | int résultatID |
| mysql_pconnect | 3.0.0 | 4.0.0 | int | int connexionID [string serveur_mysql] |
| mysql_query | 3.0.0 | 4.0.0 | int | int résultatID string requête_SQL [, int connexionID] |
| mysql_result | 3.0.0 | 4.0.0 | int | int résultatID, int numero_enreg [, mixed field] |
| mysql_select_db | 3.0.0 | 4.0.0 | int | string base_données [, int connexionID] |
| mysql_tablename | 3.0.0 | 4.0.0 | string | int résultatID, int i |
| next | 3.0.0 | 4.0.0 | mixed | array array |
| nl2br | 3.0.0 | 4.0.0 | string | string string |
| number_format | 3.0.0 | 4.0.0 | string | float number, int decimals, string dec_point, string thousands_sep |
| ob_end_clean | | 4.0.0 | void | [int flag] |
| ob_start | | 4.0.0 | void | string octal_string |
| opendir | 3.0.0 | 4.0.0 | int | string path |
| openlog | 3.0.0 | 4.0.0 | int | string ident, int option, int facility |
| ord | 3.0.0 | 4.0.0 | int | string string |
| pack | 3.0.0 | 4.0.0 | string | string format [, mixed args ...] |
| parse_str | 3.0.0 | 4.0.0 | void | string str |
| parse_url | 3.0.0 | 4.0.0 | array | string url |

| | | | | |
|---------------------------|--------|----------|--------------|---|
| passthru | 3.0.0 | 4.0.0 | void | string command [, int return_var] |
| pclose | 3.0.0 | 4.0.0 | int | int fp |
| PDF fonctions | | | | |
| pdf_add_annotation | 3.0.12 | 4.0b2 | void | int pdfID, double llx, double lly, double urx, double ury, string titre, string contenu |
| pdf_add_outline | 3.0.6 | 4.0.0 | int signetID | int pdfID, string texte_signet [, int signetID_parent [, int signet_ouvert = 1]] |
| pdf_arc | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor, double radius, double début, double fin |
| pdf_begin_page | 3.0.6 | 4.0.0 | void | int pdfID, double largeur, double hauteur |
| pdf_circle | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor, double radius |
| pdf_clip | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_close_image | 3.0.7 | 4.0b2 | void | int image |
| pdf_close | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_closepath_fill_stroke | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_closepath_stroke | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_closepath | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_continue_text | 3.0.6 | 4.0.0 | void | int pdfID, string text |
| pdf_curveto | 3.0.6 | 4.0.0 | void | int pdfID, double x1, double y1, double x2, double y2, double x3, double y3 |
| pdf_end_page | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_endpath | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_execute_image | 3.0.7 | 4.0b2-b4 | void | int pdfID, int image, double x-coor, double y-coor, double scale |
| pdf_fill_stroke | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_fill | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_get_info | 3.0.6 | 4.0b1 | info | string fichier |
| pdf_get_parameter | | 4.0.1 | string | int pdfID, string name, double modifier |
| pdf_get_value | | 4.0.1 | double | int pdfID, string name, double modifier |
| pdf_lineto | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor |
| pdf_moveto | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor |
| pdf_open_gif | 3.0.7 | 4.0b2 | int | int pdfID, string fichier |
| pdf_open_jpeg | 3.0.7 | 4.0b2 | int | int pdfID, string fichier |
| pdf_open_memory_image | 3.0.10 | 4.0b2 | int | int pdfID, int image |
| pdf_open_png | | 4.0rc2 | int | int pdfID, string fichier_png |
| pdf_open | 3.0.6 | 4.0.0 | int pdfID | int fichier, int info |
| pdf_place_image | 3.0.7 | 4.0b2 | void | int pdfID, int image, double x-coor, double y-coor, double scale |
| pdf_put_image | 3.0.7 | 4.0b2-b4 | void | int pdfID, int image |
| pdf_rect | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor, double width, double height |
| pdf_restore | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_rotate | 3.0.6 | 4.0.0 | void | int pdfID, double angle |
| pdf_save | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_scale | 3.0.6 | 4.0.0 | void | int pdfID, double x-scale, double y-scale |
| pdf_set_border_color | 3.0.12 | 4.0b2 | void | int pdfID, double red, double green, double blue |
| pdf_set_border_dash | | 4.0.1 | void | int pdfID, double black, double white |
| pdf_set_border_style | 3.0.12 | 4.0b2 | void | int pdfID, string style, double width |
| pdf_set_char_spacing | 3.0.6 | 4.0.0 | void | int pdfID, double space |

| | | | | |
|------------------------|-------|--------|--------|--|
| pdf_set_duration | 3.0.6 | 4.0.0 | void | int pdfID, double duration |
| pdf_set_font | 3.0.6 | 4.0.0 | void | int pdfID, string nom_fonte, double taille, string encodage [, int embed] |
| pdf_set_horiz_scaling | 3.0.6 | 4.0.0 | void | int pdfID, double échelle |
| pdf_set_info_author | | | void | int pdfID, string fieldname, string valeur |
| pdf_set_leading | 3.0.6 | 4.0.0 | void | int pdfID, double distance |
| pdf_set_parameter | | 4.0rc1 | void | int pdfID, string name, string value |
| pdf_set_text_matrix | 3.0.6 | 4.0b4 | void | int pdfID, array matrix |
| pdf_set_text_pos | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor |
| pdf_set_text_rendering | 3.0.6 | 4.0.0 | void | int pdfID, int mode |
| pdf_set_text_rise | 3.0.6 | 4.0.0 | void | int pdfID, double rise |
| pdf_set_transition | 3.0.6 | 4.0.0 | void | int pdfID, int transition |
| pdf_set_value | | 4.0.1 | void | int pdfID, string name, double value |
| pdf_set_word_spacing | 3.0.6 | 4.0.0 | void | int pdfID, double space |
| pdf_setdash | 3.0.6 | 4.0.0 | void | int pdfID, double white, double black |
| pdf_setflat | 3.0.6 | 4.0.0 | void | int pdfID, double value |
| pdf_setgray_fill | 3.0.6 | 4.0.0 | void | int pdfID, double gray value |
| pdf_setgray_stroke | 3.0.6 | 4.0.0 | void | int pdfID, double gray value |
| pdf_setgray | 3.0.6 | 4.0.0 | void | int pdfID, double gray value |
| pdf_setlinecap | 3.0.6 | 4.0.0 | void | int pdfID, int value |
| pdf_setlinejoin | 3.0.6 | 4.0.0 | void | int pdfID, long value |
| pdf_setlinewidth | 3.0.6 | 4.0.0 | void | int pdfID, double width |
| pdf_setmiterlimit | 3.0.6 | 4.0.0 | void | int pdfID, double value |
| pdf_setrgbcolor_fill | 3.0.6 | 4.0.0 | void | int pdfID, double red value, double green value, double blue value |
| pdf_setrgbcolor_stroke | 3.0.6 | 4.0.0 | void | int pdfID, double red value, double green value, double blue value |
| pdf_setrgbcolor | 3.0.6 | 4.0.0 | void | int pdfID, double red value, double green value, double blue value |
| pdf_show_boxed | | 4.0rc1 | int | int pdfID, string text, double x-coor, double y-coor, double width, double height, string mode |
| pdf_show_xy | 3.0.6 | 4.0.0 | void | int pdfID, string text, double x-coor, double y-coor |
| pdf_show | 3.0.6 | 4.0.0 | void | int pdfID, string text |
| pdf_skew | | 4.0rc1 | void | int pdfID, double alpha, double beta |
| pdf_stringwidth | 3.0.6 | 4.0.0 | double | int pdfID, string text |
| pdf_stroke | 3.0.6 | 4.0.0 | void | int pdfID |
| pdf_translate | 3.0.6 | 4.0.0 | void | int pdfID, double x-coor, double y-coor |
| pfsckopen | 3.0.7 | 4.0.0 | int | string hostname, int port [, int errno [, string errstr [, int timeout]]] |
| php_logo_guid | | 4.0b4 | string | void |
| php_sapi_name | | 4.0.1 | string | void |
| phpcredits | | 4.0.0 | void | int flag |
| phpinfo | 3.0.0 | 4.0.0 | int | void |
| phpversion | 3.0.0 | 4.0.0 | string | void |
| pi | 3.0.0 | 4.0.0 | double | void |
| popen | 3.0.0 | 4.0.0 | int | string command, string mode |
| pos | 3.0.0 | 4.0.0 | mixed | array array |
| pow | 3.0.0 | 4.0.0 | float | float base, float exp |
| preg_grep | | 4.0.0 | array | string pattern, array input |

| | | | | |
|------------------------------|--------|--------|---------|---|
| preg_match_all | 3.0.9 | 4.0.0 | int | string pattern, string subject, array matches [, int order] |
| preg_match | 3.0.9 | 4.0.0 | int | string pattern, string subject [, array matches] |
| preg_quote | 3.0.9 | 4.0.0 | string | string str [, string delimiter] |
| preg_replace | 3.0.9 | 4.0.0 | mixed | mixed pattern, mixed replacement, mixed subject [, int limit] |
| preg_split | 3.0.9 | 4.0.0 | array | string pattern, string subject [, int limit [, int flags]] |
| prev | 3.0.0 | 4.0.0 | mixed | array array |
| print_r | | 4.0.0 | void | mixed expression |
| print | | | print | string format [, mixed args...] |
| putenv | 3.0.0 | 4.0.0 | void | string setting |
| quoted_printable_decode | | 3.0.6 | 4.0.0 | string string str |
| quotemeta | 3.0.0 | 4.0.0 | string | string str |
| rad2deg | 3.0.4 | 4.0.0 | double | double number |
| rand | 3.0.0 | 4.0.0 | int | [int min [, int max]] |
| range | 3.0.8 | 4.0b4 | array | int low, int high |
| rawurldecode | 3.0.0 | 4.0.0 | string | string str |
| rawurlencode | 3.0.0 | 4.0.0 | string | string str |
| read_exif_data | | 4.0.1 | array | string fichier |
| readdir | 3.0.0 | 4.0.0 | string | int dir_handle |
| readfile | 3.0.0 | 4.0.0 | int | string fichier [, int use_include_path] |
| readgzfile | 3.0.0 | 4.0.0 | int | string fichier [, int use_include_path] |
| readline_add_history | | 4.0b4 | void | string line |
| readline_clear_history | | 4.0b4 | boolean | void |
| readline_completion_function | | 4.0b4 | boolean | readline_completion_function (string line) |
| readline_info | | 4.0b4 | mixed | [string varname [, string newvalue]] |
| readline_list_history | | 4.0b4 | array | void |
| readline_read_history | | 4.0b4 | boolean | string fichier |
| readline_write_history | | | 4.0b4 | boolean string fichier |
| readline | | 4.0b4 | string | [string prompt] |
| realpath | | 4.0b4 | string | string path |
| recode_file | 3.0.13 | 4.0RC1 | bool | int input, int output |
| recode_string | 3.0.13 | 4.0RC1 | string | string request, string string |
| recode | | 4.0RC1 | string | string request, string string |
| register_shutdown_function | 3.0.4 | 4.0.0 | int | string func |
| rename | 3.0.0 | 4.0.0 | int | string oldname, string newname |
| reset | 3.0.0 | 4.0.0 | mixed | array array |
| rewind | 3.0.0 | 4.0.0 | int | int fp |
| rewinddir | 3.0.0 | 4.0.0 | void | int dir_handle |
| rmdir | 3.0.0 | 4.0.0 | int | string dirname |
| round | 3.0.0 | 4.0.0 | double | double val [, int precision] |
| rsort | 3.0.0 | 4.0.0 | void | array array [, int sort_flags] |
| rtrim | 3.0.0 | 4.0.0 | string | string str |
| serialize | 3.0.5 | 4.0.0 | string | mixed value |
| session_decode | | 4.0.0 | bool | string data |
| session_destroy | | 4.0.0 | bool | void); |
| session_id | | 4.0.0 | string | session_id ([string id] |
| session_is_registered | | 4.0.0 | bool | string name |
| session_module_name | | 4.0.0 | string | [string module] |
| session_name | | 4.0.0 | string | [string name] |

| | | | |
|---------------------------|-------|--------|--|
| session_register | 4.0.0 | bool | mixed name [, mixed ...] |
| session_save_path | 4.0.0 | string | [string path] |
| session_set_cookie_params | 4.0b4 | void | int lifetime [, string path [, string domain]] |
| session_start | 4.0.0 | bool | string name |
| session_unset | 4.0b4 | void | int fp, int buffer |
| set_magic_quotes_runtime | 3.0.6 | 4.0.0 | long int new_setting |
| socket_set_blocking | 4.0b4 | int | int socket descriptor, int mode |
| set_time_limit | 3.0.0 | 4.0.0 | void int seconds |
| setcookie | 3.0.0 | 4.0.0 | int string name [, string value [, int expire [, string path [, string domain [, int secure]]]]] |
| setlocale | 3.0.0 | 4.0.0 | string string category, string locale |
| settype | 3.0.0 | 4.0.0 | int string var, string type |
| show_source | | 4.0.0 | void string fichier |
| shuffle | 3.0.8 | 4.0b4 | void array array |
| similar_text | 3.0.7 | 4.0b2 | int string first, string second [, double percent] |
| sin | 3.0.0 | 4.0.0 | float float arg |
| sizeof | 3.0.0 | 4.0.0 | int array array |
| sleep | 3.0.0 | 4.0.0 | void int seconds |
| sort | 3.0.0 | 4.0.0 | void array array [, int sort_flags] |
| soundex | 3.0.0 | 4.0.0 | string string str |
| split | 3.0.0 | 4.0.0 | array string pattern, string string [, int limit] |
| spliti | | 4.0.1 | array string pattern, string string [, int limit] |
| sprintf | 3.0.0 | 4.0.0 | string string format [, mixed args...] |
| sql_regcase | 3.0.0 | 4.0.0 | string string string string |
| sqrt | 3.0.0 | 4.0.0 | float float arg |
| srand | 3.0.0 | 4.0.0 | void int seed |
| sscanf | | 4.0.1 | mixed string str, string format [, string var1...] |
| stat | 3.0.0 | 4.0.0 | array string fichier |
| str_pad | | 4.0.1 | string string input, int pad_length [, string pad_string [, int pad_type]] |
| str_repeat | | 4.0b4 | string string input, int multiplier |
| str_replace | 3.0.8 | 4.0.0 | string string sous_texte_a_chercher, string sous_texte_a_replacer, string texte_principal |
| strcasecmp | 3.0.2 | 4.0.0 | int string str1, string str2 |
| strchr | 3.0.0 | 4.0.0 | string string haystack, string needle |
| strcmp | 3.0.0 | 4.0.0 | int string str1, string str2 |
| strcspn | 3.0.3 | 4.0.0 | int string str1, string str2 |
| strftime | 3.0.0 | 4.0.0 | string string format [, int timestamp] |
| strip_tags | 3.0.8 | 4.0b2 | string string str [, string allowable_tags] |
| stripslashes | | 4.0b4 | string string str |
| stripslashes | 3.0.0 | 4.0.0 | string string str |
| stristr | 3.0.6 | 4.0.0 | string string haystack, string needle |
| strlen | 3.0.0 | 4.0.0 | int string str |
| strnatcasecmp | | 4.0rc2 | int string str1, string str2 |
| strnatcmp | | 4.0rc2 | int string str1, string str2 |
| strncmp | | 4.0b4 | int string str1, string str2, int len |
| strpos | 3.0.0 | 4.0.0 | int string haystack, string needle [, int offset] |
| strrchr | 3.0.0 | 4.0.0 | string string haystack, string needle |
| strrev | 3.0.0 | 4.0.0 | string string string string |
| strrpos | 3.0.0 | 4.0.0 | int string haystack, char needle |

| | | | | |
|-------------------------------|--------|----------|--------|--|
| strspn | 3.0.3 | 4.0.0 | int | string str1, string str2 |
| strstr | 3.0.0 | 4.0.0 | string | string haystack, string needle |
| strtok | 3.0.0 | 4.0.0 | string | string arg1, string arg2 |
| strtolower | 3.0.0 | 4.0.0 | string | string str |
| strtotime | 3.0.12 | 4.0b2 | int | string time [, int now] |
| strtoupper | 3.0.0 | 4.0.0 | string | string string |
| strtr | 3.0.0 | 4.0.0 | string | string source, string fromcharlist, string tocharlist |
| strval | 3.0.0 | 4.0.0 | string | mixed var |
| substr_count | | 4.0rc2 | int | string haystack, string needle |
| substr_replace | | 4.0b4 | string | string string, string replacement, int start [, int length] |
| substr | 3.0.0 | 4.0.0 | string | string string, int start [, int length] |
| syslog | 3.0.0 | 4.0.0 | int | int priority, string message |
| system | 3.0.0 | 4.0.0 | string | string command [, int return_var] |
| tan | 3.0.0 | 4.0.0 | float | float arg |
| tempnam | 3.0.0 | 4.0.0 | string | string dir, string prefix |
| textdomain | 3.0.7 | 4.0.0 | int | [string library] |
| time | 3.0.0 | 4.0.0 | int | void |
| touch | 3.0.0 | 4.0.0 | int | string fichier [, int |
| trim | 3.0.0 | 4.0.0 | string | trim (string str |
| uasort | 3.0.4 | 4.0.0 | void | array array, function cmp_function |
| ucfirst | 3.0.0 | 4.0.0 | string | string str |
| ucwords | 3.0.3 | 4.0.0 | string | string str |
| uksort | 3.0.4 | 4.0.0 | void | array array, function cmp_function |
| umask | 3.0.0 | 4.0.0 | int | int mask |
| uniqid | 3.0.0 | 4.0.0 | int | string prefix [, boolean lcg] |
| unixtojd | | 4.0rc2 | int | [int timestamp] |
| unlink | 3.0.0 | 4.0.0 | int | string fichier |
| unpack | 3.0.0 | 4.0.0 | array | string format, string data |
| unserialize | 3.0.5 | 4.0.0 | mixed | string str |
| unset | | | int | mixed var |
| urldecode | 3.0.0 | 4.0.0 | string | string str |
| urlencode | 3.0.0 | 4.0.0 | string | string str |
| usort | 3.0.3 | 4.0.0 | void | array array, string cmp_function |
| utf8_decode | 3.0.6 | 4.0.0 | string | string data |
| utf8_encode | 3.0.6 | 4.0.0 | string | string data |
| var_dump | 3.0.5 | 4.0.0 | void | mixed expression |
| virtual | 3.0.0 | 4.0.0 | int | string fichier |
| vm_addalias | 3.0.0 | 4.0.0 | int | string vdomain, string basepwd, string username, string alias |
| vm_adduser | 3.0.0 | 4.0.0 | int | string vdomain, string basepwd, string newusername, string newuserpassword |
| vm_delalias | 3.0.0 | 4.0.0 | int | string vdomain, string basepwd, string alias |
| vm_deluser | 3.0.0 | 4.0.0 | int | string vdomain, string username |
| vm_passwd | 3.0.0 | 4.0.0 | int | string vdomain, string username, string password, string newpassword |
| wordwrap | | 4.0.0cvs | | string string str [, int width [, string break]] |
| xml_error_string | 3.0.6 | 4.0.0 | string | int code |
| xml_get_current_byte_index | 3.0.6 | 4.0.0 | | int int parser |
| xml_get_current_column_number | | 3.0.6 | | 4.0.0 int int parser |
| xml_get_current_line_number | 3.0.6 | 4.0.0 | | int int parser |

| | | | | |
|--|-------|--------|--------|--|
| xml_get_error_code | 3.0.6 | 4.0.0 | int | int parser |
| xml_parse_into_struct | 3.0.8 | 4.0.0 | int | int parser, string data, array values, array index |
| xml_parse | 3.0.6 | 4.0.0 | int | int parser, string data [, int isFinal] |
| xml_parser_create | 3.0.6 | 4.0.0 | int | [string encoding] |
| xml_parser_free | 3.0.6 | 4.0.0 | string | int parser |
| xml_parser_get_option | 3.0.6 | 4.0.0 | mixed | int parser, int option |
| xml_parser_set_option | 3.0.6 | 4.0.0 | int | int parser, int option, mixed value |
| xml_set_character_data_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xml_set_default_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xml_set_element_handler | 3.0.6 | 4.0.0 | int | int parser, string startElementHandler, string endElementHandler |
| xml_set_external_entity_ref_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xml_set_notation_decl_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xml_set_object | 4.0b4 | void | int | int parser, object object |
| xml_set_processing_instruction_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xml_set_unparsed_entity_decl_handler | 3.0.6 | 4.0.0 | int | int parser, string handler |
| xmlDoc | 4.0b4 | object | string | str |
| xmlDocfile | 4.0b4 | object | string | fichier |
| xmltree | 4.0b4 | object | string | str |
| zend_logo_guid | 4.0b4 | string | void | |

Transférer les tables MySQL vers le serveur web

Grâce à phpMyAdmin, on peut sauvegarder une base de données MySQL locale (obligatoirement entière), et la charger ensuite chez l'hébergeur.

En détail, cela fonctionne comme suit :

1. Sur votre ordinateur, lancez phpMyAdmin et choisissez la base de données php_pas_a_pas. Lancez la sauvegarde de la structure et des données dans un fichier local en avec l'extension .sql.

Afficher le schéma de la base

Structure seule

Structure et données

Ajouter des énoncés 'drop table'

transmettre ← **pour générer dans un fichier**

Exécuter

2. Connectez-vous sur phpMyAdmin chez l'hébergeur. Sélectionnez la base php_pas_a_pas, et lancez phpMyAdmin, et on lance le chargement en sélectionnant le fichier local.

Exécuter une ou des requêtes sur la base PHP

ou Emplacement du fichier texte:

Exécuter

Si phpMyAdmin n'est pas installé sur ce serveur, chargez-le vous-mêmes dans votre propre site. phpMyAdmin version 2.1.0 occupe environ 250 KB.

Le fichier .sql est un fichier texte composé d'une suite d'instructions MySQL telles DROP TABLE, CREATE TABLE et INSERT.

```
# Structure de la table 'avis'
#
```

```
DROP TABLE IF EXISTS avis;
CREATE TABLE avis (
  num_livre int(11) DEFAULT '0' NOT NULL,
  num_avis int(11) NOT NULL auto_increment,
  avis text NOT NULL,
  date datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
  PRIMARY KEY (num_livre, num_avis)
);
```

```
#
# Contenu de la table 'avis'
#
```

```
INSERT INTO avis VALUES ( '1', '4', 'je suis très bavarde', '2001-04-09 19:24:25');
```

Pour afficher le schéma, deux options existent : l'une permet d'omettre la génération des instructions DROP TABLE, et l'autre permet d'omettre la génération des instructions INSERT. A noter que le résultat peut être affiché à l'écran au lieu de créer un fichier (utile pour tester le résultat avant de créer le fichier).

L'inconvénient majeur des schémas dans phpMyAdmin est que l'on doit régénérer systématiquement toute la base de données, même si on n'a ajouté qu'un simple champ dans une table!

1. Lorsque la base de données est grosse, par exemple un nombre important de tables, ou une grande quantité de données, le temps d'exécution du fichier .sql chez l'hébergeur peut être long.
2. Si le site est déjà en fonction et si les internautes peuvent enregistrer des messages dans une table que vous avez modifiée localement, on ne peut pas actualiser cette table par le fichier .sql. Dans un tel cas, il faut mettre à jour la table directement chez l'hébergeur (toujours en utilisant phpMyAdmin). Cela présente un risque car vous devez répéter à l'identique vos mises à jour locales, sans erreur possible (il ne faut pas perdre les messages des internautes !).

SOMMAIRE

| | |
|---|-----------|
| POURQUOI PHP?..... | 1 |
| POURQUOI MYSQL ? | 2 |
| PRINCIPE DE FONCTIONNEMENT DE PHP/MYSQL2 | |
| EXECUTER DU PHP | 4 |
| CONFIGURATION SERVEUR WEB | |
| PERSONNEL | 5 |
| <i>Installer les logiciels</i> | 5 |
| <i>Installer easyPHP</i> | 5 |
| <i>Démarrer et arrêter Apache et MySQL</i> | 6 |
| <i>Vérifier la conformité de l'installation</i> | 8 |
| Vérifier l'installation de MySQL..... | 8 |
| INSTALLATION SUR MACINTOSH ET SUR UNIX ... | 9 |
| PHP PAS A PAS..... | 10 |
| PREMIER PROGRAMME | 10 |
| <i>Ce qu'il faut savoir avant de commencer un programme</i> | 10 |
| <i>Script PHP</i> | 10 |
| <i>Début et fin d'instructions PHP</i> | 11 |
| Analyse du script..... | 12 |
| LES VARIABLES, LES CONSTANTES, LES FONCTIONS, LES TABLEAUX | 13 |
| <i>Les commentaires</i> | 13 |
| <i>Les variables</i> | 13 |
| Portées des variables | 14 |
| <i>Constantes</i> | 14 |
| <i>Littéraux</i> | 15 |
| <i>Les tableaux</i> | 15 |
| Créer un tableau : | 15 |
| Modifier un tableau:..... | 16 |
| <i>Les nombres</i> | 16 |
| <i>Les opérateurs</i> | 16 |
| <i>L'opérateur de concaténation</i> | 17 |
| <i>Les caractères spéciaux</i> | 18 |
| <i>Les fonctions</i> | 19 |
| <i>Bloc d'instructions</i> | 19 |
| <i>Les structures de contrôles</i> | 20 |
| If..... | 20 |
| If else..... | 21 |
| <i>Portées des variables dans les fonctions</i> | 21 |
| CREER UN COOKIE..... | 22 |
| RECUPERER LES ZONES D'UN FORMULAIRE HTML DANS PHP | 26 |
| MYSQL | 28 |
| <i>Qu'est-ce qu'une base de données</i> | 28 |
| CREATION D'UNE BASE DE DONNEES AVEC PHPMYADMIN | 28 |
| <i>La base de données</i> | 28 |
| <i>Création des champs de la table</i> | 29 |
| <i>Un problème connu de phpMyAdmin</i> | 32 |
| CREER UNE TABLE AVEC UN SCRIPT PHP | 32 |
| <i>Connexion à la base de données MySQL</i> | 32 |
| <i>Création d'une nouvelle table</i> | 34 |
| <i>Formulaire HTML : récupérer des données</i> 35 | |
| LIRE LE CONTENU D'UNE TABLE | 36 |
| <i>Lire un enregistrement dans la table MySQL</i> 37 | |
| <i>Insérer un script pour chaque champ à afficher</i> | 37 |
| TRIER LES ENREGISTREMENTS..... | 37 |
| <i>Répéter les lignes d'un tableau HTML grâce à PHP</i> | 38 |
| <i>Fichiers et répertoires</i> | 1 |
| Ouvrir et lire un fichier | 39 |
| fopen()..... | 39 |
| fgets() | 39 |
| Ecrire dans un fichier | 40 |
| Créer un fichier | 40 |
| Lire un répertoire | 40 |
| <i>La date et l'heure</i> | 42 |
| Opérations sur les dates..... | 42 |
| <i>Les fichiers PDF</i> | 45 |
| ERREURS FREQUENTES..... | 46 |
| <i>Erreurs fréquentes dans c:/Program Files/Apache Group/Apache/logs/error.log</i> 46 | |
| <i>Exécution de programme php</i> | 47 |
| HEBERGEURS PROPOSANT PHP..... | 48 |
| CHOISIR L'HEBERGEUR..... | 48 |
| <i>Sites internet parlant du PHP</i> | 53 |
| <i>Autres sites</i> | 53 |
| <i>Sites internet contenant les applications nécessaires pour programmer en PHP (à télécharger)</i> | 53 |
| <i>Liste de fonctions</i> :..... | 54 |
| TRANSFERER LES TABLES MYSQL VERS LE SERVEUR WEB | 68 |