Initiation à la création de sites Web dynamiques

- rappels sur le langage HTML
- découverte du langage PHP
- bases de données MySQL en local et en ligne

C.D.D.P. Arras 16 janvier et 6 mars 2002

© David Ducrocq C.N.T.E Liévin Hénin Beaumont (janvier 2002)

Sommaire

Introduction. page 3

1. Langage HTML, rappels pages 3 à 5

   1.1. Structure de base page 3

   1.2. Les différentes balises page 3

   1.3. Rôles du serveur et du navigateur pages 4 et 5

2. Le langage PHP . pages 5 à 13

   2.1. Origines page 5

   2.2. Spécificités par rapport aux autres langages dynamiques pages 5 et 6

   2.3. Installation des différents modules pages 5 à 9

   2.4. Premiers scripts en PHP pages 9 à 12

   2.5. Faire "héberger" vos scripts pages 12 et 13

3. PHP et bases de données pages 13 à 18

   3.1. Installation et utilisation de Mysql et Phpmyadmin (en local) pages 13 à 17

   3.2. Au travail ! page 18

   3.3. Travail en ligne : le cas de Free page 18

4. Bibliographie, ressources en ligne pages 19 et 20

Conclusion page 20

Annexes (quelques instructions et fonctions du langage PHP, scripts détaillés PHP + MySQL, exercices)

NB : Les marques et noms commerciaux figurant sur ce document appartiennent à leur(s) propriétaire(s) respectif(s).

© David Ducrocq C.N.T.E Liévin Hénin Beaumont (janvier 2002)

Introduction :

Un site sur l'Internet est une entreprise coûteuse en terme de temps. Les mises à jour régulières sont parfois difficiles à réaliser. Or, un site qui n'évolue pas est condamné à disparaître ou du moins à voir sa fréquentation diminuer. L'internaute aime le changement, il ne supporte pas longtemps un site statique qui n'évolue guère. Les langages dynamiques permettent de résoudre ces problèmes. Ils facilitent les opérations de mises à jour, permettent plus d'interactivité sur les pages…

Le but de ces pages est de vous initier à l'un des langages les plus en vogue actuellement : PHP.

1. Langage HTML, rappels

Avant de nous plonger dans les méandres des langages dynamiques, nous commencerons par un bref rappel sur la syntaxe HTML.

1.1. Structure de base

Même lorsque vous n'y avez rien inséré, une page HTML comporte néanmoins quelques éléments. Il s'agit des balises qui représentent la structure de base. Ces balises vont par paires, ainsi le <html> qui inaugure le code est associé à la balise </html> qui termine la page. Ce document fera très souvent référence à cette syntaxe.

```
<html>
…etc.
<head>

…etc.

</head>
<body>

…etc
<body>
</html>
```

1.2. Les différentes balises

Il existe de nombreuses balises en HTML (Cf. quelques exemples dans le tableau suivant) :

ma phrase	donne	*ma phrase* (en italique)
ma phrase	donne	**ma phrase** (en caractères gras)
<u> ma phrase </u>	donne	ma phrase (en caractères soulignés)
Entrez	donne	Entrez (lien vers la page som.htm)

1.3. Rôles du serveur et du navigateur

L'affichage des pages sur l'Internet repose sur la notion de client-serveur
Un système client-serveur fonctionne selon le schéma suivant (architecture à 2 niveaux) :

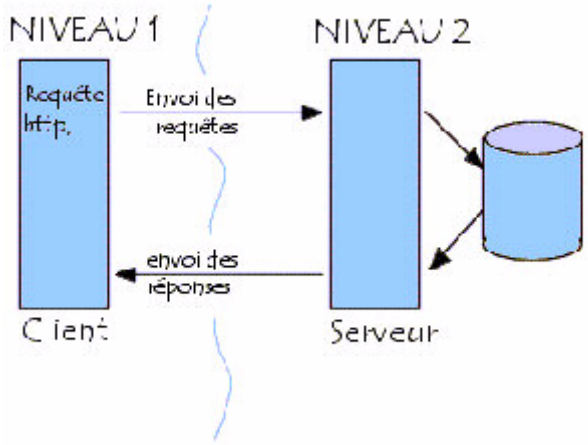


schéma : © Copyright 2000 Jean-François Pillou. Document issu de CommentCaMarche.net (soumis à la licence GNU FDL)

- Le client (navigateur sur votre ordinateur) émet une **requête** vers le serveur grâce à son adresse et le port, qui désigne un service particulier du serveur
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.
- Le navigateur, sur la machine cliente, interprète le code renvoyé par le serveur et l'affiche.

Ainsi le script html suivant (renvoyé par le serveur lorsque votre navigateur a appelé http://www.monsite/index.htm) :

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title>Traditions, anecdotes et légendes au nord de la France</title>
<meta name="Description"
content="Partez à la découverte des légendes, traditions, anecdotes
de la région Nord Pas de Calais. Découvrez ses personnages
(imaginaires, célèbres ou inconnus). Visitez des lieux méconnus.">
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<meta name="keywords"
content="traditions, anecdotes, légendes, nord pas de calais,
personnages célèbres, vidocq, faber, marie groëtte, recettes
locales">
</head>
<body bgcolor="#FFFFFF">
<p align="center"><font color="#FF0000" size="7"
face="MS Sans Serif">T</font><font size="7" face="MS Sans
Serif">raditions,
anecdotes <br>
et légendes au nord de la France</font></p>
<p align="center"> </p>
<p align="center"><a href="som.htm"><font
size="5">Entrez</font></a><font
size="5"> <img src="images/arbre2.jpg" width="209"
height="150"></font></p>
<p> </p>
<p align="center">Copyright - David Ducrocq - mai 2001</p>
</body>
</html>
```

donnera sur votre navigateur

2. Le langage PHP

Découvrons maintenant, l'objet de notre étude.

2.1. Origines

Le langage PHP a vu le jour en 1994, sur le site de Rasmus Lerdof. Ce dernier avait créé une page personnelle avec son curriculum vitæ, il souhaitait conserver une trace du passage de ses visiteurs et pour ce faire, il inventa quelques scripts. Très vite, les internautes qui découvrent le site, désirent obtenir un exemplaire du langage. C'est le début du succès pour Rasmus qui décide que son langage sera mis gratuitement à disposition des utilisateurs (du moins dans le cadre de la GNU licence issue du monde Linux). Une communauté de développeurs a contribué à améliorer le langage (actuellement version 4).

Pour la petite histoire, sachez qu'à l'origine PHP signifiait "Personal Home Pages", étant donné la puissance de ce langage les programmeurs ont opté pour "PHP : Hypertext PréProcessor".

2.2. Spécificités par rapport aux autres langages dynamiques

Il existe de nombreux langages permettant de dynamiser vos pages Web. Il y a : le Javascript, le langage ASP de Microsoft, JSP (Java Server pages), le langage Perl, les scripts CGI (Common Gateway Interface)…etc.

Mon propos ne vise pas à égratigner tel ou tel langage. Il serait stérile d'aller dans cette voie. Chaque langage possède ses fans et ses détracteurs. Néanmoins, le langage PHP réunit plusieurs avantages qui le rendent particulièrement intéressant pour un débutant ou dans un contexte scolaire.

Détaillons ces avantages :

La rapidité

Les spécialistes s'accordent à dire que ce langage génère des scripts particulièrement rapides en terme de temps de réponse du serveur. Cela vient du fait que le PHP peut être intégré au noyau Linux. Sur le Net, la plupart des serveurs fonctionnent sur une plate-forme Linux avec serveur Apache.

La sécurité

Avec le PHP, vos scripts demeurent confidentiels. Le serveur ne fait que renvoyer du code HTML au navigateur de l'internaute qui visite votre site. Ainsi, votre site ne peut être "aspiré" par les utilitaires comme Memoweb ou Webcopier. Le javascript, par contre, est visible dans la source de la page consultée.

Une compatibilité maximale

Comme je l'ai déjà dit, le serveur renvoie uniquement au navigateur du code HTML. Cela assure une compatibilité optimale avec la majorité des logiciels de navigation. Cette compatibilité est aussi particulièrement intéressante en ce qui concerne les systèmes d'exploitation. Il existe des distributions PHP pour Linux, Windows, Windows NT, Macintosh, FreeBSD, Solaris et même Amiga. Enfin, pour conclure sur ce point, le langage PHP est compatible avec de nombreuses bases de données. Un script PHP peut interroger une base Mysql, Access, Oracle, Dbase, PostgreSQL, Filepro…

La gratuité

Les scripts PHP sont majoritairement développés dans le cadre de la GNU licence. Cela est particulièrement intéressant dans un cadre scolaire. Les différents logiciels utilisés relèvent pratiquement tous, du domaine libre (Apache, distributions et librairies PHP, base de données Mysql, éditeurs spécialisés…)

L'exhaustivité

Le langage possède beaucoup plus de fonctions que son concurrent direct l'ASP de Microsoft. De plus le code source étant libre, tous les ajouts et extensions sont à la portée des développeurs. Ainsi, les bugs sont très vite corrigés.

La simplicité

Le PHP est un langage relativement simple à apprendre. On peut rapidement développer des sites fonctionnels. Les personnes qui ont quelques notions de programmation devraient rapidement s'adapter à ce langage. La syntaxe est proche de celle du langage C. Le PHP ne génère pas de phénomène de rejet comme avec Perl.

Une documentation accessible en français

Contrairement à certains autres langages, on trouve de nombreuses ressources liées au PHP sur le Web. Il existe une communauté de développeurs compétents et, en général, disponibles (Cf. paragraphe 4 "Bibliographie et ressources en ligne").

Le langage PHP est en vogue actuellement sur le Net. Les spécialistes observent une migration de nombreux sites de l'ASP (Microsoft) vers le PHP. Cependant, il faut savoir que les sites professionnels les plus exigeants nécessitent des langages plus sophistiqués (mais aussi plus difficiles à maîtriser) comme Java.

2.3. Installation des différents modules

Avant de pouvoir développer en PHP, il faut d'abord installer sur votre machine quelques applications ou modules. Ainsi, il va falloir émuler en local un serveur Web (nous utiliserons pour ce faire Apache). Nous n'aborderons ici que l'installation sous Windows. Les personnes souhaitant obtenir des

renseignements pour les installations sur d'autres systèmes, sont invitées à se référer à la documentation en ligne (paragraphe 4)

2.3.1. Jouons la simplicité : EasyPHP

La "distribution" EasyPHP est la plus simple à utiliser pour un débutant. Elle est disponible uniquement pour l'environnement Windows 95 ou supérieur. Il s'agit d'une "distribution" française.

Récupérer le fichier d'installation (version 1.5) sur le site http://www.easyphp.org Attention, le fichier "pèse" 9235 ko, comptez ½ heure de connexion (au maximum) avec un modem.

Il s'agit d'un exécutable (*.exe) qui s'installe comme n'importe quel programme. Il suffit donc de suivre la procédure d'installation.

Vous venez d'installer un serveur Apache, le module PHP (plusieurs versions sont installées 2 à 4), un serveur de bases de données MySQL et l'utilitaire phpMyAdmin (nous reviendrons sur ces deux dernières "applications" plus loin, dans le présent document).

1 icône est désormais visible sur la barre des tâches (à gauche de l'horloge)

→ Utilisation de "EasyPHP"

Pour cela, faites un clic droit sur l'icône et optez pour "Démarrer/Arrêter". Cela a pour effet d'activer le serveur Apache (ainsi que le serveur de base de données MySQL Cf. plus loin)

L'arrêt des serveurs se fait de la même façon. En choisissant "Quitter", vous fermez le gestionnaire, l'icône disparaît de la barre des tâches. Pour la retrouver : Démarrer → Programmes → EasyPHP.

→ Test du serveur Apache :

Démarrez votre navigateur, refusez la proposition de connexion (ou optez pour travail hors ligne). Entrez l'adresse suivante dans la zone adéquate http://localhost ou http://127.0.0.1 Si vous obtenez la page d'accueil du répertoire www, cela signifie que le serveur Apache et PHP fonctionnent correctement. Vous pouvez aussi cliquer droit sur l'icône et opter pour "Web local".

→ Où placer mes scripts ?

Le répertoire de travail de EasyPHP est intitulé www, il se trouve en principe dans Program Files/EasyPHP (ou dans le répertoire d'installation choisi).

Dans ce répertoire www, se trouve un répertoire projet1 (que vous pouvez renommer). Il vous faudra créer un sous-répertoire pour chacun de vos projets, afin d'obtenir un espace de travail clair et cohérent.

Vos scripts seront accessibles à http://localhost/sous-répertoire/script.php3

2.3.2. Pour aller plus loin et comprendre : l'installation manuelle

Cette partie s'adresse aux personnes qui souhaitent comprendre plus finement les différentes étapes de l'installation. Nous allons voir comment on installe "manuellement" Apache et PHP.

→ Téléchargement des fichiers nécessaires :

	Sites Web	Fichiers à récupérer
Serveur Apache	http://httpd.apache.org	apachexxxwin32.exe (dossier binaries/win32)
Apache manager 2.0.2	http://brian.threadnet.com/apmgr.html	apmgrxxx.zip
Distribution PHP	http://www.php.net	php-3xxx-win32.zip (ou version ultérieure)

Remarques. : De préférence, téléchargez toujours les dernières versions. Ces dernières sortent régulièrement. Evitez cependant les versions bêta, elles manquent parfois de "stabilité" à l'utilisation et sont réservées aux spécialistes, aux développeurs ou aux responsables des serveurs qui ont une utilisation "pointue" des logiciels. Pour les besoins de cette formation, je propose le téléchargement de la version 3 de PHP. Une version 4 existe, mais la plupart des prestataires d'hébergement ne l'offrent pas. La version 3, par contre, est disponible chez tous les prestataires d'espace.

→ Installation d'Apache

L'installation automatisée ne pose aucun problème particulier. Il vaut mieux installer à la racine de votre disque dur (exemple : c:/apache) pour éviter les longs déplacements au sein de vos répertoires.

Editez ensuite (avec le bloc-notes de Windows par exemple) le fichier *httpd.conf* situé dans le sous-répertoire \conf d'Apache et ajoutez les éléments suivants :

Spécifiez l'adresse IP de votre serveur
A la ligne **#ServerName new.host.name**, retirez le # et remplacez new.host.name par l'adresse IP choisie. Je vous conseille **127.0.0.1** ou **localhost** pour une installation en local (test ou intranet)

Indiquez au serveur où se situe votre répertoire de travail
Par défaut, il s'agit du sous-répertoire htdocs d'Apache. Vous pouvez opter pour un autre répertoire (exemple : c:\site). Dans ce cas, vous devez opérer les changements suivants :
A la ligne **DocumentRoot "C:/apache/htdocs"** remplacez **"C:/apache/htdocs"** par **"C:/site"**
et un peu plus bas dans le fichier
<Directory "**C:/site**">
Options **All**
Allowoverride **All**
Order allow,deny
Allow from all
</Directory>

Indiquez au serveur où il pourra trouver le module php.exe
Juste en dessous de la ligne **ScriptAlias /cgi-bin/"C:/Apache/cgi-bin/"** , écrivez
ScriptAlias /php3/ "c:/php3/" Je vous conseille de prévoir l'installation de PHP dans ce répertoire. Si vous décidez d'en changer, n'oubliez pas de modifier la ligne évoquée ci-dessus.

Indiquez au serveur les extensions recevables pour vos scripts PHP
Juste en-dessous de la ligne **AddType application/x-tar .tgz** , écrivez les lignes suivantes
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3 .php
AddType application/x-httpd-php3 .phtml

Faites le lien entre l'alias et les extensions
En dessous de la ligne **# Format: Action handler-name /cgi-script/location** , écrivez
Action application/x-httpd-php3 "/php3/php.exe"

Lorsque Apache rencontrera un fichier de type application/x-httpd-php, il demandera à l'analyseur c:/php3/php.exe de l'interpréter et de renvoyer la réponse en html au navigateur.

→ Test d'Apache

Pour tester notre serveur, il faut d'abord lancer Apache. Pour cela, nous allons décompresser l'utilitaire Apache Manager dans le répertoire d'installation d'Apache. Il suffit alors de double cliquer sur

apmgr.exe pour faire apparaître une icône à gauche de l'horloge dans la barre des tâches. Un clic droit sur cette icône dévoilera un menu permettant de démarrer le serveur, de le stopper, de le redémarrer (en cas de problème) et de quitter le manager. Démarrons donc le serveur Apache. Il suffit ensuite de lancer votre navigateur, de refuser la connexion à l'Internet (ou d'opter pour un travail hors ligne) et d'entrer l'adresse suivante dans la zone adéquate http://localhost ou http://127.0.0.1

Votre navigateur va vous proposer une autre connexion qu'il faut accepter (cette connexion va se faire sur votre serveur en local). Si Apache fonctionne correctement, vous devez voir apparaître une page d'accueil commençant par " Ça fonctionne ! Le serveur web Apache est installé sur ce site ! ".

→ Installation de PHP3

Elle n'est pas beaucoup plus compliquée. Décompressez le fichier récupéré dans le répertoire c:/php3 (pour rester dans la logique évoquée plus haut). Copiez ensuite le fichier php3.ini-dist dans le répertoire Windows et renommez-le php3.ini. Pour ce faire, il est nécessaire de travailler sous le Dos ou d'ouvrir ce fichier avec le bloc notes de Windows et de l'enregistrer sous le nouveau nom (en optant pour tous dans la zone de type à l'enregistrement).

Editez ensuite ce fichier renommé et copiez-le dans le répertoire Windows, modifiez les lignes suivantes :

A la ligne **extension_dir** = indiquez **extension_dir = c:_php3**\

A la ligne **;extension=php3_mysql.dll** enlevez le point virgule en début de ligne pour que la DLL soit chargée à l'exécution.

Le module PHP est désormais prêt à fonctionner, nous allons pouvoir le vérifier lorsque nous aurons créé et sauvegardé nos premiers scripts dans le répertoire dédié (sous-répertoire htdocs d'Apache, répertoire site ou autre répertoire choisi à l'installation).

2.4. Premiers scripts en PHP

Un simple éditeur de texte (Cf. bloc-notes de Windows) suffit à développer un script en PHP. Néanmoins, il existe des éditeurs spécialisés (certains sont gratuits). C'est le cas de "Context" et de "Phpcoder". Ils offrent de nombreuses options : coloration syntaxique, débuggeur, visualisation immédiate… Vous trouverez ces logiciels en téléchargement sur http://www.phpinfo.net (rubrique : Editeurs). Je vous conseille vivement d'utiliser ce type de logiciels, ils faciliteront vos développements.

2.4.1. Syntaxe de base

Le PHP est un langage de script, comme le HTML et le javascript. Il s'insère directement entre des balises. Vous pouvez mélanger du PHP et du HTML sans problème.

```
<body>
<?
echo "Bonjour le monde !";
?>
</body>
```

Remarque importante : une instruction PHP se termine toujours (sauf cas très rares) par un point virgule. Si vous l'oubliez, votre navigateur vous enverra des messages d'erreurs (parse error)

A la place des balises <? et ?> peuvent être remplacées par <?php et ?>, ou <SCRIPT LANGAGE="php"> et </SCRIPT>. On peut aussi utiliser les balises à la mode ASP de Microsoft <% et %>
N.B. : Certains prestataires d'hébergement ne supportent pas toutes les balises évoquées ci-dessus. Dans le doute, choisissez <?php et ?> qui sont acceptées par tous les serveurs et qui identifient clairement le langage utilisé.

→ Commentez vos scripts

Il est fortement conseillé de commenter vos scripts, afin d'en faciliter la compréhension. Ces commentaires vous seront très utiles, si vous décidez de reprendre un script sur lequel vous n'avez pas travaillé depuis un moment. Ces petites indications seront ignorées par le serveur et le navigateur ne les affichera pas, vos "secrets" seront bien gardés !

```
<?
// Vous placez ici votre commentaire.
?>
```

A la place des // , on peut utiliser # . Ces signes sont adaptés à un commentaire sur une seule ligne. Si vous souhaitez apporter des commentaires plus longs, utilisez la syntaxe suivante :

```
<?php
instructions PHP

/* votre commentaire
*/

suite des instructions PHP

?>
```

→ L'instruction echo

exemple :

```
<html>
<head>
</head>
<body>
<?php
echo "Bonjour le monde !";
?>
</body>
</html>
```

Elle indique au navigateur qu'il faut afficher ce qui est placé entre les guillemets. Ainsi, l'exemple ci-dessus affichera :

Bonjour le monde !

L'instruction peut être remplacée par print qui donne les mêmes résultats.

Remarquez que dans la source de la page, toutes les traces du PHP ont disparu. Le serveur nous a envoyé uniquement du HTML interprété par le navigateur. Vos scripts demeurent confidentiels qu'ils soient simples ou plus complexes. En terme de sécurité, cela représente un net avantage. L'internaute ne peut découvrir les "secrets" qui se cachent dans vos pages.

© David Ducrocq C.N.T.E Liévin Hénin Beaumont (janvier 2002)

```
<html>
<head>
</head>
<body>
Bonjour le monde !
</body>
</html>
```

A l'aide de la balise echo, vous pouvez écrire du code HTML

exemple :
```
<?php
echo "<p align='center'> Bonjour le monde !</p>";
?>
```

affichera

Bonjour le monde !

N.B. : Vous aurez sans doute remarqué que l'on a utilisé les apostrophes au lieu des guillemets pour "encadrer" la balise 'center'. Si nous avions placé des guillemets, nous aurions obtenu un message d'erreur de la part du navigateur (parse error). En effet, on ne peut placer d'autres guillemets à l'intérieur de celui qui débute et de celui qui termine l'instruction echo. Il faut les remplacer par des apostrophes. Si vous souhaitez utiliser un guillemet dans une fonction echo, vous devez le faire précéder du signe \

exemple :
```
<?php
echo "La petite fille dit :\"Bonjour le monde !\"";
?>
```

affichera

La petite fille dit : "Bonjour le monde !"

→ Visualiser vos scripts :

En local, vos scripts sont uniquement accessibles en entrant l'adresse suivante dans votre navigateur http://localhost/sous-repertoire/script.php3 ou http://127.0.0.1/sous-repertoire/script.php3 (si vous décidez de placer vos scripts dans un sous-répertoire, ce qui n'est nullement obligatoire, mais contribue à plus de clarté dans votre espace de travail).

Vous ne devez surtout pas vous rendre dans le répertoire où se situent vos scripts et double cliquer sur vos fichiers. Si vous le faites, vous obtiendrez des messages d'erreur. Les scripts ne sont "visibles" qu'à travers votre serveur, il faut donc forcer votre navigateur à l'appeler.

Réalisons un petit test, écrivez le script suivant et placez-le dans votre répertoire de travail. Vous le nommerez index.php3

```
<?php
echo "Apache et PHP sont correctement installés sur cette machine !";
?>
```

Lancez ensuite votre navigateur, refusez la première connexion et entrez l'adresse suivante : http://localhost/index.php3 ou http://127.0.0.1/index.php3

2.4.2. Aller plus loin

Vous trouverez le détail des instructions et fonctions principales du langage PHP dans le document annexe. Vous pouvez aussi vous rendre sur l'excellent site Zone suivante : http://www.zonesuivante.com qui présente clairement et simplement ces instructions. Voir aussi les scripts proposés en annexe (exercices)

2.4.2. Potentialités du langage PHP

Avec ce langage, vous pourrez :

Ouvrir et écrire dans des fichiers → exemple : réaliser un compteur de visites personnalisé.

Afficher un élément ou un autre en fonction de l'heure → structure conditionnelle (Cf. annexe)

Retoucher rapidement la charte graphique de votre site (Cf. exercice en annexe)

Interroger et compléter des bases de données (Cf. paragraphe 3)

…etc.

2.5. Faire héberger vos scripts

Avant de placer vos pages chez le prestataire d'hébergement Lambda, il faut d'abord vous assurer que ce dernier offre les extensions PHP et Mysql (Cf. paragraphe 4 "PHP et bases de données en ligne).
Il existe des hébergements gratuits donnant accès à ces ressources.

Les tableaux ci-dessous offrent un état des lieux de l'offre pour les hébergements gratuits et payants. Je vous conseille de consulter la rubrique hébergement de PHP Index (Cf. paragraphe 4) pour des renseignements actualisés

Hébergements gratuits

D'après des tableaux disponibles sur le site PHPGratuit France (http://php-gratuit.forez.com/)

(N= NON X=OUI)

Noms	Adresses	PHP	Mysql	Espace	Pub	Version PHP	Commentaires
F2s	http://www.f2s.com/	**X**	**X**	20 Mo	non	4	propose aussi les extensions FP2000
Anotherlight	http://www.anotherlight.com/	**X**	**N**	50 Mo	non	3	Bonne prestation
OVH	http://www.ovh.org	**X**	**N**	60 Mo	Oui sur la première page	4	Bon serveur !

Hébergements gratuits (suite)

Noms	Adresses	PHP	Mysql	Espace	Pub	Version PHP	Commentaires
Free	http://www.free.fr	X	X	100 Mo	non	3	mail() désactivée
Forez	http://host.forez.com	X	X	25 Mo	Par frame	4	mail() active mais chdir() désactivée
Multimania	http://www.multimania.com	X	X	illimité	Frame, fenêtre ou menu	4	Fonction mail() désactivée
Freesurf	http://www.freesurf.fr	X	X	50 Mo	oui		mail() désactivée
Citeglobe	http://www.citeglobe.com	X	N	75 Mo	A mettre dans les pages	3	mail() active

Hébergements payants

Noms	Adresses	PHP	Mysql	Espace	Pub
OVH	http://ovh.net	X	X	Au choix	non
Online.fr	http://www.online.fr	X	N	Au choix	non
Domicile	http://www.domicile.fr (serveurs France Télécom)	X	N	Au choix (à partir de 30 Mo)	non

NB : les hébergements payants offrent une sécurité maximale (protection efficace, sauvegardes régulières des scripts et des bases de données, continuité du service, liaisons sécurisées pour les paiements en ligne…etc).

Vous pouvez aussi placer vos scripts sur un Intranet (réseau interne à un établissement) ou dans une "salle pupitres".

A terme, vous pourrez créer votre propre hébergement sur le Net, avec le développement de l'ADSL (technologie déployée en ce moment par France Télécom) ou le câble. Ces technologies offrent une IP fixe et donnent donc la possibilité d'héberger.

3. PHP et bases de données

C'est à ce niveau que vous allez percevoir toute la puissance de ce langage. En effet, les bases de données vont donner de la réactivité à votre site.

3.1. Installation et utilisation de Mysql et PhpMyAdmin (en local)

Si vous avez opté pour l'installation d'EasyPHP, vous n'avez rien de particulier à faire. En effet, cette "distribution" installe en plus de PHP et Apache, MySQL et l'utilitaire PhpMyAdmin.

→ Installation manuelle :

Les personnes souhaitant installer manuellement MySQL et PhpMyAdmin trouveront ci-dessous la procédure à suivre. Cette option permet de mieux comprendre ce qui se passe et éventuellement de pouvoir résoudre certains problèmes.

Installer MySql :

Le téléchargement de MySQL se fait sur le site http://www.mysql.com Depuis le milieu de l'année 2000 (version 3.23.28), le serveur MySQL est gratuit et répond aux spécifications des logiciels libres (GNU Licence). Téléchargez la version 3.23.34 ou supérieure (pour win32).

Pour installer MySQL, il suffit de décompacter l'ensemble des fichiers dans un répertoire temporaire. Lancez ensuite, l'exécutable qui installera le serveur de base de données dans le répertoire c:\Mysql

Pour terminer l'installation, il nous faut créer 2 fichiers *.BAT pour lancer et arrêter MySQL.

Pour démarrer MySQL, créez un fichier go.bat avec le bloc notes de Windows (à l'enregistrement, optez pour type de fichiers → tous) Ce fichier go.bat contiendra le texte suivant : c:\mysql\bin\mysqld.exe

De la même façon, créez un fichier stop.bat qui contiendra :

c:\mysql\bin\mysqladmin.exe -u root shutdown

N'oubliez pas de lancer votre serveur MySQL, avant toutes manipulations de données sinon vous obtiendrez à coup sûr des messages d'erreur.

Installation de PhpMyAdmin :

PhpMyAdmin est une application écrite en PHP qui permet de gérer depuis un navigateur des bases de données MySQL.

Avec cet utilitaire, vous pouvez :

- créer et supprimer des bases de données (si vous êtes administrateur)
- créer, copier, supprimer et modifier des tables
- supprimer, éditer et ajouter des champs à vos bases
- exécuter des requêtes SQL ("interrogations" de vos bases SQL = structured query language)
- importer des fichiers texte (*.txt) dans une table. Cette option est très intéressante pour préparer vos bases hors ligne, avant de les placer sur l'Internet.
- exporter et importer des fichiers au format CSV (champs séparés par une virgule, option d'enregistrement proposée par la plupart des tableurs).
- créer des sauvegardes de vos tables (cela peut toujours servir !)

Le téléchargement de cette application est possible sur le site officiel (http://phpmyadmin.sourceforge.net/) Téléchargez la version 2.2.3 (ou supérieure). Attention, il existe deux versions (avec extension php et php3), voir en fonction de votre hébergement local ou en ligne. Privilégiez la version php3.

Avant de procéder à l'installation, vous devez avoir sur votre machine un serveur Web configuré pour interpréter le PHP (3.09 ou supérieure) et avoir un serveur de bases de données MySQL installé.

Décompressez les fichiers dans le sous-répertoire htdocs d'Apache (ou celui où vous avez placé vos premiers scripts). Dans le cas d'une installation en ligne, il est conseillé de protéger votre répertoire d'installation par un fichier .htaccess si votre prestataire d'hébergement fonctionne avec un serveur Apache (Cf. paragraphe 4 "ressources en ligne)

L'étape suivante consiste à configurer l'application à l'aide du fichier config.inc.php3 (se trouvant dans le répertoire d'installation).

Au début du fichier :

$cfgServers[1]['port']='**3300**'; (3300 par défaut, numéro de port du **n**ième serveur MySQL. Ici **n**=1)
$cfgServers[1]['host']='**localhost**';
$cfgServers[1]['adv_auth']='**false**';
$cfgServers[1]['user']='**root**';
$cfgServers[1]['password']='**mot de passe**'; *

* en local, vous pouvez laisser ce champ libre.

Si vous êtes administrateur de la base de données et avez accès à toutes les bases, laissez le champ suivant libre (c'est le cas lors d'une installation en local). Dans un autre cas, écrivez le nom de la base que vous souhaitez administrer.

$cfgServers[1]['only_db']='**nom de la base**'; *

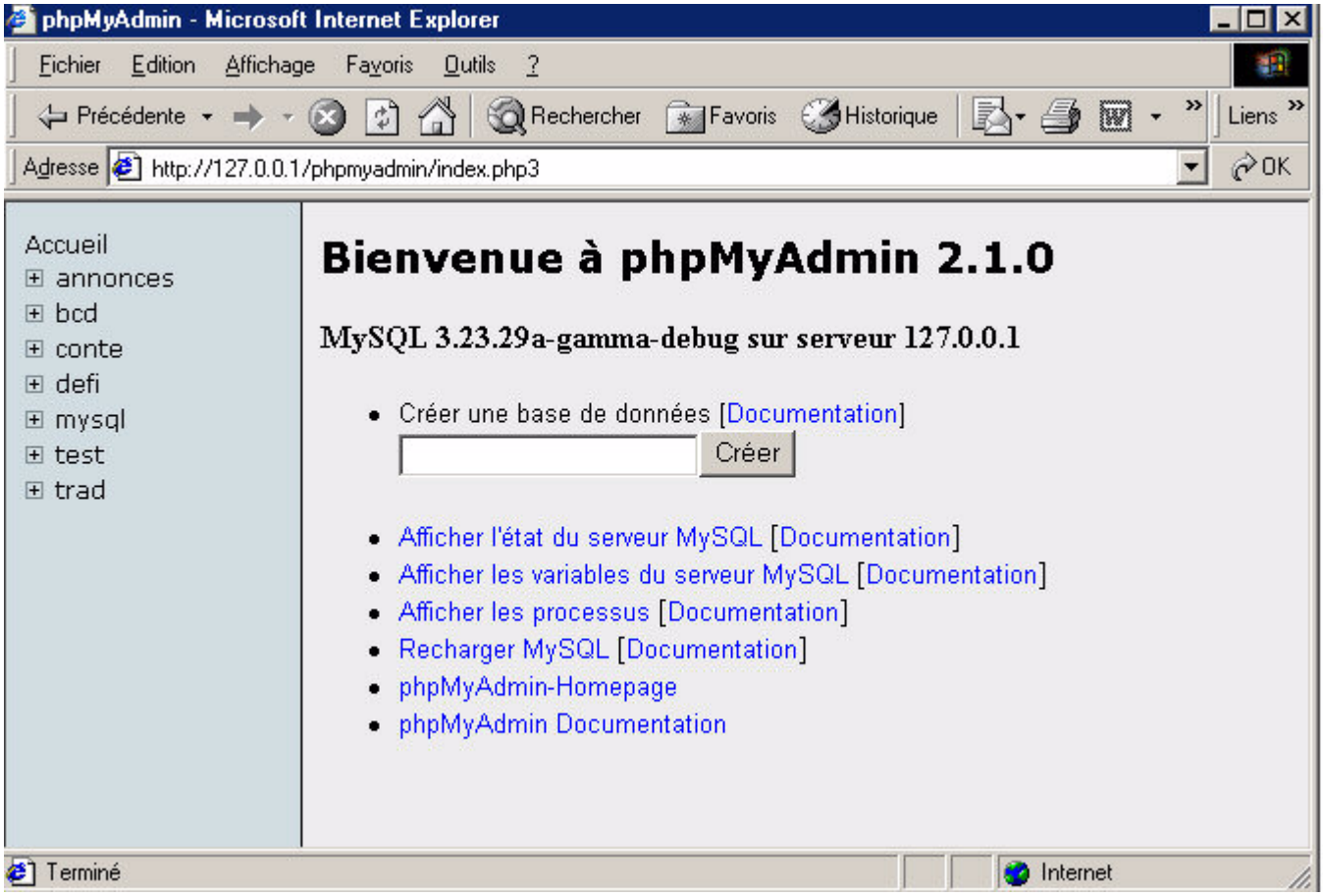
N.B. : Vous pouvez répéter l'opération pour chaque base à gérer, en incrémentant le chiffre du tableau cfgServers.

Par défaut, l'application PhpMyAdmin est disponible en langue anglaise. Nous allons faire en sorte que les menus soient écrits en français.
Toujours dans le fichier config.inc.php3, repérez la ligne suivante :
require("english.inc.php3"); → remplacez english par french → require("**french**.inc.php3");

L'utilitaire est désormais prêt à administrer vos bases. Si vous l'avez placé dans le sous-répertoire htdocs d'Apache, il est accessible en local à l'adresse http://127.0.0.1/phpMyAdmin/index.php3 ou http://localhost/phpMyAdmin/index.php3 (à condition d'avoir démarré votre serveur et votre base de données MySQL)

Utilisation de MySQL et PhpMyAdmin
(reprise partielle d'un article disponible sur Php Facile : http://www.toutestfacile.com/php/

PhpMyAdmin est une interface conviviale écrite en PHP qui permet de gérer sa base de données MySQL. On peut créer des bases des données, les supprimer, créer des tables et exécuter toutes les requêtes permises par MySQL.
En tapant l'adresse l'adresse http://127.0.0.1/phpMyAdmin/index.php3 ou http://localhost/phpMyAdmin/index.php3 (après avoir lancé Apache et MySQL), vous arrivez sur la première page de PhpMyAdmin (Cf. schéma suivant).



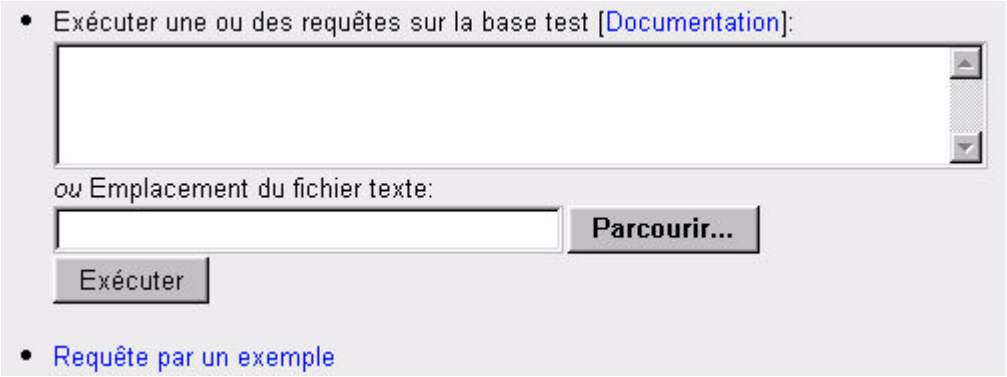
Dans la fenêtre de droite, on observe un champ qui sert à créer une base de données. Il suffit d'entrer un nom puis de cliquer sur "Créer" pour que la base de données soit créée. Evitez les accents, les caractères exotiques, les points et les espaces.

Dans la fenêtre de gauche on trouve une arborescence avec les bases de données (et les tables associées).

"annonces", "bcd", "conte", "defi", "mysql", "test" et "trad" sont des bases de données *. En cliquant dans la fenêtre de gauche sur l'une des bases de données, on obtient des détails sur celle-ci.

* en local, on peut administrer plusieurs bases de données. Sur le Web (hébergements gratuits), une seule base de données est mise à votre disposition.

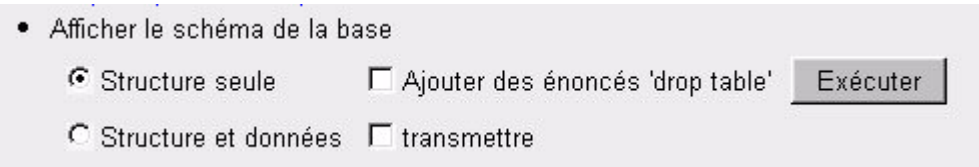
On retrouve ici, les différentes tables de la base de données "annonces", ainsi que le nombre de données présentes dans chaque table (9 entrées dans la table categorie). En dessous de ce tableau on voit :



C'est dans le champ du haut que l'on va pouvoir exécuter des requêtes du genre: "create table ..." ou "select * from ...". (Cf. annexes)

Le champ du dessous sert à indiquer un fichier où les requêtes auraient été enregistrées. On peut ainsi créer une table complète (structure + enregistrements).

En dessous on trouve :



Cette partie sert à "dumper" c'est à dire à créer les requêtes SQL nécessaires pour reconstituer la base de données. On peut sauver la "structure seule" c'est à dire ce qui a permis de créer les tables.

On peut sauver la "structure et les données" c'est à dire ce qui a permis de créer les tables ainsi que les données (option à utiliser pour sauver une base de données).

Deux options s'offrent ensuite à nous :

- la première "drop table" sert à insérer dans le fichier des instructions demandant la suppression des tables (utile pour remplacer des tables existantes).
- la seconde sert à indiquer si on veut mettre les requêtes dans un fichier (case à cocher dans ce cas) ou seulement les voir à l'écran.

On trouve ensuite un moyen de créer une table de manière conviviale:

Avec cette interface, il suffit d'entrer le nom souhaité pour une nouvelle table ainsi que le nombre de champs que l'on veut créer. Ce nombre pourra être modifié par la suite. En cliquant sur "Exécuter", une interface apparaît nous demandant les noms des champs et leurs types.

Lorsque les tables et les champs sont créés et que des données y sont entrées, on peut les afficher, les supprimer, les modifier…etc.

PhpMyAdmin est un outil très convivial, de nombreux prestataires d'hébergement l'ont adopté. C'est le cas de Free que nous allons étudier dans le paragraphe 3.3.

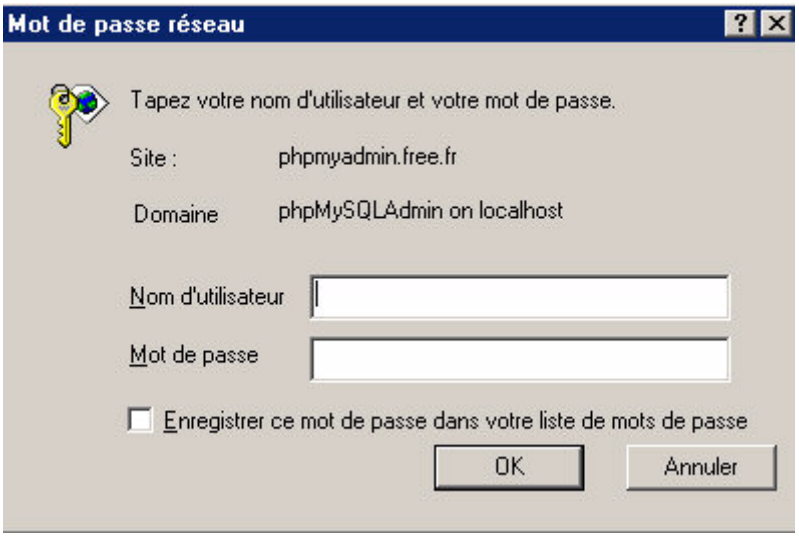
3.2. Au travail !

Vous allez maintenant vous exercer à créer une base, à y ajouter des données, à la gérer avec PhpMyAdmin. Pour cela, prenez la feuille du script détaillé figurant en annexe.

3.3. Travail en ligne : le cas de Free

Etapes à respecter pour mettre en place un site et une base de données chez Free.

1. S'inscrire gratuitement chez ce prestataire d'hébergement (http://www.free.fr).
2. Vous allez devoir compléter un formulaire. Entrez une adresse postale valide, vous allez y recevoir un courrier. Je vous conseille de demander le cédérom Free et non d'opter pour un envoi avec les "informations de configuration uniquement". Cela vous permettra d'obtenir d'une part, le cédérom d'installation (qui comprend d'excellents didacticiels) et d'autre part un document reprenant toutes les caractéristiques d'accès à votre compte.
3. Quelques jours plus tard, vous recevrez votre courrier. **Il faut compter environ 2 semaines.**
4. Allez sur le site http://www.free.fr, choisissez l'onglet "Votre espace Web" puis "Gérer votre espace".
5. Optez pour "Gérer vos pages personnelles". Dans un premier temps, vous allez devoir activer votre compte (lien j'y vais). Cela ne se fait pas par défaut, certains abonnés se contentent de l'accès à l'Internet proposé par Free et ne créent pas de pages. Votre compte (espace) sera activé dans les quatre heures, parfois plus vite. Vous pourrez alors y publier vos pages, à l'aide d'un logiciel de transfert FTP (Cf. document "Publier sur l'Internet de la page au site").
   Si votre première page se nomme index.php3, elle sera accessible à http://votrelogin.free.fr ou http://votrelogin.free.fr/index.php3 (votre login vous est attribué automatiquement à l'inscription)
6. Allez ensuite sur l'onglet "PHP – Base de données" et activez votre base de données. Chez Free, cela est fait toutes les nuits.
   Cette base et l'utilitaire PhpMyAdmin sont accessibles à l'aide du lien "Gérer votre base de données MySQL" ou en entrant http://sql.free.fr/phpMyAdmin (ou http://phpmyadmin.free.fr/phpMyAdmin) dans la barre d'adresse de votre navigateur.
   Une boîte de dialogue .htacces (protection des serveurs Apache sous Linux) va apparaître :



Entrez alors votre nom d'utilisateur (login) et votre mot de passe. Vous pourrez alors gérer votre base de la même manière que lors de l'utilisation en local.

4. Bibliographie et ressources en ligne

→ Bibliographie :

➢ *Pour débuter :*

Pages Web dynamiques avec ASP3, PHP 4 et SQL
de Jean-Marc Herellier et Philippe Mérigot
Editions Campus Press Collection "L'intro"
ISBN : 2-7440-01012-X prix : environ 150 FF
Cet ouvrage permet de se familiariser avec les langages dynamiques et de percevoir les plus apportés par rapport à un site classique.

PHP 4 et MySQL
Editions Micro Application
prix : environ 200 FF

➢ *Pour aller plus loin*

Programmation en PHP 4
de Léon Atkinson
Editions Campus Press Collection "Référence"
ISBN : 2-7440-1000-6 prix : environ 230 FF
Cet ouvrage détaille toutes les instructions et fonctions du langage PHP. Chaque fonction est accompagnée d'un exemple, malheureusement ils sont en anglais et ne sont pas toujours accompagnés des copies d'écrans. Il s'agit néanmoins d'une référence pour tous les programmeurs. Le livre est distribué avec un cédérom reprenant tous les exemples, ainsi que le code source PHP 4 et le serveur Apache pour environnements Windows et Unix (Linux).

PHP Professionnel
de J. Castagnetto, H. Rawat, S. Schumann, C. Scollo, D. Veliath
Editions Eyrolles professionnel – Wrox Collection "Solutions développeurs"
ISBN : 2-212-09235-0 prix : environ 320 FF
Sans doute l'un des meilleurs ouvrages pour découvrir toutes les subtilités du langage PHP et des bases de données MySQL. Les spécialistes s'accordent à dire qu'il s'agit d'une référence incontournable. Cet ouvrage propose 4 études de cas en détail (un site de commerce électronique, une application de gestion de forums de discussion : le célèbre Phorum, une application web de gestion d'emails, une application d'accès à une base de données biomédicale). Les codes des exemples du livre sont téléchargeables sur le site de l'éditeur.

MySQL
Editions Campus Press
Collection "Référence"
prix : environ 200 FF
La référence pour cette base de données. Pour ceux qui utilisent régulièrement et de manière très "pointue" MySQL.

→ Ressources en ligne :

PhpInfo.net : http://www.phpinfo.net (sans doute la référence française sur le Net, vous trouverez une multitude d'informations classées sur PHP / PHP3 / PHP4 /MySQL, les mises à jour sont très fréquentes)

PHP France : http://www.phpfrance.com (des cours, des exemples commentés, des forums de discussion très réactifs)

Banque de scripts PHP, MySQL : http://www.phpscripts-fr.net/ (plusieurs centaines de scripts classés, un must pour le développeur qui manque de temps)

PHP Facile : http://www.toutestfacile.com/php/ (un site très pédagogique pour apprendre le PHP)

SQL Facile : http://www.toutestfacile.com/sql/ (le site pour apprendre le SQL simplement et efficacement)

PHP Débutant : http://www.phpdebutant.com (pour réussir son entrée dans le monde PHP, MySQL, des cours bien "dosés", d'intéressants chapitres sur la protection ".htaccess" et les "Chmod")

Comment ça marche l'informatique ? http://www.commentcamarche.net (un site qui a le mérite de simplifier l'approche de l'informatique, PHP et MySQL + de nombreuses autres rubriques)

Zone suivante : http://www.zonesuivante.com (initiation en douceur à PHP et MySQL, découverte des différentes fonctions et instructions).

Php-gratuit : http://phpgratuit.free.fr (des exemples originaux, à visiter!)

Devenez un PHPmestre : http://phpmestre.forez.com (de très bons exemples analysés et commentés)

Maelstrom : http://maelstrom.nexen.net (un site clair et précis qui ne manque pas d'humour)

Conclusion

J'espère que vous avez perçu l'intérêt du langage PHP et de MySQL pour dynamiser votre site. Les sites classiques sont très bien adaptés aux pages personnelles qui n'ont pas besoin d'évoluer fréquemment (CV par exemple) ou aux petites productions. Si votre site est appelé à prendre de l'ampleur, je ne peux que vous recommander d'apprendre ce langage et de vous initier à la manipulation des bases de données. Cela ne devrait pas vous prendre trop de temps. N'oubliez pas que vous n'êtes pas seul, une communauté de développeurs sévit sur les sites indiqués dans le paragraphe 4. N'hésitez pas à les solliciter. Je reste à votre disposition pour de plus amples renseignements.

David Ducrocq david.ducrocq@laposte.net - janvier 2002
Conseiller TICE – Liévin – Hénin Beaumont (Pas-de-Calais)

© David Ducrocq C.N.T.E Liévin Hénin Beaumont (janvier 2002)

Annexe 1 - Fonctions et instructions principales en PHP

Principales instructions et fonctions en PHP :
(la plupart des informations figurant dans ces pages viennent du site "Zone suivante" Cf. Zone suivante : http://www.zonesuivante.com/zonephp/basephp.php?page=ind)

PHP comporte plus de 500 fonctions et instructions. Nous n'allons pas les explorer en détail mais simplement découvrir les principales.

Les variables

Les variables sont précédées du signe $, quel que soit leur type. La déclaration d'une variable est optionnelle. C'est PHP qui décide de son sort dans le déroulement du script.

Les variables scalaires

Les variables scalaires sont de 3 types : chaînes, entiers, réels.
Affectation
$nom= "toto";
$age= 23;

Il existe aussi les variables de types tableau et tableau associatif. *Nous ne les aborderons pas dans cette documentation succincte (Cf. ressources en ligne – pages 19 et 20 du document principal)*

Les opérateurs :

Les opérateurs de comparaison

== égalité (2 signes égal accolés)
> supérieur strict
< inférieur strict
<= inférieur ou égal
>= supérieur ou égal
! négation

Les opérateurs logiques

Les opérateurs logiques sont principalement utilisés dans les tests, par ex. dans un if (condition)
&& et
|| ou
xor ou exclusif
! négation
Note : les opérateurs **and**, **or** , **not** sont également disponibles et font la même chose.

Les opérateurs arithmétiques

+ addition
- soustraction
/ division
* multiplication
++ incrément
-- décrément

Un des opérateurs divers

L'opérateur de concaténation **.** est utilisable sur les scalaires (Cf. paragraphe sur les variables) chaînes
$a = "Salut !";
echo "$a" . " le monde";
Le résultat affiche à l'écran le message : Salut ! le monde

Structures de contrôles

Tests (structures conditionnelles)

If
Syntaxes

Test if "de base"

```
if( [condition] ) {
	...
}
```

Test if-else :

```
if( [condition]  ) {
	...
} else {
	...
}
```

Test if-elseif :

```
if( [condition]  ) {
	...
} elseif( [condition] ) {
	...
} else {
	...
}
```

Dans le cas de plusieurs tests successifs sur une même variable, on utilisera plutôt le test switch
Note: si le corps du test ne comporte qu'une instruction, les accolades {} sont optionnelles.

Switch

Le **switch** est l'équivalent du *SELECT CASE* en Basic. Il permet de confronter une variable à plusieurs valeurs prédéfinies. Il permet un code plus compact et lisible qu'un test *if-elseif-elseif...*

Syntaxes

```
switch( [variable] ) {
	case [valeur1] :
		[bloc d'instructions]
		break;
	case [valeur2] :
		[bloc d'instructions]
		break;
	...
	default:
		[bloc d'instructions]
}
```

La valeur de [variable] est comparée successivement à chaque **case**. Si il y a égalité, le bloc d'instruction est exécuté.
Il ne faut pas omettre le **break** en fin de bloc, sans quoi le reste du switch est exécuté.
Enfin, le handler **default** permet de définir des instructions à effectuer par défaut, c'est à dire si aucun **case** n'a "fonctionné".

© David Ducrocq C.N.T.E Liévin Hénin Beaumont (janvier 2002)

Annexe 2 - Petits scripts en PHP (exercices)

1. Installez EasyPHP puis démarrez.

2. Ecrivez votre premier script index.php3(Cf. document général page 12)

3. Faire évoluer la charte graphique de votre site(Cf. page suivante)
Cet exemple va nous permettre d'utiliser l'instruction include qui permet d'insérer un fichier (.php3, .htm, .txt …) dans une page.
Essayez de modifier les fichiers droits.txt, haut.php3 et bas.php3 pour changer l'interface des pages exemples.

4. Utilisation d'un petit formulaire :
Nous allons récupérer les données d'un formulaire pour les utiliser dans une page.
A l'aide d'un éditeur, écrivez le script suivant et sauvegardez-le sous le nom form1.php3 dans un sous-répertoire de votre espace de travail.

```
<form method="post" action="form2.php3">
Nom : <input type="text" name="nom" size="10"><br>
Prénom : <input type="text" name="prenom" size="10">
Age : <input type="text" name="age" size="10">
<input type="submit" value="OK">
</form>
```

Créez maintenant le script de "réception" form2.php3 en utilisant la syntaxe suivante.

```
<?
Echo "Bonjour $prenom $nom, vous avez $age ans";
?>
```

Allons plus loin, ajoutez un champ sexe dans votre formulaire (size= "1") sur la page form1.php3. Précisez à l'utilisateur que les deux mots acceptés sont m ou f.
Modifiez la page form2.php3 afin que le fond d'écran soit rose si la variable $sexe est f et bleu si cette variable est m. Pour ce faire, utilisez la structure conditionnelle (if … else) disponible à la page précédente et inspirez-vous de la page haut.php3 utilisée dans l'exercice 4 ("Faire évoluer la charte graphique de votre site ").
NB : en HTML la couleur rose est codée #FF80C0 et la couleur bleue #0080FF. Pour éviter, les saisies non reconnues dans le champ sexe, il faut améliorer ce script avec un menu déroulant proposant F et M.

5. Protéger une page avec un mot de passe.
Pour ce script, nous allons utiliser l'instruction header qui permet de rediriger vers une autre page.
Créez d'abord une page basique qui se nommera protegee.htm vous y écrirez "Vous êtes sur une page protégée par un mot de passe".
Reprenez la page du formulaire form1.php3 utilisée à l'exercice précédent, ne conservez qu'un champ (Cf. Mot de passe : <input type="password" name="pass" size="8">)
Ecrivez le script suivant au sein de votre page form2.php3 :

```
<? if($pass==toto)      {
                        // redirige vers la page protégée
                        header("Location:protegee.htm");
                        exit;
                        }
else  {
      echo "Vous n'êtes pas autorisé(e) à visualiser la page protégée" ;
      }
?>
```

Remarque : vous noterez que la source de la page n'affiche pas la variable $pass, le mot de passe demeure secret. C'est l'un des avantages des langages "côté serveur", ce type de script existe en javascript mais le mot de passe est très souvent visible en clair dans la source de la page.

Faire évoluer la charte graphique d'un site (détails des différents scripts)

script haut.php3

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title>Site dynamique en PHP</title>
```

```
</head>
<?
// **** Gestion de la couleur du fond
echo "<body bgcolor=\"FF8000\">";
?>
```

script index.php3

```
<?
include "haut.php3";
?>
<p align="center"><font size="5"></font> </p>
<p align="center"><font size="5">Page 1</font></p>
<p> </p>
<p align="center"><a href="page2.php3">lien vers la page 2</a></p>
<p> </p>
<p> </p>
<p> </p>
<?
echo "<p align='center'>";
include "droits.txt";
include "bas.php3";
?>
```

script bas.php3 (de l'HTML pur)

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

fichier droits.txt

```
Copyright - Ma pomme - mai 2001
```

Protection par mot de passe (détails des différents scripts)

Script form1.php3

```
<form method="post" action="form2.php3">
Mot de passe : <input type="password" name="pass" size="10">
<input type="submit" value="OK">
</form>
```

Script form2.php3

```
<? if($pass==toto)      {
                        // redirige vers la page protégée
                        header("Location:protegee.htm");
                        exit;
                        }
else  {
      echo "Vous n'êtes pas autorisé(e) à visualiser la page protégée" ;
      }
?>
```

Un répertoire téléphonique

1. Création de la table mySQL en local avec PhpMyAdmin

Avec PhpMyadmin, vous pouvez créer une table de deux façons. Nous allons les détailler. Il faut tout d'abord démarrer l'utilitaire (Cf. page 15 bas). Ensuite, il faut créer une base de données, nous l'appellerons "test" (Cf. page 16 du document).
Créez la table en écrivant sa structure à l'endroit adéquat (Cf. page 17 haut "Exécuter une ou des requêtes sur la base test")

Création de la table repertoire

```
CREATE TABLE repertoire (
  id int(3) DEFAULT '1' NOT NULL auto_increment,
  nom varchar(50) NOT NULL,
  prenom varchar(50) NOT NULL,
  tel varchar(14) NOT NULL,
  PRIMARY KEY (id)
);
```

Remarques : chaque ligne (sauf la première et l'avant dernière) se termine par une virgule. La dernière ligne est terminée par un point virgule. Evitez les caractères exotiques, les espaces, les accents dans le nom de la base, des tables et des champs. Les points sont à éviter aussi, certains prestataires d'hébergement n'acceptent pas les noms de table avec un point (c'est le cas de Freesurf). Nous pouvions aussi écrire cette structure dans un fichier texte et importer ce dernier (Cf. page 17 – zone " *ou* Emplacement du fichier texte:" et bouton parcourir)

La deuxième solution consiste à créer la table "manuellement" en utilisant les différents "menus" de l'utilitaire PhpMyAdmin. Pour cela, on utilise la zone "Créer une table sur la base…" (Cf. page 17 bas). A ce niveau, on doit estimer le nombre de champs de la table (en sachant que cela n'est pas figé). Il suffira ensuite de créer les champs un à un en remplissant les zones adéquates. L'utilitaire étant très bien réalisé, je vous laisse expérimenter seul(e) ces étapes.

Notre table est créée, il nous faut maintenant y entrer des données et les consulter.

2. Ecriture du script d'ajout des données

Ecrivons maintenant le script suivant avec l'éditeur. Nous l'enregistrerons sous le nom form.php3

```
<html>
<?php
 if ($action=="ajout")
  {
  //connexion à la base de données et à la table
  $connexion=mysql_connect("localhost","root","");
  $db=mysql_select_db("test", $connexion);
  // ajout des données issues du formulaire dans la table
  $resultat_sql=  mysql_query("insert into repertoire values  (\"$nom\",\"$prenom\",\"$tel\")", $connexion);
 // "fermeture" de la connexion à la base de données
 mysql_close($connexion);
  // Message
 echo "J'ai ajouté $nom  $prenom  $tel";
  }
 else
  {
  echo "<table>";
  echo "<form action=form.php3 method=post>";
  echo "<input type=hidden name=action value=ajout>";
  echo "<tr><td>Nom :</td><td><input type=text name=nom size=30></td></tr>";
  echo "<tr><td>Prenom :</td><td><input type=text name=prenom size=30></td></tr>";
  echo "<tr><td>Téléphone :</td><td><input type=text name=tel size=14></td></tr>";
  echo "<tr><td colspan=2 align=center><input type=submit value=Ajouter></td></tr>";
  echo "</form>";
  echo "</table>";
  }
```

```
?>
<html>
```

Détaillons ce script :

Par défaut, un formulaire est proposé à l'internaute, il est classique. Si le bouton "Ajouter" est cliqué par l'utilisateur l'ajout des données dans la table repertoire commence (Cf. if ($action=="ajout")…)

Cette partie du script débute par la connexion à la base de données, 4 paramètres sont utilisés.

```
$connexion=mysql_connect("localhost","root","");
$db=mysql_select_db("test", $connexion);
```

L'adresse du serveur sur lequel la base de données est hébergée (ici localhost, nous sommes en local). Localhost est aussi utilisé si le serveur qui gère la base de données est le même que celui qui héberge le serveur Apache. Beaucoup de prestataires d'hébergement, pour des raisons de sécurité, font appel à deux machines différentes. En ligne, localhost sera remplacé par une syntaxe qui est propre à chaque prestataire (Free accepte localhost)
L'utilisateur (ici root : nous sommes en local) qui correspond au login que vous transmettra votre prestataire d'hébergement.
Le mot de passe (vide pour nous "" car nous sommes en local) qui correspond au mot de passe qui vous sera donné par votre prestataire d'hébergement.
Le nom de la base de données (ici test). Chez la plupart des prestataires d'hébergement ce nom est similaire au login.

Remarque : pour la simplicité de l'exemple, nous écrivons les différents paramètres "en clair" dans le script. Pour des raisons de sécurité, il faudra faire appel en ligne à l'instruction require (similaire à include) qui permettra de placer ces données sensibles dans un répertoire protégé par un .htaccess.

La fonction php qui permet de se connecter à la base est mysql_connect.

```
$connexion=mysql_connect("localhost","root","");
```

Ensuite, on sélectionne la base sur laquelle on désire travailler avec mysql_select_db

```
$db = mysql_select_db("test", $connexion);
```

Ensuite, on utilise la fonction mysql_query qui va nous permettre d'exercer une requête sur la base (ici une requête, une demande d'insertion de données : insert)

```
$resultat_sql=  mysql_query("insert into repertoire values  (\"$nom\",\"$prenom\",\"$tel\")", $connexion);
```

La fonction mysql_close permet de fermer la connexion à la base de données.

```
mysql_close($connexion);
```

Les données sont insérées dans la table, on peut le vérifier avec PhpMyAdmin. On termine par l'affichage des données entrées pour éviter d'afficher une page blanche qui risque d'inquiéter l'utilisateur.
echo "…etc.

3. Afficher les données

Il nous faut maintenant écrire un script pour visualiser les données entrées dans la table "repertoire". Nous allons l'appeler list.php3

```
<html>
<?php

  //connexion à la base de données et à la table
  $connexion=mysql_connect("localhost","root","");
  $db=mysql_select_db("test", $connexion);
  // on récupère toutes les données de la table repertoire
  $resultat_sql=  mysql_query("select * from repertoire", $connexion);
  // on détermine le nombre de données dans cette variable $resultat_sql
  // à l'aide de la fonction mysql_num_rows
  $num = mysql_num_rows($resultat_sql);

  // on affiche la totalité de ces résultats en utilisant une boucle
```

```
 // on initialise la variable $i à 0
 $i = 0;
 // on demande que la boucle s'effectue de 0 jusqu'au nombre $num (nombre de données
 // dans la table)
 while ($i<$num)
 {
 // on extrait les données de $resultat_sql (variable qui est en fait un tableau de données)
 $nom = mysql_result($resultat_sql,$i,"nom");
 $prenom = mysql_result($resultat_sql,$i,"prenom");
 $tel = mysql_result($resultat_sql,$i,"tel");
 // on procède à l'affichage des données récupérées
 // la balise <br> permet d'effectuer un passage à la ligne après chaque ligne de
 // données extraites (nom, prénom, téléphone)
 echo "$nom $prenom $tel<br>";
 $i++;
 }

// "fermeture" de la connexion à la base de données
mysql_close($connexion);

?>
<html>
```

Commentaire :

Le résultat de $resultat_sql= mysql_query("select * from repertoire", $connexion); est stocké dans un tableau :

id	nom	prenom	tel
1	nom1	prenom1	tel1
2	nom2	prenom2	tel2
…	…	…	…

La fonction mysql_num_rows permet de compter le nombre de ligne de ce tableau (attention ici les têtes de colonnes ne sont données que pour faciliter la compréhension, elles n'existent pas dans la variable $resultat_sql)

La fonction mysql_result permet d'extraire les données pour chaque champ de ce tableau.
Veillez à respecter la syntaxe suivante : $champ1 = mysql_result($resultat_sql,$i,"nom du champ1");

4. Modification et suppression de données avec PhpMyAdmin

L'utilitaire PhpMyAdmin permet très simplement de modifier, de supprimer les données à l'intérieur d'une table.
Il suffit de demander l'affichage des données et d'utiliser les liens explicites proposés au bout de toutes les lignes de données.

5. Suppression de données par l'utilisateur

Nous allons renommer notre script list.php3 en list2.php3, afin de conserver intactes les différentes étapes de nos travaux.

Ajoutons d'abord la ligne : $id = mysql_result($resultat_sql,$i,"id");
au dessus de $nom = mysql_result($resultat_sql,$i,"nom");

Cette ligne nous permet de récupérer l'id (identifiant) de chaque ligne de données.

Nous allons utiliser cet identifiant (id) dans la ligne echo "… se trouvant au sein de la boucle d'affichage des résultats.

On modifie : echo "$nom $prenom $tel
";

en echo "$nom $prenom $tel - Suppression
";

Remarquez le passage de la variable dans l'URL. On demande au script de nous afficher un lien menant vers la page supp.php3, avec la variable $id passée en paramètre. Cette variable sera traitée avec la valeur transmise dans la page cible.

Il nous reste à écrire le script supp.php3

```
<html>
<?php

  //connexion à la base de données et à la table
  $connexion=mysql_connect("localhost","root","");
  $db=mysql_select_db("test", $connexion);
  // on récupère les données de la table repertoire liées à l'id=$id et on les supprime
  $resultat_sql=  mysql_query("delete from repertoire where id='$id'", $connexion);

 // "fermeture" de la connexion à la base de données
 mysql_close($connexion);

echo "Données effacées";
?>
<html>
```

6. Modification des données par l'utilisateur

Nous allons renommer notre script list2.php3 en list3.php3, afin de conserver intactes les différentes étapes de nos travaux.

Proposer la modification des données est à peine plus compliqué.

Modifions : echo "$nom $prenom $tel - Suppression
";
en
echo "$nom $prenom $tel - Modification - Suppression
";

Il nous reste à écrire le script update.php3

```
<html>
<?php
 if ($action=="modifier")
  {
  //connexion à la base de données et à la table
  $connexion=mysql_connect("localhost","root","");
  $db=mysql_select_db("test", $connexion);
  // modification des données issues du formulaire dans la table
  $resultat_sql=   mysql_query("update repertoire SET nom='$nom',prenom='$prenom', tel='$tel' WHERE id='$id'",
$connexion);

 // "fermeture" de la connexion à la base de données
 mysql_close($connexion);
  // Message
 echo "Données modifiées";
  }
else
{
//connexion à la base de données et à la table
$connexion=mysql_connect("localhost","root","");
$db=mysql_select_db("test", $connexion);
```

```
// on récupère les données de la ligne $id
$resultat_sql=  mysql_query("select * from repertoire where id='$id'", $connexion);

  // on détermine le nombre de données dans cette variable $resultat_sql
  // à l'aide de la fonction mysql_num_rows
  $num = mysql_num_rows($resultat_sql);

  // on affiche la totalité de ces résultats en utilisant une boucle
  // on initialise la variable $i à 0
  $i = 0;
  // on demande que la boucle s'effectue de 0 jusqu'au nombre $num (nombre de données
  // dans la table répondant à la demande ici une seule puisque l'id de chaque ligne est unique)
  while ($i<$num)
  {
  // on extrait les données de $resultat_sql (variable qui est en fait un tableau de données)
  $id = mysql_result($resultat_sql,$i,"id");
  $nom = mysql_result($resultat_sql,$i,"nom");
  $prenom = mysql_result($resultat_sql,$i,"prenom");
  $tel = mysql_result($resultat_sql,$i,"tel");
  $i++;
  }
//on utilise ces données extraites dans un formulaire de modification
echo "<table>";
  echo "<form action=update.php3 method=post>";
  echo "<input type=hidden name=action value=modifier>";
  echo "<tr><td>Nom :</td><td><input type=text name=nom size=30 value='$nom'></td></tr>";
  echo "<tr><td>Prenom :</td><td><input type=text name=prenom size=30 value='$prenom'></td></tr>";
  echo "<tr><td>Téléphone :</td><td><input type=text name=tel size=14 value='$tel'></td></tr>";
  echo "<tr><td colspan=2 align=center><input type=submit value=Modifier></td></tr>";
  echo "<input type=hidden name=id value='$id'>";
  echo "</form>";
  echo "</table>";
 // "fermeture" de la connexion à la base de données
 mysql_close($connexion);
}
?>
<html>
```

J'espère que cette série de scripts vous aura permis de percevoir les potentialités offertes par le couple PHP / MySQL.