

PHP *les bases*

PHP / mySql *une approche pratique basée sur la réalisation étape par étape du site de vente en ligne d'une librairie de Bandes Dessinées*

par Xavier Braive et François Plégades

Notes

| |
|--|
| |
|--|

PHP *les bases*

Table des matières

Table des matières

| | |
|--|----|
| Principes généraux - Rappel..... | 10 |
| Service statique d'une page HTML..... | 10 |
| Service dynamique d'une page HTML..... | 12 |
| Pages dynamiques pour le Web = +de possibilités !..... | 17 |
| PHP: simple, intuitif et complet(able)..... | 17 |
| Les bases du langage..... | 19 |
| Formats et conventions..... | 20 |
| Fichiers..... | 20 |
| Balises..... | 20 |
| Instructions..... | 21 |
| Chaîne de Caractères..... | 22 |
| Exemple:..... | 22 |
| Sortie vers le client..... | 23 |
| Exemple:..... | 23 |
| Echappement..... | 24 |
| Echappements particuliers..... | 24 |
| Commentaire..... | 24 |
| Bases du Langage..... | 25 |
| Variables..... | 28 |
| syntaxe..... | 28 |

Notes

| |
|--|
| |
|--|

PHP *les bases*

| | |
|----------------------------------|----|
| Assignation..... | 28 |
| Constantes..... | 29 |
| syntaxe..... | 29 |
| Assignation..... | 29 |
| Application..... | 30 |
| Corrigé..... | 31 |
| Référencement..... | 32 |
| Transtypage explicite..... | 33 |
| Variables variables..... | 33 |
| Tableaux (Listes)..... | 34 |
| Tableaux multidimensionnels..... | 37 |
| Portée des variables..... | 40 |
| Passage de paramètres http..... | 41 |
| GET..... | 41 |
| POST..... | 42 |
| FILE..... | 43 |
| Application..... | 44 |
| Corrigé formLivre.php..... | 45 |
| Corrigé doLivre.php..... | 46 |
| Opérateurs..... | 48 |
| Arithmétiques:..... | 48 |
| Incrémentation:..... | 48 |
| Chaîne (concaténation):..... | 48 |
| Logique (Comparaison)..... | 49 |

Notes

PHP *les bases*

| | |
|--|----|
| Combinaisons Logiques..... | 49 |
| Application..... | 50 |
| Corrigé..... | 51 |
| Les contrôle de flux (conditionnels et boucles)..... | 52 |
| Contrôles de Flux..... | 53 |
| Les Tests..... | 53 |
| If..... | 53 |
| Switch..... | 54 |
| Application..... | 56 |
| Corrigé..... | 57 |
| Traitements en Boucle..... | 58 |
| While (tant que condition vraie)..... | 58 |
| Do While..... | 59 |
| For..... | 59 |
| Foreach..... | 60 |
| Exemple..... | 60 |
| APPLICATION..... | 61 |
| Corrigé..... | 62 |
| Fonctions..... | 65 |
| Syntaxe..... | 65 |
| Portée..... | 66 |
| Exemple..... | 66 |
| Application..... | 68 |
| Corrigé..... | 69 |

Notes

PHP *les bases*

| | |
|--|-----|
| Corrigé 2..... | 70 |
| Librairies et inclusion de fichier..... | 71 |
| Application..... | 71 |
| Corrigé..... | 72 |
| Application..... | 73 |
| Corrigé..... | 74 |
| Les fonctions et variables internes du PHP..... | 79 |
| Fonctions PHP..... | 80 |
| Mathématiques..... | 80 |
| Fonctions de chaîne:..... | 82 |
| Fonctions de dates:..... | 91 |
| Exemple :..... | 94 |
| Fonctions associées aux tableaux:..... | 97 |
| Tri..... | 100 |
| Application..... | 101 |
| Corrigé..... | 102 |
| Variables PHP..... | 104 |
| Variables Serveur (ici apache, donc peuvent différer)... | 105 |
| Navigation et maintien des données..... | 109 |
| Redirection..... | 110 |
| Application : Détection Langue..... | 110 |
| Corrigé..... | 111 |
| Lecture / écriture de fichiers..... | 112 |
| Ouverture fichier:..... | 112 |

Notes

PHP *les bases*

| | |
|--|-----|
| Lecture:..... | 113 |
| Détection la fin du fichier..... | 114 |
| Ecriture:..... | 115 |
| Fermeture..... | 115 |
| Application : Liste d'Auteurs..... | 116 |
| Corrigé..... | 117 |
| Application : Boutique BDs..... | 118 |
| Corrigé..... | 120 |
| Cookies..... | 123 |
| Session..... | 126 |
| APPLICATION : SESSION..... | 128 |
| Corrigé..... | 131 |
| Accès bases de données MySql..... | 133 |
| Bases de données relationnelles et SQL..... | 134 |
| MYSQL..... | 135 |
| Type de données de MYSQL..... | 135 |
| Types numériques..... | 135 |
| Types numériques entier..... | 136 |
| Types numériques décimaux..... | 136 |
| Types dates..... | 137 |
| Types chaînes de caractères..... | 138 |
| Type binaire BLOB (Binary Large Object)..... | 139 |
| Type ENUM et SET..... | 139 |
| SQL..... | 140 |

Notes

PHP *les bases*

| | |
|---|-----|
| DDL (Data Definition Language)..... | 141 |
| DML (Data Manipulation Language)..... | 141 |
| DQL (Data Query Language)..... | 141 |
| DCL (Data Control Language)..... | 141 |
| Commande de contrôle transactionnel..... | 142 |
| Les contraintes d'intégrité (CONSTRAINT)..... | 142 |
| Création d'une Base de Données mySql..... | 145 |
| Sélection base..... | 148 |
| Envoi d'une requête SQL..... | 148 |
| Requêtes SQL..... | 148 |
| Sélections..... | 148 |
| Condition numérique..... | 149 |
| Condition chaîne..... | 149 |
| Combinaisons..... | 149 |
| Limitation..... | 149 |
| Ordre..... | 149 |
| Récupérer le résultat..... | 151 |
| en utilisant les clés..... | 151 |
| en utilisant les index..... | 152 |
| en utilisant un extract:..... | 152 |
| en utilisant un list..... | 152 |
| Insertion..... | 152 |
| Suppression..... | 153 |
| Modification..... | 153 |

Notes

PHP *les bases*

| | |
|---|-----|
| Gestion des erreurs..... | 155 |
| APPLICATION..... | 156 |
| Corrigé..... | 158 |
| APPLICATION..... | 159 |
| Correction..... | 166 |
| Fonctions graphiques..... | 170 |
| GD (Fonctions graphiques)..... | 171 |
| Fonction de création d'image..... | 171 |
| Gestion des couleurs de la palette image..... | 173 |
| Gestion des polices | 173 |
| correction..... | 176 |
| Tracé de graphique..... | 177 |
| Tracé des lignes..... | 177 |
| Application Graphique..... | 178 |
| Correction..... | 181 |
| Réglages..... | 184 |
| php.ini..... | 185 |
| Répertoire www..... | 187 |
| Répertoire tmp..... | 188 |
| Références..... | 189 |
| LES BALISES HTML..... | 190 |
| SQL (Structured Query Langage)..... | 196 |

Notes

PHP *les bases*

Introduction

Ce chapitre présente le principe général d'un service dynamique de page web et les principaux flux de données.

Ceci doit vous permettre de bien comprendre l'environnement client-serveur dans lequel le PHP s'inscrit.

Comprendre le rôle de chaque composant d'un site web Internet Intranet permet de mieux concevoir des applications optimisées, tout en tenant compte des possibilités et des limites liées aux techniques mises en oeuvre côté client et côté serveur.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Principes généraux - Rappel

Service statique d'une page HTML.

Un site internet statique est constitué d'un ensemble de pages html, qui sont de simples fichiers au format texte, avec une extension htm ou html.

Ces fichiers sont placés sur un serveur web, qui est une machine sur laquelle tourne un logiciel capable de répondre à des requêtes http (ex.: Apache).

Une requête http contient plusieurs informations, dont l'adresse (URL) du fichier demandé par le client.

Le rôle du serveur est donc de :

- accepter des requêtes de Clients distants
- décortiquer l'information contenue dans l'URL,
- rechercher le fichier demandé dans le système de fichiers de hôte (la machine sur laquelle le logiciel serveur tourne),
- lire ce fichier
- renvoyer le résultat de cette lecture au bon Client.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Sur la machine Cliente, un logiciel Client (IE/FF) permet de :

- saisir une URL ou valider un formulaire,
- envoyer la requête http correspondante,
- attendre la réponse du serveur,
- recevoir la chaîne de caractères renvoyée par le serveur,
- interpréter les balises html pour assurer
- la mise en forme du texte,
- l'affichage des éléments multimédia,
- la mise en page des éléments (texte,images,...)
- interpréter du code javascript pour:
- gérer des événements (*click, survol, sélection...*)
- agir sur les éléments de la page (*masquer, déplacer,...*)

Notes

PHP *les bases*

Service dynamique d'une page HTML.

Dans le cas d'un service dynamique, le Client ne demande plus une page html, mais l'exécution d'un script sur le serveur. C'est le résultat de cette exécution qui est ensuite envoyé au Client.

Ceci suppose que :

- le serveur possède un interpréteur de script adhoc
- les fichiers de scripts portent une extension permettant au serveur de les exécuter sur l'interpréteur adhoc.

En savoir +

le lien entre l'extension et l'interpréteur se fait dans httpd.config

L'avantage est que ces langages de script peuvent lire ou écrire dans des fichiers externes ou des bases de données, ce qui leur permet d'enregistrer des informations fournies par le Client et de générer à la volée des chaînes de caractères constituant (éventuellement) une page html complète, basée sur des données externes.

Notes

PHP *les bases*

En savoir +

Protocole HTTP

Soit une URL de la forme :

HTTP://www.bdphilia.fr:80/auteurs/index.html#haut?log=Toto&mdp=123

Le navigateur va interpréter cette URL de la façon suivante:

***HTTP://** - Protocole de communication utilisé (Ici Transfert Hypertexte).*

***www.bdphilia.net** - serveur à contacter (via un DNS)*

***:80** - port de connexion sur le serveur. (80 par défaut pour http)*

***/coucou/** - répertoire du document demandé.*

La racine du site est /

***index.html** -document demandé au serveur. (Par défaut index.html ou index.htm)*

***#haut** - repère d'affichage dans la page (ancree)*

***?log=Toto&mdp=123** -arguments sous forme nom=valeur séparés par des &.*

*Les **espaces** et certains caractères particuliers (comme & ou les caractères ascii 128 à 255) sont codé sous forme "%valeur ascii en hexadécimal du caractère"*

(par exemple ê = ascii 234 ; 234=EA en hexadécimal donc ê s'écrit %EA)

Une fois l'interprétation faite il envoie la requête au serveur sous forme :

GET/index.html#haut ?log=Toto&mdp=123 HTTP/1.1

*Accept: image/gif, image/jpeg, */**

Accept-language: fr,en-us

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Notes

PHP *les bases*

Host: monsite.fr

Connexion: Keep-Alive

D'une manière générale la requête suit la règle

Méthode URI Version-HTTP

En-tête général

En-tête de requête

En-tête d'entité

Ligne blanche obligatoire si le corps d'entité existe

Corps d'entité

- La ligne 1 indique la méthode que le client utilise, à quel document elle doit s'appliquer et quelle est la version HTTP qui est utilisée.

Les méthodes peuvent être GET, POST, HEAD, PUT, LINK, UNLINK, DELETE, OPTION et TRACE

- Les en-têtes généraux donnent des informations générales telles que la date ou le fait que la connexion doit ou ne doit pas être maintenue

- Les en-têtes de requête donnent les informations sur la configuration du client et sur les formats de documents acceptés

- Les en-têtes d'entité décrivent quel est le format des données qui sont envoyées (encodage, taille, type etc).

Le client ne les utilise que lorsqu'il envoie un corps d'entité c'est à dire lors des méthode POST ou PUT.

Le serveur va répondre en recherchant la ressource demandée, en créant une en-tête de serveur (header) et en joignant le document demandé, séparé des en-têtes par une ligne blanche :

HTTP/1.1 200 OK

Data: Mon, 06 Feb 2006 09:06:11 GMT

Notes

PHP *les bases*

Server: Apache/1.3.33 (Win32)

Last-Modified: Thu, 04 Jan 2006 14:11:45 GMT

ETag: "2f5cd-964-381e1bd6"

Accept-Ranges: bytes

Content-length: 327

Connection: close

Content-type: text/html

Ligne blanche de séparation header/document

Page html demandé (index.html)

D'une manière générale la requête suit la règle :

Version-HTTP - Code d'état - Explication

En-tête général

En-tête de réponse

En-tête d'entité

Ligne blanche obligatoire si le corps d'entité existe

Corps d'entité

- La ligne 1 indique la version d'HTTP qui est utilisée, le code de résultat de la requête (200 *requête* accomplie avec succès) et une explication humainement compréhensible du code d'état (OK).

- *Les en-têtes généraux donnent comme pour le client des informations générales telles que la date ou le fait que la connexion est ou n'est pas maintenue*

- *Les en-têtes de réponse donnent les informations sur la configuration du serveur, informe le client sur les*

Notes

PHP *les bases*

méthodes supportées et les autorisations nécessaires, ou demande au client de réessayer plus tard.

- Les en-têtes d'entité décrivent quel est le format des données qui sont envoyées (encodage, taille, type etc).

Attention

C'est le serveur au niveau des en-têtes d'entité qui gère le type de document envoyé. Le client ne vérifie pas la nature du document par rapport à sa demande. Il fait confiance au serveur et s'appuie sur les en-têtes d'entité du serveur pour afficher correctement le document.

Autrement dit le client peut demander une image gif et recevoir une image jpeg en remplacement de l'image gif sans que cela lui pose de problèmes ou demander une image .php et recevoir une image gif ou jpeg.

Notes

PHP *les bases*

Pages dynamiques pour le Web = +de possibilités !

- optimiser les développements (ex.:fiches produits)
- Interactivité avec les internautes (ex. chat)
- Traitement des formulaires
- Personnalisation du site (myTruc)
- Connexion aux bases de données
- Accès aux systèmes d'information existants
- Création de moteurs de recherche

PHP: simple, intuitif et complet(able)

- Transtypage intuitif
- Facile d'accès
- De plus en plus rationnel
- Nombreuses possibilités d'extensions (XML, PDF, Mail,...)
- Régulièrement mises à jour, augmentées et documentées
- Support Bases de données (MySQL, Oracle,LDAP...)

Notes

| |
|--|
| |
|--|

PHP *les bases*

En savoir +

Historique

1994: création de PHP/FI par Rasmus Lerdorf Cette première version est un ensemble de fonctions C, avec un analyseur syntaxique.

1995 Première version de PHP Cette version permettait déjà quelques traitements simples, et l'utilisation de formulaire.

1996 PHP/FI 2.0 Premières contributions externes Accès aux bases de données Support des formulaires HTML.

Création d'un groupe de développement.

1996-1998 réécriture complète du langage: version 3.0

Amélioration des performances. Croissance exponentielle du nombre d'utilisateurs et de contributeurs.

1998 – 2000: Zend Scripting Engine: Réécriture du moteur de scripts. Nouvelle amélioration des performances

très notable. Ajout de nombreux nouveaux modules.

2002 Zend Engine 2: Amélioration du support Objet

Outils de développement (Zend Studio) Bibliothèques de codes

2004 PHP 5

2008 PHP 6

Notes

PHP *les bases*

Les bases du langage

Ce chapitre vous introduit aux bases du langage PHP.

Vous y verrez la syntaxe classique ainsi que les différents éléments tel que variables, opérateurs et fonctions.

Grâce à une série d'exercices de base, vous devez savoir, en fin de chapitre, écrire des scripts simples en utilisant les composants du langage PHP.

Notes

PHP *les bases*

Formats et conventions

Fichiers

Les fichiers sont au format txt (texte simple), ont une extension .php et peuvent être édités avec notepad, PFE, Ultraedit, notepad++, etc.

Balises

Les balises indiquent au processeur PHP de traiter les instructions comme étant du PHP.

```
<?php ... . ?>
```

La balise de fin indique de repasser au HTML; donc on *peut* alterner (mais ce n'est pas forcément souhaitable d'un point de vue maintenabilité):

```
<?php Instruction ?>  
<br> bla bla <hr>  
<?php Instruction ?>
```

Notes

PHP *les bases*

En savoir +

Si la directive `short_open_tag` du `php.ini` est à ON, on peut omettre `php`.
Il faut la laisser à OFF si l'on veut assurer la portabilité du code.

Instructions

Les ligne d'instruction sont séparées par point-virgule (;)

En savoir +

CoEc. : Indentation et longueur de lignes :

Utilisez une indentation de 4 espaces, sans tabulations.

Il est recommandé que la longueur des lignes ne dépasse pas 75 à 85 caractères.

Il n'y a pas de règles fixes par rapport aux retours à la ligne, mais vous pouvez utiliser votre bon jugement.

En cas de doutes, vous pouvez toujours demander sur la liste de diffusion PEAR Quality Assurance (Ecrivez en anglais).

Notes

PHP *les bases*

Chaîne de Caractères

Une chaîne de caractères est délimitée par des guillemets simples ou doubles.

Exemple:

```
"ceci est une chaîne"
```

```
'ceci est une chaîne aussi'
```

En savoir +

Les guillemets doubles interprètent les variables.

Notes

PHP *les bases*

Sortie vers le client

L'envoi de la requête http de réponse se fait par :

```
echo
```

Exemple:

```
<?php
echo "bonjour le Monde !";
?>
```

En savoir +

Équivalent (pour 1 ligne et short_open_tags): <?= "Bonjour" ?>

Affichage d'objets non scalaires : print_r()

Sortie type shell (attention la fin de la chaîne doit suivre un retour à la ligne)

```
<?php
$var = <<<FIN
<html> . . .
FIN; //pas d'espace ni caractères avant FIN
echo $var;
?>
```

Notes

PHP *les bases*

Echappement

Le signe d'échappement (\) permet de ne pas interpréter un caractère:

```
<?php
echo "je dis \"bonjour le Monde !\"";
?>
```

Echappements particuliers

\n nouvelle ligne (newline)

\t tabulation

\r retour chariot (Carriage return)

\xdd caractère encodé Latin-1 en hexadecimal dd

\\ écrit le caractère \ (antislash)

\\$ écrit le caractère \$

Commentaire

On peut insérer des commentaires par

```
/*...
...
...*/
```

ou // en début de ligne.

Notes

PHP *les bases*

Bases du Langage

PHP est un langage de type impératif, permettant une programmation procédurale ou orientée objet.

Un langage impératif se caractérise principalement par une description des différentes instructions à exécuter pour parvenir au résultat voulu.

Un programme PHP sera donc une suite d'instructions, dont certaines portions pourront être conditionnelles ou répétitives.

Ces instructions réalisent des opérations élémentaires sur des données, stockées dans des emplacements mémoires spécifiques : les variables et les constantes.

En savoir +

`is_scalar()` ne considère pas les valeurs des types ressource comme scalaires, étant donné que les ressources sont des types abstraits, basés sur des entiers. Ceci est susceptible de changer.

Notes

PHP *les bases*

Ces variables peuvent contenir des données de type :

- scalaire
 - numériques (entiers, décimaux),
 - logiques (vrai/faux) ou
 - chaîne de caractères (texte).
- tableau
- objet
- ressource
 - pointeur de fichier
 - connexion bdd

Notes

PHP *les bases*

Les opérations effectuées utilisent :

- des opérateurs
 - arithmétiques (addition,...)
 - logiques (comparaison,...)
 - de chaîne (concaténation,...)
- des fonctions qui peuvent être :
 - natives
 - php (mathématiques, recherche, tri,...)
 - module externe (GD, PDF,...)
 - définies par l'utilisateur

Notes

PHP *les bases*

Variables

syntaxe

Les variables PHP doivent :

- commencer par un \$ suivi d'une lettre ou d'un underscore (_), suivis de lettres, chiffres ou _
- ne doivent pas être déclarées et
- ne sont **pas fortement typées**, elles peuvent contenir:
 - une chaîne de caractère,
 - une valeur numérique (entière ou décimale) ou
 - une valeur booléenne (true = 1 false = 0).
 - une valeur non scalaire
- sont **sensibles à la casse** (\$ville différent de \$Ville).

Assignment

```
$ville      = "Bruxelles"; (string)
$ZIP        = 85750; (integer ou int)
$taux       = 6.55957; (float ou double)
$reponse    = false; (boolean)
```

remarques :

- *echo d'une valeur false n'affiche rien.*
- *utiliser une variable inexistante déclenche une notice. (désactivable par @)*

Notes

PHP *les bases*

En savoir +

on peut détruire une variable avec unset().

Constantes

syntaxe

Une constante n'utilise pas de \$ et s'écrit par convention en majuscules.

Une tentative de modification de sa valeur génère une notice.

Assignment

Elle est définie par :

```
define(NOM,valeur);
```

En savoir +

Si le 3ème paramètre (optionnel) est TRUE, la constante sera insensible à la casse.

Notes

PHP les bases

Application

A partir des variables suivantes :

| | |
|----------|----------------|
| sNom | = Braive |
| sPrenom | = Xavier |
| sRue | = de la Sirène |
| iNo | = 2 |
| iCP | = 4000 |
| sVille | = Liège |
| fTotalHT | = 457.00 |

Créer le tableau HTML correspondant à :

| | |
|---------------------------|-------------------------------------|
| Client | Xavier Braive |
| Adresse | 2, rue de la Sirène 4000 - Liège |
| Montant de la Commande HT | 457 € |

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

```
<?php
$sNom      = 'Braive';
$sPrenom   = 'Xavier';
$sRue      = 'de la Sirène';
$iNo       = 2;
$iCP       = 4000;
$sVille    = 'Liège';
$fTotalHT  = 457.00;
?>
<html><body>
<table border="1">
<tr>
<th>Client</th><td><?php echo "$sPrenom
$sNom";?></td>
</tr>
<tr>
<th>Adresse</th><td>
<?php echo "$iNo,rue $sRue<br>$iCP-
$sVille";?></td>
</tr>
<tr>
<th>Montant de la Commande HT</th>
<td><?php echo $fTotalHT; ?> &euro;</td>
</tr>
```

Notes

P H P *les bases*

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

Référencement

On peut créer un lien entre deux variables :

```
$choix = &$ville;
```

le contenu de choix sera lié au contenu de ville
(et inversement!):

```
<?php
$ville = "Paris";
$choix = &$ville;
echo $choix;
$ville = "Tokyo";
echo $choix;
$choix = "Mexico";
echo $choix;
echo $ville;
?>
```

donnera : Paris Tokyo Mexico Mexico

Notes

PHP *les bases*

Transtypage explicite

On peut forcer le type d'une variable en affichant le type voulu entre parenthèses avant sa valeur.

int ou integer, string, float ou double, boolean.

```
<?php
$var = false;
echo "$var<br />"; //affiche rien
$var=(int) $var;
echo $var; //affiche 0
?>
```

Variables variables

Les variables variables permettent de définir un nom de variable de façon dynamique :

```
<?php
$ville1 = "Tokyo";
$ville2 = "Nagoya";
$ville3 = "Kobe";
$numero = 2;
echo ${ "ville$numero" };
?>
```

Donnera Nagoya.

Notes

PHP les bases

Tableaux (Listes)

Les tableaux sont des variables pouvant contenir plusieurs valeurs.

Ces valeurs sont repérées par un indice ou une clé (chaîne).

Un tableau pouvant contenir plusieurs valeurs est aussi appelé une liste.

Si chaque élément d'un tableau est lui-même un tableau, on aura un tableau à plusieurs dimensions.

Si les notions de liste et de tableau vous semblent abstraites, vous pouvez penser à l'analogie suivante :

Un tableau à 1 dimension peut être vu comme un classeur contenant des pages.

Les pages sont numérotées à partir de zéro.

Sur chacune des pages est inscrit le nom d'un client, par exemple.

ceci se traduit en PHP par :

```
$clients[0] = 'françois';
$clients[1] = 'xavier';
```

Notes

| |
|--|
| |
|--|

PHP les bases

OU

```
$clients[] = 'françois';
$clients[] = 'xavier';
```

OU

```
$clients = array('françois','xavier');
```

Lecture d'un élément de tableau :

```
echo $clients[1]
//affiche xavier
```

Dans certains cas, l'utilisation des indices peut être laborieuse, et impose le respect de conventions.

Si nous décidons qu'une personne est définie par son nom puis son prénom, nous pouvons écrire :

```
$personne = array('Plégades','François');
```

Ceci impose de se souvenir de l'ordre des données.

Notes

| |
|--|
| |
|--|

PHP *les bases*

L'utilisation de clés (ou étiquettes) permet une manipulation plus aisée:

```
$personne['nom'] = 'Plégades';  
$personne['prenom'] = 'François';
```

OU

```
$personne = array(  
    'nom'=>'Plégades',  
    'prenom'=>'François'  
);
```

permet de récupérer le nom par :

```
echo $personne['nom'];
```

Attention : `$personne[0]` n'existe pas !

Notes

PHP *les bases*

Tableaux multidimensionnels

Dans notre tableau Clients, nous pouvons remplacer les chaînes par des tableaux, à clés et/ou à indices :

```
$personnel = array(  
    'nom'=>'Plégades',  
    'prenom'=>'François'  
);  
$personne2 = array(  
    'nom'=>'Braive',  
    'prenom'=>'Xavier'  
);
```

```
$clients = array( $personnel, $personne2);
```

Pour connaître le prénom du deuxième client :

```
echo $clients[1]['prenom'];
```

Notes

PHP *les bases*

Dans le cas d'un tableau purement à indices :

```
$personnel = array('Plégades','François');  
$personne2 = array('Braive','Xavier');
```

```
$clients = array( $personnel,$personne2);
```

Pour connaître le prénom du deuxième client :

```
echo $clients[1][1];
```

Notes

PHP *les bases*

Dans le cas d'un tableau purement à clés :

```
$personnel = array(  
    'nom'=>'Plégades',  
    'prenom'=>'François'  
);  
$personne2 = array(  
    'nom'=>'Braive',  
    'prenom'=>'Xavier'  
);
```

```
$clients = array(  
    'CL001'=>$personnel,  
    'CL002'=>$personne2  
);
```

Pour connaître le prénom du deuxième client :

```
echo $clients['CL002']['prenom'];
```

Notes

PHP *les bases*

Portée des variables

Les variables en PHP ont une portée globale au niveau du script.

Cela signifie qu'elles existent à partir du moment où elles sont déclarées ou assignées jusqu'à la fin du script.

La fin du script détruit toutes les variables et libère leur espace mémoire.

Si les variables sont créées à l'intérieur d'une fonction ou d'un objet leur portée est locale à la fonction ou à l'objet.

Elles ne sont accessibles que dans la fonction ou l'objet et sont détruites à la fin de la fonction ou de l'objet.

Les variables globales du script ne sont pas accessibles depuis l'intérieur d'une fonction ou d'un objet.

PHP a créé des variables systèmes qui sont accessibles de partout : dans le script, dans les fonctions ou dans les objets. Ce sont des **superglobales**.

Notes

PHP *les bases*

Ce sont en général des tableaux qui permettent de récupérer les valeurs d'environnement (passées dans les URL, stockées en sessions ou cookies, informations sur le Client, etc.)

Passage de paramètres http

Les deux méthodes principales sont GET et POST.

GET

Les paramètres passés en GET transitent par l'entête de requête HTTP.

Ils sont visibles dans la barre d'adresse du navigateur et peuvent être limités en longueur.

Ils sont récupérés dans le tableau à clé \$_GET.

```
http://www.auppd.net/fichClient.php?id=CL002
```

L'appel peut se faire directement en tapant l'URL (rare) ou par un hyperlien dans la page, ou un formulaire GET.

```
<?php
//fichClient.php
echo $_GET['id'];
?>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

POST

Les paramètres passés en POST transitent par le corps de requête HTTP.

Ils sont invisibles dans la barre d'adresse du navigateur et acceptent de grandes quantités de données.

Ils sont récupérés dans le tableau à clé \$_POST.

```
<form action="fichClient2.php" method="post">
  <input type="text" name="id">
  . . .
</form>
```

```
<?php
//ficheClient.php
echo $_POST['id'];
?>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

FILE

Les paramètres passés en FILE transmettent un fichier, qui est stocké dans un répertoire temporaire puis détruit à la fin du script.

```
<form enctype = "multipart/form-data"
action= "do_ajClient.php"method = "post">
<input type="file" name="fic" >
<input type="submit">
</form>

-----

<?
//do_ajClient.php
print_r($_FILES['fic']);
move_uploaded_file
($_FILES['fic']['tmp_name'], 'c:\photo1.jpg');
?>
```

Donnera :

```
Array([fic]=> Array(
[name]          => P1010685.JPG
[type]          => image/jpeg
[tmp_name]      => C:\EASYPHP\\tmp\php213.tmp
[error]         => 0
[size]          => 62798)
```

Notes

PHP *les bases*

Application

Fichier formLivre.php

-Créer un formulaire permettant de sélectionner :

- une référence Livre par saisie
- une langue (français/anglais) par boutons radio
- une couleur de fond dans une liste déroulante

Fichier doLivre.php

- Créer un tableau \$livres, pour lequel chaque livre est repéré par une référence de type BD000001, et contient les informations suivantes:

- titre
- auteur
- editeur
- prix
- stock

- Afficher sous forme de tableau HTML les informations du Livre dont la référence est passée par formulaire.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé formLivre.php

```
<html><body>
<form action="doLivre.php" method = "post">
  <table border = "2" bgcolor = "ffffcc">
    <tr><td>Référence</td><td><input type =
      "text" value = "bd000001"
      name="c_ref"></td>
    </tr>
    <tr>
      <td>Langue :</td><td><input type =
        "radio" name = "c_lang" value="fr"
        checked="checked">Français<br>
        <input type = "radio" name = "c_lang"
        value="en">English</td>
    </tr>
    <tr><td>Fond :</td><td><select
      name="c_fond">
        <option value="FFFFCC">j jaune
        pâle</option>
        <option value="FFCCCC">rose</option>
      </select></td>
    </tr>
  </table>
  <input type="submit">
</form>
</body></html>
```

Notes

PHP *les bases*

Corrigé doLivres.php

```
<?php
$libelles = array(
    'l_titre'=>array('fr'=>'titre','en'=>'title'),
    'l_auteur'=>array('fr'=>'auteur','en'=>'author'),
    'l_editeur'=>array('fr'=>'éditeur','en'=>'editor'),
    'l_prix'=>array('fr'=>'prix','en'=>'price')
);

$livres = array(
    'bd000001'=>array(
        'titre'=>'Le Lotus Bleu',
        'auteur'=>'Hergé',
        'editeur'=>'Casterman',
        'prix'=>5.6
    ),
    'bd000002'=>array(
        'titre'=>'Gaston Lagaffe',
        'auteur'=>'Franquin',
        'editeur'=>'Dupuis',
        'prix'=>9.6
    )
);

$ref      = $_POST['c_ref'];
$lang     = $_POST['c_lang'];
$fond     = $_POST['c_fond'];
?>
[suite...]
```

Notes

PHP *les bases*

```
<html>
<body bgcolor="<?php echo $fond;?>">
<table border="1">
  <tr>
    <th><?php echo
$libelles['l_titre'][$lang];?></th>
    <td><?php echo
$livres[$ref]['titre'];?></td>
  </tr>
  <tr>
    <th><?php echo
$libelles['l_auteur'][$lang];?></th>
    <td><?php echo
$livres[$ref]['auteur'];?></td>
  </tr>
  <tr>
    <th><?php echo
$libelles['l_editeur'][$lang];?></th>
    <td><?php echo
$livres[$ref]['editeur'];?></td>
  </tr>
  <tr>
    <th><?php echo
$libelles['l_prix'][$lang];?></th>
    <td><?php echo
$livres[$ref]['prix'];?>&euro;</td>
  </tr>
</table>
</body>
</html>
```

Notes

PHP *les bases*

Opérateurs

Arithmétiques:

+, -, *, /, % (modulo)

exemple:

```
<?= (20%7) ?>
```

donne 6

Incrémentation:

`$x++;` est équivalent à `$x = $x + 1;`

`$x--;` est équivalent à `$x = $x - 1;`

`$x +=10;` est équivalent à `$x = $x + 10;`

`$x (op)=val` est équivalent à `$x=$x (op) val`

Chaîne (concaténation):

La concaténation permet de coller deux chaînes:

`"bon" . "jour"`

On peut utiliser l'autoconcaténation (`.=`)

```
$mot="Bon";
```

```
$mot .= "jour"; // $mot contient "Bonjour"
```

Notes

PHP *les bases*

Logique (Comparaison)

< plus petit
<= plus petit ou égal
> plus grand
>= plus grand ou égal
== comparaison large (égalité de valeurs)
=== comparaison stricte (valeurs et types) !!
!= différent de

Combinaisons Logiques

| | | |
|-------------|---------|---|
| \$a and \$b | Vrai si | \$a ET \$b sont vrais. (&&) |
| \$a or \$b | Vrai si | \$a OU \$b est vrai () |
| \$a xor \$b | Vrai si | \$a OU \$b est vrai, mais pas les deux. |
| !\$a | Vrai si | \$a est faux. |

Notes

PHP *les bases*

Application

Sur base de l'exercice précédent, afficher une ligne supplémentaire pour la TVA (5.5%) et pour le TTC.

Notes

PHP *les bases*

Corrigé

```
.....
$lang    = $_POST['c_lang'];
$fond    = $_POST['c_fond'];
define(TVA_R,5.5);
$ht = $livres[$ref]['prix'];
$tva = $ht * TVA/100;
$ttc = $ht + $tva;
?>
.....
    <tr>
        <th><?php echo
$libelles['l_prix'][$lang];?></th>
        <td><?php echo
$livres[$ref]['prix'];?>&euro;</td>
        <th><?php echo
$libelles['l_tva'][$lang];?></th>
        <td><?php echo $tva;?>&euro;</td>
        <th><?php echo
$libelles['l_ttc'][$lang];?></th>
        <td><?php echo $ttc;?>&euro;</td>
    </tr>
</table>
.....
?>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Les contrôle de flux (conditionnels et boucles)

Cette partie doit vous permettre d'utiliser les contrôles conditionnels pour orienter vos scripts en fonctions des contextes et d'utiliser les différents types de boucles pour automatiser les étapes répétitives en interrogation ou affichage de données.

A la fin de ce chapitre vous connaîtrez les syntaxes et possibilités du PHP au niveau du contrôle de flux.

Notes

PHP *les bases*

Contrôles de Flux

Permettent de modifier le cours d'un programme en fonction de certaines conditions.

Les Tests

If

Ces tests utilisent le résultat de comparaisons logiques:

```
if (condition) {  
    instructions...  
}
```

On peut prévoir des instructions alternatives:

```
if ($b>$c) {  
    $msg='oui';  
}else{  
    $msg='non';  
}
```

En savoir +

On peut aussi écrire :

```
$msg= ($b>$c)?'oui':'non';
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

On peut prévoir des instructions alternatives conditionnelles:

```
if ($a > $b) {  
    echo "a est plus grand que b";  
} elseif ($a == $b) {  
    echo "a est égal à b";  
} else {  
    echo "a est plus petit que b";  
}
```

Switch

Dans certains cas, cette structure est plus efficace que des imbrications de si. Le break force la sortie sans passer par le traitement par défaut:

```
switch($variable) {  
    case "valeur1":  
        instructions1;  
        break;  
  
    default:  
        instructions par défaut;  
}
```

Notes

PHP *les bases*

En savoir +

CoEc. : Les instructions de contrôle doivent avoir un espace entre le mot clé de l'instruction et la parenthèse ouvrante, afin de les distinguer des appels de fonctions.

Il est vivement recommandé de toujours utiliser des accolades, même dans les situations où elles sont techniquement optionnelles.

Leur présence augmente la lisibilité du code et réduit le risque d'erreur logique lors de l'ajout de nouvelles lignes de code.

```
if ((condition1) || (condition2)) {  
    action1;  
} elseif ((condition3) && (condition4)) {  
    action2;  
} else {  
    defaultaction;  
}  
switch (condition) {  
case 1:  
    action1;  
    break;  
  
default:  
    defaultaction;  
    break;  
}
```

Notes

PHP *les bases*

Application

Dans l'exercice précédent, prévoir une case à cocher "j'accepte les conditions générales".

Si l'utilisateur ne coche pas la case, il ne voit pas le résultat et reçoit un message "vous n'avez pas accepté les CGV".

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

```
....
$tva = $ht * TVA_R/100;
$ttc = $ht + $tva;
if (@$_POST['c_cgv']=='ok') {
?>
<html>

....
</tr>
</table>
<?php
}else{
    echo "vous n'avez pas accepté les CGV";
}
?>
</body>
</html>
```

Notes

P H P *les bases*

Traitements en Boucle

Les traitements en boucles permettent d'exécuter des instructions de manière répétitive. Il en existe plusieurs types:

While (tant que condition vraie)

Peut ne pas être exécutée (si la condition n'est pas vérifiée au départ)

```
while (condition) {  
    instructions;  
}
```

exemple:

```
$i = 0;  
while ($i<=20){  
    echo ($i); $i++;  
}
```

si \$i = 21 au départ, rien ne sera exécuté

Notes

PHP *les bases*

Do While

La condition est évaluée en fin de boucle (donc exécutée au moins une fois)

```
$i = 0;
do{
echo ($i);
$i++;
} while ($i <= 20);
```

si \$i = 21 au départ, la boucle sera quand même exécutée 1 fois.

For

La boucle FOR s'exécute un nombre déterminé de fois.

```
for ($i = 10; $i <= 20; $i++) {
print $i . "<br />";
}
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Foreach

Il est souvent nécessaire de récupérer via une boucle le contenu d'un tableau. La boucle foreach parcourt automatiquement la totalité d'un tableau, en récupérant à chaque pas la clé (éventuellement) et la valeur .

```
foreach ($tableau as $valeur){  
    instructions  
}
```

OU

```
foreach ($tableau as $cle => $valeur){  
    instructions  
}
```

Exemple

```
$tab=array(  
    'nom'=>'Braive',  
    'prenom'=>'Xavier');  
foreach ($tab as $cle=>$val){  
    echo "$cle contient $val.<hr />";  
}
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

APPLICATION

Sur base de l'exercice précédent, si la référence rentrée est 'inventaire', afficher la totalité du stock et sa valeur sous forme de tableau, avec le sous-total pour chaque titre.

Option : alterner les couleurs de lignes pour améliorer la lisibilité.

| Titre | Stock | PU | Sous Total |
|----------------|-------|-----|------------|
| Le Lotus Bleu | 10 | 5.6 | 56 |
| Gaston Lagaffe | 2 | 9.6 | 19.2 |
| Natacha | 5 | 100 | 500 |
| Total | | | 575.2 |

Notes

PHP *les bases*

Corrigé

```
<?php
$libelles = array(
    'l_titre'=>array('fr'=>'titre','en'=>'title'),
    [...]
);

$livres = array(
    'bd000001'=>array('titre'=>'Le Lotus
Bleu','auteur'=>'Hergé','editeur'=>'Casterman','prix
'=>5.6,'stock'=>10),
    'bd000002'=>array('titre'=>'Gaston
Lagaffe','auteur'=>'Franquin','editeur'=>'Dupuis','p
rix'=>9.6,'stock'=>2),
    [...]
);

$ref      = $_POST['c_ref'];
$lang     = $_POST['c_lang'];
$fond     = $_POST['c_fond'];
define('TVA_R',5.5);
switch($ref){
    case 'inventaire':
        $msg      = '';
        $total    = 0;
        foreach ($livres as $livre) {
            $sstot =      $livre['stock'] *
                        $livre['prix'];
            $total += $sstot;
        }
    }
}
```

Notes

PHP *les bases*

```
        $msg .= "<tr>
        <td>{$livre['titre']}

## Notes


```

PHP *les bases*

```
<tr>
    <th>{$libelles['l_ttc'][$lang]}</th>
    <td>$ttc&euro;</td>
</tr>";
}
?>

<html>
<body bgcolor=<?=$fond?>>
<table border="1">
<?php
if(@$_POST['c_cgv']=='ok'){
    echo $msg;
}else{
    echo
"<tr><th>vous n'avez pas accepté les CGV<th></tr>";
}
?>
</table>
</body>
</html>
```

Notes

PHP *les bases*

Fonctions

Une fonction isole une partie du code pour le rendre réutilisable. Elle utilise des arguments qui lui sont transmis lors de son appel, et peut renvoyer un résultat avec return.

Syntaxe

```
<?php
function euros ($francs){
    $resultat = $francs / 6.55957;
    return($resultat);
}
echo euros(10);
echo euros(20);
?>
```

Une fonction peut accepter plusieurs arguments, séparés par une virgule.

On peut définir une valeur par défaut pour les arguments. Cette valeur sera utilisée si l'argument n'est pas passé.

Notes

PHP *les bases*

Portée

Attention: les **variables définies dans une fonction ont une portée locale**. (pas accessibles en dehors de la fonction).

Pour pouvoir utiliser les variables externes à la fonction il faut les déclarer dans la fonction comme "global" ou les récupérer dans la variable superglobale \$GLOBALS.

Exemple

```
<?php
$taux = 6.55957;
function euros ($francs)
{
    global $taux;
    //$taux = &$GLOBALS['taux'];
    //$taux = $GLOBALS['taux'];
    $resultat = $francs / $taux;
    return($resultat);
}
echo euros(10);
?>
```

Notes

PHP *les bases*

En savoir +

CoEc. :

Appels de Fonctions

Les fonctions doivent être appelées sans espace entre le nom de la fonction, la parenthèse ouvrante, et le premier paramètre ; avec un espace entre la virgule et chaque paramètre et aucun espace entre le dernier paramètre, la parenthèse fermante et le point virgule.

```
$var = truc($un, $deux, $trois);
```

Comme montré ci-dessus, il doit y avoir un espace de chaque côté du signe égal utilisé pour affecter la valeur de retour de la fonction à une variable.

Dans le cas d'un bloc d'instructions similaires, des espaces supplémentaires peuvent être ajoutés pour améliorer la lisibilité :

```
$short      = truc($un);  
$long_variable = truc($deux);
```

Définitions des fonctions

La déclaration des fonctions respecte l'indentation classique des accolades :

```
function truc($arg1, $arg2 = "")  
{  
    if (condition) {  
        instruction;  
    }  
    return $val;  
}
```

Les arguments possédant des valeurs par défaut vont à la fin de la liste.

Notes

PHP *les bases*

Application

Ecrire une fonction qui calcule un prix TTC sur base de 2 arguments:

- le prix HT donné via l'URL
- le taux (5.5 ou 19.6 %) fixé via l'URL
- option : améliorer pour utiliser un code tva (1 ou 2).
- option : la fonction renvoie le montant de la TVA et le TTC.

Notes

PHP *les bases*

Corrigé

```
<?php
function ttc ($ht, $taux)
{
    $resultat = $ht * (1+ $taux / 100);
    return($resultat);
}
$ht = $_GET['ht'];
$tva = $_GET['tva'];
$ttc = ttc($ht,$tva);
?>
<html>
<head><title>calcul ttc</title></head>
<body>
<?php echo $ht;?> euros hors taxe donnent
avec une TVA de <?php echo $tva;?> % <?php
echo $ttc;?> euros TTC.
</body>
</html>
```

Notes

PHP *les bases*

Corrigé 2

```
<?php
$tva[1]=5.5;
$tva[2]=19.6;
function ttc($ht, $codeTva=2)
{
    global $tva;
    $mTva = $ht * $tva[$codeTva] / 100;
    $mTtc = $ht+$mTva;
    $resultat = array('tva'=>$mTva, 'ttc'=>$mTtc);
    return($resultat);
}

$ht      = $_GET['ht'];
$taux    = $_GET['tva'];
$prix    = ttc($ht,$taux);
?>
<html>
<body>
HT:<?php echo $ht;?><br>
TVA:<?php echo $prix['tva'];?><hr>
TTC:<?php echo $prix['ttc'];?>
</body>
</html>
```

Notes

PHP *les bases*

Librairies et inclusion de fichier

Une librairie est un fichier php qui ne contient que des fonctions. Son extension est libre (ex. : .lib). Par sécurité, il vaut mieux terminer le nom par ".php" afin que le fichier soit toujours interprété par le parseur PHP.

On peut y faire appel soit par la fonction **require()** :

```
<?php
require("ma_librairie.lib.php");
?>
```

Si le fichier est absent, une erreur fatale surviendra.

soit par la fonction **include()** :

```
<?php
include("ma_librairie.lib.php");
?>
```

Ces fonctions impliquent l'inclusion à chaque appel (les variables sont remises à zéro). Si l'on ne désire qu'une seule inclusion on peut utiliser **require_once()** ou **include_once()**

Application

Externaliser la fonction TTC dans un fichier f_compta.lib.php

Notes

PHP les bases

Corrigé

Fichier f_compta.lib.php:

```
<?php
function ttc ($ht, $taux)
{
    $resultat = $ht * (1+ $taux / 100);
    return($resultat);
}
?>
```

Fichier principal:

```
<?php
require ("f_compta.lib.php");
$ht      = $_GET["ht"];
$tva      = $_GET["tva"];
$ttc      = ttc($ht,$tva);
?>
<html>
    <head><title>calcul ttc</title></head>
    <body>
        <?php echo $ht;?> euros hors taxe donnent
        avec une TVA de <?php echo $tva;?> % <?php
        echo $ttc;?> euros TTC.
    </body>
</html>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Application

Objectif: optimiser l'utilisation du HTML par l'emploi de fonctions de formatage en librairie externe `l_html.lib.php`.

`f_tete` crée l'entête html (`<html><body>`), avec en paramètres le titre (title) la couleur de fond et de texte.

`f_pied` crée la fin de fichier (`</body></html>`).

`f_br` crée un retour `
`

`f_titre` crée un titre h, avec en paramètre le texte du titre et le niveau.

`f_texte` affiche le texte en Arial, taille 3, avec un paramètre d'alignement 0,1 ou 2 pour gauche, centré, droite.

`f_tablo` crée un tableau html à partir d'une variable tableau.

Option : créer des fonctions permettant de générer un formulaire.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

Fichier l_html.lib.php :

```
<?php
$ali = array("left", "center", "right");

function f_tete($titre, $bg="FFFFFF", $fg="000000")
{
    echo "<html>
<head>\n<title>$titre</title>\n</head>
<body bgcolor=\"#$bg\" text=\"#$fg\">\n";
}
function f_titre($texte, $niv=2)
{
    echo "<h$niv> $texte</h$niv>\n";
}
function f_texte($texte, $cal=0, $size=3, $pol="arial")
{
    global $ali;
    echo "<p align=\"$ali[$cal]\">
<font size=\"$size\" face=\"$pol\">$texte</font>
</p>\n ";
}
function f_br()
{
    echo "<br />\n";
}
function f_pied($menu='')
{

```

Notes

PHP *les bases*

```
include($menu);
echo "</body>\n</html> ";
}

function f_tablo
($array,$border=1,$pair='FFFFFF',$impair='CCCCC',
$cali=1)
{
    global $ali;
    $coul=array($pair,$impair);
    $i=0;
    $str = "<table border=\"\$bord\"
align=\"\$ali[$cali]\">";
    foreach($array as $val){
        $str .= '<tr bgcolor="'. $coul[$i].' ">';
        foreach ($val as $elem){
            $str .= "<td>$elem</td>";
        }
        $str .= '</tr>';
        $i=!$i;
    }
    $str .= '</table>';
    echo ($str);
}

function
f_startForm($action='', $method='POST', $cali){
    global $form, $ali;
    $form = '<form action="'. $action. '"
method="'. $method. '">
<table align="'. $ali[$cali].' ">';
```

Notes

PHP *les bases*

```
}  
function  
f_ajChamp($type='T',$name='', $lib='', $value='', $checked='', $ev='ev') {  
    global $form;  
    $chk = array('', 'checked="checked"');  
    $sel = array('', 'selected="selected"');  
    switch ($type) {  
        case 'T':  
            $form .= "<tr><td>$lib</td><td><input  
type=\"text\" name=\"$name\"  
value=\"$value\"></td></tr>\n";  
            break;  
  
        case 'H':  
            $form .= "<input type=\"hidden\"  
name=\"$name\" value=\"$value\">";  
            break;  
  
        case 'P':  
            $form .= "<tr><td>$lib</td><td><input  
type=\"password\" name=\"$name\"></td></tr>\n";  
            break;  
  
        case 'R':  
            $form .= "<tr><td>$lib</td><td>";  
            foreach($value as $svalue=>$slib ) {  
                $test=($svalue==$checked);  
                $form .= "<input type=\"radio\"
```

Notes

PHP *les bases*

```
name=\"$name\" value=\"$svalue\"
$chk[$test]>$slib<br>";
    }
    $form .= "</td></tr>\n";
    break;

    case 'S':
        $form .= "<tr><td>$lib</td><td><select
name=\"$name\">
    <option
value=\"\">sélectionnez</option>\n";
        foreach($value as $svalue=>$slib ) {
            if($ev=='vv') {
                $svalue=$slib;
            }
            $test=($svalue==$checked);
            $form .= "<option value=\"$svalue\"
$sel[$test]>$slib</option>\n";
        }
        $form .= "</select></td></tr>\n";
        break;

    case 'A':
        $form .=
"<tr><td>$lib</td><td><textarea
name=\"$name\">$value</textarea></td></tr>\n";
        break;
}
}
```

Notes

PHP *les bases*

```
}

function f_affForm($libAct='OK') {
    global $form;
    $form .= "<tr><td colspan=2 align=right><input
type=\"submit\" name=modeAction
value=$libAct></td></tr>
    </table>";
    echo $form;
}

?>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Les fonctions et variables internes du PHP

Ce chapitre doit vous permettre d'aborder les grandes fonctions du PHP ainsi que les variables mises à disposition par le langage.

Vous y verrez la syntaxe des principales fonctions mathématiques, de texte et de date

.

Vous verrez aussi les variables prédéfinies par PHP ainsi que celles mises à disposition par le serveur (apache).

En fin de chapitre vous devez connaître les possibilités offertes par le PHP pour gérer les calculs, l'écriture et les dates.

(Il ne s'agit pas de connaître par coeur ces fonctions mais de savoir quelles possibilités sont mises à disposition pour écrire des scripts en PHP et ou retrouver leur syntaxe dans l'Aide)

Notes

| |
|--|
| |
|--|

PHP *les bases*

Fonctions PHP

Mathématiques

Abs(\$nombre) -- Valeur absolue

Abs(-10) => 10

Acos(\$valeur) -- arc cosinus (radians) $-1 \leq \$valeur \leq 1$

acos(1) => 0 (angle nul)

Asin(\$valeur) -- arc sinus (radians) $-1 \leq \$valeur \leq 1$

asin(1) => 1.5707963267949 (soit $\pi / 2$, ou 90°)

Atan(\$valeur) -- arc tangente

Cos(\$angle) -- cosinus \$angle en radians

Sin(\$angle) -- Sinus \$angle en radians

Tan(\$angle) -- Tangente \$angle en radians

deg2rad(\$angle) -- Convertit les degrés en radians

deg2rad(180) => 3.1415926535898

rad2deg(\$angle) -- Convertit de radians en degrés

DecHex(\$nombre) -- Convertit de décimal en hexadécimal

Notes

| |
|--|
| |
|--|

PHP *les bases*

`dechex(255) => FF`

hexdec(\$nombre) -- Convertit de hexadécimal en décimal

Ceil(\$nombre) -- Arrondit au nombre supérieur

`Ceil(2.8) => 3`

Floor(\$nombre) -- Arrondit à l'entier inférieur

`Floor(2.8) => 2`

round(\$nombre, précision) -- Arrondit.

`round(1.23456789,3) => 1.235`

max(\$a,\$b,\$c,...) -- La plus grande valeur d'une liste à virgule ou d'un tableau. `Max(2,5) => 5`

min(\$a,\$b,\$c,...) -- La plus petite valeur. `min(2,5) => 2`

mt_rand()-- Génère une valeur aléatoire

(4x+rapide que rand).

`mt_rand()` => de 0 à maxrand (nombre aléatoire le plus grand possible)

`mt_rand(10,15)` => de 10 à 15

pow(\$nombre,puissance) --Puissance `pow(2,3)` ($2^3 => 8$)

Notes

PHP *les bases*

Sqrt(\$nombre) -- Racine carrée. Sqrt(9) => 3

number_format(\$nombre,précision,separateur_decimal,séparateur_millier)

-- formate pour l'affichage avec comme arguments
nombre,nombre de chiffres après la virgule, séparateur
décimal,séparateur milliers.

```
<?php
$nombre = 1234.5678;
$e_number = number_format($nombre);
// 1,234.5678
$nfrancais = number_format($nombre,2,',','');
// 1 234,56
?>
```

Fonctions de chaîne:

chop(\$chaîne) ou **rtrim(\$chaîne)**-- Enlève les espaces de fin de chaîne.

ltrim(\$chaîne) -- idem en début de chaîne.

trim(\$chaîne)-- idem en début et fin de chaîne

chr(\$ascii) -- Caractère sur code ASCII. [Chr(69) => E]

Notes

PHP *les bases*

ord (\$char)-- Valeur ASCII du caractère.[ord(E) => 69]

crypt (\$chaine,chaine_cryptée)-- Chiffre une chaîne avec un DES.

Note : Il n'existe pas de fonction de déchiffrement, car la fonction crypt() utilise un algorithme à un seul sens (injection).

```
Crypt ("motdepasse")
```

=> d.XaEFv1z4fEA Crypté aléatoirement

```
Crypt ("motdepasse", "d.XaEFv1z4fEA")
```

=> Crypte selon la clé générée la première fois, donc comparaison possible.

explode("separateur",\$chaine) -- Scinde une chaîne en morceaux, grâce à un délimiteur, et crée un tableau.

```
$info = explode("#", "france#33");  
echo $info[1]; => 33
```

implode("separateur",\$tableau) ou join -- Regroupe tous les éléments d'un tableau dans une chaîne, avec une chaîne de jointure.

htmlentities(\$chaine) -- Convertit tous les caractères spéciaux en entité HTML.

Notes

PHP *les bases*

print(\$chaine) ou **echo** -- Affiche une chaîne.

Comme echo, mais c'est une fonction, qui renverra 0 ou 1 suivant succès ou échec.

printf -- Affiche une chaîne formatée. **sprintf** -- retourne une chaîne formatée.

La chaîne de format est composée de 0 ou plus directives: généralement des caractères qui sont copiés tels quels (hormis %), et des spécifications, chacune d'elle disposant de son propre paramètre.

Cela s'applique aussi à sprintf(). Chaque conversion consiste en un signe pourcentage (%), suivi d'un ou plusieurs éléments parmi ceux-ci:

1. Une option de remplissage, qui indique quel caractère sera utilisé pour le remplissage jusqu'à la taille finale de la chaîne. Le caractère de remplissage peut être un espace ou le caractère zéro (0).). La valeur par défaut est l'espace. Une autre valeur peut être spécifiée en la préfixant par un guillemet simple (').
2. Un spécificateur d'alignement qui indique si le résultat

Notes

| |
|--|
| |
|--|

PHP *les bases*

doit être aligné à gauche ou à droite. Par défaut, le résultat est aligné à gauche. Le caractère - fera que le résultat est aligné à droite.

3. Argument optionnel, spécificateur de taille indique le nombre minimum de caractères que la conversion devrait retourner.

4. Argument optionnel, spécificateur de précision indique le nombre de décimales utilisé pour afficher un nombre à virgule flottante. Cette option n'a d'effet que sur les nombres à virgule de type float.

5. Un spécificateur de type qui indique le type avec lequel l'argument sera traité. Plusieurs types possibles :

Les types possibles sont :

% - un caractère de pourcentage littéral. Aucun argument n'est nécessaire.

b - l'argument est traité comme un entier, et présenté comme un nombre binaire.

c - l'argument est traité comme un entier, et présenté

Notes

| |
|--|
| |
|--|

PHP *les bases*

comme le caractère de code ASCII correspondant.

d - l'argument est traité comme un entier, et présenté comme un nombre décimal signé.

u - l'argument est traité comme un entier, et présenté comme un nombre décimal non signé.

f - l'argument est traité comme un nombre à virgule flottante

, et présenté comme un nombre à virgule flottante.

o - l'argument est traité comme un entier, et présenté comme un nombre octal.

s - l'argument est traité et présenté comme une chaîne de caractères.

x - l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en minuscules).

X - l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en majuscules).

Notes

| |
|--|
| |
|--|

PHP *les bases*

Exemples:

```
<?php
$arbre = "Palmier";
$nombre = 5;
$format = "Il y a %d singes dans le %s";
printf($format,$nombre , $arbre);

$jour = 1;
$mois = 4;
$annee = 2003;
$isodate = sprintf("%04d-%02d-%02d", $annee,
$mois, $jour);

$r = "15";
$v = "128";
$b = "16";
$scoul_hexa = sprintf("#%02X%02X%02X",$r,$v,$b);

$format = "%01.2f &euro;";
$money1 = 68.75;
$money2 = 54.35;
$money = $money1 + $money2;
// affichera "123.1";
$formate = sprintf($format, $money);
// affichera "123.10 •"
?>
```

Notes

P H P *les bases*

addslashes(\$string) -- échape avec des slashes les caractères spéciaux dans les chaînes de caractères.

stripSlashes(\$string) -- Enlève les slashes ajoutés par la fonction addslashes()

strlen(\$string) -- Retourne la longueur de la chaîne.

strpos(\$string, char) -- Trouve la première position d'un caractère ou d'une chaîne de caractères *char* dans une chaîne *\$string*.

Attention : sensible à la casse

```
echo strpos("abcd", "cd"); //donne 2
```

strstr(\$string, char) -- Renvoie la chaîne à partir de la première occurrence d'un caractère ou d'une chaîne de caractères *char*

Attention : sensible à la casse

```
$email = 'courrier@cmbp.info';  
$domaine = strstr($email, '@');  
print $domaine; //donne @cmbp.info
```

strtolower(\$string) -- Met tous les caractères en minuscules.

Notes

PHP *les bases*

strtoupper(\$string) -- Met tous les caractères en majuscules.

ucfirst(\$string) -- Force le premier caractère d'une chaîne en majuscule.

ucwords(\$string) -- Force le premier caractère de chaque mot d'une chaîne en majuscule.

str_replace(rech, rempl, \$string, \$nb) -- Remplace toutes les occurrences *rech* d'une chaîne *\$string* par *rempl* .

- En PHP 4.0.5 et plus récent, tous les paramètres de `str_replace()` peuvent être des tableaux

Si *rech* et *rempl* sont des tableaux, alors `str_replace()` prendra une valeur de chaque tableau, et l'utilisera pour faire le remplacement dans *\$string*.

Si *rempl* a moins de valeurs que *rech* , alors une chaîne vide sera utilisée pour effectuer les remplacements.

Si *rech* est un tableau et que *rempl* est une chaîne, alors la chaîne de remplacement sera utilisée pour chaque élément de *rech* .

Notes

| |
|--|
| |
|--|

PHP *les bases*

Depuis PHP 5.0, le nombre de valeurs de *rech* trouvées et remplacées est retourné dans un 4ème paramètre *\$nb* passé par référence.

```
<?php
$bodytag = str_replace("#000000", "black",
"<body text=#000000>");

$rech=array("ç","à","é","è","ê","ô"," ", "'");
$rempl=array("c","a","e","e","e","o");

$nom="François Plégades de l'étôile";
echo str_replace($rech,$rempl,$nom,$nb);
echo "<br /> $nb";
?>
```

strtr -- Remplace toutes les occurrences d'un caractère par un autre.

```
<?php
$addr = strtr($addr, "èêç", "eec");
?>
```

strtr() peut être appelé avec uniquement deux arguments (une chaîne et un tableau).

Dans ce cas, elle se comporte d'une autre façon :

Notes

PHP *les bases*

```
$trans = array("Bonjour" => "salut", "salut" => "bonjour");  
echo strtr("Bonjour à tous, j'ai dit salut",$trans) . "\n";  
Cet exemple va afficher : "Salut à tous, j'ai dit bonjour".
```

substr -- Retourne une partie de la chaîne, avec index de départ et longueur en paramètres.

```
echo substr("bonjour",3,2); => jo
```

Fonctions de dates:

date(format,timestamp) -- retourne une date sous forme d'une chaîne, au format donné par la chaîne *format*. La date est fournie par le paramètre *timestamp*, sous la forme d'un timestamp. Par défaut, la date courante est utilisée.

Les caractères suivants sont utilisés pour spécifier le format

- a - "am" (matin) ou "pm" (après-midi)
- A - "AM" (matin) ou "PM" (après-midi)
- B - Heure Internet Swatch
- d - Jour du mois, sur deux chiffres (éventuellement

Notes

PHP *les bases*

avec un zéro) : "01" à "31"

- D - Jour de la semaine, en trois lettres (et en anglais)

: par exemple "Fri" (pour Vendredi)

- F - Mois, textuel, version longue; en anglais, i.e.

"January" (pour Janvier)

- g - Heure, au format 12h, sans les zéros initiaux i.e. "1" à "12"

- G - Heure, au format 24h, sans les zéros initiaux i.e. "0" à "23"

- h - Heure, au format 12h, "01" à "12"

- H - heure, au format 24h, "00" à "23"

- i - Minutes; "00" à "59"

- l (i majuscule) - "1" si l'heure d'été est activée, "0" si heure d'hiver .

- j - Jour du mois sans les zéros initiaux: "1" à "31"

- l - ('L' minuscule) - Jour de la semaine, textuel, version longue; en anglais, i.e. "Friday" (pour Vendredi)

- L - Booléen pour savoir si l'année est bissextile ("1") ou pas ("0")

- m - Mois; i.e. "01" à "12"

- M - Mois, en trois lettres (et en anglais) : par exemple

Notes

| |
|--|
| |
|--|

PHP *les bases*

"Apr" (pour Avril)

- n - Mois sans les zéros initiaux; i.e. "1" à "12"
- O - Différence d'heures avec l'heure de Greenwich, exprimée en heures; i.e. "+0200"
- r - Format de date RFC 822; i.e. "Thu, 21 Dec 2000 16:01:07 +0200" (ajouté en PHP 4.0.4)
- s - Secondes; i.e. "00" à "59"
- S - Suffixe ordinal d'un nombre, en anglais, sur deux lettres : i.e. "th", "nd"
- t - Nombre de jours dans le mois donné, i.e. "28" à "31"
- T - Fuseau horaire de la machine ; i.e. "MET"
- U - Secondes depuis une époque
- w - Jour de la semaine, numérique, i.e. "0" (Dimanche) to "6" (Samedi)
- W - Numéro de semaine dans l'année ISO-8601 : les semaines commencent le lundi (ajouté en PHP 4.1.0)
- Y - Année, 4 chiffres; i.e. "1999"
- y - Année, 2 chiffres; i.e. "99"

Notes

| |
|--|
| |
|--|

PHP *les bases*

- z - Jour de l'année; i.e. "0" à "365"
- Z - Décalage horaire en secondes (i.e. "-43200" à "43200")

Exemple :

```
<?php
echo "La date du jour est : ";
echo date("F j, Y, g:i a");
echo "<br />". date("m.d.y");
echo "<br />". date("j, m, Y");
echo "<br />". date("Ymd");
echo "<br />Pour écrire il faut
écrire il faut échapper les lettres sinon
echo "<br />". date('h-i-s, j-m-y, it is
w Day z '); //Pour
écrire il faut échapper les lettres sinon
echo "<br />". date('\C\' \e\s\t \l\e jS
\j\o\u\r. ');
echo "<br />". date("D M j G:i:s T Y");
echo "<br />". date('H:m:s \m \e\s\t
\l\e \m\o\i\s');
echo "<br />". date("H:i:s");
```

Notes

PHP *les bases*

```
echo "<br />Et en notation  
fran&ccedil;aise :";  
echo "<br />". date("d/m/y");  
echo "<br />". date("d/m/Y");  
?>
```

ce qui donne

La date du jour est : July 23, 2003, 9:31 am

07.23.03

23, 07, 2003

20030723

Pour écrire il faut échapper les lettres sinon ...

09-31-35, 23-07-03, 3131 3135 3 Wedam03 203

C'est le 23rd jour.

Wed Jul 23 9:31:35 Paris, Madrid 2003

09:07:35 m est le mois

09:31:35

Et en notation française :

23/07/03

23/07/2003

Notes

PHP *les bases*

time() retourne l'heure courante, mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).

mktime(heure,minute,seconde,mois,jour,année,heureHiver)

Retourne le timestamp UNIX d'une date.

Si on est en heure d'hiver le dernier argument est mis à 1, si c'est heure d'été à 0 et si on ne sait pas à -1 (par défaut)

microtime() retourne la chaîne "msec sec" avec sec qui est mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT), et msec qui est le nombre de microsecondes de cette heure.

```
<?php
$date=explode(" ",microtime());
$ref=$date[0];
echo "Nombre de secondes : $date[1]
<br />Nombre de microseconde : $date[0]<br />";
echo "Ce script à mis exactement : ";
$date=explode(" ",microtime());
echo ($date[0]-$ref). "microseconde pour
s'executer.";
?>
```

Notes

PHP *les bases*

Fonctions associées aux tableaux:

is_array(\$tableau) : vrai si c'est un tableau

count(\$tableau) ou **sizeof(\$tableau)** : donne la longueur du tableau

array_keys(\$tableau) : donne la liste des clés.

array_values(\$tableau) : donne la liste des valeurs.

in_array("recherche", \$tableau) renvoie vrai si recherche est dans \$tableau.

array_search("recherche", \$tableau) renvoie l'indice de la chaîne recherchée.

Attention le premier élément du tableau renvoie 0.

Pour éviter une confusion avec "faux", utiliser ===

list() envoie les valeurs de tableau dans des variables mais fonctionne uniquement avec des tableaux à indexation numérique, et suppose que l'indexation commence à 0

```
list($a, $b, $c, $d, $e) = $tableau;
```

\$a contiendra \$tableau[0]

Notes

| |
|--|
| |
|--|

PHP *les bases*

extract(\$tableau) crée des variables en correspondance clés - valeurs.

```
$livre['titre'] = 'Astérix';  
extract $livre;  
echo $titre;
```

array_push(\$tableau,"élément") ajoute un élément à la fin d'un tableau

array_pop(\$tableau) enlève le dernier élément.

array_shift(\$tableau) enlève le premier élément,

array_unshift(\$tableau,"élément") ajoute un élément au début d'un tableau.

array_slice(\$tableau, debut, longueur)

array_slice(\$jours, 1, 2) renverra (indice 1, longueur 2)
soit

```
array("mardi", "mercredi");
```

array_splice(tableau, debut, longueur, remplacement)
effectue un remplacement ou une insertion.

current(\$tableau) renvoie l'élément courant.

key(\$tableau) renvoie l'index courant

Notes

PHP *les bases*

next(\$tableau) avance de 1 élément et le renvoie

prev(\$tableau) recule de 1 élément et le renvoie

reset(\$tableau) revient au début

end(\$tableau) va à la fin

each(\$tableau) permet d'itérer: chaque fois qu'il est lancé sur un tableau, il crée un tableau contenant 4 clés: 0,1,key et value.

Exemple:

```
<?php
$menu = array("entree" => "soupe miso",
"plat" => "sashimi","dessert" =>"glace haricots
rouges"
);
/*crée le tableau ("0" => "entree", "1" => "soupe
miso","key" => "entree", "value" => "soupe miso")
*/
$temp = each($menu);
//renvoie "entree"
echo $temp[0];
//renvoie "soupe miso"
echo $temp[1];
//renvoie "entree"
echo $temp["key"];
```

Notes

PHP *les bases*

```
//renvoie "soupe miso"  
echo $temp["value"];
```

array_map("nom_fonction",\$tableau) permet d'appliquer une fonction utilisateur aux éléments d'un tableau.

Tri

sort(\$tableau, type_tri) opère un tri numérique ou alphabétique,

(rsort(\$tableau) tri à l'envers)

```
<?php
```

```
$animaux = array("antilope", "zèbre", "serpent", "babouin");
```

```
// renvoie ("antilope", "babouin", "serpent","zèbre")
```

```
sort($animaux);
```

```
?>
```

array_multisort(\$tab1, sens_tri, type_tri, \$tab2, ...) sert à trier simultanément plusieurs tableaux, ou bien à trier un tableau multi-dimensionnel, suivant l'une ou l'autre de ses dimensions. Les clés sont préservées.

sens_tri = SORT_ASC (par défaut) ou SORT_DESC

Notes

PHP les bases

type_tri = SORT_REGULAR (par default),
SORT_NUMERIC, SORT_STRING

```
<?php
$ar = array (
    array ("10", 100, 100, "a"),
    array (1, 3, "2", 1)
);
array_multisort
($ar[0], SORT_ASC, SORT_STRING,
$ar[1], SORT_NUMERIC, SORT_DESC);
?>
```

ksort(\$tableau) effectue un tri sur les étiquettes

Application

Créer un tri par titre, auteur ou éditeur sur le tableau \$livres, sur base de la variable suivante, organisée en titre,auteur,prix,stock :

```
$liste="Le Lotus Bleu#Hergé#Casterman#5.6#10
Gaston Lagaffe#Franquin#Dupuis#100#2
Natacha#Walthéry#Dupuis#9.6#5";
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

```
<?php
function xp($t,$sep='#')
{
    return(explode($sep,$t));
}
function a_lc($tab)
{
    $nc = count($tab[0]);
    $nl = count($tab);
    for($i=0;$i<$nc;$i++){
        for($j=0;$j<$nl;$j++){
            $nt[$i][$j] = $tab[$j][$i];
        }
    }
    return($nt);
}
function tabHTML($tab)
{
    echo '<table border=1>';
    foreach ($tab as $bato){
        echo '<tr>';
        foreach ($bato as $data){
            echo "<td>$data</td>";
        }
        echo '</tr>';
    }
    echo '</table>';
}
```

Notes

PHP *les bases*

```
$liste="Le Lotus Bleu#Hergé#Casterman#5.6#10
Gaston Lagaffe#Franquin#Dupuis#100#2
Natacha#Walthéry#Dupuis#9.6#5";
$tab = explode("\n",$liste);
$tab2 = array_map("xp",$tab);
tabHTML($tab2);
$ti = a_lc($tab2);
tabHTML($ti);
echo "<hr><h3>tri sur le stock</h3>";
array_multisort ($ti["4"],SORT_ASC,
SORT_NUMERIC,$ti["0"],$ti["1"],$ti["2"],$ti["3"]);
echo "<hr>";
tabHTML(a_lc($ti));
echo "<hr><h3>tri sur le prix</h3>";
array_multisort ($ti["3"],SORT_ASC,
SORT_NUMERIC,$ti["0"],$ti["1"],$ti["2"],$ti["4"]);
echo "<hr>";
tabHTML(a_lc($ti));
echo "<hr>";
echo "<hr><h3>tri sur l'auteur</h3>";
array_multisort ($ti["1"],SORT_ASC,
SORT_STRING,$ti["0"],$ti["2"],$ti["3"],$ti["4"]);
echo "<hr>";
tabHTML(a_lc($ti));
echo "<hr>";
sort ($tab2);
tabHTML($tab2);
array_multisort ($tab2[0],SORT_ASC, SORT_STRING);
tabHTML($tab2);
?>
```

Notes

PHP *les bases*

Variables PHP

`$_COOKIE` Un tableau associatif des variables passées au script courant via les HTTP cookies.

`$_GET` Un tableau associatif des variables passées au script courant via les HTTP GET.

`$_POST` Un tableau associatif des variables passées au script courant via les HTTP POST.

`$_FILES` Un tableau associatif contenant les informations sur les fichiers téléchargés avec la méthode HTTP POST.

`$_ENV` Un tableau associatif des variables passées au script par l'environnement parent.

`$_SERVER` Un tableau associatif des variables passées au script par le serveur HTTP. Ces variables sont analogues aux variables décrites ci-dessus.

`$_REQUEST` Un tableau associatif constitué du contenu des variables `$_GET`, `$_POST`, `$_COOKIE`, et `$_FILES`.

`$_SESSION` Un tableau associatif des variables enregistrées dans la session attachée au script.

`$GLOBALS` Un tableau associatif contenant les références

Notes

| |
|--|
| |
|--|

PHP *les bases*

sur toutes les variables globales actuellement définies dans le contexte d'exécution global du script.

Les noms des variables sont les clés du tableau.

Ces variables sont "superglobal" ou globales automatiques.

Cela signifie qu'elles sont simplement disponibles dans tous les contextes d'exécution (fonctions ou méthodes).

Vous n'avez pas besoin d'utiliser `global` pour y accéder à l'intérieur d'une fonction ou d'un objet.

Variables Serveur (ici apache, donc peuvent différer)

Accessibles via la superglobale `$_SERVER`

PHP_SELF La référence du script en cours.

GATEWAY_INTERFACE Numéro de révision de l'interface CGI du serveur : i.e. 'CGI/1.1'.

SERVER_NAME Le nom du serveur hôte qui exécute le script. Si le script est exécuté sur un hôte virtuel, ce sera la valeur définie pour cet hôte virtuel.

SERVER_SOFTWARE Chaîne d'identification du serveur, qui est donnée dans les en-têtes lors de la réponse aux

Notes

PHP *les bases*

requêtes.

SERVER_PROTOCOL Nom et révision du protocole de communication : i.e. 'HTTP/1.0';

REQUEST_METHOD Méthode de requête utilisée pour accéder à la page; i.e. 'GET', 'HEAD', 'POST', 'PUT'.

QUERY_STRING La chaîne de requête, si elle existe, qui est utilisée pour accéder à la page.

DOCUMENT_ROOT La racine sous laquelle le script courant est exécuté, comme défini dans la configuration du serveur.

HTTP_ACCEPT Contenu de l'en-tête Accept: de la requête courante, s'il y en a une.

HTTP_ACCEPT_CHARSET Contenu de l'en-tête Accept-Charset: de la requête courante, s'elle existe. Par exemple : 'iso-8859-1,*,utf-8'.

HTTP_ACCEPT_ENCODING Contenu de l'en-tête Accept-Encoding: de la requête courante, si elle existe. Par exemple : 'gzip'.

Notes

PHP *les bases*

HTTP_ACCEPT_LANGUAGE Contenu de l'en-tête Accept-Language de la requête courante, si elle existe.
Par exemple : 'en'.

HTTP_CONNECTION Contenu de l'en-tête Connection: de la requête courante, si elle existe.
Par exemple : 'Keep-Alive'.

HTTP_HOST Contenu de l'en-tête Host: de la requête courante, si elle existe.

HTTP_REFERER L'adresse de la page (si elle existe) qui a conduit le client à la page courante. Cette valeur est affectée par le client, et tous les clients ne le font pas.

HTTP_USER_AGENT Contenu de l'en-tête User-Agent: de la requête courante, si elle existe. C'est une chaîne qui décrit le client HTML utilisé pour voir la page courante.
Par exemple : Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586).
Entre autres choses.

REMOTE_ADDR L'adresse IP du client qui demande la page courante.

Notes

PHP *les bases*

REMOTE_HOST Le nom du client qui demande la page courante.

REMOTE_PORT Le port utilisé par la machine cliente pour communiquer avec le serveur web.

SCRIPT_FILENAME Le chemin absolu jusqu'au script courant.

SCRIPT_NAME Contient le nom du script courant.

REQUEST_URI L'URI qui a été fourni pour accéder à cette page. Par exemple : '/index.html'.

Notes

PHP *les bases*

Navigation et maintien des données

Ce chapitre doit vous permettre de comprendre les différentes possibilités offertes par PHP pour naviguer d'une page à une autre (redirection, inclusion) ou d'une session à une autre (déconnexion/reconnexion) tout en conservant l'état des données.

A la fin de ce chapitre vous saurez choisir le mode le plus adapté à la préservation de vos données au sein des scripts PHP (écriture sur serveur, variables de session ou cookies) en fonction du contexte d'utilisation.

Notes

PHP *les bases*

Redirection

```
header ("location:xxx.html") ;
```

Attention, **le code ne doit rien afficher d'autre avant** (aucun echo, print etc.), puisqu'on manipule l'entête http.

Application : Détection Langue

- A partir d'une page index1.php, créer une redirection automatique vers une page en français (index_fr.php) ou en anglais (index_en.php), suivant la configuration du navigateur.

- Si un choix explicite existe (configuration), respecter ce choix. (i.e. index1.php?lang=fr devra rediriger sur index_fr même si on est dans un environnement anglais.)

`$_SERVER["HTTP_ACCEPT_LANGUAGE"]` contient les langues.

Extraire les 2 premiers caractères avec `substr()`

Vérifier qu'un choix explicite n'a pas été fait:

`isset($lang)` est vrai si la variable lang est initialisée.

Vérifier que le fichier existe (`file_exists`) avant de rediriger.

Notes

PHP *les bases*

Corrigé

index.php

```
<?php
if (!isset($_GET["lang"])) {
    $lang =
        substr($_SERVER["HTTP_ACCEPT_LANGUAGE"], 0, 2);
} else {
    $lang=$_GET["lang"];
}
if(file_exists(index_$lang.htm)) {
    $fic = "index_$lang.htm";
} else {
    $fic = "index_fr.htm";
}
header("location:index_$lang.htm");
?>
```

index_en.htm

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

index_fr.htm

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

Notes

| |
|--|
| |
|--|

PHP *les bases*

Lecture / écriture de fichiers

Les opérations de lecture / écriture sur les fichiers externes nécessitent 3 opérations:

- Ouverture
- Lecture/écriture
- Fermeture

Ouverture fichier:

`$fichier=fopen("fichier.txt", "mode");`

mode peut prendre plusieurs valeurs:

r = lecture r+ = lecture et écriture.

L'écriture se fait par remplacement bit à bit

w = écriture par écrasement w+ = lecture et écriture

a = append = ajout L'écriture se fait à la fin du fichier

Pour certain système comme win32 on doit ajouter b après la lettre de mode pour ouvrir le fichier en mode binaire (rb, wb, ou ab).

Notes

PHP *les bases*

Lecture:

```
$texte = fgets($fichier, n);
```

OU

```
$texte = fread($fichier, n);
```

n = nbre max de caractères à lire,

p.ex. 3000 ou jusqu'à la fin de la ligne (\n)

Pour une lecture en mode binaire on utilise

```
fread($fichier, n)
```

Dans ce cas comme on ne connaît généralement pas la longueur du fichier qui en mode binaire fait une seule ligne on utilise

```
filesize("nom_fichier.ext")
```

pour récupérer la longueur.

```
$binaire=fread($fic,filesize('xav.jpg'));
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

fgetcsv(\$fichier, n, [sep], [dél])

Lit une ligne d'un fichier csv.

\$fichier est le pointeur de fichier créé avec fopen,

n : le nombre de caractères maximum lu,

sep : séparateur de champ utilisé (par défaut,)

dél : délimiteur de chaîne utilisé (par défaut c'est le ").

Détecter la fin du fichier

feof(\$fichier)

est vrai quand on arrive à la fin du fichier. feof(\$fichier)

permet d'utiliser une boucle pour lire un fichier ligne par

ligne :

```
while (!feof($fichier) {  
    echo fgets($fichier, 1000);  
}
```

Notes

PHP *les bases*

Ecriture:

fputs(\$fichier,"nouveau texte")

Ecrit la chaîne "nouveau texte" dans le fichier \$fichier

Fermeture

fclose(\$fichier);

En savoir +

Bloquer le fichier

flock(\$fichier,ver)

- Mise en place d'un verrou en lecture : ver = LOCK_SH
(valeur de 1 pour les versions plus anciennes que PHP 4.0.1).
 - Mise en place d'un verrou exclusif en écriture : ver = LOCK_EX
(valeur de 2 pour les versions plus anciennes que PHP 4.0.1).
 - Libération d'un verrou : ver = LOCK_UN
(valeur de 3 pour les versions plus anciennes que PHP 4.0.1).
- Attention : Le verrou se met en place à partir du pointeur ouvert avec fopen. si vous êtes en mode "w" la troncature

Notes

PHP *les bases*

du fichier sera faite avant la pose du verrou.

Déplacer le curseur du fichier

fseek(\$fichier,pos,from) modifie le curseur de position dans le fichier \$fichier. La nouvelle position mesurée en octets à partir du début du fichier, est obtenue en additionnant la distance pos à la position from.

From peut prendre les valeurs suivantes :

SEEK_SET - La position finale vaut pos octets.

SEEK_CUR - La position finale vaut la position courante +pos octets.

SEEK_END - La position finale vaut la position de fin du fichier + pos (pos peut être négatif).

Si from n'est pas spécifiée, il vaut par défaut SEEK_SET c'est à dire 0.

ftell(\$fichier) retourne la position courante du pointeur dans le fichier

rewind(\$fichier) replace le pointeur du fichier au début

Application : Liste d'Auteurs

A partir d'une base de données sous forme texte bdp.dat, (séparateur *) générez le fichier auteurs.sel.php contenant la liste des auteurs (dédoublonnée et trié) sous forme de liste déroulante.

Notes

PHP *les bases*

Corrigé

```
<?php
$nl = "\r\n";
$format = '<option value="%s">%s</option>'.$nl;
$fichier = fopen('bdp.dat','r');
while (!feof ($fichier)){
    $bd = explode('*',fgets($fichier,5000));
    $aut = $bd[1];
    $t_aut[$aut] = $aut;
}
ksort($t_aut);
fclose ($fichier);

$Fliste = fopen('auteurs.sel.php','w');
$strOpt = '';
foreach($t_aut as $v){
    $strOpt .= sprintf($format,$v,$v);
}

$strOpt =
"<select  name=\"c_aut\">$nl$strOpt$nl</select>";

fputs($Fliste,$strOpt);

fclose($Fliste);
?>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

Application : Boutique BDs

Création d'une page menu et d'une page détails à partir d'une base sous forme texte.

A partir d'une base de données sous forme texte bdp.dat, (séparateur *) créez une page menu affichant tous les titres des bandes dessinées de la base.

En cliquant sur un lien "détail" on accède à une page détails qui donne le titre de la BD, l'image de sa couverture si elle existe (utilisez la fonction `file_exists()`), le résumé de la BD s'il existe, l'auteur, le prix.

Prévoir un formulaire pour envoyer les données (ref,titre,prix,quantité) à un système de panier d'achats.

[Option]

Créer un lien "ajouter une BD " qui permettra d'insérer un nouveau livre dans la base, via un formulaire.

1) Page menu

ouvrir le fichier base texte (bdp.dat)

pour chaque ligne, créer une ligne de tableau html, contenant 2 cellules: titre et détail. "Détail" est un hyperlien transmettant la référence à la fiche détail.

Notes

| |
|--|
| |
|--|

PHP *les bases*

2) Page détail

ouvrir le fichier base texte (bdp.dat)

lire ligne par ligne jusqu'à rencontrer la référence récupérée en paramètre.

afficher les données.

on crée un tableau d'affichage

si l'image de la couverture existe on l'affiche sinon on

met une image par défaut (default.jpg).

De même si un résumé existe on l'affiche sinon on

écrit "pas de résumé"

S'il y a un prix on l'affiche.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

1) Page menu (menu.php)

```
<?php
$format = '<tr bgcolor="%s"><td>%s</td><td><a
href="bdp_fiche.php?ref=%s">voir</a></td></tr>';
$c=array('FFCCFF','CCCCFF');
$strLignes=''; // variable d'affichage
$f=fopen('bdp.dat','r');

$bd=explode(" ",fgets($f,5000));
$i=0;
while (!feof($f)){
    $ref=$bd[0];
    $titre=$bd[3];
    $i = !$i++;
    $strLignes.=sprintf($format,$c[$i],$titre,$ref);
    $bd=explode(" ",fgets($f,5000));
}
fclose($f);
?>
<html>
<body>
<h1>BDPhilia</h1>
<table border="1" align="center">
<?php echo $strLignes;?>
</table>
</body>
</html>
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

2) page détails (fiches.php)

```
<?php
extract($_GET);
if(!strlen($ref)){
    header('location:menuBDP.php');
}
$f=fopen('bdp.dat','r');

$bd=explode(" ",fgets($f,5000));

while (!feof($f) AND $ref != $bd[0]){
    $bd=explode(" ",fgets($f,5000));
}
fclose($f);
$ref      =$bd[0];
$titre    =$bd[3];
$auteur   =$bd[1];
$editeur  =$bd[9];
$prix     =$bd[5];
if (empty($bd[12])){
    $resume="Pas de résumé disponible";
}else{
    $resume=nl2br(htmlentities($bd[12]));
}

$img=(file_exists("couv/$ref.jpg"))?"$ref.jpg":'default.jpg';

if (empty($prix)){
```

Notes

PHP *les bases*

```
$subm = '"Non dispo" disabled';
}else{
    $subm = 'ajouter au panier';
}
//----- fin traitement
?>
<table border="1">
<tr><th colspan="2"><?php echo $titre;?></th></tr>
<tr>
<td></td>
<td><?php echo $resume;?></td>
</tr>
<tr><td>Auteur :</td><td><?php echo
$auteur;?></td></tr>
<tr><td>Editeur :</td><td><?php echo
$editeur;?></td></tr>
<tr><td>Prix :</td><td><?php echo $prix;?></td></tr>
<tr>
<td colspan="2">
<form method="post" action="ajoute.php">
<input type="hidden" name="ref" value="<?php echo
$ref;?>">
<input type="hidden" name="titre" value="<?php echo
$titre;?>">
<input type="hidden" name="prix" value="<?php echo
$prix;?>">
<input type="text" name = "qte" size="3" value="1">
<input type="submit" <?php echo $subm;?>>
</form>
</td></tr>
```

Notes

PHP *les bases*

```
</table>
```

```
</body>
```

```
</html>
```

Cookies

Les cookies sont des fichiers texte côté client.

setcookie() définit un cookie à envoyer avec le header.

Les cookies doivent être envoyés avant tout autre header .

setcookie (nom , valeur, expiration ,répertoire, domaine, sécurisé)

nom - C'est le nom du cookie (obligatoire)

valeur - contenu à stocker dans le cookie

expiration - date d'expiration du cookie (ou *null*)

répertoire - repertoire de validité du cookie dans le site

(ou *null*)

domaine - domaine de validité du cookie (ou *null*)

sécurisé - true si on veut que le cookie ne soit envoyé que

lors d'une connexion sécurisée (SSL ou S-HTTP) sinon false (valeur par défaut)

Notes

PHP *les bases*

La fonction `setcookie()` retourne `true` en cas de succès et `false` en cas d'échec.

Les cookie voyagent dans les entête http (headers). Il est donc **totallement interdit d'envoyer quoi que ce soit à l'affichage** avant le `setcookie()`.

De plus le transfert sur le poste client et sa récupération transitent via des URLs, il faut donc que la structure du cookie soit compatible avec le codage des URLs.

Les variables complexes tel que tableaux ou objets ne sont pas compatibles avec les URLs qui n'acceptent que des chaines de caractères linéarisées.

Il faut donc mettre sous forme simples chaînes de caractères linéarisées (sérialisation) les variables complexes pour les stocker dans un cookie et faire l'opération inverse (dessérialisation) pour les récupérer.

Pour cela on utilise la fonction **`serialize($var)`** et la fonction **`unserialize($cookie)`**

Notes

PHP *les bases*

Attention : Avant de dessérialiser les cookies il faut supprimer les échappements (\) des caractères spéciaux mis en place par la sérialisation.

```
$var=unserialize(stripslashes($cookie));
```

Exemple:

```
setcookie ('nom', 'Xavier');  
setcookie ('nom', 'Xavier', time()+3600);  
/* expire dans 1 h */  
setcookie ('nom','Xavier',time()+3600,  
'/xx/', '.xxx.com',1);  
/*expire dans 1 h et n'est accessible que  
pour les pages du répertoire xx du site  
xxx.com si elles sont en connexion  
sécurisée*/
```

Pour récupérer un cookie :

```
$_COOKIE["nom_du_cookie"]
```

Pour supprimer un cookie :

```
setcookie ("monCookie ", $valeur,time()-  
3600);
```

Notes

PHP *les bases*

Session

Les sessions ont été introduites dans PHP4. Ce sont des variables persistantes côté serveur, associées au client. (Cette association se fait par un cookie de session PHPSESSID qui contient un nombre aléatoire.)

`session_start()` démarre la session ou l'ouvre si elle existe déjà.

Il génère alors le cookie d'identification. Il est donc indispensable qu'aucun flux d'affichage ne soit envoyé avant `session_start()`

Exemple: compteur sur une page.

```
<?php
// initialiser la session session_start();
// l'utiliser à travers $_SESSION
$_SESSION["compteur"]++;
$compteur=$_SESSION["compteur"];
echo("Vous êtes passé $compteur fois!");
?>
```

Pour utiliser la valeur de la session sur une autre page, il suffit de rappeler la session avec `session_start()`

Notes

PHP *les bases*

En savoir +

Fonctions associées:

session_name() assigne ou lit un nom de session.

Le nom de session est le nom utilisé dans le cookie de session

session_id() assigne ou lit le id de la session courante

session_id() permet de définir l'id de session qui va être utilisé.

On peut ainsi permettre l'accès à tous les scripts se connectant à une session commune.

session_save_path("chemin") permet de définir le chemin et le répertoire où les sessions seront enregistrées.

L'assignation d'un nom d'un id ou d'un chemin à la session doit se faire avant l'appel de la session avec session_start().

session_destroy() détruit la session et toutes ses variables.

```
<?php
session_id("chat");
session_name("chat_commune");
session_start();
echo $_COOKIE["chat_commune"];
echo "<br />";
echo $_SESSION["commune"];
echo "<br />";
$_SESSION["commune"].=$_GET["qui"]." dit : Je suis".$_GET["qui"]."<br />\n"
?>
```

Notes

PHP *les bases*

APPLICATION : SESSION

1) A partir de l'exercice précédent, créer un panier d'achats avec une variable session.

Créer le programme "ajoute.php" qui permet d'ajouter un article à la commande (référence, quantité, titre, auteur, éditeur et prix) depuis la page "detail.php".

Une page "Panier.php" affiche le contenu de la commande, les sous-totaux et le total.

2) Gérer le nombre d'articles: ajouter un article déjà commandé incrémente la quantité de cet article.

3) de la même façon, gérer la suppression d'articles.

ajoute.php

enlev.php

Notes

| |
|--|
| |
|--|

PHP *les bases*

Plan

1) ajoute.php

créer une variable de session panier

recupérer le tableau transmis par le formulaire de la page détail.

si panier n'existe pas:

on met le tableau dans panier

si panier existe:

on vérifie si l'article est déjà commandé

si c'est le cas on augmente la quantité

sinon on ajoute l'article à panier

on redirige vers au panier.php

2) panier.php

on récupère la variable de session panier

on affiche ligne par ligne chaque article de panier

à la fin de chaque ligne on fait un hyperlien vers enlev en transmettant la position de la ligne (numéro du compteur de la boucle d'affichage) ou la référence.

3) enlev.php

Notes

PHP *les bases*

on récupère la variable de session panier

si la quantité est plus grande que 1

on enlève 1 à la quantité

sinon

on supprime l'article (unset ou array_splice si indice)

Notes

PHP *les bases*

Corrigé

1) Fichier ajoute.php (\$c_ref,\$c_qte,\$c_tit,\$c_prx)

```
<?php
session_start();
$panier = &$_SESSION['panier'];
extract($_POST);

if(isset($panier[$c_ref])){
    $panier[$c_ref][0] += $c_qte;
}else{
    $panier[$c_ref] = array($c_qte,$c_tit,$c_prx);
}

header('location:panier.php');
?>
```

2) Le Panier (panier.php)

```
<?php
$formatLigne = "<tr>
<td>%s</td><td>%d</td>
<td align=right>%0.2f &euro;</td>
<td align=right>%0.2f &euro;</td>
<td><a href=suppr.php?ref=%s>X</a></td>
</tr>\n";
$formatDernLigne ="<tr>
<td colspan=3>Total</td>
<td align=right>%0.2f &euro;</td>
</tr></table>";
```

Notes

PHP *les bases*

```
session_start();
$panier = $_SESSION['panier'];
echo '<table border="1">';
//print_r($panier);
foreach($panier as $ref => $article){
    list($qte,$titre,$prix) = $article;
    $soustotal = $prix * $qte;
    $total += $soustotal;

    printf($formatLigne,htmlentities(stripslashes($titre
    )), $qte, $prix, $soustotal, $ref);
}
printf( $formatDernLigne, $total);
?>
<a href=menu01.php>Menu</a>
```

Le fichier suppr.php

```
<?php
$ref = $_GET['ref'];
session_start();
$panier = &$_SESSION['panier'];

if($panier[$ref][0] > 1){
    $panier[$ref][0]--;
}else{
    unset($panier[$ref]);
}
header('location:panier.php');
?>
```

Notes

PHP *les bases*

Accès bases de données MySql

Ce chapitre doit vous permettre de comprendre le fonctionnement d'une base de données MySql et les possibilités d'interfaçage avec PHP.

Vous verrez et utiliserez les fonctions implémentées par le PHP pour interroger, actualiser ou modifier les bases MySql.

A la fin de ce chapitre vous saurez utiliser une base de données MySql et l'interfacer avec PHP.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Bases de données relationnelles et SQL

Une base de données relationnelles est une organisation logique des données dans un emplacement spécifique.

Cet emplacement est sécurisé et les utilisateurs de la base se voient attribuer des privilèges allant de la simple consultation à la gestion complète de la base.

Les données incluses dans la base sont organisées sous forme logique dans des tables.

Chaque table comprend un certain nombre de champs, que l'on nomme aussi colonnes.

Les champs regroupent des données de même sens (nom, prix, référence, etc) et donc de même type (texte, nombre, caractères, etc).

Certains champs ont des contraintes liées à leur nature (référence unique, unicité des données, contrainte de remplissage, contrainte de vérification ou contrainte de liaison avec une autre table).

Les opérations sur la base se font via des requêtes utilisant un langage structuré particulier : le SQL (Structured Query Language)

Notes

| |
|--|
| |
|--|

PHP *les bases*

MYSQL

Mysql est une base de données relationnelle basée sur le langage SQL.

Les droits des utilisateurs sont enregistrés dans des tables internes et leur authentification repose sur des mots de passe cryptés.

Mysql n'intègre pas par défaut de mécanisme de transaction (commit/rollback) sauf pour les tables de type "InnoDB".

Type de données de MYSQL

Les bases de données Mysql acceptent des données de type numérique, date, chaîne de caractères ou de type binaire.

Types numériques

Les données de types numériques sont signées par défaut. On peut en le précisant par UNSIGNED les utiliser sous forme non signée.

On peut aussi donner le nombre total de chiffres (M) et pour les nombres à virgule le nombre de chiffre après la virgule(D). Si l'option ZEROFILL est présente les nombres sont complétés à gauche par des zéros.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Si un nombre n'est pas conforme à la limite fixée par le type, il est transformé en la plus grande ou plus petite valeur permise.

Types numériques entier

BIGINT (M) UNSIGNED ZEROFILL 8 octets

Entier long (-9223372036854775808 à 9223372036854775807 ou 0 à 18446744073709551615)

INTEGER (M) UNSIGNED ZEROFILL 4 octets

Entier (-2147483648 à 2147483647 ou 0 à 4294967295)

MEDIUMINT (M) UNSIGNED ZEROFILL 3 octets

Entier (-8388608 à 8388607 ou 0 à 16777215)

SMALLINT (M) UNSIGNED ZEROFILL 2 octets

Entier court (-32768 à 32767 ou 0 à 65535)

TINYINT (M) UNSIGNED ZEROFILL 1 octet

Entier court (-128 à 127 ou 0 à 255)

Types numériques décimaux

FLOAT (M,D) ZEROFILL 4 octets

Réel

DOUBLE (M,D) ZEROFILL 8 octets

Réel en double précision

DECIMAL (M,D) M octets

Décimal codé en chaîne

Notes

| |
|--|
| |
|--|

PHP *les bases*

Types dates

DATE 3 octets

Date entre 1000-01-01 et 9999-12-31

DATETIME 8 octets

Date et heure entre 1000-01-01 00:00:00 et 9999-12-31 23:59:59

TIME 3 octets

Durée entre -838:59:59 et 838:59:59

TIMESTAMP (M) 4 octets

Temps au format Epoch début le 1 janvier 1970 à 00h00 et fin courant 2037

Suivant que M vaut 14, 12, 10, 8, 6, 4 ou 2

YYYYMMDDHHMMSS

YYMMDDHHMMSS

YYMMDDHHMM

YYYYMMDD

YYMMDD

YYMM

YY

YEAR (2 ou 4) 1 octet

Année sur 2 ou 4 chiffres

Notes

PHP *les bases*

Types chaînes de caractères

Mysql traite au choix les chaînes de caractères de longueur fixe ou variable

L'option BINARY permet de distinguer les minuscules et majuscules dans les opérations de comparaison et de tri.

CHAR (M) BINARY M octets

Chaîne de longueur fixe M caractères complétée le cas échéant par des blancs.

M <=255

VARCHAR (M) BINARY L + 1 octet

Chaîne de longueur variable L ne pouvant dépasser M caractères. M <=255

TINYTEXT L + 1 octet

Chaîne de L caractères . L <=255

TEXT L + 2 octets

Chaîne de L caractères . L <=65535

MEDIUMTEXT L + 3 octets

Chaîne de L caractères . L <=16777215

LONGTEXT L + 4 octets

Chaîne de L caractères . L <=4294967295 Les octets ajoutés en plus de la longueur pour les types de chaînes variables spécifient la limite de longueur

Notes

PHP *les bases*

Type binaire BLOB (Binary Large Object)

Les objets binaires BLOB sont des éléments codés que l'on sauve sans se soucier de leur nature, directement sous forme binaire. C'est le cas par exemple pour les images.

TINYBLOB L + 1 octet

Objet binaire de L octets. L <=255

BLOB L + 2 octets

Objet binaire de L octets. L <=65535

MEDIUMBLOB L + 3 octets

Objet binaire de L octets. L <=16777215

LOB L + 4 octets

Objet binaire de L octets. L <=4294967295 Les octets ajoutés en plus de la longueur pour les types binaires variables spécifient la limite de stockage (de 255 octets à 4GO)

Type ENUM et SET

Les données de type ENUM et SET permettent de stocker la position d'un élément dans une liste prédéfinie. Ces types permettent donc de gagner de la place.

ENUM (val1, val2, ...,valn) 1 ou 2 octets

Position d'un élément parmi 65535.

Si la liste fait moins de 255 éléments le codage est fait sur 1 octet sinon sur 2

Notes

PHP *les bases*

SET (val1, val2, ..., valn) 1 à 8 octets

Groupe d'éléments parmi un ensemble listé d'au maximum 64 (8 octets fois 8 bits)

Le codage de SET se fait sur les bits

Elément 1 code 0001 (1)

Elément 2 code 0010 (2)

Elément 3 code 0100 (4)

Elément 4 code 1000 (8)

(9) qui se code 1001 donnera la valeur "Elément 1, Elément 4"

et l'ensemble "Elément 1, Elément 2, Elément 3, Elément 4"

sera codé 1111

Pour résumé les types ENUM et SET sont équivalent à des listes à choix simples (ENUM) ou à choix multiples (SET).

SQL

Langage recouvrant toutes les opérations sur les bases de données, le SQL est regroupé en catégories de commandes qui permettent d'accomplir les différentes fonctions liées aux bases : construire les objets de la base, manipuler les objets, remplir les tables, actualiser ou supprimer les données existantes, lancer des requêtes ou contrôler les utilisateurs et leurs droits.

Notes

PHP *les bases*

DDL (Data Definition Language)

Permet de créer et de restructurer les objets de la base tels que les tables ou les index

CREATE TABLE

ALTER TABLE

DROP TABLE

CREATE INDEX

ALTER INDEX

DROP INDEX

DML (Data Manipulation Language)

Permet de manipuler les données contenues dans les objets de la base

INSERT

UPDATE

REPLACE

DELETE

DQL (Data Query Language)

Permet de faire des requêtes sur la base

SELECT

DCL (Data Control Language)

Permet de gérer l'accès aux données de la base, de contrôler et de distribuer les privilèges des utilisateurs

Notes

PHP *les bases*

ALTER PASSWORD

GRANT

REVOKE

CREATE SYNONYM

Commande de contrôle transactionnel

Pour des raisons de gain de vitesse et de charge des serveurs et de leur processeur, Mysql n'a pas mis en place des contrôles transactionnels sauf sur les tables de type "InnoDB".

COMMIT

ROLLBACK

SAVEPOINT

Les contraintes d'intégrité (CONSTRAINT)

Les contraintes d'intégrité servent dans une base de données relationnelle à assurer la cohérence et la sûreté des opérations définies via SQL sur les tables de données de la base.

On a :

PRIMARY KEY (Clé primaire)

Identification d'un ou de plusieurs champs comme référence

Notes

| |
|--|
| |
|--|

PHP *les bases*

unique de la table. Cette contrainte empêche les doublons dans la clé

UNIQUE (unicité)

Vérifie l'unicité des données dans le champ ou la contrainte s'applique. UNIQUE interdit les doublons.

FOREIGN KEY (Clé externe)

Fait référence à une clé primaire d'une autre table et vérifie sa présence lors de la saisie. (pas implémenté dans Mysql sauf pour les tables "InnoDB")

NOT NULL (obligation de remplissage)

Interdit l'absence de données dans le champ ou elle s'applique

Base de données MYSQL

Les bases de données Mysql sont organisées en répertoires portant le nom de la base protégés par login mot de passe.

Chaque répertoire contient les tables de la base

Les opérations pour utiliser une base Mysql consiste à se connecter au serveur qui vérifie la validité du login et du

Notes

PHP *les bases*

mot de passe. La connexion est établie avec les privilèges liés à l'utilisateur. On sélectionne ensuite la base et envoie les requêtes SQL.

L'évolution des bases de données MySql a entraîné de nombreuses transformations dans leur moteur (MySql 4.1 et ultérieur). De ce fait, l'extension à utiliser en PHP est devenu **mysqli_** à la place de **mysql_**

Connexion au serveur Mysql:

```
$cnx= mysql_connect('adresse','login',  
'password')
```

avec Mysql 4.1 et PHP 5

```
$cnx= mysqli_connect('adresse', 'login',  
'password','base')
```

La différence consiste à donner le nom de la base par défaut lors de la connexion (facultatif)

Notes

PHP *les bases*

Création d'une Base de Données mySql

Dans PHPmyAdmin, créer une base BDP, avec une table livres qui contiendra les champs suivants:

- ref (varchar 8, clé primaire)
- titre (tinytext)
- auteur (tinytext)
- prix (float)
- editeur(tinytext)
- resume (longtext)

Renseigner 3 livres (onglet 'insérer').

En savoir +

Création d'une Base de Données mySql avec PHP

```
<?php
```

```
//on se connecte
```

```
$cnx=mysql_connect("localhost","root","") or die("prob connexion!");
```

```
//on crée la base via une commande sql
```

```
$sql="CREATE DATABASE IF NOT EXISTS bd ";
```

```
mysql_query($sql) or die("Création de la base impossible!");
```

Notes

PHP les bases

```
//on sélectionne la base
mysql_select_db("bd");
//on écrit la commande SQL
$sql="CREATE TABLE reference
(ref char(8) primary key ,
titre tinytext ,
auteur varchar(50),
prix float(10,2),
editeur varchar(50),
resume text,
);" ;
//on envoie la commande SQL
mysql_query($sql) or die("Création de la table 'reference' impossible!");
echo "Base et table cr  e  es!";
?>

Cr  ation d'une Base mySql 4.1 avec PHP5

<?php
//on se connecte
$cnx=mysqli_connect("localhost","root","") or
die("prob connexion!");
//on cr  e la base via une commande sql
$sql="CREATE DATABASE IF NOT EXISTS bd ";
mysqli_query($cnx,$sql) or die("Cr  ation de la base impossible!");
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

```
//on sélectionne la base
mysqli_select_db($cnx,"bd");
//on écrit la commande SQL
$sql="CREATE TABLE reference
(ref char(8) primary key ,
titre tinytext ,
auteur varchar(50),
prix float(10,2),
editeur varchar(50),
resume text,
);" ;
//on envoie la commande SQL
mysqli_query($cnx,$sql) or die("Création de la table 'reference' impossible!");
echo "Base et table créées!";
?>
```

Notes

PHP *les bases*

Sélection base

Pour sélectionner une base on utilise :

```
mysql_select_db("nom_de_la_base" [, $cnx])
```

ou en php5 mysql 4.1

```
mysqli_select_db($cnx, "nom_de_la_base" )
```

Envoi d'une requête SQL

Pour envoyer une requête SQL à la base on utilise

```
mysql_query($SQL, $cnx)
```

ou en php5 mysql 4.1

```
mysqli_query($cnx, $SQL)
```

La fonction retourne une ressource (mysql) ou un objet (mysqli) de résultat dans un buffer (cas d'un SELECT) ou true/false pour indiquer la réussite ou l'échec de la requête.

Requêtes SQL

Sélections

\$sql = "SELECT ref FROM livres" // sur un élément

\$sql = "SELECT * FROM livres" //sur tout

Notes

| |
|--|
| |
|--|

PHP *les bases*

Condition numérique

`$sql = "SELECT * FROM livres WHERE PRIX < 15")`

Condition chaîne

`$sql = "SELECT * FROM livres WHERE auteur LIKE 'hergé' "`

Pour une correspondance partielle (recherche):

`WHERE titre LIKE '%riz%'` sera vrai pour "Arizona Joe" et "Pour une poignée de riz"

Combinaisons

AND ou OR possibles, avec ou sans parenthèses.

Limitation

la mention `LIMIT 0,30` renvoie les 30 premiers résultats.

Ordre

la mention `ORDER BY` permet de classer la requête sur un ou plusieurs champs (`ASC` ou `DESC`)

Notes

PHP *les bases*

En savoir +

Fonctions sql de mysql

count(*) ou **count**(champ) renvoie le nombre de ligne répondant à la requête (* renvoie tout, "champ" renvoie les non nul du champ référence)

avg(champ) renvoie la moyenne des valeur de "champ"

max(champ) renvoie la valeur la plus grande de "champ"

min(champ) renvoie la valeur la plus petite de "champ"

sum(champ) renvoie la somme des valeur de "champ"

concat(prenom, ' ', nom) renvoie la concaténation des champs listés

lower(champ) renvoie le texte de champ en minuscule

upper(champ) renvoie le texte de champ en majuscule

ltrim(champ) **rtrim**(champ) **trim**(champ) nettoie le texte de champ des espace blanc du début de la fin ou des deux

now() renvoie la date et l'heure

adddate(depart, **INTERVAL nb DAY**) renvoi la date correspondant à depart + le nb de jours. Utilisable avec les jours DAY les semaines WEEK les mois MONTH les trimestres QUARTER ou les années YEAR

subdate(dept, **INTERVAL nb DAY**) date correspondant à dep - le nb de jours.

unix_timestamp(champ) renvoi la conversion de champ en timestamp unix

lpad(champ, nbchar, remp) valeur de champ avec nbchar caractères. Si la valeur est moindre remplit jusqu'à nbchar avec remp à gauche.

rpad(champ, nbchar, remp) renvoie la valeur de champ avec nbchar caractère. Si la valeur est moindre rempli jusqu'à nbchar avec remp à droite.

Notes

PHP *les bases*

Récupérer le résultat

Le résultat est un recordset, à parser dans un tableau avec

```
$tableau =  
mysql_fetch_array(pointeur[,type_de_tableau])
```

Le type de tableau est défini par une constante PHP

| | |
|-------------|---|
| MYSQL_ASSOC | tableau à étiquettes |
| MYSQL_NUM | tableau à indices |
| MYSQL_BOTH | tableau à étiquettes et indices (par défaut) |

Pour mysql 4.1 et php5 on ajoute i après mysql et l après MySQL pour les types.

en utilisant les clés

```
$resultat = mysql_query($sql);  
while ($li = mysql_fetch_array($resultat)) {  
    echo '<p>'.$li['titre'].'</p>';  
}
```

Notes

PHP les bases

en utilisant les index

```
while ($ligne= mysql_fetch_array($resultat)) {
    echo"<p>"$ligne[1].$ligne[2]."</p>";
}
```

en utilisant un extract:

```
while ($ligne= mysql_fetch_array($resultat)) {
    extract($ligne);
    echo"<p>$ref $titre $prix </p>";
}
```

en utilisant un list

```
while ($ligne= mysql_fetch_array($resultat)){
    list($ref,$titre,$prix)=$ligne;
    echo"<p>$ref $titre $prix </p>";
}
```

Insertion

```
$sql ="INSERT INTO livres SET  
ref='$ref',titre='$titre',prix='$prix' "
```

OU

```
$sql = "INSERT INTO livres VALUE
('$ref','$titre','$prix' )"
```

en mettant les valeurs correspondant à la totalité des

Notes

| |
|--|
| |
|--|

PHP *les bases*

champs de la table dans l'ordre de la table

ou

```
$sql = "INSERT INTO livres (ref,titre,prix)
VALUES
('$ref','$titre','$prix' )"
```

Pensez à mettre entre quote simple les valeurs à entrer car mysql accepte des valeurs numériques entre quotes mais refuse les valeurs chaînes qui ne sont pas encadrées par ' '.

Les valeurs transmises par les formulaires sont le plus souvent des valeurs chaînes (par exemple un nombre entré dans un champ de type text)

Suppression

```
$sql = "DELETE FROM livres WHERE idx=$idx";
```

Ne pas oublier la clause WHERE sinon la table sera vidée

Modification

```
$sql =
"UPDATE article SET stock='$stock' WHERE ref
= '$ref' "
```

Notes

PHP *les bases*

Ne pas oublier la clause WHERE sinon toutes les valeurs de stock seront mises à \$stock

En savoir +

Jointures de tables

Pour optimiser la structure de la base, on peut répartir les informations dans plusieurs tables, et effectuer des jointures lors des requêtes:

Au lieu d'utiliser les auteurs de façon explicite dans le champ auteur de la table livres, nous allons utiliser un code auteur, et modifier le type de champ à "integer".

Parallèlement, nous créons une table "auteurs", avec un champ "c_aut" pour le code et un champ "n_aut" pour le nom de l'auteur.

1 = Hergé

2 = Walthéry

3 = Franquin etc.

La requête de sélection devient:

```
SELECT l.*,a.n_aut
```

```
FROM livres l,auteurs a
```

```
WHERE l.auteur = a.c_aut
```

Jointures externes

Elles permettent de récupérer la totalité des informations d'une table avec une partie des informations de l'autre table

Notes

| |
|--|
| |
|--|

PHP *les bases*

```
SELECT a.n_aut, l.*  
FROM livres l LEFT JOIN auteurs a  
ON l.auteur = a.c_aut  
WHERE l.ref IS NOT NULL  
ORDER BY a.c_aut
```

sous requêtes

Elles ne sont pas possible sur les versions avant Mysql 4.1

```
SELECT titre FROM livres  
WHERE auteur_id IN (SELECT id FROM `auteurs` WHERE nom LIKE '%FLO%')  
ORDER BY titre
```

Gestion des erreurs

On utilise 2 fonctions qui permettent la gestion des erreurs

mysql_errno() qui renvoie le numéro de l'erreur et

mysql_error() qui envoie le message de l'erreur.

Avec mysql 4.1 et php5 mysqli_errno() et mysqli_error().

Exemple :

```
@mysql_query($sql) or  
$erreurs.=mysql_error()."<br />\n";  
.....
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

code

```
.....;  
if ($erreurs){  
echo $erreurs;  
}else{  
echo "tout est ok";  
}
```

APPLICATION

Transférer la base texte (séparateur *) dans une base mysql.

Notes

P H P *les bases*

Plan

On crée à l'aide de PHPMyAdmin la base "bdp" avec une table "livres" contenant les champs :

ref (text, 8 caractères et clé primaire),

titre (text),

auteur (text),

editeur(text),

resume (longtext)

prix (float)

On analyse ligne par ligne le fichier bdp.dat de la base texte et on recopie les items correspondant aux champs dans la base MySql.

Comme la référence de connexion à la base sera la même pour les exercices suivant externalisé la partie connexion

dans un fichier connexion.inc.php

Notes

| |
|--|
| |
|--|

PHP *les bases*

Corrigé

Fichier connexion.inc.php

```
<?php
$cnx=mysql_connect("localhost","root","")
or die("Erreur de connexion : ".mysql_error());
mysql_select_db("bdp",$cnx)
or die ("Erreur selection de base :".mysql_error());
?>
```

Fichier de transfert

```
<?php
include("../connexion.inc.php");
$erreurs="";
$compte=0;
$fichier=fopen("../bdp.dat","r");
while (!feof($fichier)){
    $data=fgets($fichier,5000);
    $data=addslashes($data);
    $tmp=explode(" ",$data);
    $ref=$tmp[0];
    $aut=$tmp[1];
    $tit=$tmp[3];
    $editr=$tmp[9];
    $prix=$tmp[5];
    $resu=(!empty($tmp[12]))?$tmp[12]:'Pas de résumé
    disponible!';
    $sql="INSERT INTO livres VALUES
    ('$ref','$tit','$aut','$prix','$edit','$resu')";
    $compte++;
}
```

Notes

PHP *les bases*

```
@mysql_query($sql) or $erreurs.= mysql_errno()  
    ."_".mysql_error()." ligne $compte.<br />\n";  
}  
fclose($fichier);  
  
if (empty($erreurs)){  
    echo "Transfert fait!";  
}else{  
    echo $erreurs;  
}  
?>
```

APPLICATION

-Générer le menu et les fiches produit à partir de la base de données "bdp".

Notes

PHP *les bases*

2) Fichier menuSql.php

```
<table border=1>
<?php
include('connexion.inc.php');
$resu = mysql_query("select ref,titre from livres");

while($enreg = mysql_fetch_array($resu) ){
extract($enreg);
    echo "<tr>
    <td>$titre</td>
    <td>
        <a href=\"ficheSQL.php?ref=$ref\">
            détails
        </a>
    </td>
    </tr>\n";
}
?>
</table>
```

Notes

PHP *les bases*

2) Fichier ficheSQL.php

```
<?php
$ref = $_GET['ref'];
include('connexion.inc.php');
$sql = "select * from livres where ref = '$ref'";
$resu = mysql_query($sql);
$enreg = mysql_fetch_array($resu) ;
extract($enreg);

$iref = $ref;
if(!file_exists("couv/$iref.jpg")){
    $iref='default';
}
?>
<h1><?=$titre?></h1>
<h2>par <?=$auteur?></h2>
<h3>aux éditions <?=$editeur?></h3>


<form action="ajoute.php" method="post">
Quantité :
<input type="text" name="c_qte" value="1" size="3">
<input type="hidden" name="c_ref" value="<?=$ref?>">
<input type="hidden" name="c_tit"
value="<?=$titre?>">
<input type="hidden" name="c_prx"
value="<?=$prix?>">
<input type="submit" value="Ajouter à ma Commande">
</form>
```

Notes

PHP *les bases*

En savoir +

BLOB

Certains fichiers à mettre dans les bases sont ni du texte ni des nombres. Afin de préserver leur structure on les stocke sous forme binaire c'est à dire sous forme de suite de zéros et de uns copié bit à bit. C'est le cas pour par

exemple les fichiers image. Ces types de codage sont appelés BLOB (Binary Large Object)

Pour inclure un BLOB dans une base Mysql il faut que le champ soit défini comme étant un champ de type TINYBLOB BLOB MEDIUMBLOB ou LONGBLOB.

Attention la capacité de stockage prévue par le type doit être supérieur à la taille de l'objet binaire à sauver.

Une fois le champ créé on ouvre le fichier en mode binaire avec fopen() on le lit avec fread() et on le stocke avec un SQL INSERT

De la même manière on récupère le BLOB avec un SQL SELECT puis on le met dans une variable que l'on sauve avec fputs()

Code PHP pour inclure un BLOB

```
<?php
```

```
//on ouvre le fichier en mode binaire
```

```
$f=fopen("zebre.jpg","rb");
```

```
//on le place dans une variable
```

```
$data=fread($f,filesize("zebre.jpg"));
```

```
fclose($f);
```

```
if (!get_magic_quotes_runtime()){
```

Notes

PHP *les bases*

```
$data=addslashes($data);  
}  
$cnx=mysql_connect("localhost","root","") or die("prob de connexion") ;  
mysql_select_db("bdp",$cnx) or die("prob de selection de base") ;  
$sql= "insert into image SET img='$data' " ;  
@mysql_query($sql) or die("problème insertion");  
echo "Transfert fait";  
?>
```

Code PHP pour récupérer un BLOB

```
<?  
$cnx=mysql_connect("localhost","root","") or die(mysql_error()) ;  
mysql_select_db("bdp",$cnx) or die(mysql_error());  
$sql= "select img from image where id=1 " ;  
$fait=@mysql_query($sql) or die(mysql_error());  
$tmp=mysql_fetch_array($fait);  
$bin= $tmp["img"];  
/-- $bin contient le fichier image  
$f=fopen("image.jpg", "wb") ;  
fputs($f,$bin) ;  
fclose($f);  
header("Content-Type: image/jpeg");  
echo $bin;  
?>
```

Notes

PHP *les bases*

Application : Moteur de recherche

Il s'agit de créer un moteur de recherche sur plusieurs critères:

choix de l'éditeur dans une liste déroulante sans doublons,

choix de l'auteur idem,

choix sur mot clé contenus dans le résumé,

choix d'un prix mini et d'un prix maxi et

choix d'un type de requête en ET ou en OU

La requête SQL doit être créée automatiquement et la

liste correspondante doit être affichée.

recherche.php

gestion_base.php

art.php

Notes

| |
|--|
| |
|--|

PHP *les bases*

1) formulaire de recherche

- on crée un formulaire
- choix auteur : on crée une liste déroulante remplie par la base : requête sur le champ auteur avec DISTINCT
- on affiche le retour ligne par ligne dans les options du select.
- même chose pour l'éditeur.
- on crée un champ texte de recherche sur texte dans le résumé
- champs prix plus grand que et prix plus petit que
- on offre le choix du ET et du OU

2) traitement du formulaire

- on construit la requête SQL
- on initialise le SQL (SELECT * FROM reference)
- on met dans un tableau chaque élément du formulaire de recherche sous forme SQL : "auteur LIKE ...", "editeur LIKE ...", titre LIKE '%...%' etc.
- si le tableau contient au moins un élément, on le linéarise avec le séparateur AND ou OR et on exécute le sql.
- sinon, on affiche un message invitant à saisir 1 critère.

Notes

| |
|--|
| |
|--|

PHP *les bases*

Correction

```
<?php
$fOpt = '<option value="%s">%s</option>';
include('connexion.inc.php');
//création listes options auteurs et éditeurs
$listes = array('auteur','editeur');
foreach($listes as $quoi){
    $sql=
    "select distinct $quoi from livres order by $quoi";
    $resu = mysql_query($sql);
    while($x = mysql_fetch_array($resu) ){
        ${"strOpt$quoi"} .=
        sprintf($fOpt,$x[0],$x[0]);
    }
    ${"strOpt$quoi"} = "<select name=c_$quoi>
    <option value=\"\">Sélectionnez</option>
    ".${"strOpt$quoi"}."</select>\n";
}
?>
<form action="menuRechSQL.php" method="post">
<table border=1>
<tr>
<td>Auteur</td><td><?=$strOptauteur?></td>
</tr>
<tr>
<td>Editeur</td><td><?=$strOptediteur?></td>
</tr>
<tr>
<td>Mot Clé</td><td><input type="text"
name="c_cle"></td>
```

Notes

PHP *les bases*

```
</tr>
<tr>
<td>Prix Max</td><td><input type="text"
name="c_pmax"></td>
</tr>
<tr>
<td>Mode</td>
<td>
<input type="radio" name="c_mode" value="AND"
checked>ET<br>
<input type="radio" name="c_mode" value="OR">OU
</td>
</tr>
<tr>
<td colspan=2><input type="submit"
value="chercher"></td>
</tr>
</table>
</form>
<?php
if(!$ _POST){
    //accès direct
    echo "veuillez sélectionner un ou plusieurs
critères et appuyer sur le bouton";
}else{
    //deuxième accès (pour recherche)
    extract($_POST);
    //vérifier tous les champs
    if(strlen($c_auteur)){
        $crit[] = "auteur = '$c_auteur'";
```

Notes

PHP *les bases*

```
}
if(strlen($c_editeur)){
    $crit[] = "editeur = '$c_editeur'";
}
if(strlen($c_cle)){
    $crit[] = "titre like '%$c_cle%'";
}
if(strlen($c_pmax)){
    $crit[] = "prix < $c_pmax";
}
//vérifier si au moins un critère a été renseigné
if(!count($crit)){
    echo "merci de renseigner au moins un critère";
}else{
    //assembler les morceaux de requête, avec c_mode
    //comme chaîne de jointure
    $where = join(" $c_mode ", $crit);
    $sql = "select
ref,titre,auteur,editeur,prix from livres where
$where";

    echo "$sql<hr>";
    $resu = mysql_query($sql);
    echo "<table border=1>";
    while($enreg = mysql_fetch_array($resu) ){
        extract($enreg);
        echo "<tr>

<td>$titre</td><td>$auteur</td><td>$editeur</td>
<td>$prix</td>
```

Notes

PHP *les bases*

```
        <td><a
href=\"ficheSQL.php?ref=$ref\">détails</a></td>
        </tr>\n";
    }
    echo "</table>";
}
}
?>
</table>
```

Notes

PHP *les bases*

Fonctions graphiques

Ce chapitre doit vous permettre de voir les principales fonctions graphiques pour manipuler les images dans les principaux formats web.

Notes

| |
|--|
| |
|--|

PHP *les bases*

GD (Fonctions graphiques)

Les fonctions graphiques permettent de générer des images à la volée en PNG et JPEG.

La première série de fonctions concerne la création de l'image en mémoire.

On définit son type avec `header()` puis on crée l'image en mémoire soit à partir d'une image existante avec `imageCreateFromXXX()`, soit en créant une nouvelle image avec `imageCreate()`.

On travaille l'image et une fois le travail fait on envoie l'image avec `imageXXX` puis on libère l'espace mémoire avec `imageDestroy()` .

Fonction de création d'image

`header("content-type: image/XXX")` en-tête définissant pour le navigateur le type d'image (XXX) jpeg ou png

`$img = imageCreate($large,$haut)` crée une image de \$large de largeur et de \$haut de hauteur et la place dans \$img.

Notes

PHP *les bases*

L'image est une image utilisant une palette de couleurs.

Toutes les opérations sur l'image se feront avec \$img

\$img=imageCreateFromJPEG(\$fichier) ou

\$img=imageCreateFromPNG(\$fichier) Crée une image à partir d'un fichier image JPEG ou PNG.

imageSX(\$img) et imageSY(\$img) renvoie la largeur et la hauteur de l'image \$img

imageJPEG(\$img,fichier,qualite) ou

imagePNG(\$img,fichier) envoie l'image en affichage ou la sauve si un nom de fichier est spécifié.

L'argument qualité pour les images JPEG permet de définir le taux de compression. Pour afficher directement l'image dans une page html sans la sauver sur le serveur il suffit d'appeler le fichier de création image PHP dans la balise image de la page :

Ex pour un fichier créant une image MonImage et nommé MonImage.php on a dans la page html devant afficher cette image :

Notes

| |
|--|
| |
|--|

PHP *les bases*

`imageDestroy($img)` Libère l'espace mémoire de l'image sur le serveur

Gestion des couleurs de la palette image

On travaille sur une palette de couleurs, soit déjà existante (cas d'une image ouverte), soit à créer.

`imageColorAllocate($img,$r,$g,$b)` permet de définir une couleur qui est mise dans la palette de l'image. La fonction retourne l'identifiant de la couleur ou -1 en cas d'erreur (impossibilité de mettre la couleur dans la palette)

`imageColorResolve($img,$r,$g,$b)` retourne l'identifiant de la couleur si celle-ci est dans la palette ou place la couleur en tant que nouvelle couleur dans la palette. En cas de manque de place dans la palette c'est l'identifiant de la couleur la plus proche qui est retourné.

Gestion des polices

on utilise soit les polices de la bibliothèque GD

`imageString($img,$pol,$posX,$posY,$texte,$couleur)`

Ecrit le texte `$texte` dans la couleur `$couleur` en position `posX`

Notes

PHP *les bases*

et posY dans l'image \$img avec la police GD pol.

Par défaut la bibliothèque comprend 5 polices
(pol correspond au numéro)

imageFontHeight(police) : hauteur de la police en pixels

imageFontWidth(police) : largeur de la police en pixels.

Ex :

```
<?php
$string = "bonjour $nom";
//l'entête
header("Content-Type: image/png");
//créer une image de taille largeur 500 hauteur 75
$img = ImageCreate(500, 75);
//couleurs
$rouge = ImageColorAllocate($img, 255, 0, 0);
$noir = ImageColorAllocate($img, 0, 0, 0);
//texte $string de couleur $noir positionné à x=50
et y =30
//en police système n° 4
ImageString($img, 4, 50, 30, $string, $noir);
//génération
ImagePNG($img);
?>
```

Notes

P H P *les bases*

Application : Copyright

Modifier la Fiche BD pour que l'image affichée porte mention du copyright.

1) Création

on récupère la ref

on définit le header

on crée l'image à partir du bon fichier jpeg

on écrit "copyright BDPhilia" en bas à gauche.

on envoie l'image

2) Intégration dans Fiche.php

on redéfinit la source de l'image en transmettant la ref.

Notes

| |
|--|
| |
|--|

PHP *les bases*

correction

1) fichier img.php

```
<?php
$ref = $_GET['ref'];

header("Content-Type: image/jpeg");

$im = ImageCreateFromJpeg("couv/$ref.jpg");

$rrouge = ImageColorAllocate($im, 255, 0, 0);
$noir = ImageColorAllocate($im, 0, 0, 0);

ImageString($im, 5, 30, 20, 'copyright BDP', $noir);

ImageJpeg($im);
?>
```

2) Fiche.php

```
...

...
```

Notes

PHP *les bases*

Tracé de graphique

On peut aussi tracer des formes directement (ligne, rectangle, cercle).

Cette possibilité permet de créer des graphiques dynamiques que l'on affiche sous forme d'image.

Se souvenir que le format jpeg est flou du fait de sa forte compression.

Il est donc préférable si l'on veut une qualité de graphique importante (aplatissement) de travailler en PNG

Tracé des lignes

`imageline($img, posX1, posY1, posX2, posY2, $coul)`

trace une ligne du point posX1, posY1 au point posX2, posY2 dans la couleur \$coul.

`imagerectangle ($img, posX1, posY1, posX2, posY2,$coul)`

trace un rectangle du point haut gauche posX1, posY1 au point bas droit posX2, posY2 dans la couleur de tracé \$coul.

Notes

P H P *les bases*

imagefill (\$img, posX, posY, \$coul) Remplit à partir du point posX, posY l'image avec la couleur \$coul tant qu'il trouve des pixels adjacents de même couleur que celui de position posX, posY.

imagearc

(\$img, poscx, poscy, \$larg, \$haut,anglD, anglF,\$coul)
trace une ellipse de centre poscx,poscy de couleur de tracé \$coul. La largeur de l'ellipse est \$largeur et sa hauteur \$hauteur. Le tracé commenc à la position d'angle anglD et se poursuit jusqu'à la position d'angle anglF (exprimé en degrés).

Pour tracer un cercle on donne une même valeur à \$largeur et \$hauteur, 0 à anglD et 360 à anglF.

Application Graphique

Tracer un graphe correspondant à une liste de points .

L'image est appelée dans la page de réglage (liste des points séparés par une virgule)

Notes

| |
|--|
| |
|--|

PHP *les bases*

1) fichier affichage

- on crée le formulaire qui s'auto-appelle
- on crée un champ de saisie texte (pré rempli avec
- liste si la variable existe)
- on crée un select loupe avec option ($\frac{1}{4}$, $\frac{1}{2}$, 1, 2 , 4)
- on appelle l'image en passant les paramètres de liste de point et la valeur de la loupe

2) création du graphe

- on récupère la liste des points et la valeur de la loupe
- on crée le header
- on définit la position de l'axe des x
- on crée l'image
- on définit les couleurs de la palette
- on définit le rapport vertical et horizontal (horizontal dépend de la loupe)
- on trace le rectangle qui contient le graphe
- on trace l'axe des x
- on définit les coordonnées du point de départ du tracé
- on définit le nombre de points possibles à tracer sans

Notes

PHP *les bases*

sortir

- du rectangle d'affichage
- on trace le graphe point par point
- on met les repère de graduation
- on trace les verticales de grille
- on écrit la légende
- on envoie l'image

Notes

PHP *les bases*

Correction

1) Fichier d'affichage graphisme.php

```
<?php
if(isset($_POST["liste"])){
    $liste=$_POST["liste"];
}else{
    $liste="";
}
if(isset($_POST["loupe"])){
    $loupe=$_POST["loupe"];
}else{
    $loupe="1";
}
?>
<html>
<body>
<FORM action="#" method="post" name="graph">
<table>
<tr><td>Entrez la liste des points : </td>
<td>
<INPUT TYPE="TEXT" name="liste" value="<?=$liste; ?
>">
</td></tr>
<tr><td>Loupe : </td><td><SELECT name="loupe">
<OPTION value="0.25">1/4</OPTION>
<OPTION value="0.5">1/2</OPTION>
<OPTION value="1" selected="selected">1</OPTION>
<OPTION value="2">2</OPTION>
```

Notes

PHP *les bases*

```
<OPTION value="4">4</OPTION></SELECT></td></tr>
</table>
<INPUT TYPE="SUBMIT" name="submit" value="Voir">
</FORM>

<?php
echo "<img
src=\"graph01.php?liste=$liste&loupe=$loupe\">";
?>
</body></html>
```

2) Fichier de création d'image graph01.php

```
<?php
$font=dirname(__FILE__)."/ariali.ttf";
$data = explode(",",$_GET["liste"]);
$loupe= $_GET["loupe"];
Header("Content-Type: image/PNG");
$axeX = 100;
$im = imagecreate(250, 200);
$white = ImageColorAllocate($im, 240, 240, 240);
$gris = ImageColorAllocate($im, 200, 200, 200);
$black = ImageColorAllocate($im, 0, 0, 0);
$red = ImageColorAllocate($im, 255, 0, 0);
$h_scale = 20*$loupe;
$v_scale = 2;
imagerectangle($im, 25, 25, 225, 175, $black);
ImageFill($im, 50, 50, $gris);
//axes
ImageLine($im, 25, $axeX, 225, $axeX, $black);
$x0 = 25;
```

Notes

PHP *les bases*

```
$y0 = 25+$axeX;
$nbpt=min(count($data),floor(200/$h_scale));
for ($i=0;$i<$nbpt;$i++){
    $x1 = 25+$h_scale*$i;
    $y1 = $axeX - $v_scale * $data[$i];

    //tracé graphe
    ImageLine($im,$x0,$y0,$x1,$y1,$red);
    //tracé tick vertical
    ImageLine($im,$x1,$axeX,$x1,$axeX+5,$black);
    //tracé grille
    $style=array($black,$black,$black,$black,$black,$gris,
    $gris,$gris);
    imagesetstyle($im,$style);
    if (!(($i%2) AND $i!=0)){
        ImageLine($im,$x1,25,$x1,175,IMG_COLOR_STYLED);
    }
    //légende
    ImageString($im, 4, $x1, $axeX + 10, $i, $black);
    $x0 = $x1;
    $y0 = $y1;
}
imaggottext ( $im, 12, 0, 25, 190, $black,
$font, "Graphe des datas en liste");
ImagePNG($im);
?>
```

Notes

PHP *les bases*

Réglages

php.ini

répertoires

Notes

| |
|--|
| |
|--|

PHP *les bases*

php.ini

C'est le fichier de réglage de PHP. Il se situe dans le répertoire du serveur Apache. Il se présente sous forme d'un fichier texte.

Les lignes peuvent être désactivées ou activées en plaçant ou supprimant un point-virgule en début de ligne.

Les données d'initialisation du PHP sont transmises sous forme "*directive = valeur*"

Les noms des directives sont sensible à la casse

Les valeurs peuvent être une chaîne de caractères (string), un nombre, une constante PHP (e.g. E_ALL ou M_PI), une

des constantes INI (On, Off, True, False, Yes, No et None) ou une expression (e.g. E_ALL & ~E_NOTICE), ou une chaîne de caractères entre guillemets ("foo").

Il est permis d'utiliser les opérateurs sur les bits (bitwise) et les parenthèses

(| bitwise OR , & bitwise AND , ~ bitwise NOT , !

Notes

| |
|--|
| |
|--|

PHP *les bases*

boolean NOT)

La valeur vrai peut être activée par 1, On, True ou Yes.

La valeur faux peut être activée par 0, Off, False ou No.

Une chaîne vide est définie en écrivant rien après le signe
page 200

égal ou grâce au mot clé *none*

`foo =` (foo contient une chaîne vide)

`foo = none` (foo contient une chaîne vide)

`foo = "none"` (foo contient la chaîne 'none')

Le `php.ini` permet de régler entre autre :

- la possibilité d'utiliser les délimiteurs courts `<? et ?>`

ainsi que les délimiteurs asp `<% %>` (*short_open_tag* et *asp_tags*)

- l'affichage des messages d'erreurs en fonction du type d'erreur (*error_reporting*)

- l'ordre de mémorisation des variables GET, POST,

Notes

| |
|--|
| |
|--|

PHP *les bases*

Cookie,

Environnement et interne (Built-in) (G, P, C, E & S)

(*variables_order*)

- la possibilité de récupérer directement les variables sans passer par les tableaux superglobaux (*register_globals*)
- l'échappement automatique des guillemets pour les méthodes

GET POST Cookie (*magic_quotes_gpc*) et pour les données de runtime comme SQL ou `exec()`

(*magic_quotes_runtime*)

- les adresses et répertoires
- les activations des extensions (bibliothèques php)

Répertoire www

Pour que vos pages PHP soient interprétées, il est impératif de placer vos fichiers dans le répertoire www. Le serveur Apache est configuré pour ouvrir automatiquement un fichier index lorsque vous saisissez l'adresse "http://

Notes

| |
|--|
| |
|--|

PHP *les bases*

localhost/" (à condition évidemment que le serveur Apache soit en route). Cette page sert de page d'accueil au web local et permet de vérifier le bon fonctionnement des scripts.

Il est conseillé de créer un répertoire par projet dans le répertoire www afin d'avoir une vision plus claire des développements.

Répertoire tmp

C'est le répertoire des fichiers temporaires. c'est là que les variables de session sont enregistrées. Le répertoire est nettoyé par le serveur dès que la durée fixée est dépassée.

`phpinfo()`

Cette fonction affiche en retour la totalité des réglages du php. Elle permet donc rapidement de voir si une bibliothèque est active ou d'avoir les valeurs des variables serveur et php

Notes

PHP *les bases*

Références

Balises HTML

SQL

Notes

| |
|--|
| |
|--|

PHP *les bases*

LES BALISES HTML

balise Définition

attributs rôle

<html> </html> Englobe le document html dans sa totalité

<head> </head> Détermine la zone d'en-tête

<title> </title> Titre du document

<script> </script> Insert un script (Javascript ou VBscript)

<body> </body> Détermine la zone d'affichage

background = "url d'image" image de fond

bgcolor="#FFFFFF" Couleur du fond en hexadecimal

link="#0000FF" Couleur des liens

text="#000000" Couleur du texte

Vlink="#00FF00" Couleur des liens visités

<h1> </h1> Niveau de titre dans le texte (de 1 à 6)

** ** obsolète remplacé par un style

color="#FF0000" Couleur du texte

face="verdana, arial, helvetica, sans-serif" Police à utiliser

Notes

PHP *les bases*

| | |
|---|---|
| <code>size = "3"</code> | Taille du texte (entre 1 et 7) |
| <code> </code> | Texte en gras (Bold) |
| <code><i> </i></code> | Texte en italique (Italie) |
| <code><u> </u></code> | Texte souligné (Underline) |
| <code><s> </s></code> | Texte barré (Strike) |
| <code><sub> </sub></code> | Mis en indice (Subscript) |
| <code><sup> </sup></code> | Mis en exposant (Superscript) |
| <code><p> </p></code> | Marque de paragraphe (retour + saut de ligne) |
| <code> </code> | Retour à la ligne |
| <code><blockquote> </blockquote></code> | Mise en retrait d'un paragraphe |
| <code></code> | Insertion d'une image |
| <code>src="url"</code> | URL de l'image (absolue ou relative) |
| <code>alt</code> | Texte de remplacement |
| <code>border</code> | Bordure d'image (à éviter) |
| <code>height</code> | Hauteur de l'image |
| <code>width</code> | Largeur de l'image |
| <code>Align=(left right top middle bottom)</code> | Alignement de l'image par rapport au texte |

Notes

PHP *les bases*

| | |
|--|--|
| <i>hspace</i> | Espace vertical entre image et texte (en largeur) |
| <i>vspace</i> | Espace horizontal entre image et texte (en hauteur) |
| <a> | Lien |
| <i>href</i> | URL du lien |
| <i>name</i> | Étiquette d'ancrage (référence d'une ancre) |
| <object> </object> | Insertion d'un objet (animation, vidéo, son) activeX |
| <i>data</i> | URL de l'objet |
| <i>border ; width ; height ; Align ; hspace ; vspace</i> | Comme pour |
| <param /> | Paramètre de <object> |
| <embed> </embed> | Insertion d'un objet multimédia |
| <i>src</i> | URL du fichier multimédia |
| <i>align height width</i> | Comme |
| <i>Loop=(true ou false)</i> | Répétition (true) ou lecture unique (false) |
| | Liste à puces |
| <i>type =(“disc” ou “circle” ou “square”)</i> | Type de puce |

Notes

PHP *les bases*

** ** Liste ordonnée - -

Type= ("a" ou "A" ou "I" ou "I") Type de numérotation

** ** Valeur d'un élément de la liste

value Valeur initiale de la numérotation

** ** Isole un sous ensemble pour lui appliquer des paramètres. Balise de type inline

tableau

<table> </table> Mise en page d'un tableau

border Epaisseur en pixels de la bordure (0 par défaut)

width Largeur du tableau (en pixels ou en % de la fenêtre)

align Alignement dans la largeur de la page

bgcolor Couleur de fond du tableau

cellpadding Espace entre le contenu d'une cellule et les bords

cellspacing Espace entre les cellules

<tr> </tr> Ligne de cellules d'un tableau

align=(left center right) Alignement horizontal

Notes

| |
|--|
| |
|--|

PHP *les bases*

valign=(top middle bottom) alignement vertical

bgcolor couleur des cellules de la ligne

<td> </td>

Align valign bgcolor

height

width

colspan Fusion des cellules de la même ligne

rowspan Fusion des cellules de la même colonne

formulaire

<form> </form> Création d'un formulaire

name Nom du formulaire

method Mode d'envoi du formulaire (POST ou GET)

action URL du programme de traitement du formulaire

<input /> Contrôle dans un formulaire

Type=(text, password, checkbox, radio, submit, reset, button, file, image, hidden) Type du contrôle mis à disposition de l'utilisateur

Notes

PHP *les bases*

`checked="checked"` Pour les types radio ou checkbox
coche le bouton par défaut

`Maxlength` Pour les types text ou password nombre de caractères maximum saisissable

`name` Nom du contrôle

`value` Valeur initiale du contrôle

<select> </select> Liste déroulante

`name` Nom de la liste

`multiple = (true ou false)` Permet les choix multiples

<option /> Intitulé d'un élément de liste déroulante

`value` Valeur de l'élément (souvent différente de l'intitulé)

`Selected="selected"` Sélectionné par défaut

<textarea> </textarea> Zone de saisie de texte acceptant plusieurs lignes

`name` Nom de la zone

`cols` Largeur de la zone en nombre de caractères

`rows` Hauteur de la zone en nombre de lignes calques

<div> </div> Balise de regroupement utilisée pour les calques. Balise de type block

Notes

PHP *les bases*

id Référence du calque

`style = "position:absolute; left:41px; top:465px; width:177px; height:74px; z-index: 1"` mise en forme et positionnement du calque. c'est le style qui permet de mettre précisément les objets dans la page. Le z-index est l'ordre d'empilement des objets.

SQL (Structured Query Langage)

Les principales instructions SQL pour manipuler les données d'une base de données :

SELECT Sélection de données (*Permet de créer un Record*

Set (Requête))

INSERT Insertion de données (*Permet d'ajouter un enregistrement*

à la base)

UPDATE Mise à jour de données (*Permet de mettre à jour un enregistrement de la base*)

REPLACE Remplacement d'un enregistrement par ses

Notes

PHP *les bases*

nouvelles valeurs

DELETE Suppression de données (*Permet de supprimer un enregistrement de la base*)

Instruction SELECT

SELECT Nom des champs à extraire Séparé par , *
pour tous les champs

FROM Nom des tables de référence Séparé par ,

WHERE Critères de sélection Analysé par opérateur de comparaison Lié par opérateur de conjonction

ORDER BY Référence de tri et ordre Référence =
champ sur lequel se porte le tri Ordre = croissant (ASC)
ou décroissant (DESC)

LIMIT ligne de début, nombre de renvoi

limitation du nombre de réponses renvoyées par la base

page 206

Opérateurs de comparaison :

Notes

| |
|--|
| |
|--|

PHP *les bases*

= Egal (égalité stricte qui n'accepte pas l'alias % pour les chaînes de caractères)

>= Plus grand ou égal

> Plus grand que <= Plus petit ou égal

< Plus petit que <> Différent de

Opérateurs logiques :

LIKE Recherche les cellules du champ sélectionné correspondantes à une chaîne de caractère. A l'aide du signe % on peut :

Recherche les cellules du champ sélectionné finissant par (%chars)

Recherche les cellules du champ sélectionné commençant par (chars%)

Recherche les cellules du champ sélectionné contenant (%chars%)

BETWEEN Recherche les cellules du champ sélectionné comprises entre deux valeurs

IN Recherche les cellules du champ sélectionné

Notes

PHP *les bases*

comprises

dans une liste de valeurs

NOT IN Recherche les cellules du champ sélectionné
non comprises dans une liste de valeurs

IS NULL Recherche les champs vides

IS NOT NULL Recherche les champs qui ne sont pas
vides

page 207

Opérateurs de conjonction

Permet de combiner plusieurs critères de sélection

AND Toutes les conditions sont remplies (TRUE)

OR Une au moins des conditions est remplie

NOT Négation d'une condition (permet d'exclure
certaines
sélections)

L'utilisation du mot clé **DISTINCT** permet d'éliminer les

Notes

| |
|--|
| |
|--|

PHP *les bases*

doublons.

```
SELECT nom_de_champ1, DISTINCT(nom_de_champ2),  
..... FROM nom_de_table WHERE conditions ORDER BY  
nom_de_champ sens
```

Pour des sélections croisées sur plusieurs table

```
SELECT alias1.nom_de_champ1, DISTINCT(  
alias2.nom_de_champ ), ..... FROM nom_de_table1  
alias1, nom_de_table2 alias2 WHERE conditions ORDER  
BY alias2.nom_de_champ sens
```

Attention ne pas oublier le WHERE (conditions de croisement des tables) sinon toutes les combinaisons seront effectuées. (le nombre d'enregistrement de la table 1 multiplié par le nombre d'enregistrement de la table 2)

Instruction INSERT

```
INSERT INTO nom_de_table(nom_de_champ1,  
nom_de_champ2, ....) VALUES (valeur1, valeur2, ....);
```

Si on ne donne pas la liste des champs entre parenthèse après le nom_de_table tous les champs de la table seront utilisés dans l'ordre de la lecture

Notes

| |
|--|
| |
|--|

PHP *les bases*

ou

```
INSERT INTO nom_de_table SET
```

```
nom_de_champ1=valeur1, nom_de_champ2=valeur2, etc
```

Insertion via une sous-requête

```
INSERT INTO nom_de_table (nom_de_champ1,...)
```

```
SELECT sous-requête sql
```

insertion multiple

```
INSERT INTO nom_de_table
```

```
(nom_de_champ1,nom_de_champ2)
```

```
VALUES (valeur1a,valeur2a),(valeur1b,valeur2b)
```

Instruction UPDATE

```
UPDATE nom_de_table SET nom_de_champ = valeur,....
```

```
WHERE conditions
```

Instruction REPLACE

```
REPLACE INTO nom_de_table (nom_de_champ1,...)
```

```
VALUES valeur1,...
```

ou

```
REPLACE INTO nom_de_table (nom_de_champ1,...)
```

```
SELECT sous-requête sql
```

Notes

| |
|--|
| |
|--|

PHP *les bases*

ou

REPLACE INTO nom_de_table

SET nom_de_champ1= valeur1,...

Instruction DELETE

DELETE FROM nom_de_table WHERE conditions

Attention ne pas oublier le WHERE conditions sinon la table est vidée (suppression de tous les enregistrements)

Notes

| |
|--|
| |
|--|